

# Oracle® Tuxedo

## Installation Guide



Release 22c  
F61906-14  
January 2024



Oracle Tuxedo Installation Guide, Release 22c

F61906-14

Copyright © 1996, 2024, Oracle and/or its affiliates.

Primary Author: Priya Pathak

Contributing Authors: Tulika Das

Contributors: Maggie Li

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Documentation Accessibility	x
-----------------------------	---

## 1 Preparing to Install the Oracle Tuxedo System

---

1.1 Oracle Universal Installer (OUI)	1-1
1.2 Modes of Installation	1-1
1.3 Installation Media	1-2
1.4 Special Instructions for UNIX Users	1-2
1.4.1 Providing a UNIX Installer Location with Root Privileges	1-2
1.4.2 Failed to Connect to Server Error	1-3
1.4.3 Providing a UNIX Group Name	1-3
1.5 About Oracle Universal Installer Log Files	1-4
1.6 Oracle Internationalization	1-4
1.6.1 Installation Dialogs Language	1-4
1.7 Oracle Tuxedo Web Distribution	1-4
1.8 Oracle Tuxedo Software Components	1-5
1.9 Hardware and Software Prerequisites	1-6
1.10 System Requirements	1-7
1.11 Software Requirements	1-7
1.12 Interprocess Communication Resources Configuration	1-7
1.13 Oracle Installation Program	1-8
1.13.1 Cancelling Installation	1-8
1.14 Install Types	1-8
1.15 Installation Road Map	1-9

## 2 Installing Oracle Tuxedo Using GUI-Mode Installation

---

2.1 What Is GUI-Mode Installation?	2-1
2.2 Before You Start	2-1
2.3 Starting GUI-Mode Installation on a UNIX System	2-2
2.4 Starting GUI-Mode Installation on a Windows System	2-2
2.5 Running GUI-Mode Installation	2-3

2.6	Assigning File Ownership on a UNIX System	2-6
2.7	Understanding the Oracle Tuxedo Shortcuts on a Windows System	2-7
2.8	Reviewing the Windows Server Registry Content	2-7
2.9	What Do I Do Next?	2-10

### 3 Installing Oracle Tuxedo Using Console-Mode Installation

---

3.1	What Is Console-Mode Installation?	3-1
3.2	Starting Console-Mode Installation on a UNIX System	3-1
3.3	Running Console-Mode Installation	3-2
3.4	Assigning File Ownership on a UNIX System	3-8
3.5	What Do I Do Next?	3-8

### 4 Installing Oracle Tuxedo Using Silent Installation

---

4.1	What Is Silent Installation?	4-1
4.2	Before You Start	4-1
4.3	Using Silent Installation: Main Steps	4-1
4.3.1	Creating a Response File	4-1
4.3.1.1	Using Record Mode	4-2
4.3.1.2	Required Variables in Tuxedo Response File	4-2
4.4	Installing with a Response File	4-3
4.4.1	Specifying a Response File	4-4
4.4.2	Optional Parameters When Specifying a Response File	4-4
4.5	UNIX Template File	4-5
4.5.1	Listing 4-1 UNIX response File	4-18
4.6	What Do I Do Next?	4-30

### 5 Upgrading the Oracle Tuxedo System to Tuxedo 22c Release 1(22.1.0.0.0)

---

5.1	Upgrade Paths	5-1
5.2	Preparing Your Machine for an Upgrade	5-1
5.3	Selecting an Upgrade Procedure	5-2
5.4	Backing Up Files	5-2
5.5	Performing a Simple Upgrade	5-3
5.6	Performing Additional Upgrade Steps to Windows Server	5-3
5.7	Upgrading from Tuxedo Previous Releases to Oracle Tuxedo 22c Release 1 (22.1.0.0.0)	5-4
5.8	Performing a Hot Upgrade	5-5
5.8.1	Pre-Upgrade Notes	5-5
5.8.2	Hot-Upgrade Procedure	5-6

5.8.3	Hot-Upgrade Example	5-6
5.9	Rebuilding an Application	5-10
5.9.1	Changing from Object Files to Libraries	5-11
5.10	Installing Oracle Jolt 22c with Oracle WebLogic Server	5-12

## 6 Performing Post-Installation Tasks

---

6.1	Understanding the Oracle Tuxedo Directory Structure	6-1
6.2	Understanding the Oracle Service Architecture Leveraging Tuxedo Directory Structure	6-4
6.3	Understanding the Oracle Tuxedo Architecture	6-5
6.4	UBBCONFIG File	6-6
6.5	MASTER Machine	6-7
6.6	TUXCONFIG File	6-7
6.7	TUXCONFIG Environment Variable	6-7
6.8	TUXDIR Environment Variable	6-7
6.9	Setting Up Your Environment	6-7
6.10	Setting Environment Variables on a UNIX System	6-9
6.11	Starting the tlisten Process	6-9
6.12	tlisten Invocation	6-10
6.13	tlisten Password	6-11
6.14	Editing a UBBCONFIG File	6-11
6.15	Using the TYPE Parameter in UBBCONFIG	6-13
6.16	Checking IPC Requirements	6-13
6.17	Creating the Universal Device List and the Transaction Log	6-14
6.18	Defining the TLOG	6-15
6.19	Creating the UDL and UDL Entries for TLOG	6-15
6.20	Running simpapp to Verify Your Installation	6-16
6.21	Running simpapp to Verify the Oracle Tuxedo CORBA C++ Software Installation	6-16
6.21.1	Running simpapp to Verify the Oracle Tuxedo CORBA C++ Software Installation on a UNIX System	6-17
6.22	Running buildtms for Oracle Tuxedo Applications That Use XA Resource Managers	6-18
6.22.1	Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation	6-18
6.22.1.1	Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation on a UNIX System	6-19
6.23	Uninstalling Oracle Tuxedo	6-21
6.24	Reinstalling Oracle Tuxedo	6-22

## 7 Oracle Tuxedo 22c Release Platform Data Sheets

---

7.1	Platform Support Policy	7-1
7.2	Database Support Policy	7-2

7.3	Supported Platforms	7-3
7.4	Oracle Linux 7 (64-bit) on x86-64	7-3
7.4.1	Available Oracle Tuxedo 22c Install Types for Oracle Linux 7 (64-bit)	7-4
7.4.2	Hardware Requirements for Oracle Linux 7 (64-bit)	7-4
7.4.3	Software Requirements for Oracle Linux 7 (64-bit)	7-4
7.4.4	Network Requirements for Oracle Linux 7 (64-bit)	7-5
7.4.5	Disk Space Requirements for Oracle Linux 7 (64-bit)	7-5
7.4.6	Temporary Storage Space Requirements for Oracle Linux 7 (64-bit)	7-5
7.4.7	Tunable Parameters for Oracle Linux 7 (64-bit)	7-5
7.5	Oracle Linux 8 (64-bit) on x86-64	7-5
7.5.1	Available Oracle Tuxedo 22c Release Install Types for Oracle Linux 8 (64-bit)	7-5
7.5.2	Hardware Requirements for Oracle Linux 8 (64-bit)	7-6
7.5.3	Software Requirements for Oracle Linux 8 (64-bit)	7-6
7.5.4	Network Requirements for Oracle Linux 8 (64-bit)	7-7
7.5.5	Disk Space Requirements for Red Hat Enterprise Linux 8 (64-bit)	7-7
7.5.6	Temporary Storage Space Requirements for Oracle Linux 8 (64-bit)	7-7
7.5.7	Tunable Parameters for Oracle Linux 8 (64-bit)	7-7
7.6	Oracle Linux 9 (64-bit) on x86-64	7-7
7.6.1	Software Requirements for Oracle Linux 9 (64-bit)	7-7
7.7	Oracle Solaris 11 (64-bit) on SPARC	7-8
7.7.1	Available Oracle Tuxedo 22c Release Install Types for Oracle Solaris 11 (64-bit)	7-8
7.7.2	Hardware Requirements for Oracle Solaris 11 (64-bit)	7-9
7.7.3	Software Requirements for Oracle Solaris 11 (64-bit)	7-9
7.7.4	Network Requirements for Oracle Solaris 11 (64-bit)	7-9
7.7.5	Disk Space Requirements for Oracle Solaris 11 (64-bit)	7-9
7.7.6	Temporary Storage Space Requirements for Oracle Solaris 11 (64-bit)	7-10
7.7.7	Tunable Parameters for Oracle Solaris 11 (64-bit)	7-10
7.8	Red Hat Enterprise Linux 7 (64-bit) on x86-64	7-10
7.8.1	Available Oracle Tuxedo 22c Release Install Types for Red Hat Enterprise Linux 7 (64-bit)	7-10
7.8.2	Hardware Requirements for Red Hat Enterprise Linux 7 (64-bit)	7-10
7.8.3	Software Requirements for Red Hat Enterprise Linux 7 (64-bit)	7-11
7.8.4	Network Requirements for Red Hat Enterprise Linux 7 (64-bit)	7-11
7.8.5	Disk Space Requirements for Red Hat Enterprise Linux 7 (64-bit)	7-11
7.8.6	Temporary Storage Space Requirements for Red Hat Enterprise Linux 7 (64-bit)	7-11
7.8.7	Tunable Parameters for Red Hat Enterprise Linux 7 (64-bit)	7-11
7.9	Red Hat Enterprise Linux 8 (64-bit) on x86-64	7-11
7.9.1	Hardware Requirements for Red Hat Enterprise Linux 8 (64-bit)	7-12
7.9.2	Available Oracle Tuxedo 22c Release Install Types for Red Hat Enterprise Linux 8 (64-bit)	7-12

7.9.3	Software Requirements for Red Hat Enterprise Linux 8 (64-bit)	7-12
7.9.4	Disk Space Requirements for Red Hat Enterprise Linux 8 (64-bit)	7-13
7.9.5	Temporary Storage Space Requirements for Red Hat Enterprise Linux 8 (64-bit)	7-13
7.9.6	Tunable Parameters for Red Hat Enterprise Linux 8 (64-bit)	7-13
7.10	Red Hat Enterprise Linux 9 (64-bit) on x86-64	7-13
7.10.1	Software Requirements for Red Hat Enterprise Linux 9 (64-bit)	7-14
7.11	SUSE Linux Enterprise Server 12.5 (64-bit) on x86-64	7-14
7.11.1	Available Oracle Tuxedo 22c Release Install Types for SUSE Linux Enterprise Server 12.5 (64-bit)	7-14
7.11.2	Hardware Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)	7-14
7.11.3	Software Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)	7-15
7.11.4	Network Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)	7-15
7.11.5	Disk Space Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)	7-15
7.11.6	Temporary Storage Space Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)	7-15
7.11.7	Tunable Parameters for SUSE Linux Enterprise Server 12.5 (64-bit)	7-15
7.12	AlmaLinux 8 (64-bit) on x86-64	7-15
7.12.1	Hardware Requirements for AlmaLinux 8 (64-bit)	7-16
7.12.2	Available Oracle Tuxedo 22c Release Install Types for AlmaLinux 8 (64-bit)	7-16
7.12.3	Software Requirements for AlmaLinux 8 (64-bit)	7-16
7.12.4	Disk Space Requirements for AlmaLinux 8 (64-bit)	7-17
7.12.5	Temporary Storage Space Requirements for AlmaLinux 8 (64-bit)	7-17
7.12.6	Tunable Parameters for AlmaLinux 8 (64-bit)	7-17
7.13	IBM AIX 7.x (64-bit)	7-17
7.13.1	Available Oracle Tuxedo 22c Install Types for IBM AIX 7.x (64-bit)	7-18
7.13.2	Hardware Requirements for IBM AIX 7.x (64-bit)	7-18
7.13.3	Software Requirements for IBM AIX 7.x (64-bit)	7-18
7.13.4	Network Requirements for IBM AIX 7.x (64-bit)	7-19
7.13.5	Disk Space Requirements for IBM AIX 7.x (64-bit)	7-19
7.13.6	Temporary Storage Space Requirements for IBM AIX 7.x (64-bit)	7-19
7.13.7	Tunable Parameters for IBM AIX 7.x (64-bit)	7-20
7.14	Microsoft Windows 10 (32-bit, Client-only) on x86-64	7-20
7.15	Microsoft Windows 11 (32-bit, Client-only) on x86-64	7-20
7.16	Microsoft Windows Server 2019 (32-bit, Client-only) on x86-64	7-20
7.16.1	Available Oracle Tuxedo 22c Release Install Types for Microsoft Windows Server 2019 (32-bit)	7-21
7.16.2	Hardware Requirements for Microsoft Windows Server 2019 (32-bit)	7-21
7.16.3	Software Requirements for Microsoft Windows Server 2019 (32-bit)	7-21
7.16.4	Disk Space Requirements for Microsoft Windows Server 2019 (32-bit)	7-22
7.16.5	Network Requirements for Microsoft Windows Server 2019 (32-bit)	7-22
7.17	Microsoft Windows Server 2019 (64-bit) on x86-64	7-22

7.17.1	Available Oracle Tuxedo 22c Release Install Types for Microsoft Windows Server 2019 (64-bit)	7-23
7.17.2	Hardware Requirements for Microsoft Windows Server 2019 (64-bit)	7-23
7.17.3	Software Requirements for Microsoft Windows Server 2019 (64-bit)	7-23
7.17.4	Network Requirements for Microsoft Windows Server 2019 (64-bit)	7-24
7.17.5	Disk Space Requirements for Microsoft Windows Server 2019 (64-bit)	7-24
7.17.6	Tunable Parameters for Microsoft Windows Server 2019 (64-bit)	7-24
7.18	Microsoft Windows Server 2022 (64-bit) on x86-64	7-25
7.19	HP-UX 11i v3 (64-bit) on Itanium	7-25
7.19.1	Available Oracle Tuxedo 22c Install Types for HP-UX 11i v3 (64-bit)	7-25
7.19.2	Hardware Requirements for HP-UX 11i v3 (64-bit) on Itanium	7-25
7.19.3	Software Requirements for HP-UX 11i v3 (64-bit) on Itanium	7-26
7.19.4	Network Requirements for HP-UX 11i v3 (64-bit) on Itanium	7-26
7.19.5	Disk Space Requirements for HP-UX 11i v3 (64-bit) on Itanium	7-26
7.19.6	Temporary Storage Space Requirements for HP-UX 11i v3 (64-bit) on Itanium	7-26
7.19.7	Tunable Parameters for HP-UX 11i v3 (64-bit) on Itanium	7-26

## 8 Oracle Jolt 22c Release Overview and Installation Information

---

8.1	About Oracle Jolt	8-1
8.2	Oracle Jolt Components	8-2
8.3	Jolt Server Listener	8-3
8.4	Jolt Server Handler	8-3
8.5	Metadata Repository Server	8-4
8.6	Metadata Repository	8-4
8.7	Jolt Internet Relay	8-4
8.8	Jolt Class Library	8-5
8.9	JoltBeans	8-6
8.10	Jolt ECID	8-6
8.11	Oracle Jolt Client Personalities	8-7
8.12	Oracle Jolt Capabilities	8-7
8.13	Oracle Jolt Client Support	8-8
8.14	Oracle Jolt Client Requirements	8-9
8.15	Oracle Jolt Client Class Library	8-9
8.16	Oracle Jolt Release Interoperability	8-9
8.17	Oracle Jolt Documentation	8-10

## 9 File and Database Management and Disk Space Allocation

---

9.1	Introduction	9-1
9.2	How the Oracle Tuxedo System Manages Files	9-1
9.3	Arranging for Raw Disk Space	9-2



9.4	How the Oracle Tuxedo Filesystem is Organized	9-2
9.5	Space for Queue Spaces	9-3
9.6	Space for Application Servers	9-3

## 10 IPC Resource Configuration on a UNIX System

---

10.1	Parameter Sets Controlling IPC Resources	10-1
10.1.1	Shared Memory	10-2
10.1.2	Semaphores	10-2
10.1.3	Message Queues and Messages	10-3
10.1.4	What Is Application Deadlock?	10-4
10.1.5	Performance Implications of Blocking Conditions	10-4
10.1.6	Tunable Message Parameters	10-4
10.1.7	Other Kernel Tunables	10-5

# Preface

The installation guide provides detailed information for installing Oracle Tuxedo 22c Release 1 (22.1.0.0.0).

- [Documentation Accessibility](#)

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

# 1

## Preparing to Install the Oracle Tuxedo System

Preparing to Install the Oracle Tuxedo System The following sections provide information that you need to know before installing the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) product software:

- [Oracle Universal Installer \(OUI\)](#)
- [Modes of Installation](#)
- [Installation Media](#)
- [Special Instructions for UNIX Users](#)
- [About Oracle Universal Installer Log Files](#)
- [Oracle Internationalization](#)
- [Oracle Tuxedo Web Distribution](#)
- [Oracle Tuxedo Software Components](#)
- [Hardware and Software Prerequisites](#)
- [System Requirements](#)
- [Software Requirements](#)
- [Interprocess Communication Resources Configuration](#)
- [Oracle Installation Program](#)
- [Install Types](#)
- [Installation Road Map](#)

### 1.1 Oracle Universal Installer (OUI)

The Oracle Tuxedo 22c Release 1 (22.1.0.0.0) product software installer is based on the Oracle Universal Installer (OUI). To install the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) product software on your host, you must know how to use OUI to install Oracle products.



#### See Also:

[Introduction to Oracle Universal Installer](#)

### 1.2 Modes of Installation

You can use OUI to install Oracle products in any of the three following modes:

**Interactive**

Use OUI interactive mode to use the graphical user interface to walk through the installation, providing information in the installation dialogs when prompted. This method is most useful when installing a small number of products in different setups on a small number of hosts.

**Console**

Use OUI console mode to bypass the graphical user interface and execute the Oracle Installation program with a console interactive interface. This method is intended for UNIX systems with non-graphical consoles.

**Silent**

Use OUI silent installation mode to bypass the graphical user interface and supply the necessary information in a response file. This method is most useful when installing the same product multiple times on multiple hosts. By using a response file, you can automate the installation of a product for which you know the installation parameters.

## 1.3 Installation Media

 **Note:**

When you invoke `runInstaller.sh` (UNIX), you should invoke it from the directory where this command is present, or you must specify the complete path to `runInstaller.sh` (UNIX).

## 1.4 Special Instructions for UNIX Users

The following sections describe special instructions that apply when you are installing certain products on a UNIX system.

- [Providing a UNIX Installer Location with Root Privileges](#)
- [Failed to Connect to Server Error](#)
- [Providing a UNIX Group Name](#)

### 1.4.1 Providing a UNIX Installer Location with Root Privileges

If you are installing Oracle Products with OUI for the first time, you are prompted to run a shell script from another terminal window before proceeding with the installation.

 **Note:**

When running OUI in silent mode, `orainstRoot.sh` is required prior to configuration assistants, OUI skips configuration assistants during the installation. You must run `orainstRoot.sh` as root and then run the skipped configuration assistants after the silent installation is complete.

To successfully run the required shell script:

1. Leave the OUI window open and open another terminal window.

2. In the new terminal window, use the substitute user command to log in with root privileges: `su -root`.
3. Change directory to the Oracle home into which you are currently installing your Oracle software product.
4. Run the shell script `orainstRoot.sh`.
5. When the script is finished and you are returned to the command prompt, exit from the new terminal window and return to OUI to continue installation.

**Note:**

Do not exit the installation to run the shell script. Exiting the installation removes this script. You are prompted to run the script only when you install Oracle products for the first time with OUI.

## 1.4.2 Failed to Connect to Server Error

If you receive an Xlib error or a "Failed to connect to Server" error when you are running OUI on the Solaris operating system, do the following:

1. Define the following environment variable on the host computer where you are running OUI:

```
%setenv DISPLAY <machine name>:0.0
```

2. Replace `<machine name>` with the name of the computer that displays the OUI.
3. On the computer that displays the OUI, enter the following command, which allows other computers to display information on the computer monitor: `%xhost +`
4. Rerun the `runInstaller.sh` script after you have set the `DISPLAY` environment variable.

**Note:**

You can run OUI without specifying the `DISPLAY` variable by running in silent mode using a response file.

## 1.4.3 Providing a UNIX Group Name

If you are installing a product on a UNIX system, the Installer also prompts you to provide the name of the group that owns the base directory.

You must choose a UNIX group name that has permissions to update, install, and remove Oracle software. Members of this group must have write permissions for the chosen base directory.

Only users who belong to this group are able to install or remove software on this host.

## 1.5 About Oracle Universal Installer Log Files

When you install or deinstall products using OUI, important information about each installation is saved not only in the inventory, but also in a series of log files, located in the following directory:

```
$ORACLE_HOME/cfgtoollogs
```

You can use these log files to troubleshoot installation problems. These files are also crucial for removing and configuring the various software components you install. OUI displays the name and location of the current session log file on the Install page. Each installation or configuration utility provides a separate folder containing the logs inside the `$ORACLE_HOME/cfgtoollogs` folder.

### Note:

The logs used to remove products are different from `installActions<timestamp>.log` generated during the install process. The `installActions<timestamp>.log` is easier to read and can be used to view the operations performed at installation time.

## 1.6 Oracle Internationalization

This section helps you set your OUI (Oracle User Interface ) language.

- [Installation Dialogs Language](#)

### 1.6.1 Installation Dialogs Language

OUI runs in the operating system language. OUI uses the language that Java detects the system locale value, and sets that to the default language. The OUI dialogs are displayed in this language if available. If specific the OUI dialogs are not translated in the language of the operating system, these dialogs are shown in English.

OUI displays the translated GUI only if the variable `NLS_ENABLED` has been set to `TRUE` in the `oraparam.ini` file. If the `NLS_ENABLED` variable is set to `FALSE`, all text is shown in English.

### Note:

The dialogs displayed for internationalization can only be customized parts; some of them are embedded in OUI.

## 1.7 Oracle Tuxedo Web Distribution

Oracle Tuxedo is available for download from the Oracle corporate [Web site](#).

Platform-specific installer files for the Oracle Tuxedo product software are available for download from the Oracle corporate Web site.

## 1.8 Oracle Tuxedo Software Components

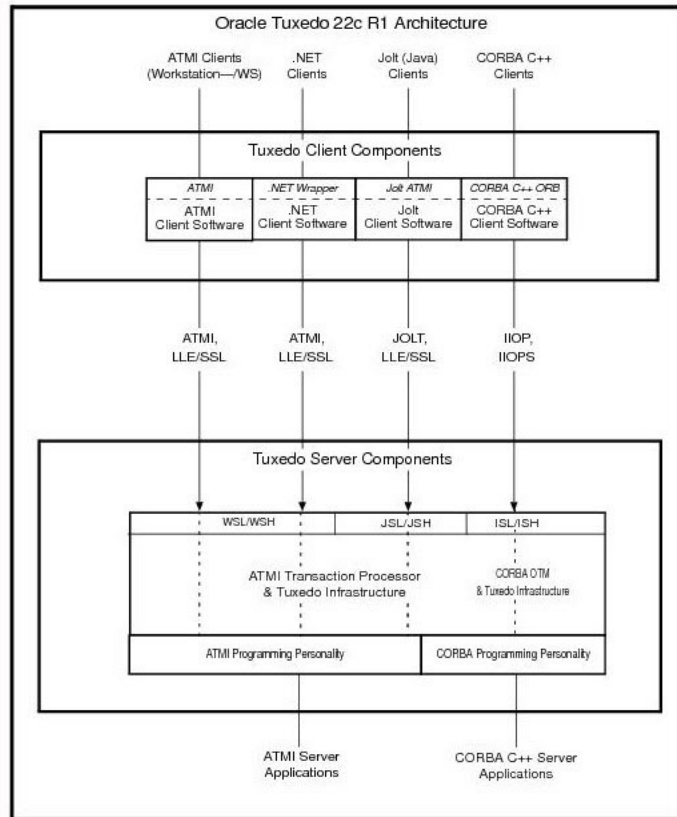
Included with the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) software distribution is the Oracle Jolt product software, which enables Java clients to call Oracle Tuxedo services, and the Oracle SNMP Agent product software, which enables Oracle Tuxedo applications to be managed from a centralized enterprise management console, and Oracle Service Architecture Leveraging Tuxedo (Oracle SALT), which provides native Tuxedo Web service.

The Oracle Tuxedo 22c Release 1 (22.1.0.0.0) distribution contains the following components:

- Oracle Tuxedo Server components:
  - Oracle Application-to-Transaction Monitor Interface (ATMI) server software
  - Oracle Common Object Request Broker Architecture (CORBA) C++ server software
  - Oracle Jolt server software
  - Oracle SNMP Agent software
- Oracle Tuxedo 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 22c Release 1 (22.1.0.0.0) client software
- Oracle Service Architecture Leveraging Tuxedo (Oracle SALT):
  - Oracle Service Architecture Leveraging Tuxedo (Oracle SALT) software

The following figure illustrates the high-level components of Oracle Tuxedo:

Figure 1-1 Oracle Tuxedo Client and Server Components



**See Also:**

[Client and Server Components](#)

## 1.9 Hardware and Software Prerequisites

The Oracle Tuxedo software must be installed on each server machine that participates in an Oracle Tuxedo application, also known as a Tuxedo domain. A Tuxedo application is a business software program, built upon the Tuxedo system, which is defined and controlled by a single configuration file known as the `UBBCONFIG` file. The Tuxedo configuration file is described on reference page [UBBCONFIG\(5\) in 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.
- 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](#).

## 1.10 System Requirements

The system requirements for Oracle Tuxedo 22c Release 1 (22.1.0.0.0) mentioned in the following table:

**Table 1-1 The System Requirements for Oracle Tuxedo 22c Release 1 (22.1.0.0.0)**

Component	Requirement
Platform*	Any platform identified in <a href="#">Oracle Tuxedo 22c Release Platform Data Sheets</a> .
Hard disk drive	As stated in the data sheet for the target platform in <a href="#">Oracle Tuxedo 22c Release Platform Data Sheets</a> . For additional information about disk space requirements, see <a href="#">File and Database Management and Disk Space Allocation</a> .
Memory	As stated in the data sheet for the target platform in <a href="#">Oracle Tuxedo 22c Release Platform Data Sheets</a> .

 **See Also:**

[Oracle Jolt 22c Release Overview and Installation Information](#).

## 1.11 Software Requirements

Java Runtime Environment(JRE) is required to install Oracle Tuxedo 22c using console mode. JRE 1.8 and above are recommended. Ensure that you install the required JRE and set the environment variable `JAVA_HOME` accordingly before installing Oracle Tuxedo.

## 1.12 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. On a UNIX system, you use methods native to the UNIX system to adjust the IPC parameters and maximize the performance of an Oracle Tuxedo application. Since most UNIX systems are

shipped with default values that are too low for an Oracle Tuxedo application, you need to adjust the IPC parameters, using the methods given in [Oracle Tuxedo 22c Release Platform Data Sheets](#). For the recommended IPC values, see [IPC Resource Configuration on a UNIX System](#).

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 and using the `tmloadcf(1)` command, see [Performing Post-Installation Tasks](#)

**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 the information mentioned in the following link, [IPC Resource Configuration on a UNIX System](#)

## 1.13 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 UNIX systems.

- [Cancelling Installation](#)

### 1.13.1 Cancelling Installation

Clicking the “Cancel” or the “close” window button in GUI mode causes an incomplete Oracle Tuxedo 22c Release 1 (22.1.0.0.0) installation. In that case, you must re-install Oracle Tuxedo 22c Release 1 (22.1.0.0.0).

If you install Oracle Tuxedo 22c Release 1 (22.1.0.0.0) in a previous Tuxedo product directory, and you want to return to your original configuration, you must re-install your previous Tuxedo version. It is strongly suggested that you install the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) to a new Oracle Home product directory.

## 1.14 Install Types

An install type is a bundle of product software components related by function. Oracle Tuxedo 22c Release 1 (22.1.0.0.0) offers the following install types:

- Full Install - consists of all the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) server and client components of Tuxedo, SALT, and TSAM Plus agent.
- Server Install - consists of all the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) server components of Tuxedo, SALT, and TSAM Plus agent.
- Client Install - consists of all the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) 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 22c Release 1 (22.1.0.0.0) client software

In addition to selecting an install type during an Oracle Tuxedo 22c Release 1 (22.1.0.0.0) installation, a user can further customize the installation by selecting (adding) or deselecting (deleting) one or more software components from the install type. Customizing is only possible for GUI-mode installation methods described in [Installing Oracle Tuxedo Using GUI-Mode Installation](#).

## 1.15 Installation Road Map

You are now ready to begin your installation. To install Oracle Tuxedo 22c Release 1 (22.1.0.0.0), see one of the following sections:

- [Installing Oracle Tuxedo Using GUI-Mode Installation](#).
- [Installing Oracle Tuxedo Using Console-Mode Installation](#).
- [Installing Oracle Tuxedo Using Silent Installation](#)

If you are upgrading from Oracle Tuxedo Release (12.2.2.0.0) or earlier software, see [Upgrading the Oracle Tuxedo System to Tuxedo 22c Release 1\(22.1.0.0.0\)](#)

If you want to uninstall your Oracle Tuxedo software, see [Uninstalling Oracle Tuxedo](#)

# 2

## Installing Oracle Tuxedo Using GUI-Mode Installation

The following sections describe how to install Oracle Tuxedo using graphical user interface (GUI) mode installation:

- [What Is GUI-Mode Installation?](#)
- [Before You Start](#)
- [Starting GUI-Mode Installation on a UNIX System](#)
- [Starting GUI-Mode Installation on a Windows System](#)
- [Running GUI-Mode Installation](#)
- [Assigning File Ownership on a UNIX System](#)
- [Understanding the Oracle Tuxedo Shortcuts on a Windows System](#)
- [Reviewing the Windows Server Registry Content](#)
- [What Do I Do Next?](#)

### 2.1 What Is GUI-Mode Installation?

The graphical user interface mode installation is the graphics-based method of executing the Oracle Installation program. It can run on the UNIX systems. To run GUI-mode installation, the console attached to the machine on which you are installing the software must support a Java-based GUI.



#### Note:

To install Oracle Tuxedo on a UNIX system with a non-graphics console, use silent install mode.

### 2.2 Before You Start

If you are upgrading from Tuxedo 12.2.2 or earlier, see [Upgrading from Tuxedo Previous Releases to Oracle Tuxedo 22c Release 1 \(22.1.0.0.0\)](#), and follow the instructions given there. Then return here to continue your Oracle Tuxedo installation.

If you choose to remove the older software, refer to [Upgrading from Tuxedo Previous Releases to Oracle Tuxedo 22c Release 1 \(22.1.0.0.0\)](#), 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 22c Release 1 (22.1.0.0.0) software to a different Oracle Home directory than the Oracle Home directory where the older version of the Oracle Tuxedo software is installed.

## 2.3 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 22c Release Platform Data Sheets](#).
2. Log in to the UNIX system as the Oracle Tuxedo administrator.
3. Ensure that you have enough free space for the Oracle Tuxedo installation.

For disk space requirements, see [Oracle Tuxedo 22c Release Platform Data Sheets](#). For important disk space allocation information, see [File and Database Management and Disk Space Allocation](#).

4. Install Oracle Tuxedo by downloading from the Oracle Web site:
  - a. Go to [Download Oracle Tuxedo](#) and download the Oracle Tuxedo 22c Release 1 (22.1.0.0.) installation file specific to your platform.
  - b. Go to the directory where you downloaded the installer, unzip the installer file, and then go to the `Disk1/install` directory, run the following command to invoke the installation procedure: `./runInstaller.sh`

### Note:

GUI mode is the default for Oracle Tuxedo 22c Release 1 (22.1.0.0.0) installation. If a GUI interface is not available on your UNIX platform, an error message displays and the installation process is aborted.

If a GUI interface is not available on your UNIX platform, you may use console mode or silent mode installation. For more information, see [What Is Console-Mode Installation?](#) and [What Is Silent Installation?](#)

5. Proceed to [Running GUI-Mode Installation](#).

## 2.4 Starting GUI-Mode Installation on a Windows System

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

1. For the Install Types of Microsoft Windows Server 2019 (64-bit), select a Windows system that meets the hardware and software requirements described in the [Oracle Tuxedo 22c Release 1 \(22.1.0.0.0\) Platform Data Sheet](#).
2. Log in to the Windows system as the `Administrator` or as a member of the `Administrator` group. You need administrative privileges to install the Oracle Tuxedo server components on a Windows system. However, if you are going to install only the Oracle Tuxedo client components, then you do not need administrative privileges.

3. Ensure that you have enough free space for the Oracle Tuxedo installation.

For disk space requirements, see [Disk Space Requirements for Microsoft Windows Server 2019 \(64-bit\)](#).

4. Install Oracle Tuxedo by downloading from the Oracle Web site:

- a. Go to <http://www.oracle.com/technology/software/index.html> and download the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) installation file specific to your platform.
- b. Go to the directory where you downloaded the Oracle Tuxedo installer, unzip the installer file, and then go to the Disk1\install directory, run the following command to start the installation: `setup.bat`

 **Note:**

If you have an old version of Oracle Tuxedo installed on the machine, you should deinstall it. After completing the deinstallation, click the **exit** button to exit the OUI installer process. Next, run the `setup.bat` command to start a new installation. If you already have Oracle Tuxedo 22c version installed, then you need to uninstall it first, and then exit the current installation process. Further, you will need to re-launch the installer by running the `set.bat` command.

5. Proceed to [Running GUI-Mode Installation](#)
6. Log-in again to the system after Oracle Tuxedo 22c Release 1 (22.1.0.0.0) installation has completed.

## 2.5 Running GUI-Mode Installation

The Oracle Tuxedo 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
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 are installed by default to English.

The Oracle Tuxedo installer program prompts you to enter specific information about your system and configuration.

For GUI-Mode installation screens, see Oracle Universal Installer Installation Screens.

**Table 2-1** Instructions

Window	Action
Welcome	Click <b>Next</b> to proceed with the installation. You may cancel the installation at any time by clicking Cancel.
Specify Inventory directory (UNIX only)	On UNIX operating systems, this screen appears if this is the first time you are installing any Oracle product on this host. Specify the location where you want to create your central inventory. Ensure that the operating system group name selected on this screen has write permissions to the central inventory location.

Table 2-1 (Cont.) Instructions



Window	Action
Choose Install type	<p>Select the install type that you want installed on your system. The following choices are available:</p> <ul style="list-style-type: none"> <li>• Full Install (the default) - all the Oracle Tuxedo server and Tuxedo, SALT, and TSAM Plus agent client components.</li> <li>• Server Install - all Tuxedo, SALT, and TSAM Plus agent server components of</li> <li>• Client Install - Tuxedo client software components only</li> <li>• Custom Install - select specific Tuxedo, SALT, and TSAM Plus agent server and client software components. The following table entry provides a summary of options for the Custom Install.</li> </ul> <p>Select an install type and then click <b>Next</b> 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).</p> <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;">  <b>See Also:</b> Install Types.         </div>
Specify Oracle Home	<p>Name: Enter a name for the Oracle home. This name identifies the program group associated with a particular home and the Oracle services installed on this home. The Oracle home name must be between 1 to 127 characters long, and can include only alphanumeric characters and underscores.</p> <p>Enter the full path to an Oracle home, or select an Oracle home from the drop-down list of existing Oracle homes. The Oracle home location is the directory where products are installed. Data files may or may not be installed within an Oracle home. You can use the Browse button to choose a directory to install your product. For Windows platforms, you must provide a valid path that is not in the Windows directory. Different homes cannot share the same location.</p> <p>It is recommended that you designate an Oracle home location that is an empty or non-existing directory. If you select a directory for the Oracle home location that is not empty or already exists, a warning message displays. In addition, you can specify whether or not to proceed.</p> <p>For silent installations, if a non-empty, existing directory is specified, an error is logged in the console and in the <code>installActions&lt;timestamp&gt;.log</code> file. Also, the OUI aborts. To override this condition, use the <code>-force</code> flag on the command line. The effect of using the <code>-force</code> flag is the same as selecting Yes while installing in interactive mode. You receive a warning message, but the installation continues.</p> <p>Tuxedo Product Installation Directory is "<code>ORACLE_HOME/tuxedo21.1.0.0.0</code>" by default (UNIX) and cannot change after the <code>ORACLE_HOME</code> is determined. Tuxedo installer doesn't allow installation of two versions of the same tuxedo product into the same <code>ORACLE_HOME</code>.</p> <p>If the selected <code>ORACLE_HOME</code> had the same version tuxedo installed, the installer displays a warning.</p>

Table 2-1 (Cont.) Instructions

Window	Action
Customize Install	<p>Select (add) or deselect (clear) one or more software components from the selected install type, or choose one of the other four install types or Custom Set from the drop-down list menu and customize its software components. For a description of the JRLY component, see <a href="#">Jolt Internet Relay</a>.</p> <p>Observe the following software component mappings:</p> <ul style="list-style-type: none"> <li>• <i>Tuxedo Server</i>— contains ATMI server software; CORBA C++ server software; Oracle Jolt server software; and Oracle SNMP Agent software</li> <li>• <i>Tuxedo ATMI Client</i>— contains Oracle ATMI Workstation (/WS) client software</li> <li>• <i>Tuxedo CORBA Client</i>— contains Oracle CORBA C++ client software (C++ client ORB) including environmental objects</li> <li>• <i>Tuxedo Jolt JRLY</i>— contains Oracle Jolt Relay software-</li> <li>• <i>Tuxedo .NET Client</i>— contains Oracle .Net client software</li> <li>• <i>Oracle Tuxedo System and Applications Monitor Plus (Oracle TSAM Plus) agent</i> — contains Oracle Tuxedo System and Applications Monitor Plus (Oracle TSAM Plus) agent software</li> <li>• <i>Oracle Services Architecture Leveraging Tuxedo (Oracle SALT)</i>— contains Oracle Services Architecture Leveraging Tuxedo (Oracle SALT) software</li> </ul> <p>After selecting or deselecting one or more software components from the selected install type, click Next to continue with the installation. The appropriate encryption software for LLE and/or SSL is automatically included.</p>
Confirm enable TSAM plus agent	This screen appears if the previously selected bundles include TSAM plus agent component. Enable it to use TSAM plus manager monitor and manage Tuxedo.
SSL Installation Choice	<p>This window appears only if you have installed Server, ATMI and/or CORBA Client install type components. Oracle Tuxedo provides a Lightweight Directory Access Protocol (LDAP) based certificate retrieval mechanism that supports certificate-based authentication when using SSL. The Oracle Tuxedo certificate retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.</p> <p>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.</p>
LDAP Configuration for SSL Support	<p>Enter the following LDAP configuration information and then click Next to continue:</p> <ul style="list-style-type: none"> <li>• LDAP Service Name: Fully qualified domain name of the LDAP server (for example, pcwiz.mydomain.com)</li> <li>• LDAP PortID: Port number through which the local machine communicates with the LDAP server (for example, 389)</li> <li>• LDAP BaseObject: Distinguished name of the base object for search in the LDAP server (for example, o=beasys.com)</li> </ul>
LDAP filter file choose	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 software, the default LDAP filter file is generated after installation is completed).
Select samples installation	If the selected install feature sets include server component, this screen appears to confirm the install the samples.



**Table 2-1 (Cont.) Instructions**

Window	Action
Configure <code>tlisten</code> Choice	<code>tlisten</code> is needed for MP/clustered mode configuration and Oracle TSAM Plus. If you need to configure <code>tlisten</code> , select "Yes", and then input the parameters for <code>tlisten</code> configuration on the screens that follow.
Configure <code>tlisten</code> password	<p>This window appears only if you select to configure <code>tlisten</code>. Enter a <code>tlisten</code> 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.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> <b>Note:</b></p> <p>Machines on the same domain must have the same <code>tlisten</code> password defined.</p> </div> <p>Oracle Tuxedo uses the <code>tlisten</code> password to protect the local machine from administrative requests and operations that are not authorized. Whenever administrative communication is received on the local machine through <code>tlisten(1)</code> gateway processes, Oracle Tuxedo authenticates using the <code>tlisten</code> password.</p> <p>Enter and re-enter a <code>tlisten</code> password and then click Next to continue.</p>
SSL Parameter Input	If the SSL encryption method is chosen, you must enter Security Principal name, location, and password.
Summary	Click <b>Install</b> to start the installation.
Installing	Nothing to do
Install End	Click <b>Exit</b> to exit the installation program.



**Note:**

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

## 2.6 Assigning File Ownership on a UNIX System

If you installed the Oracle Tuxedo software on a UNIX system, Oracle strongly recommends that you create a separate user account for the Oracle Tuxedo administrator and give ownership of the Oracle Tuxedo files to that account. To change ownership of the Oracle Tuxedo 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 administrator account, replace `adm_group` with the Oracle Tuxedo 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 software.

## 2.7 Understanding the Oracle Tuxedo Shortcuts on a Windows System

When installing the Oracle Tuxedo software on a Windows system, the installer program automatically creates an Oracle home directory and a Tuxedo directory under the Oracle home directory. The Tuxedo directory (the environment variable `TUXDIR`) will be added to the system environment after installation.

## 2.8 Reviewing the Windows Server Registry Content

The Windows Server Registry is the repository for all hardware, software, and application configuration settings for the Windows system. During the Oracle Tuxedo installation, the installer program writes general installation information as well as IPC Resource values to the Registry. The structure of the Registry relevant to Oracle Tuxedo is as follows:

```
HKEY_LOCAL_MACHINE\Software\ORACLE\Tuxedo\<version>\...
```

- Developer
- Environment
- Security

To view this structure, choose Start → Run to launch the Run dialog box, enter Registry Editor, and click **OK** to launch the Registry Editor window.

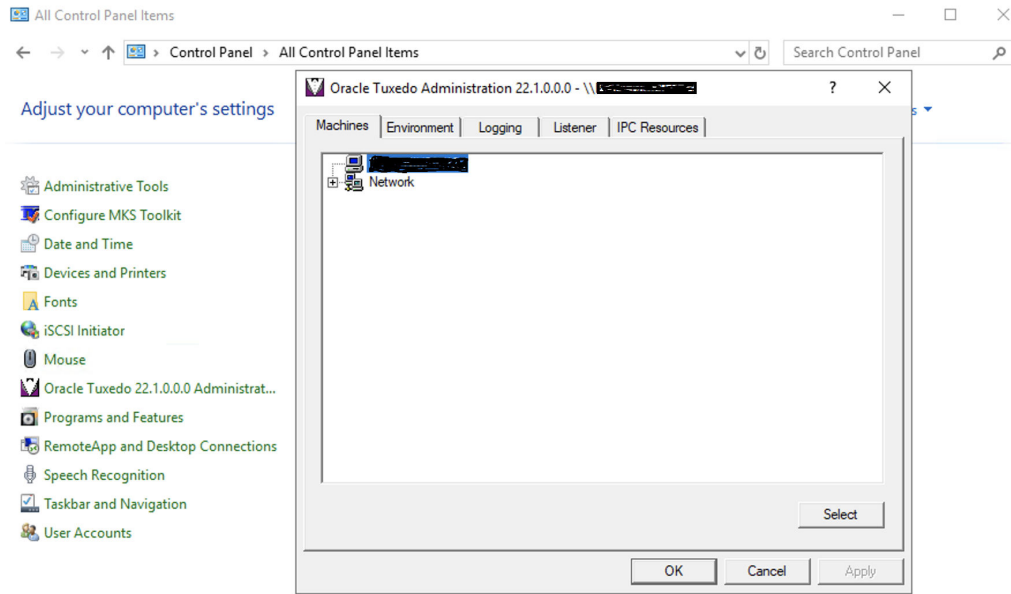
### Windows Service Change

If Oracle Tuxedo 22c Release 1 is installed successfully, then Windows Service ORACLE ProcMGR V22.1.0.0.0 and TUXEDO 22.1.0.0.0 Listener on Port 3050 are installed.

### Windows Control Panel Change

If Oracle Tuxedo 22c Release 1 (22.1.0.0.0) is installed successfully, then Oracle Tuxedo 22.1.0.0.0 Control Panel is installed as below:

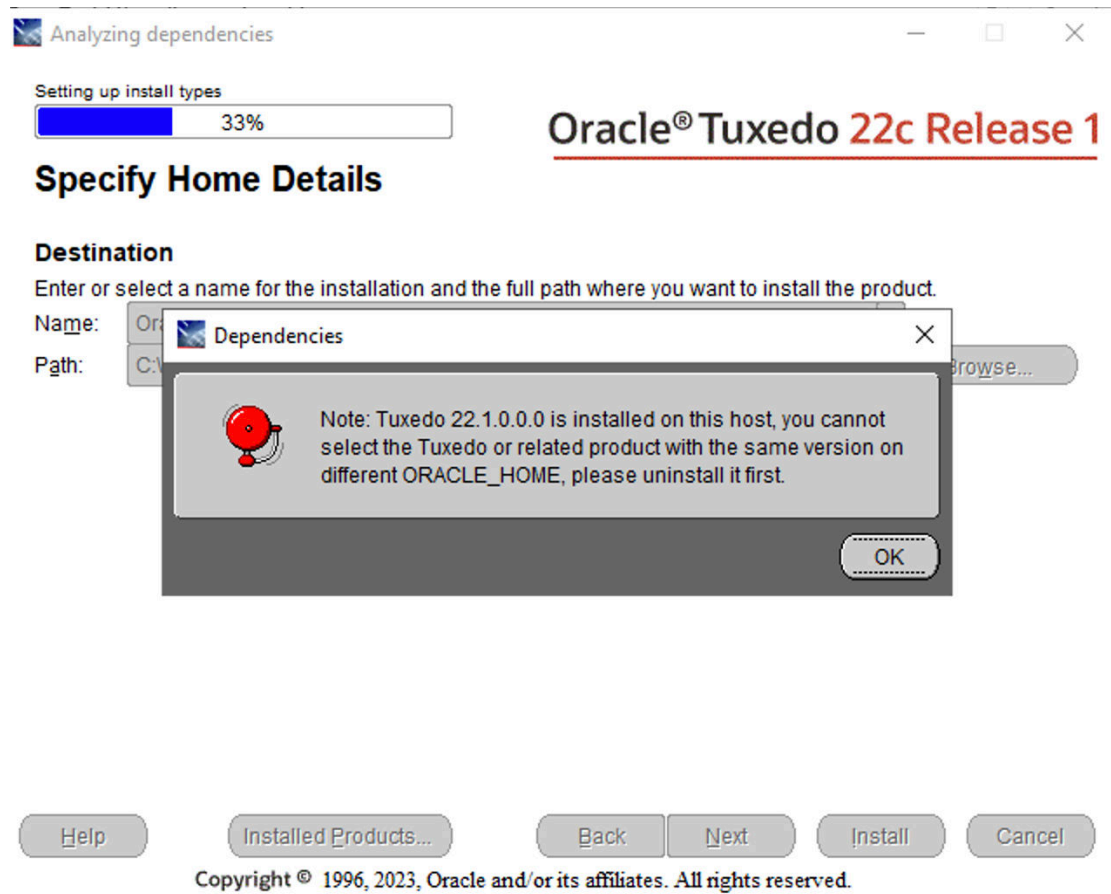
**Figure 2-1 Control Panel Change**



**Earlier Installation of the Same Version Detected**

If your installation detects an earlier installation of Oracle Tuxedo 22.1.0.0.0 in your system, then the following error message is displayed:

Figure 2-2 Earlier Version Detected Error Message



### Developer Key

The Developer key stores product information, including the major and minor version numbers of the release, and user and company names.

### Environment Key

The Environment key stores the locations referenced by the Oracle Tuxedo environment variables set on your Windows system. It also stores other values such as IPC resource settings.

### Security Key

The Security key holds the access permissions for Oracle Tuxedo processes and services. The following permissions are mandatory:

- Any user who runs `tlisten(1)` must have read access permissions.
- The account under which the Oracle ProcMGR service is running must have read access permissions.

Oracle recommends that the Administrator have full control permissions.

## 2.9 What Do I Do Next?

To configure your Oracle Tuxedo software, verify that your software is installed correctly, or to uninstall Oracle Tuxedo software.



### See Also:

[Performing Post-Installation Tasks](#)

# 3

## Installing Oracle Tuxedo Using Console-Mode Installation

The following sections describe how to install Oracle Tuxedo using console-mode installation:

- [What Is Console-Mode Installation?](#)
- [Starting Console-Mode Installation on a UNIX System](#)
- [Running Console-Mode Installation](#)
- [Assigning File Ownership on a UNIX System](#)
- [What Do I Do Next?](#)

### 3.1 What Is Console-Mode Installation?

The Oracle Tuxedo installer provides a console installation mode without GUI requirement. Console-mode installation provides a console interactive interface, and then calls the OUI silent-mode to install the product. Console-mode installation supports different platforms, for more information see [Platform Datasheet](#)

### 3.2 Starting Console-Mode Installation on a UNIX System

To start the Console-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 22c Release Platform Data Sheets](#).
2. Log in to the UNIX system as the Oracle Tuxedo administrator.
3. Ensure that you have enough free space for the Oracle Tuxedo installation.

For disk space requirements, see [Oracle Tuxedo 22c Release Platform Data Sheets](#). For important disk space allocation information, see [File and Database Management and Disk Space Allocation](#).

4. Install Oracle Tuxedo by downloading from the Oracle Web site:
  - a. Go to [Download Oracle Tuxedo](#), and download the Oracle Tuxedo 22c Release 1 (21.1.0.0) installation file specific to your platform.
  - b. Go to the directory where you downloaded the installer, unzip the installer file, and then go to the `Disk1/install` directory, set up environment variable `JAVA_HOME` to `JRE` or `JDK`, on the command line input `./runInstaller.sh -console` to invoke the installation procedure.

 **Note:**

Java version should be 1.8 or higher. JAVA\_HOME and Java version are checked. If JAVA\_HOME is not set or the java version is less than 1.8, installation is aborted.

Other options are not supported with `-console` (such as `-invPtrLoc`).

5. Proceed to [Running Console-Mode Installation](#).

## 3.3 Running Console-Mode Installation

The Oracle Tuxedo installer program prompts you to enter specific information about your system and configuration.

**Table 3-1 Instructions**


Window	Action
Specify Inventory directory (Unix only)	<p>On UNIX operating systems, this screen appears if this is the first time you are installing any Oracle product on this host. Specify the location where you want to create your central inventory. Ensure that the operating system group name selected on this screen has write permissions to the central inventory location.</p> <pre> Choose Inventory ----- -&gt;1- Central Inventory 2- Private Inventory 0- Go back Enter a number:1 /etc/oraInst.loc exists                     </pre> <div data-bbox="776 1339 1370 1516" style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; margin-top: 20px;"> <p> <b>Note:</b></p> <p>Entering "0" returns you to the Introduction screen.</p> </div>

Table 3-1 (Cont.) Instructions


Window	Action
	<p>If you select "private inventory", input your inventory pointer file as follows:</p> <pre>Choose Inventory ----- 1- Central Inventory -&gt;2- Private Inventory 0- Go back Enter a number: 2</pre> <p>Enter your private inventory pointer file: /nfs/users/beadev/tmp/oraInventory/oraInst.loc.example0</p> <p>If the input pointer file does not exist, then you are prompted to input the inventory directory and a group to create it:</p> <pre>-----/nfs/users/beadev/tmp/oraInventory/ oraInst.loc.example0does not exist, please create it. Enter private inventory directory: /nfs/users/beadev/tmp/ oraInventory Enter inst_group:dba -----</pre> <div data-bbox="776 993 1365 1444" style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; margin-top: 20px;"> <p> <b>Note:</b></p> <p>Inventory pointer file should be a legal file, which means: this file exists and you have the privilege to read, or this file doesn't exist but the parent directory of this file is writable.</p> <p>Private inventory directory should be a legal directory, which means this directory exists and you have the privileges to write, or this directory doesn't exist but the parent directory of this directory is writable.</p> <p>Inst_group should not be null.</p> </div>



Table 3-1 (Cont.) Instructions


Window	Action
Specify Oracle Home	<pre> Choose Oracle Home ----- -&gt;1- Create new Oracle Home 2- Use existing Oracle Home 0- Go back Enter a number: 1 Enter ORACLE_HOME: /home/beadev/OraHome2015 Enter ORACLE_HOME_NAME: OraHome2015 ----- </pre> <div data-bbox="776 741 1372 1108" style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> <b>Note:</b></p> <p>Entering "0" returns you to the Choose Inventory screen.</p> <p>If you enter a number 1, please specify ORACLE_HOME and ORACLE_HOME_NAME.</p> <p>Ensure:</p> <ul style="list-style-type: none"> <li>• ORACLE_HOME does not exist, and the parent directory must be writable.</li> <li>• ORACLE_HOME_NAME must not be null.</li> </ul> </div> <p>If you enter 2, the existing Oracle Home list is displayed. You choose a number from the list. Ensure that the selected Oracle Home is writable and if it has installed Tuxedo, it must be the same as the installing Tuxedo version. Otherwise, you must choose another Oracle Home.</p>

Table 3-1 (Cont.) Instructions




Window	Action
<p>Choose Install Set</p>	<pre>Choose Install Set ----- Please choose the Install Set to be installed by this installer. -&gt;1- Full Install 2- Server Install 3- Client Install 0- Go back Enter a number: 1 -----</pre> <div data-bbox="776 772 1364 1087" style="border: 1px solid #0070C0; background-color: #E6F2FF; padding: 10px; margin-top: 20px;"> <p> <b>Note:</b></p> <p>"0" returns you to the Choose Oracle Home screen.</p> <p>Custom Install is only supported in GUI mode; it is not supported in Console mode.</p> </div>
<p>Confirm enable TSAM plus agent</p> <p>This screen appears if the previously selected bundles include TSAM plus agent component. Enable it to use TSAM plus manager monitor and manage Tuxedo.</p>	<pre>Confirm Enable TSAM Plus Agent ----- Would you like to enable TSAM(Tuxedo System and Applications Monitor) Plus Agent? -&gt;1- Yes 2- No 0- Go back Enter a number: 1 -----</pre> <div data-bbox="776 1533 1364 1711" style="border: 1px solid #0070C0; background-color: #E6F2FF; padding: 10px; margin-top: 20px;"> <p> <b>Note:</b></p> <p>"0" returns you to the Choose Install Type screen.</p> </div>

Table 3-1 (Cont.) Instructions

Window	Action
<p>SSL Support Choice</p> <p>This window appears only if you have installed Server, ATMI and/or CORBA Client install type components.</p>	<pre> SSL Support Choice ----- Would you like to Support SSL -&gt;1- Yes 2- No 0- Go back Enter a number: 2 -----  If 1, you should choose to configure LDAP for SSL support. </pre> <div data-bbox="776 741 1370 915" style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>"0" returns you to the Confirm Enable TSAM Plus Agent screen.</p> </div>
<p>LDAP Support Choice</p>	<pre> ----- LDAP Support Choice ----- Would you like to configure LDAP for SSL Support? -&gt;1- Yes 2- No 0- Go back Enter a number: 1 -----  Enter Your LDAP Settings For SSL Support -----  Input LDAP Service Name: a Input LDAP PortID: b Input LDAP BaseObject: c LDAP Filter File: Press &lt;Enter&gt; to accept the default: /home/xpxu/ OraHome001/udataobj/security/bea_ldap_filter.dat or enter your own: ----- </pre> <p>If 1, then input the required arguments. Ensure that LDAP service Name, PortID, BaseObject must not be null and your own LDAP Filter File must be a legal file.</p>

**Table 3-1 (Cont.) Instructions**



Window	Action
<p>Select samples installation</p> <p>If the selected install feature sets include server components, this screen appears to confirm install the samples.</p>	<pre>Samples Installation Choice ----- Would you like to install Oracle Tuxedo Samples? 1- Yes -&gt;2- No 0- Go back Enter a number: 1 -----</pre> <div data-bbox="776 709 1364 884" style="border-left: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; margin-top: 20px;"> <p> <b>Note:</b></p> <p>"0" returns you to the SSL Installation Choice screen.</p> </div>
<p>Configure tlisten Choice</p> <p>This window appears only if you have installed the Server Install.</p>	<p>tlisten is required for MP/clustered mode configuration and Oracle TSAM Plus. If you require to configure tlisten, select "Yes", and then input the parameters for tlisten configuration on the screens that follow.</p>
<p>Configure tlisten password</p> <p>This window appears only if you select to configure tlisten. Ensure that Password must not be null.</p>	<pre>Tlisten Configuration Choice ----- Would you like to configure Oracle Tuxedo tlisten? -&gt;1- Yes 2- No 0- Go back Enter a number: 1 Enter Password: Verify Password: -----</pre> <p>Enter and re-enter a tlisten password.</p> <div data-bbox="776 1539 1364 1713" style="border-left: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; margin-top: 20px;"> <p> <b>Note:</b></p> <p>Enter "0" returns you to the screen of Samples Installation Choice.</p> </div>

Table 3-1 (Cont.) Instructions

Window	Action
Summary	<pre>Pre-Installation Summary ----- Install type: "Full Install" Install Folder: "/home/xpxu/OraHome2015" ----- Press "Enter" to start installation, Or Enter "0" to return to previous screen:</pre>
Installing	At this point, "Starting Oracle Universal Installer ..." is displayed along with the setup in progress.
Install End	Click Exit to exit the installation program.

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

## 3.4 Assigning File Ownership on a UNIX System

If you install the Oracle Tuxedo software on a UNIX system, then Oracle strongly recommends that you create a separate user account for the Oracle Tuxedo administrator and give ownership of the Oracle Tuxedo files to that account. To change ownership of the Oracle Tuxedo 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 administrator account, replace `adm_group` with the Oracle Tuxedo 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 software.

## 3.5 What Do I Do Next?

To configure your Oracle Tuxedo software, verify that your software is installed correctly, or to uninstall Oracle Tuxedo software.



### See Also:

[Performing Post-Installation Tasks](#)

# 4

## Installing Oracle Tuxedo Using Silent Installation

The following sections describe how to install Oracle Tuxedo using silent installation:

- [What Is Silent Installation?](#)
- [Before You Start](#)
- [Using Silent Installation: Main Steps](#)
- [Installing with a Response File](#)
- [UNIX Template File](#)
- [What Do I Do Next?](#)

### 4.1 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.

### 4.2 Before You Start

If you are upgrading from Oracle Tuxedo 12.2.2 or earlier, see [Upgrading the Oracle Tuxedo System to Tuxedo 22c Release 1\(22.1.0.0.0\)](#), and follow the instructions given there. Then return here to continue your Oracle Tuxedo installation.

### 4.3 Using Silent Installation: Main Steps

- [Creating a Response File](#)

#### 4.3.1 Creating a Response File

You can create a new response file, based on the installation options you select, by using the OUI record mode.

When you use record mode, OUI records the installation session to a response file. You specify the name of the response file on the command line. The recorded response file is generated immediately after the Summary page; you do not need to actually install your Oracle product to create the response file. That is, you can start the installation in Record mode and proceed through the installation options until you get to the Summary page. On the Summary Page, click Exit to stop the installation from proceeding with the installation. However, all the options you selected are saved in the resulting response file.

You can use the newly created response file to run identical installation sessions on other computers in your organization.

Record mode can be also used during a silent installation. In this case, the variable values specified in the original source response file are recorded to the new response file.

- [Using Record Mode](#)
- [Required Variables in Tuxedo Response File](#)

### 4.3.1.1 Using Record Mode

To record a new response file:

1. At the command prompt, use the `cd` command to change to the directory that contains the OUI executable file (`setup.bat` or `runInstaller.sh`) for your installation.
2. Enter the following command:  
On UNIX:

```
./runInstaller.sh -record -destinationFile <response_file_name>
```

Replace the `<response_file_name>` with the complete path for the new response file. For example:

On UNIX:

```
./runInstaller.sh -record -destinationFile /private/temp/  
install_oracle11g.rsp
```

3. Use the OUI user interface to select and record your installation options. When OUI displays the Summary page, you can either continue with the installation or exit.

OUI saves your new response file using the path and file name you specified on the command line.

### 4.3.1.2 Required Variables in Tuxedo Response File

The variables defined in this section are specific for Tuxedo installer.



#### Note:

For String type variables, you must surround the value using the double quote.

#### **ORACLE\_HOME**

The location where products are to be installed. You must enter a value for `ORACLE_HOME` for a complete silent installation.

#### **ORACLE\_HOME\_NAME**

The name of the current Oracle home. You must enter a value for `ORACLE_HOME_NAME` for a complete silent installation.

**INSTALL\_TYPE**

You can set the installation type variable to determine the installation type of the currently selected top level component. `Install_type` is selected from the following table:

Install Type	Description
Full Install	Installs all of the Tuxedo Server and Client files
Server Install	Installs the Tuxedo server files only.
Client Install	Installs the Tuxedo Client files only
Custom Install	Installs the components selected by customer

**CONFIG\_TLISTEN**

Configures the `tlisten` parameters during installation.

Configures encryption method, 0 is LLE, 1 is SSL.

If `CONFIG_TLISTEN=true`, `ENCRYPT_CHOICE` must be set.

**TLISTEN\_PASSWORD**

Configures `tlisten` password.

Configures `tlisten` port.

If `CONFIG_TLISTEN=true`, `TLISTEN_PORT` must be set.

 **Note:**

When using record mode (`setup.bat -record` or `runInstaller.sh -record command`), you must set password (with double quotation marks) in the generated response file. For example, `TLISTEN_PASSWORD="password"`.

**LDAP\_SUPPORT\_SSL**

Configures LDAP support for SSL.

**LDAP\_CONFIG**

Configures LDAP parameters (Service Name, PortID, BaseObject).

If `LDAP_SUPPORT_SSL= true`, `LDAP_CONFIG` must be set.

**LDAP\_FILTER\_FILE**

Configures LDAP filter file.

If `LDAP_SUPPORT_SSL= true`, `LDAP_CONFIG` must be set.

## 4.4 Installing with a Response File

Many Oracle software products provide tools and procedures for running OUI from the command line without displaying OUI screens or responding to questions during the installation.

This is called silent installation.



Instead of prompting you to select a series of installation options, OUI installs the software using a predefined set of options. These options are stored in a response file (.rsp).

 **Note:**

If you attempt to perform a silent installation on a UNIX computer where no Oracle products have been installed, an error messages displays. Before you can perform a silent installation in this situation, you must first run the `oralnstRoot.sh` script located in the `/oraInventory` directory. You must run this script with root privileges. This enables OUI to set up the Central Inventory on a clean host.

- [Specifying a Response File](#)
- [Optional Parameters When Specifying a Response File](#)

### 4.4.1 Specifying a Response File

To start OUI and specify the response file, enter the following command on the command line in the directory where the executable file is installed:

- On UNIX:

```
./runInstaller.sh -responseFile <filename> <optional_parameters>
```

 **Note:**

- To install Oracle Tuxedo in complete silent mode, you must specify `<optional_parameters> to-silent`.
- You must specify the complete `responseFilePath`. If you do not, OUI assumes the location is relative to the `oraparam.ini` file associated with the launched OUI.

For help on command line usage, enter the following on the command line in the directory where the executable file is stored:

- On UNIX:

```
./runInstaller.sh -help
```

### 4.4.2 Optional Parameters When Specifying a Response File

Optional parameters you can use with the `-responseFile` flag are:

- `-nowelcome` - Use the `-nowelcome` flag with the `-responseFile` flag to suppress the Welcome dialog that appears during installation.

- `-silent` - Use the `-silent` flag with the `-responseFile` flag to run OUI in complete silent mode. Note that the Welcome dialog is suppressed automatically.
- `-waitforcompletion` - Use the `-waitforcompletion` flag with `-silent` flag to wait for completion instead of spawning the java engine and exiting.
- In a file named `silentInstall<timestamp>.log` for hosts without an Oracle inventory. This file is generated in the `/tmp` directory on UNIX.
- In the inventory logs directory for hosts that already had an inventory.

**Note:**

Using the `-nowelcome` option with the `-silent` option is unnecessary since the Welcome screen does not appear when you use the `-silent` option.

## 4.5 UNIX Template File

This sample UNIX response file (`installer.properties`) in Listing 4-1 UNIX response File applies to a silent installation of Oracle Tuxedo 22c Release 1 (22.1.0.0.0).

```
#####
## Copyright (c) 1999, 2022 Oracle. All rights reserved. ##
## ##
## Specify values for the variables listed below to customize ##
## your installation. ##
## ##
## Each variable is associated with a comment. The comment ##
## identifies the variable type. ##
## ##
## Please specify the values in the following format: ##
## ##
## Type Example ##
## String "Sample Value" ##
## Boolean True or False ##
## Number 1000 ##
## StringList {"String value 1","String Value 2"} ##
## ##
## The values that are given as <Value Required> need to be ##
## specified for a silent installation to be successful. ##
## ##
## ##
## This response file is generated by Oracle Software ##
## Packager. ##
#####

RESPONSEFILE_VERSION=2.2.1.0.0

#-----
---
```

```
#Name : UNIX_GROUP_NAME
#Datatype : String
#Description: UNIX group to be set for the inventory directory. Valid
only in UNIX platforms.
#Example: UNIX_GROUP_NAME = "install"

#-----
-----
UNIX_GROUP_NAME="cr0208"

#-----
-----
#Name : FROM_LOCATION
#Datatype : String
#Description: Complete path to the products.xml.
#Example: FROM_LOCATION = "../stage/products.xml"

#-----
-----
FROM_LOCATION="/scratch/beadev/OUI_INSTALLER/TUX12cR2/stage/Disk1/
stage/products.xml"

#-----
-----
#Name : FROM_LOCATION_CD_LABEL
#Datatype : String
#Description: This variable should only be used in multi-CD
installations. It includes the label of the compact disk where the
file "products.xml" exists. The label can be found in the file
"disk.label" in the same directory as products.xml.
#Example: FROM_LOCATION_CD_LABEL = "CD Label"

#-----
-----
FROM_LOCATION_CD_LABEL=<Value Unspecified>

#-----
-----
#Name : ORACLE_HOME
#Datatype : String
#Description: Complete path of the Oracle Home.
#Example: ORACLE_HOME = "C:\OHOME1"

#-----
-----
ORACLE_HOME="/scratch/beadev/OraHome_5"

#-----
-----
#Name : ORACLE_BASE
#Datatype : String
```

```
#Description: Complete path of the Oracle Base.
#Example: ORACLE_BASE = "C:\app"

#-----
---
ORACLE_BASE=<Value Unspecified>

#-----
---
#Name : ORACLE_HOME_NAME
#Datatype : String
#Description: Oracle Home Name. Used in creating folders and services.
#Example: ORACLE_HOME_NAME = "OHOME1"

#-----
---
ORACLE_HOME_NAME="OracleHome5"

#-----
---
#Name : SHOW_WELCOME_PAGE
#Datatype : Boolean
#Description: Set to true if the Welcome page in OUI needs to be shown.
#Example: SHOW_WELCOME_PAGE = false

#-----
---
SHOW_WELCOME_PAGE=true

#-----
---
#Name : SHOW_CUSTOM_TREE_PAGE
#Datatype : Boolean
#Description: Set to true if the custom tree page in OUI needs to be shown.
#Use this page to select or de-select dependencies. This page appears only
in a custom install type.
#Example: SHOW_CUSTOM_TREE_PAGE = false

#-----
---
SHOW_CUSTOM_TREE_PAGE=true

#-----
---
#Name : SHOW_COMPONENT_LOCATIONS_PAGE
#Datatype : Boolean
#Description: Set to true if the component locations page in OUI needs to
be shown.
#This page only appears if there are products whose installed directory can
be changed.
#If you set this to false you will prevent the user from being able to
```

```
specify alternate directories.  
#Example: SHOW_COMPONENT_LOCATIONS_PAGE = false  
  
#-----  
-----  
SHOW_COMPONENT_LOCATIONS_PAGE=true  
  
#-----  
-----  
#Name : SHOW_SUMMARY_PAGE  
#Datatype : Boolean  
#Description: Set to true if the summary page in OUI needs to be  
shown.  
#The summary page shows the list of components that will be installed  
in this session.  
#Example: SHOW_SUMMARY_PAGE = true  
  
#-----  
-----  
SHOW_SUMMARY_PAGE=true  
  
#-----  
-----  
#Name : SHOW_INSTALL_PROGRESS_PAGE  
#Datatype : Boolean  
#Description: Set to true if the install progress page in OUI needs  
to be shown.  
#This page shows the current status in the installation. The current  
status includes the product being installed and the file being copied.  
#Example: SHOW_INSTALL_PROGRESS_PAGE = true  
  
#-----  
-----  
SHOW_INSTALL_PROGRESS_PAGE=true  
  
#-----  
-----  
#Name : SHOW_REQUIRED_CONFIG_TOOL_PAGE  
#Datatype : Boolean  
#Description: Set to true if the required config assistants page in  
OUI needs to be shown.  
#This page shows the list of required configuration assistants that  
are part of this installation.  
#It shows the status of each assistant, including any failures with  
detailed information on why it failed.  
#Example: SHOW_REQUIRED_CONFIG_TOOL_PAGE = true  
  
#-----  
-----  
SHOW_REQUIRED_CONFIG_TOOL_PAGE=true
```

```
#-----  
---  
#Name : SHOW_CONFIG_TOOL_PAGE  
#Datatype : Boolean  
#Description: Set to true if the config assistants page in OUI needs to be  
shown.  
#This page shows the list of configuration assistants that are part of this  
installation and are configured to launch automatically.  
#It shows the status of each assistant, including any failures with  
detailed information on why it failed.  
#Example: SHOW_CONFIG_TOOL_PAGE = true  
  
#-----  
---  
SHOW_CONFIG_TOOL_PAGE=true  
  
#-----  
---  
#Name : SHOW_RELEASE_NOTES  
#Datatype : Boolean  
#Description: Set to true if the release notes of this installation need to  
be shown at the end of installation.  
#This dialog is launchable from the End of Installation page and shows the  
list of release notes available for the products just installed.  
# This also requires the variable SHOW_END_SESSION_PAGE variable to be set  
to true.  
#Example: SHOW_RELEASE_NOTES = true  
  
#-----  
---  
SHOW_RELEASE_NOTES=true  
  
#-----  
---  
#Name : SHOW_ROOTSH_CONFIRMATION  
#Datatype : Boolean  
#Description: Set to true if the Confirmation dialog asking to run the  
root.sh script in OUI needs to be shown.  
#Valid only for UNIX platforms.  
#Example: SHOW_ROOTSH_CONFIRMATION = true  
  
#-----  
---  
SHOW_ROOTSH_CONFIRMATION=true  
  
#-----  
---  
#Name : SHOW_END_SESSION_PAGE  
#Datatype : Boolean  
#Description: Set to true if the end of session page in OUI needs to be  
shown.  
#This page shows if the installation is successful or not.
```

```
#Example: SHOW_END_SESSION_PAGE = true

#-----
SHOW_END_SESSION_PAGE=true

#-----
#Name : SHOW_EXIT_CONFIRMATION
#Datatype : Boolean
#Description: Set to true if the confirmation when exiting OUI needs
to be shown.
#Example: SHOW_EXIT_CONFIRMATION = true

#-----
SHOW_EXIT_CONFIRMATION=true

#-----
#Name : NEXT_SESSION
#Datatype : Boolean
#Description: Set to true to allow users to go back to the File
Locations page for another installation. This flag also needs to be
set to true in order to process another response file (see
NEXT_SESSION_RESPONSE).
#Example: NEXT_SESSION = true

#-----
NEXT_SESSION=false

#-----
#Name : NEXT_SESSION_ON_FAIL
#Datatype : Boolean
#Description: Set to true to allow users to invoke another session
even if current install session has failed. This flag is only relevant
if NEXT_SESSION is set to true.
#Example: NEXT_SESSION_ON_FAIL = true

#-----
NEXT_SESSION_ON_FAIL=true

#-----
#Name : NEXT_SESSION_RESPONSE
#Datatype : String
#Description: Set to true to allow users to go back to the File
Locations page for another installation. This flag also needs to be
```

```
set to true in order to process another response file (see
NEXT_SESSION_RESPONSE).
#Example: NEXT_SESSION_RESPONSE = "nextinstall.rsp"

#-----
---
NEXT_SESSION_RESPONSE=<Value Unspecified>

#-----
---
#Name : DEINSTALL_LIST
#Datatype : StringList
#Description: List of components to be deinstalled during a deinstall
session.
#Example: DEINSTALL_LIST = {"Tuxedo","12.2.2.0.0"}

#-----
---
DEINSTALL_LIST={"Tuxedo","12.2.2.0.0"}

#-----
---
#Name : SHOW_DEINSTALL_CONFIRMATION
#Datatype : Boolean
#Description: Set to true if deinstall confirmation is needed during a
deinstall session.
#Example: SHOW_DEINSTALL_CONFIRMATION = true

#-----
---
SHOW_DEINSTALL_CONFIRMATION=true

#-----
---
#Name : SHOW_DEINSTALL_PROGRESS
#Datatype : Boolean
#Description: Set to true if deinstall progress is needed during a
deinstall session.
#Example: SHOW_DEINSTALL_PROGRESS = true

#-----
---
SHOW_DEINSTALL_PROGRESS=true

#-----
---
#Name : CLUSTER_NODES
#Datatype : StringList
#Description: This variable represents the cluster node names selected by
the user for installation.
#Example: CLUSTER_NODES = {"node1","node2"}
```



```
#-----  
-----  
CLUSTER_NODES={}  
  
#-----  
-----  
#Name : ACCEPT_LICENSE_AGREEMENT  
#Datatype : Boolean  
#Description: By setting this variable to true, you are accepting the  
license agreement. This variable is used only for silent  
installations.  
#Example: ACCEPT_LICENSE_AGREEMENT = true  
  
#-----  
-----  
ACCEPT_LICENSE_AGREEMENT=false  
  
#-----  
-----  
#Name : METALINK_LOCATION  
#Datatype : String  
#Description: This variable represents the Oracle metalink location.  
  
#-----  
-----  
METALINK_LOCATION=<Value Unspecified>  
  
#-----  
-----  
#Name : METALINK_USERNAME  
#Datatype : String  
#Description: This variable represents the Oracle metalink user name.  
  
#-----  
-----  
METALINK_USERNAME=<Value Unspecified>  
  
#-----  
-----  
#Name : METALINK_PASSWORD  
#Datatype : String  
#Description: This variable represents the corresponding Oracle  
metalink password.  
  
#-----  
-----  
METALINK_PASSWORD=<Value Unspecified>  
  
#-----  
-----
```

```
-----
#Name : PROXY_HOST
#Datatype : String
#Description: The proxy host used to connect to Oracle metalink.
#Example: PROXY_HOST =

#-----
---
PROXY_HOST=""

#-----
---
#Name : PROXY_PORT
#Datatype : String
#Description: The proxy port used to connect to Oracle metalink.
#Example: PROXY_PORT =

#-----
---
PROXY_PORT=""

#-----
---
#Name : PROXY_REALM
#Datatype : String
#Description: The realm for the proxy used to connect to Oracle metalink.
#Example: PROXY_REALM =

#-----
---
PROXY_REALM=<Value Unspecified>

#-----
---
#Name : PROXY_USER
#Datatype : String
#Description: The username for the proxy used to connect to Oracle
metalink.
#Example: PROXY_USER =

#-----
---
PROXY_USER=""

#-----
---
#Name : PROXY_PWD
#Datatype : String
#Description: The password for the proxy used to connect to Oracle
metalink.
#Example: PROXY_PWD =
```

```
#-----  
-----  
    PROXY_PWD=<Value Unspecified>  
  
#-----  
-----  
    #Name : DONT_PROXY_FOR  
    #Datatype : String  
    #Description: The dont proxy for list.  
    #Example: DONT_PROXY_FOR =  
  
#-----  
-----  
    DONT_PROXY_FOR=<Value Unspecified>  
  
#-----  
-----  
    #Name : TOPLEVEL_COMPONENT  
    #Datatype : StringList  
    #Description: The top level component to be installed in the current  
    session.  
    #Example: TOPLEVEL_COMPONENT = {"Tuxedo", "22.1.0.0.0"}  
  
#-----  
-----  
    TOPLEVEL_COMPONENT={"Tuxedo", "22.1.0.0.0"}  
  
#-----  
-----  
    #Name : SHOW_SPLASH_SCREEN  
    #Datatype : Boolean  
    #Description: Set to true if the initial splash screen in OUI needs  
    to be shown.  
    #Example: SHOW_SPLASH_SCREEN = true  
  
#-----  
-----  
    SHOW_SPLASH_SCREEN=true  
  
#-----  
-----  
    #Name : SELECTED_LANGUAGES  
    #Datatype : StringList  
    #Description: Languages in which the components will be installed.  
    #Component : Tuxedo  
  
#-----  
-----  
    SELECTED_LANGUAGES={"en"}
```

```
#-----  
---  
#Name : COMPONENT_LANGUAGES  
#Datatype : StringList  
#Description: Languages supported by this component.List of supported  
languages : {"en","ja","zh_CN"}  
#Component : Tuxedo  
  
#-----  
---  
  
COMPONENT_LANGUAGES={"en"}  
  
#-----  
---  
#Name : INSTALL_TYPE  
#Datatype : String  
#Description: Installation type of the component.  
#Component : Tuxedo  
  
#-----  
---  
  
INSTALL_TYPE="Full Install"  
  
#-----  
---  
#Name : ENABLE_TSAM_AGENT  
#Datatype : Boolean  
#Description: To enable TSAM(Tuxedo System and Applications Monitor) Plus  
Agent for Tuxedo server, set this variable to true.  
#Component : tsamAgent  
  
#-----  
---  
  
ENABLE_TSAM_AGENT=true  
  
#-----  
---  
#Name : CONFIG_TLISTEN  
#Datatype : Boolean  
#Description: To configure the tlisten paramters at installation phase, set  
this variable to true.  
#Component : tuxedoServerBase  
  
#-----  
---  
  
CONFIG_TLISTEN=true  
  
#-----  
---
```

```
#Name : MIN_CRYPT_BITS_CHOOSE
#Datatype : Number
#Description: This variable is for configure Min Encryption Bits,
#0 means min encryption bit set to 0 ,
#1 means min encryption bit set to 56 ,
#2 means min encryption bit set to 128,
#3 means min encryption bit set to 256.
#Component : tuxedoServerBase

#-----
-----

MIN_CRYPT_BITS_CHOOSE=0

#-----
-----

#Name : MAX_CRYPT_BITS_CHOOSE
#Datatype : Number
#Description: This variable is for configure Max Encryption Bits,
#0 means max encryption bit set to 0 ,
#1 means max encryption bit set to 56 ,
#2 means max encryption bit set to 128,
#3 means max encryption bit set to 256.
#Component : tuxedoServerBase

#-----
-----

MAX_CRYPT_BITS_CHOOSE=3

#-----
-----

#Name : LDAP_SUPPORT_SSL
#Datatype : Boolean
#Description: This variable is for configure LDAP support for SSL
#Component : tuxedoServerBase

#-----
-----

LDAP_SUPPORT_SSL=true

#-----
-----

#Name : INSTALL_SAMPLES
#Datatype : Boolean
#Description: This variable is for configure whether need to install
Tuxedo samples
#Component : tuxedoServerBase

#-----
-----
```

```
INSTALL_SAMPLES=true

#-----
---
#Name : ENCRYPT_CHOICE
#Datatype : Number
#Description: This variable is for configure encrypt method, 0 is LLE, 1 is
SSL.
#Component : tuxedoServerBase

#-----
---

ENCRYPT_CHOICE=0

#-----
---
#Name : LDAP_FILTER_FILE
#Datatype : String
#Description: This variable is for configure LDAP filter file
#Component : tuxedoServerBase

#-----
---

LDAP_FILTER_FILE="/scratch/beadev/OraHome_5/tuxedo22.1.0.0/udataobj/
security/bea_ldap_filter.dat"

#-----
---
#Name : LDAP_CONFIG
#Datatype : StringList
#Description: This variable is for configure LDAP parameters(Service Name,
PortID, BaseObject)
#Component : tuxedoServerBase

#-----
---

LDAP_CONFIG={"ldap service name","ldap portid","ldap base object"}

### 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
```

- [Listing 4-1 UNIX response File](#)

## 4.5.1 Listing 4-1 UNIX response File

```
#####
## Copyright (c) 1999, 2022 Oracle. All rights reserved. ##
## ##
## Specify values for the variables listed below to customize ##
## your installation. ##
## ##
## Each variable is associated with a comment. The comment ##
## identifies the variable type. ##
## ##
## Please specify the values in the following format: ##
## ##
## Type Example ##
## String "Sample Value" ##
## Boolean True or False ##
## Number 1000 ##
## StringList {"String value 1","String Value 2"} ##
## ##
## The values that are given as <Value Required> need to be ##
## specified for a silent installation to be successful. ##
## ##
## ##
## This response file is generated by Oracle Software ##
## Packager. ##
#####
```

```
RESPONSEFILE_VERSION=2.2.1.0.0
```

```
#-----
-----
#Name : UNIX_GROUP_NAME
#Datatype : String
#Description: UNIX group to be set for the inventory directory. Valid
only in UNIX platforms.
#Example: UNIX_GROUP_NAME = "install"

#-----
-----
UNIX_GROUP_NAME="cr0208"

#-----
-----
#Name : FROM_LOCATION
#Datatype : String
#Description: Complete path to the products.xml.
#Example: FROM_LOCATION = "../stage/products.xml"

#-----
```

```
-----
FROM_LOCATION="/scratch/beadev/OUI_INSTALLER/TUX12cR2/stage/Disk1/stage/
products.xml"

#-----
---
#Name : FROM_LOCATION_CD_LABEL
#Datatype : String
#Description: This variable should only be used in multi-CD installations.
It includes the label of the compact disk where the file "products.xml"
exists. The label can be found in the file "disk.label" in the same
directory as products.xml.
#Example: FROM_LOCATION_CD_LABEL = "CD Label"

#-----
---
FROM_LOCATION_CD_LABEL=<Value Unspecified>

#-----
---
#Name : ORACLE_HOME
#Datatype : String
#Description: Complete path of the Oracle Home.
#Example: ORACLE_HOME = "C:\OHOME1"

#-----
---
ORACLE_HOME="/scratch/beadev/OraHome_5"

#-----
---
#Name : ORACLE_BASE
#Datatype : String
#Description: Complete path of the Oracle Base.
#Example: ORACLE_BASE = "C:\app"

#-----
---
ORACLE_BASE=<Value Unspecified>

#-----
---
#Name : ORACLE_HOME_NAME
#Datatype : String
#Description: Oracle Home Name. Used in creating folders and services.
#Example: ORACLE_HOME_NAME = "OHOME1"

#-----
---
ORACLE_HOME_NAME="OracleHome5"
```



```
#-----  
-----  
#Name : SHOW_WELCOME_PAGE  
#Datatype : Boolean  
#Description: Set to true if the Welcome page in OUI needs to be  
shown.  
#Example: SHOW_WELCOME_PAGE = false  
  
#-----  
-----  
SHOW_WELCOME_PAGE=true  
  
#-----  
-----  
#Name : SHOW_CUSTOM_TREE_PAGE  
#Datatype : Boolean  
#Description: Set to true if the custom tree page in OUI needs to be  
shown.  
#Use this page to select or de-select dependencies. This page appears  
only in a custom install type.  
#Example: SHOW_CUSTOM_TREE_PAGE = false  
  
#-----  
-----  
SHOW_CUSTOM_TREE_PAGE=true  
  
#-----  
-----  
#Name : SHOW_COMPONENT_LOCATIONS_PAGE  
#Datatype : Boolean  
#Description: Set to true if the component locations page in OUI  
needs to be shown.  
#This page only appears if there are products whose installed  
directory can be changed.  
#If you set this to false you will prevent the user from being able  
to specify alternate directories.  
#Example: SHOW_COMPONENT_LOCATIONS_PAGE = false  
  
#-----  
-----  
SHOW_COMPONENT_LOCATIONS_PAGE=true  
  
#-----  
-----  
#Name : SHOW_SUMMARY_PAGE  
#Datatype : Boolean  
#Description: Set to true if the summary page in OUI needs to be  
shown.  
#The summary page shows the list of components that will be installed  
in this session.  
#Example: SHOW_SUMMARY_PAGE = true
```

```
#-----  
---  
    SHOW_SUMMARY_PAGE=true  
  
#-----  
---  
    #Name : SHOW_INSTALL_PROGRESS_PAGE  
    #Datatype : Boolean  
    #Description: Set to true if the install progress page in OUI needs to be  
shown.  
    #This page shows the current status in the installation. The current status  
includes the product being installed and the file being copied.  
    #Example: SHOW_INSTALL_PROGRESS_PAGE = true  
  
#-----  
---  
    SHOW_INSTALL_PROGRESS_PAGE=true  
  
#-----  
---  
    #Name : SHOW_REQUIRED_CONFIG_TOOL_PAGE  
    #Datatype : Boolean  
    #Description: Set to true if the required config assistants page in OUI  
needs to be shown.  
    #This page shows the list of required configuration assistants that are  
part of this installation.  
    #It shows the status of each assistant, including any failures with  
detailed information on why it failed.  
    #Example: SHOW_REQUIRED_CONFIG_TOOL_PAGE = true  
  
#-----  
---  
    SHOW_REQUIRED_CONFIG_TOOL_PAGE=true  
  
#-----  
---  
    #Name : SHOW_CONFIG_TOOL_PAGE  
    #Datatype : Boolean  
    #Description: Set to true if the config assistants page in OUI needs to be  
shown.  
    #This page shows the list of configuration assistants that are part of this  
installation and are configured to launch automatically.  
    #It shows the status of each assistant, including any failures with  
detailed information on why it failed.  
    #Example: SHOW_CONFIG_TOOL_PAGE = true  
  
#-----  
---  
    SHOW_CONFIG_TOOL_PAGE=true
```

```
#-----  
-----  
#Name : SHOW_RELEASE_NOTES  
#Datatype : Boolean  
#Description: Set to true if the release notes of this installation  
need to be shown at the end of installation.  
#This dialog is launchable from the End of Installation page and  
shows the list of release notes available for the products just  
installed.  
# This also requires the variable SHOW_END_SESSION_PAGE variable to  
be set to true.  
#Example: SHOW_RELEASE_NOTES = true  
  
#-----  
-----  
SHOW_RELEASE_NOTES=true  
  
#-----  
-----  
#Name : SHOW_ROOTSH_CONFIRMATION  
#Datatype : Boolean  
#Description: Set to true if the Confirmation dialog asking to run  
the root.sh script in OUI needs to be shown.  
#Valid only for UNIX platforms.  
#Example: SHOW_ROOTSH_CONFIRMATION = true  
  
#-----  
-----  
SHOW_ROOTSH_CONFIRMATION=true  
  
#-----  
-----  
#Name : SHOW_END_SESSION_PAGE  
#Datatype : Boolean  
#Description: Set to true if the end of session page in OUI needs to  
be shown.  
#This page shows if the installation is successful or not.  
#Example: SHOW_END_SESSION_PAGE = true  
  
#-----  
-----  
SHOW_END_SESSION_PAGE=true  
  
#-----  
-----  
#Name : SHOW_EXIT_CONFIRMATION  
#Datatype : Boolean  
#Description: Set to true if the confirmation when exiting OUI needs  
to be shown.  
#Example: SHOW_EXIT_CONFIRMATION = true  
  
#-----
```

```
-----  
SHOW_EXIT_CONFIRMATION=true  
  
#-----  
---  
#Name : NEXT_SESSION  
#Datatype : Boolean  
#Description: Set to true to allow users to go back to the File Locations  
page for another installation. This flag also needs to be set to true in  
order to process another response file (see NEXT_SESSION_RESPONSE).  
#Example: NEXT_SESSION = true  
  
#-----  
---  
NEXT_SESSION=false  
  
#-----  
---  
#Name : NEXT_SESSION_ON_FAIL  
#Datatype : Boolean  
#Description: Set to true to allow users to invoke another session even if  
current install session has failed. This flag is only relevant if  
NEXT_SESSION is set to true.  
#Example: NEXT_SESSION_ON_FAIL = true  
  
#-----  
---  
NEXT_SESSION_ON_FAIL=true  
  
#-----  
---  
#Name : NEXT_SESSION_RESPONSE  
#Datatype : String  
#Description: Set to true to allow users to go back to the File Locations  
page for another installation. This flag also needs to be set to true in  
order to process another response file (see NEXT_SESSION_RESPONSE).  
#Example: NEXT_SESSION_RESPONSE = "nextinstall.rsp"  
  
#-----  
---  
NEXT_SESSION_RESPONSE=<Value Unspecified>  
  
#-----  
---  
#Name : DEINSTALL_LIST  
#Datatype : StringList  
#Description: List of components to be deinstalled during a deinstall  
session.  
#Example: DEINSTALL_LIST = {"Tuxedo","21.1.1.0.0"}  
  
#-----
```

```
---
DEINSTALL_LIST={"Tuxedo","21.1.1.0.0"}

#-----
#-----
#Name : SHOW_DEINSTALL_CONFIRMATION
#Datatype : Boolean
#Description: Set to true if deinstall confirmation is needed during a
deinstall session.
#Example: SHOW_DEINSTALL_CONFIRMATION = true

#-----
#-----
SHOW_DEINSTALL_CONFIRMATION=true

#-----
#-----
#Name : SHOW_DEINSTALL_PROGRESS
#Datatype : Boolean
#Description: Set to true if deinstall progress is needed during a
deinstall session.
#Example: SHOW_DEINSTALL_PROGRESS = true

#-----
#-----
SHOW_DEINSTALL_PROGRESS=true

#-----
#-----
#Name : CLUSTER_NODES
#Datatype : StringList
#Description: This variable represents the cluster node names
selected by the user for installation.
#Example: CLUSTER_NODES = {"node1","node2"}

#-----
#-----
CLUSTER_NODES={}

#-----
#-----
#Name : ACCEPT_LICENSE_AGREEMENT
#Datatype : Boolean
#Description: By setting this variable to true, you are accepting the
license agreement. This variable is used only for silent
installations.
#Example: ACCEPT_LICENSE_AGREEMENT = true

#-----
#-----
ACCEPT_LICENSE_AGREEMENT=false
```

```
#-----  
---  
#Name : METALINK_LOCATION  
#Datatype : String  
#Description: This variable represents the Oracle metalink location.  
  
#-----  
---  
METALINK_LOCATION=<Value Unspecified>  
  
#-----  
---  
#Name : METALINK_USERNAME  
#Datatype : String  
#Description: This variable represents the Oracle metalink user name.  
  
#-----  
---  
METALINK_USERNAME=<Value Unspecified>  
  
#-----  
---  
#Name : METALINK_PASSWORD  
#Datatype : String  
#Description: This variable represents the corresponding Oracle metalink  
password.  
  
#-----  
---  
METALINK_PASSWORD=<Value Unspecified>  
  
#-----  
---  
#Name : PROXY_HOST  
#Datatype : String  
#Description: The proxy host used to connect to Oracle metalink.  
#Example: PROXY_HOST =  
  
#-----  
---  
PROXY_HOST=""  
  
#-----  
---  
#Name : PROXY_PORT  
#Datatype : String  
#Description: The proxy port used to connect to Oracle metalink.  
#Example: PROXY_PORT =
```

```
#-----  
-----  
    PROXY_PORT=""  
  
#-----  
-----  
    #Name : PROXY_REALM  
    #Datatype : String  
    #Description: The realm for the proxy used to connect to Oracle  
    metalink.  
    #Example: PROXY_REALM =  
  
#-----  
-----  
    PROXY_REALM=<Value Unspecified>  
  
#-----  
-----  
    #Name : PROXY_USER  
    #Datatype : String  
    #Description: The username for the proxy used to connect to Oracle  
    metalink.  
    #Example: PROXY_USER =  
  
#-----  
-----  
    PROXY_USER=""  
  
#-----  
-----  
    #Name : PROXY_PWD  
    #Datatype : String  
    #Description: The password for the proxy used to connect to Oracle  
    metalink.  
    #Example: PROXY_PWD =  
  
#-----  
-----  
    PROXY_PWD=<Value Unspecified>  
  
#-----  
-----  
    #Name : DONT_PROXY_FOR  
    #Datatype : String  
    #Description: The dont proxy for list.  
    #Example: DONT_PROXY_FOR =  
  
#-----  
-----  
    DONT_PROXY_FOR=<Value Unspecified>
```

```
#-----  
---  
#Name : TOPLEVEL_COMPONENT  
#Datatype : StringList  
#Description: The top level component to be installed in the current  
session.  
#Example: TOPLEVEL_COMPONENT = {"Tuxedo","22.1.0.0.0"}  
  
#-----  
---  
TOPLEVEL_COMPONENT={"Tuxedo","22.1.0.0.0"}  
  
#-----  
---  
#Name : SHOW_SPLASH_SCREEN  
#Datatype : Boolean  
#Description: Set to true if the initial splash screen in OUI needs to be  
shown.  
#Example: SHOW_SPLASH_SCREEN = true  
  
#-----  
---  
SHOW_SPLASH_SCREEN=true  
  
#-----  
---  
#Name : SELECTED_LANGUAGES  
#Datatype : StringList  
#Description: Languages in which the components will be installed.  
#Component : Tuxedo  
  
#-----  
---  
SELECTED_LANGUAGES={"en"}  
  
#-----  
---  
#Name : COMPONENT_LANGUAGES  
#Datatype : StringList  
#Description: Languages supported by this component.List of supported  
languages : {"en","ja","zh_CN"}  
#Component : Tuxedo  
  
#-----  
---  
COMPONENT_LANGUAGES={"en"}  
  
#-----
```



```
---
#Name : INSTALL_TYPE
#Datatype : String
#Description: Installation type of the component.
#Component : Tuxedo

#-----
-----

INSTALL_TYPE="Full Install"

#-----
-----

#Name : ENABLE_TSAM_AGENT
#Datatype : Boolean
#Description: To enable TSAM(Tuxedo System and Applications Monitor)
Plus Agent for Tuxedo server, set this variable to true.
#Component : tsamAgent

#-----
-----

ENABLE_TSAM_AGENT=true

#-----
-----

#Name : CONFIG_TLISTEN
#Datatype : Boolean
#Description: To configure the tlisten paramters at installation
phase, set this variable to true.
#Component : tuxedoServerBase

#-----
-----

CONFIG_TLISTEN=true

#-----
-----

#Name : MIN_CRYPT_BITS_CHOOSE
#Datatype : Number
#Description: This variable is for configure Min Encryption Bits,
#0 means min encryption bit set to 0 ,
#1 means min encryption bit set to 56 ,
#2 means min encryption bit set to 128,
#3 means min encryption bit set to 256.
#Component : tuxedoServerBase

#-----
-----

MIN_CRYPT_BITS_CHOOSE=0

#-----
-----
```

```
#Name : MAX_CRYPT_BITS_CHOOSE
#Datatype : Number
#Description: This variable is for configure Max Encryption Bits,
#0 means max encryption bit set to 0 ,
#1 means max encryption bit set to 56 ,
#2 means max encryption bit set to 128,
#3 means max encryption bit set to 256.
#Component : tuxedoServerBase
```

```
#-----
---
```

```
MAX_CRYPT_BITS_CHOOSE=3
```

```
#-----
---
```

```
#Name : LDAP_SUPPORT_SSL
#Datatype : Boolean
#Description: This variable is for configure LDAP support for SSL
#Component : tuxedoServerBase
```

```
#-----
---
```

```
LDAP_SUPPORT_SSL=true
```

```
#-----
---
```

```
#Name : INSTALL_SAMPLES
#Datatype : Boolean
#Description: This variable is for configure whether need to install Tuxedo
samples
#Component : tuxedoServerBase
```

```
#-----
---
```

```
INSTALL_SAMPLES=true
```

```
#-----
---
```

```
#Name : ENCRYPT_CHOICE
#Datatype : Number
#Description: This variable is for configure encrypt method, 0 is LLE, 1 is
SSL.
#Component : tuxedoServerBase
```

```
#-----
---
```

```
ENCRYPT_CHOICE=0
```

```

#-----
#-----
#Name : LDAP_FILTER_FILE
#Datatype : String
#Description: This variable is for configure LDAP filter file
#Component : tuxedoServerBase

#-----
#-----

LDAP_FILTER_FILE="/scratch/beadev/OraHome_5/tuxedo22.1.0.0/udataobj/
security/bea_ldap_filter.dat"

#-----
#-----
#Name : LDAP_CONFIG
#Datatype : StringList
#Description: This variable is for configure LDAP parameters(Service
Name, PortID, BaseObject)
#Component : tuxedoServerBase

#-----
#-----

LDAP_CONFIG={"ldap service name","ldap portid","ldap base object"}

### 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

```

## 4.6 What Do I Do Next?

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

- [Assigning File Ownership on a UNIX System](#)

To configure your Oracle Tuxedo software, verify that your software is installed correctly, or to uninstall Oracle Tuxedo software, see [Performing Post-Installation Tasks](#).

# 5

## Upgrading the Oracle Tuxedo System to Tuxedo 22c Release 1(22.1.0.0.0)

The following sections provide procedures for upgrading your Oracle Tuxedo 12c Release 2 (12.2.2) to Oracle Tuxedo 22c using simple and hot upgrade procedures:

- [Upgrade Paths](#)
- [Preparing Your Machine for an Upgrade](#)
- [Selecting an Upgrade Procedure](#)
- [Backing Up Files](#)
- [Performing a Simple Upgrade](#)
- [Performing Additional Upgrade Steps to Windows Server](#)
- [Upgrading from Tuxedo Previous Releases to Oracle Tuxedo 22c Release 1 \(22.1.0.0.0\)](#)
- [Performing a Hot Upgrade](#)
- [Rebuilding an Application](#)
- [Installing Oracle Jolt 22c with Oracle WebLogic Server](#)

### 5.1 Upgrade Paths

The following figure illustrates the existing Oracle Tuxedo products that can be upgraded to Oracle Tuxedo 22c Release 1 (22.1.0.0.0):

**Figure 5-1 Upgrade Paths**



When you shut down the domain (application) targeted for the upgrade, you must shut down the domain and perform a *simple upgrade*. If you cannot shut down the domain targeted for the upgrade, perform a *hot upgrade*, that is, add the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) system software to the existing Oracle Tuxedo domain without shutting down the domain.

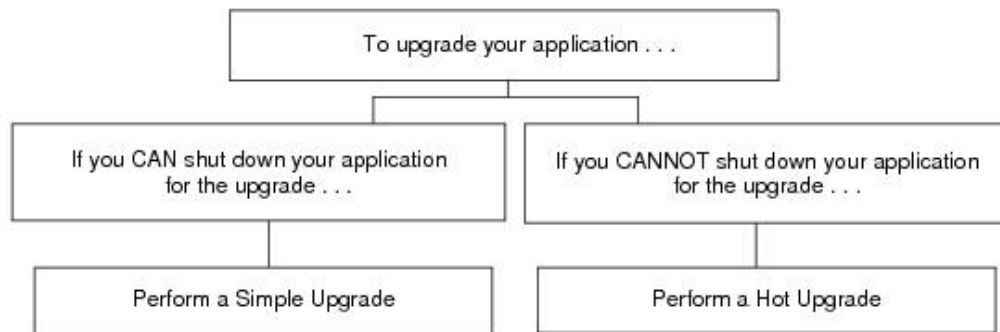
### 5.2 Preparing Your Machine for an Upgrade

Before you install the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) 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 22c Release Platform Data Sheets](#).

## 5.3 Selecting an Upgrade Procedure

If you are installing Oracle Tuxedo 22c Release 1 (22.1.0.0.0) on a platform that contains an earlier release of Oracle Tuxedo software, then make sure you read the following procedures carefully.



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](#).

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 22c Release 1 (22.1.0.0.0) without having to shut down your application:

- Versions 12.1.1.0.0, 12.1.3.0.0, and 12.2.2.0.0 of Oracle Tuxedo

You can add the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) software to existing Oracle Tuxedo domains without shutting down the existing applications, and you can also add new Oracle Tuxedo 22c Release 1 (22.1.0.0.0) application servers to those domains without shutting down the existing applications. For instructions on performing a hot upgrade, see [Performing a Hot Upgrade](#).

## 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 are overwritten when the new Oracle Tuxedo software is installed.

Follow these steps to avoid having to modify these files again:

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 `buildXAJIS`. It is located in the `tux_prod_dir/udataobj` directory, where `tux_prod_dir` is the directory in which you installed the Oracle Tuxedo.
2. Back up the `CATNAMES`, `jrepository`, and `jrly.config` files. The `CATNAMES` file is located in the `tux_prod_dir/locale` directory. The `jrepository` file is located in the `tux_prod_dir/udataobj/jolt/repository/` directory structure and the `jrly.config` file is located in the `tux_prod_dir/udataobj/jolt/relay/` directory structure.

3. Move any files that you added to the `tux_prod_dir` directory to a temporary location.
4. After the installation is complete, restore these files to their original locations.

## 5.5 Performing a Simple Upgrade

A simple upgrade consists of the following basic steps:

1. If you already have the `DMTLOG` and `TLOG`, ensure that there are no pending transaction logs in these files.
2. Shut down the application.
3. Back up files as described in [Backing Up Files](#).
4. Update the text version of the configuration file on the `MASTER` machine by running the `tmunloadcf(1)` command.
5. Back up the old binary version of the configuration file.
6. If upgrading an Oracle Tuxedo 9.1 machine, uninstall the Oracle Tuxedo system software as described in [Uninstalling Oracle Tuxedo](#).
7. Install the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) software on at least the `MASTER` and backup `MASTER` machines.
8. Reload the text version of the configuration file by running `tmloadcf(1)` on the `MASTER` machine.
9. Recompile and relink your application if you use an older release. Upgrading Tuxedo 12.2.2 is not mandatory. For more on Upgrade Considerations, see [Upgrade Considerations](#).
10. If you already have `DMTLOG` and `TLOG`, remove them.
11. Reboot the application.
12. Shut down and upgrade the other machines in your configuration as time permits.



### Note:

Follow these steps if you are using Tuxedo applications that use `XAUTHSVR`:

1. Enter the value of `PASSWORD` in a plain text.
2. To encrypt the `PASSWORD` value in the `XAUTHSVR` configuration file, run `'22c installation dir>/bin/tpldapconf -f XAUTHSVR configuration file>'file`.

## 5.6 Performing Additional Upgrade Steps to Windows Server

For a Windows machine running on Oracle Tuxedo 22c Release 1 (22.1.0.0.0) or earlier software, performs the following additional steps before installing the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) software to ensure that you are starting with a clean system.

1. Stop the Tuxedo `TListen` and Oracle `procMGR` services. From the Start menu, click Start → All Programs → Administrative Tools → Services. The Services window appears.

2. Scroll to the entry for the `TListen` service, select it, and then click the **Stop** button. The Status value should change from Started to a blank entry. Then scroll to the `Oracle procMGR` (or IPC Helper) service, select it, and click the **Stop** button. In some cases, you may see an error; however, the service's Status value should change from Started to a blank entry. Next, click the **Close** button.
3. Run the Windows **Add/Remove** program to remove the prior Oracle Tuxedo software version. From the Start menu, choose Start → Settings → Control Panel → Add/Remove Programs. The Add/Remove Programs Properties window appears.
4. Scroll to the entry for the prior Oracle Tuxedo, select it, and click the **Add/Remove** button.
5. In response to the prompt, confirm that you want to uninstall the software. After the program finishes, the Remove Programs From Your Computer window appears.
6. If the uninstall program was not able to remove all directories (usually because the files were added after the original installation), you can click the **Details** button to find out which directories remain on your system. If the files in the directories contain changes that you made, such as a modified sample file, move the modified files to a temporary location.
7. Reboot your system after the uninstall completes.

## 5.7 Upgrading from Tuxedo Previous Releases to Oracle Tuxedo 22c Release 1 (22.1.0.0.0)

If you are upgrading from Oracle Tuxedo previous releases to Oracle Tuxedo 22c Release 1 (22.1.0.0.0), please note the following:

- In order to allow an Oracle Tuxedo 22c Release 1 (22.1.0.0.0) server to interoperate with Tuxedo 6.5 software, the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) server `servopts -toption` 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 Oracle Tuxedo 22c Release 1 (22.1.0.0.0) 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_OLDSECHECK` environment variable of the domain gateway in this Oracle Tuxedo 22c Release 1 (22.1.0.0.0). 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](#).

- In this release, `tpretain()` API method throws an exception when an error is encountered. You need to change your existing code to invoke `tpretain()` API method in a try-catch block; otherwise, you fail to successfully compile the code.

## 5.8 Performing a Hot Upgrade

Before performing a hot upgrade, back up files as described in [Backing Up Files](#).

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](#)
- [Hot-Upgrade Procedure](#)
- [Hot-Upgrade Example](#)

### 5.8.1 Pre-Upgrade Notes

When you migrate the `MASTER` to Oracle Tuxedo 22c Release 1 (22.1.0.0.0) (through a hot upgrade), you cannot migrate it back to the acting backup until you upgrade the backup to Oracle Tuxedo 22c Release 1 (22.1.0.0.0). Therefore, migrating the `MASTER` is irreversible; once the migration is complete, the `MASTER` cannot be reverted to its earlier Oracle Tuxedo release.

When you run `tmadmin(1)` on an Oracle Tuxedo 22c Release 1 (22.1.0.0.0) machine with a previous release `MASTER`, the `UPGRADE` file shown in [Hot-Upgrade Example](#) must reside in the current directory. To ensure that the files are in the current directory, you might, for example, run `tmadmin` from the Tuxedo application directory: `$APPDIR` (UNIX) directory.

Follow these guidelines when migrating Oracle Tuxedo data storage areas:

- You must remove the `oldTUXCONFIG` file. As `TUXCONFIG` is propagated automatically when the backup machine is run, no action is required on your part.
- 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 22c Release 1 (22.1.0.0.0) on the `MASTER` machine and backup `MASTER`, and rebuilding all clients and servers in the directory on each machine that is defined by `$APPDIR`, in parallel with the directory in which the old-release clients and servers were created. In other words, both an old-release application and an Oracle Tuxedo 22c Release 1 (22.1.0.0.0) application must 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 do not to recreate them. Furthermore, most applications do not need to add



queue space extent in the future or are 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.

If you already have `DMTLOG` and `TLOG`, make sure that no pending transaction logs are in them before shutting down the application. In addition, ensure that you remove the `DMTLOG` and `TLOG` (if they exist) after the application is shut down.

## 5.8.2 Hot-Upgrade Procedure

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

1. Ensure 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 an Oracle Tuxedo 9.1 machine, uninstall the Oracle Tuxedo system software as specified in [Uninstalling Oracle Tuxedo](#).
3. Shut down the backup `MASTER` machine.
4. Install the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) 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 Tuxedo 22c Release 1 (22.1.0.0.0).
6. Shut down the machine that is now acting as the backup `MASTER`.
7. Install Oracle Tuxedo 22c Release 1 (22.1.0.0.0) on the old `MASTER`.
8. Using Oracle Tuxedo 22c Release 1 (22.1.0.0.0), reboot the machine that you shut down in step 7.

For more information about hot upgrades, see [Hot-Upgrade Example](#).

## 5.8.3 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:

1. 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`
2. A machine called `MACH2` that is configured as the backup.
  - A corresponding application directory called `$APPDIR2`
  - A corresponding configuration file called `$TUXCONFIG2`

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

1. Ensure that your environment has been set up as follows (for MP mode, you must configure the `UBBCONFIG` file in the `$APPDIR` of the MASTER machine).

- a. Refer to the `ubbmp` file of the sample bank application for reference.
- b. In the resource section of the configuration, the `BACKUP MASTER` machine must be specified. Follow the steps below:

- Under the `*RESOURCES` section, select the value of `MASTER` as `MACH1, MACH2` (where, `MACH2` is the backup master machine), the following is the example:

```
MASTER MACH1, MACH2
```

- c. In the configuration, the `MODEL` option must be set to `MP` under the `*RESOURCES` section.
- d. In the configuration, the `MIGRATE` option must be set.

```
OPTIONS          LAN, MIGRATE
```

- e. Ensure that the Network section is configured with both `NADDR` and `NLSADDR` (both must be on separate ports).

```
*NETWORK
MACH1 NADDR="//<MACH1_Hostname>:<naddr_port_number>"
#BRIDGE="/scratch/<user_name>"
NLSADDR="//<MACH1_Hostname>:<nls_port_number>"
MACH2 NADDR="//<MACH2_Hostname>:<naddr_port_number>"
#BRIDGE="/scratch/user_name"
NLSADDR="//<MACH2_Hostname>:<nls_port_number>"
```

- f. Both the `MASTER` and `BACKUP MASTER` machines must be installed with the old Tuxedo version.
  - g. Start `Tlisten` on both the `MASTER` and `BACKUP MASTER` machines. In the section of `UBBCONFIG Network`, you must specify the same address as `NLSADDR` for the `tlisten`.
  - h. Follow the `README` of the sample application to generate the server and client files, in this case `"simperv"` and `"simpcl"`.
2. Run the `Tmboot` command on `MACH1` after generating the `tmconfig` file from the above configuration file.

```
MACH1> tmloadcf -y ubbmp
```

3. The `Tmboot` on Machine1 (`MASTER` machine) generates a `tmconfig` on the `BACKUP MASTER` machine (`MACH2`)
4. Use the following command to shut down the backup `MASTER` machine from `MACH1`:

```
MACH1> tmshutdown -B MACH2 -l MACH2
```

5. Use the following command to kill the `tlisten` process on the backup master machine (`MACH2`)

```
ps -ef | grep tlisten
```

```
kill -9 <pid of the tlisten got from above command>
```

6. Install the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) software on the backup `MASTER` machine.
7. Cd `$Appdir` of `MACH2` and `rm tuxconfig` file.
8. Prior to running `tmconfig` to update the `APPDIR`, `TUXCONFIG`, and `TUXDIR` for `MACH2` on the `UBBCONFIG` file available in the `MACH1`, export `EDITOR=vi` to enable editing in the `vi` editor:

```
MACH1> tmconfig
choose option 2)MACHINES
choose option 5)Update
```

`TA_LMID`<the machine LMID for which you want an update> (You can find the value of `LMID` in `UBBCONFIG` file `*MACHINES` section)

```
TA_APPDIR Value of the new appdir
TA_TUXDIR Value of new tuxdir
TA_TUXCONFIG value of new TUXCONFIG
```

```
TA_LMID MACH2
TA_APPDIR      /scratch/<user_name>/oraHome_22c_load11/
tuxedo22.1.0.0.0/samples/atmi/simpapp
TA_APPDIR      /scratch/<user_name>/oraHome_22c_load11/
tuxedo22.1.0.0.0
TA_APPDIR      /scratch/<user_name>/oraHome_22c_load11/
tuxedo22.1.0.0.0/samples/atmi/tuxconfig
```

- Run the command `wq!` to save and exit
  - Select `Y` for the confirmation. For more information about `tmconfig`, see [How to Run tmconfig](#).
  - Run `tmunloadcf` to verify successful update.
9. To set `PATH`, `LD_LIBRARY`, and `TUXDIR` to the latest release of Tuxedo, run `tux.env` from the newly installed Tuxedo product in `MACH2`.
  10. Ensure `TUXCONFIG` and `APPDIR` point to the latest version of Tuxedo.
  11. Use the following command to create a file `UPGRADE` inside the `$APPDIR` of `MACH2`

```
touch UPGRADE
```

12. In MACH2, export the following Security environment variables:

```
export LLE_DEPRECATION_WARN_LEVEL=NONE
export TM_USE_OLD_CIPHER=Y
export TM_ALLOW_NOTLS=Y
export TM_MIN_PUB_KEY_LENGTH=0
export TM_SECURITY_CONFIG=NONE
```

13. Start the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) `tlisten` in MACH2.

14. Reboot the backup MASTER machine using the following command from MACH1:

```
MACH1> tmboot -B MACH2 -l MACH2
```

verify the `tmconfig` file generated in this process is located in `$APPDIR`

15. Migrate the MASTER machine running the old release to the MASTER machine running the latest release.

After running the `MASTER` command, you must not encounter any errors and `PSR` displays the services running on the machine.

```
MACH2> tadmin
          master
          Y
          psr
          q
```

16. Use the following command to kill the `tlisten` from MACH1:

```
ps -ef | grep tlisten
kill -9 <pid of the tlisten got from above command>
```

17. Ensure that Oracle Tuxedo 22c Release 1 (22.1.0.0.0) software is installed on the OLD MASTER machine (MACH1).

18. Delete the `tuxconfig` file from `$APPDIR` of the MACH1 (if it exists).

19. Run `tmconfig` file on the MACH 2 to reset the `APPDIR`, `TUXCONFIG`, and `TUXDIR` for MACH1 on the `UBBCONFIG` file, and then export `EDITOR=vi` before running `tmconfig` file to enable the modification to be completed with the `vi` editor:

- MACH2> `tmconfig`  
choose option 2)MACHINES  
choose option 5)Update

Pass the following variables in key-value pairs with `tab` to update the details of a specific machine.

- `TA_LMID MACH1`  
`TA_APPDIR` Value of the new `appdir`  
`TA_TUXDIR` Value of new `tuxdir`  
`TA_TUXCONFIG` value of new `TUXCONFIG`

- Run the command `wq!` to save and exit.
  - Select `Y` for the confirmation. For more information about `tmconfig`, see [How to Run tmconfig](#).
  - Run `tmunloadcf` to verify successful update.
20. Set the `PATH`, `LD_LIBRARY`, and `TUXDIR` to the latest release of Tuxedo in `MACH1` by running the `tux.env` file from the newly installed Tuxedo product.
  21. Ensure `TUXCONFIG` and `APPDIR` point to the latest version of Tuxedo.
  22. Export the following environment variables in `MACH1`:

```
MACH1>
export LLE_DEPRECATED_WARN_LEVEL=NONE
export TM_USE_OLD_CIPHER=Y
export TM_ALLOW_NOTLS=Y
export TM_MIN_PUB_KEY_LENGTH=0
export TM_SECURITY_CONFIG=NONE
```

23. Start the Oracle Tuxedo 22c `tlisten` in `MACH1`.
24. Using the following command to reboot the OLD MASTER machine (`MACH1`) from the `MACH2`:

```
MACH2> tmboot -B MACH1 -l MACH1
```

Verify `tmconfig` is generated in `$APPDIR` in this process.

25. Delete the `UPGRADE` command from `MACHINE2 $APPDIR`

## 5.9 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 22c Release 1 (22.1.0.0.0).

### Note:

- Rebuild is mandatory when upgrading from a previous Oracle Tuxedo version to Oracle Tuxedo 22c Release 1 (22.1.0.0.0). For more information about Upgrading Consideration, see Upgrade Considerations.
- The TMS server must be rebuilt when upgrading a previous Oracle Tuxedo version to Oracle Tuxedo 22c Release 1 (22.1.0.0.0).

You should also remove any old `TUXCONFIG` files and reload them using the newly installed `tmloadcf(1)`.

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, can continue to use executable files from an earlier release. For example, even if your MASTER and BACKUP MASTER machines are running Oracle Tuxedo 12.2.2, 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 12c Release 2 (12.2.2) only on Workstation clients, and you do not want to upgrade any of your Oracle Tuxedo system server machines to Oracle Tuxedo 22c Release 1 (22.1.0.0.0), 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 machine is running Oracle Tuxedo 6.5 or later.

- [Changing from Object Files to Libraries](#)

## 5.9.1 Changing from Object Files to Libraries

Some Oracle Tuxedo functions are delivered in object files in the Oracle Tuxedo release 12.1.3 (or lower). In this Oracle Tuxedo release, these object files are packaged in libraries. When rebuilding the corresponding executables, you need to link with the related libraries instead of the object files as listed in the following table:

**Table 5-1 Change from Object Files to Libraries**

Object File	Library	Link option (Unix/Linux)	Usage
cleanupsrv.o	libtux.so / libtux.dll / libtux.sl	-ltux	cleanupsrv
enqdeqmq.o	libtuxmq.so / libtuxmq.dll / libtuxmq.sl	-ltuxmq	TMQUEUE_MQM
tm_mqi.o	libtuxmq.so / libtuxmq.dll / libtuxmq.sl	-ltuxmq	TM_MQI
tm_mqi_st.o	libtuxmq.so / libtuxmq.dll / libtuxmq.sl	-ltuxmq	TM_MQI
tm_mqo.o	libtuxmq.so / libtuxmq.dll / libtuxmq.sl	-ltuxmq	TM_MQO
tgiopservice.o	liborb.so / liborb.dll / liborb.sl	-lorb	TMIFRSVR
TMQFORWARD.o	libqm.so / libqm.dll / libqm.sl	-lqm	TMQFORWARD
TMQUEUE.o	libqm.so / libqm.dll / libqm.sl	-lqm	TMQUEUE
tpfwinit.o	libnative.so / libnative.dll / libnative.sl	-lnative	TMNTS
tsam_ext.o	libtmagent.so / libtmagent.dll / libtmagent.sl	-ltmagent	Tuxedo service using tsambegin() / tsamend().

Table 5-1 (Cont.) Change from Object Files to Libraries

Object File	Library	Link option (Unix/Linux)	Usage
usort.o	libusort.so / libusort.dll / libusort.sl	-lusort	

## 5.10 Installing Oracle Jolt 22c with Oracle WebLogic Server

This section provides supplementary installation instructions for customers using Oracle Jolt 22c Release 1 (22.1.0.0.0) with Oracle WebLogic Server 9.x, 10.x, 12.1.x, or 14.1.1 and Oracle Tuxedo 9.0, 9.1, 10.0, 10.3, 11.1.1.x, 12.1.1, 12.1.3, 12.2.2 or 22.1.0.0.0.

1. You can install the Jolt client software on an Oracle WebLogic Server 9.x, 10.x, 12.1.x or 14.1.1 machine by copying the Jolt client software from an Oracle Tuxedo 22c Release 1 (22.1.0.0.0) machine to the target Oracle WebLogic Server machine.

 **Note:**

If the Oracle Tuxedo 22c Release 1 (22.1.0.0.0) machine and the target Oracle WebLogic Server machine are the same machine, then you only need to include the Jolt client library class files in your CLASSPATH variable setting.

2. 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.

# 6

## Performing Post-Installation Tasks

The following sections describe the tasks you perform after installing Oracle Tuxedo 22c.

- [Understanding the Oracle Tuxedo Directory Structure](#)
- [Understanding the Oracle Service Architecture Leveraging Tuxedo Directory Structure](#)
- [Understanding the Oracle Tuxedo Architecture](#)
- [UBBCONFIG File](#)
- [MASTER Machine](#)
- [TUXCONFIG File](#)
- [TUXCONFIG Environment Variable](#)
- [TUXDIR Environment Variable](#)
- [Setting Up Your Environment](#)
- [Setting Environment Variables on a UNIX System](#)
- [Starting the tlisten Process](#)
- [tlisten Invocation](#)
- [tlisten Password](#)
- [Editing a UBBCONFIG File](#)
- [Using the TYPE Parameter in UBBCONFIG](#)
- [Checking IPC Requirements](#)
- [Creating the Universal Device List and the Transaction Log](#)
- [Defining the TLOG](#)
- [Creating the UDL and UDL Entries for TLOG](#)
- [Running simpapp to Verify Your Installation](#)
- [Running simpapp to Verify the Oracle Tuxedo CORBA C++ Software Installation](#)
- [Running buildtms for Oracle Tuxedo Applications That Use XA Resource Managers](#)
- [Uninstalling Oracle Tuxedo](#)
- [Reinstalling Oracle Tuxedo](#)

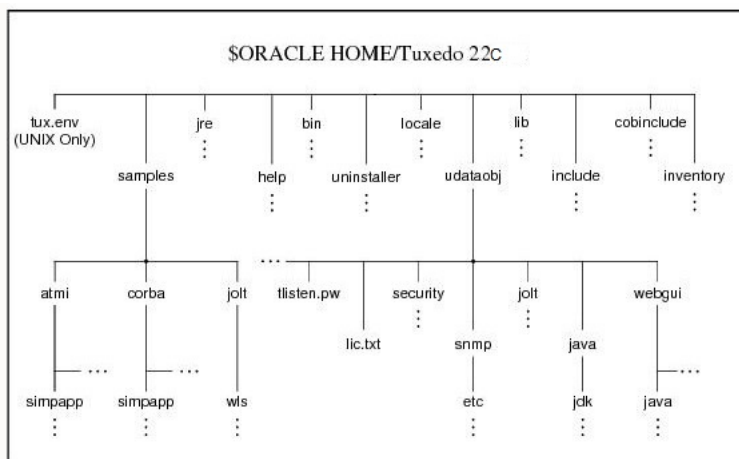
### 6.1 Understanding the Oracle Tuxedo Directory Structure

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.

The following figure illustrates the Oracle Tuxedo 22c Directory Structure:



**Figure 6-1 Oracle Tuxedo 22c Release Directory Structure**




The product directory shown here, is the default for Oracle Tuxedo 22c Release 1 (22.1.0.0.0). The default name can be changed during installation. The top-level directories and files of the Oracle Tuxedo directory structure are briefly described in the following table:

**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.
cobininclude	Contains .cbl file entries for use in COBOL programs.
include	Contains C and C++ language header files, as well as OMG IDL files. Includes sub-directories unicode and xercesc; may include sub-directory rpc, depending on the platform.
jmx	Contains the files for TSAM Plus.
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.
locale	Contains sub directories to support the localization of system messages. C sub-directory contains message catalogs for the default locale (U.S. English).

**Table 6-1 (Cont.) Top-level Directories and Files of the Oracle Tuxedo Directory Structure**

Directory Name	Description
<p>samples</p> <div data-bbox="574 436 867 821" style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> <b>Note:</b></p> <p>Samples are not installed unless they are selected during Oracle Tuxedo 22c Release installation.</p> </div>	<p>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 sub-directories</p> <ul style="list-style-type: none"> <li>• <code>atmi</code> A collection of simple applications that demonstrate the many features of the Oracle Tuxedo Application-to-Transaction Monitor Interface (ATMI) server software.</li> <li>• <code>corba</code> A collection of simple applications that demonstrate the many features of the Oracle Tuxedo Common Object Request Broker Architecture (CORBA) C++ server software.</li> <li>• <code>jolt</code> A collection of simple applications that demonstrate the many features of the Oracle Jolt server software.</li> <li>• <code>tdc</code> Samples for Tuxedo distributed cache</li> </ul>
<p><code>tux.env</code> (file)</p>	<p>UNIX systems only: Contains Oracle Tuxedo environment variables for UNIX installations and serves as a model for setting those variables.</p>
<p><code>udataobj</code></p>	<p>Contains other directories and files required by Oracle Tuxedo. The <code>udataobj</code> directory contains the following sub-directories and files:</p> <ul style="list-style-type: none"> <li>• <code>security</code> Contains the default Lightweight Directory Access Protocol (LDAP) filter file (<code>bea_ldap_filter.dat</code>) and LLE related and SSL related encryption files.</li> <li>• <code>Jolt</code> Contains the files for the Oracle Jolt software components that you selected to install.</li> <li>• <code>snmp</code> Contains the <code>etc</code> directory, which in turn contains the SNMP Agent configuration files, MIB files, and other miscellaneous files.</li> <li>• <code>java</code> Contains the classes and Java archive files needed to run Java applications.</li> <li>• <code>tlisten.pw</code> (file) Contains the <code>tlisten</code> administrative password that you entered during the installation.</li> </ul>

**Table 6-1 (Cont.) Top-level Directories and Files of the Oracle Tuxedo Directory Structure**

Directory Name	Description
uninstaller	Contains code required to uninstall the Oracle Tuxedo software.

## 6.2 Understanding the Oracle Service Architecture Leveraging Tuxedo Directory Structure

During the SALT software installation, all SALT files are decompressed within the Oracle Tuxedo `TUXDIR` directory.



### Note:


For each Oracle product, a file called `comps.xml` (located in `$ORACLEHOME/inventory/ContentsXML`), lists installation information.

The following table lists brief descriptions of SALT directories and files.

**Table 6-2 SALT Directory and File Structure**

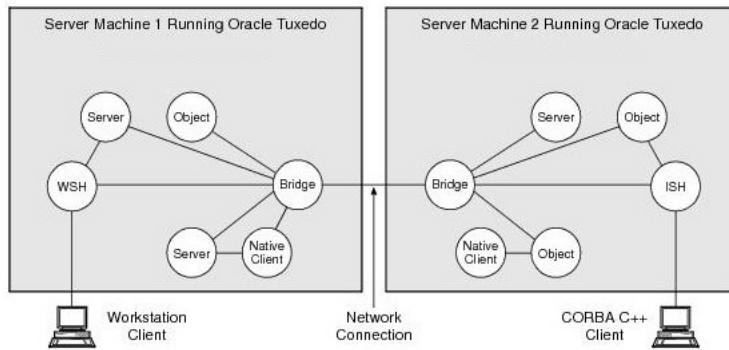
Directory Name	Description
bin	Contains SALT executable programs, including: <ul style="list-style-type: none"> <li>GWWS</li> <li>tmwsdlgen</li> <li>wsadmin</li> <li>wsloadcf</li> <li>tmwsdlflt (internally used)</li> </ul>
include	Contains C language header files for SALT plug-in development and administrative development, including: <ul style="list-style-type: none"> <li>auth_pi_ex.h</li> <li>custtype_pi_ex.h</li> <li>libsoapeng.*</li> <li>redmap_pi_ex.h</li> <li>wsadmflds.h</li> <li>wsoapflds.h</li> </ul>
lib	Contains C language header files for SALT plug-in development and administrative development, including: <ul style="list-style-type: none"> <li>libwsbase.* (shared library suffix is variant on different platforms)</li> <li>libwsconf.*</li> <li>libsoapeng.*</li> </ul>
locale	Contains the localization of system messages used by SALT. C subdirectory contains the following SALT specific message catalogs for the default locale (U.S. English): GWWS, WSBASE, WSCFG, WSCMD.

**Table 6-2 (Cont.) SALT Directory and File Structure**

Directory Name	Description
samples/salt	<p>A collection of simple applications that demonstrate the SALT features.</p> <div style="border-left: 2px solid #0070C0; padding-left: 10px; margin-left: 150px;"> <p> <b>Note:</b></p> <p>samples/salt is installed only if requested during installation</p> </div>
udataobj	<p>Contains the following information files for SALT:</p> <ul style="list-style-type: none"> <li>• 3rd_party_licenses_salt.txt</li> </ul> <p>License information for 3rd party software used by SALT.</p> <ul style="list-style-type: none"> <li>• wsadmflds</li> </ul> <p>SALT administrative FML32 field definitions</p> <ul style="list-style-type: none"> <li>• wsssoapflds</li> </ul> <p>SALT pre-defined FML32 fields for outbound Web Service call.</p>
inventory	<p>Contains information about installed products for use with the Oracle Universal Installer.</p>
udataobj/salt	<p>Contains sub-directories and files required by SALT, including:</p> <ul style="list-style-type: none"> <li>• policy subdirectory</li> </ul> <p>Contains SALT pre-defined WS-* policy files.</p> <ul style="list-style-type: none"> <li>• WS subdirectory</li> </ul> <p>Contains various external XML Schema files required by SALT.</p> <ul style="list-style-type: none"> <li>• xalan subdirectory</li> </ul> <p>Contains Apache Xalan Java 2.7 class library files and corresponding license information.</p> <ul style="list-style-type: none"> <li>• xslt files</li> </ul> <p>wsdl*.xsl files are used by SALT WSDL converter (wsdlcvt).</p> <ul style="list-style-type: none"> <li>• SALT XML Schema files</li> </ul> <p>*.xsd files are XML Schema files of SALT configuration file format.</p>

## 6.3 Understanding the Oracle Tuxedo Architecture

The following image 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.

## 6.4 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 File](#) in Oracle Tuxedo 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.

## 6.5 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.

## 6.6 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; the Tuxedo system is booted on the `MASTER` machine.

## 6.7 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 `MACHINES` 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.

## 6.8 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.

## 6.9 Setting Up Your Environment

You need to set several environment variables before using Oracle Tuxedo to build and run Oracle Tuxedo applications: [Table 6-3](#), [Table 6-4](#), [Table 6-5](#) and [Table 6-6](#).

**Table 6-3 Oracle Tuxedo Core Environment Variables**

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 are booted on this server machine. APPDIR may be set to more than one application directory.
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 <code>tmloadcf(1)</code> command on the UBBCONFIG configuration file.

**Table 6-4 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 6-5 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.

The following table lists Java Environment Variables.

**Table 6-6 Java Environment Variables**

Environment Variable	Description
JAVA_HOME	Absolute pathname of the Java Development Kit (JDK) 1.8.x installation directory on this machine; needed to build and run Java applications on this machine.*
JDKDIR	Set to JAVA_HOME value.

**Table 6-6 (Cont.) Java Environment Variables**

Environment Variable	Description
CLASSPATH	Absolute pathnames to classes and Java archive files on this machine; needed to run Java applications on this machine.



**Note:**

Oracle Tuxedo 22c distribution does not include a JDK

## 6.10 Setting Environment Variables on a UNIX System

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

**Table 6-7 Environment Variables**

Environment Variables
TUXDIR=pathname_of_Oracle_Tuxedo_product_directory
APPDIR=pathname_of_Oracle_Tuxedo_application_directory
TUXCONFIG=pathname_of_TUXCONFIG_file
PATH=\$APPDIR:\$TUXDIR/bin:/bin:\$PATH
LD_LIBRARY_PATH=\$APPDIR:\$TUXDIR/lib:/lib:/usr/lib:\$LD_LIBRARY_PATH
export TUXDIR APPDIR TUXCONFIG WEBJAVADIR PATH LD_LIBRARY_PATH

Examples of TUXDIR, APPDIR, and TUXCONFIG are:

```
TUXDIR=/home/Oracle/tuxedo22cR1
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.

## 6.11 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 the following table.

**Table 6-8 tlisten Process**

Machine	Administrator	Method
UNIX server	UNIX system administrator	Manually starting <code>tlisten</code> process from a command-line shell.
UNIX server	Oracle Tuxedo application administrator	As a <code>cron</code> job.
	Oracle Tuxedo application administrator	Manually starting a <code>tlisten</code> process from a command-line shell.

## 6.12 `tlisten` Invocation

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

```
%TUXDIR%\bin\tlisten -l nlsaddr [-u appuid] (Windows)
$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 [UBCONFIG\(5\)](#) in Oracle Tuxedo File Formats, Data Descriptions, MIBs, and System Processes.

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`. This is done 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 not mandatory since the job is already owned by the correct account.

For more information about the [tlisten\(1\)](#) command, see reference page in Oracle Tuxedo Command Reference.

## 6.13 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)` gateway processes, Oracle Tuxedo authenticates them by means of the password.

A `tlisten` password can't be in clear-text format. You can generate a `tlisten` password by running the following command:

- Unix/Linux: `tlistpwd $TUXDIR`
- Windows: `tlistpwd %TUXDIR%`

A common password is required for two machines in an 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 each other.

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.

## 6.14 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 Windows system, this application is found in the directory `%TUXDIR%\samples\atmi\simpapp`; on a UNIX system, it is found in the directory `$TUXDIR/samples/atmi/simpapp`.

The following example `ubbsimple` file has been modified from the `ubbsimple` file delivered on a UNIX system to include example pathname values for both Windows and UNIX systems.

```
#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/tuxedo12cR1"
<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. The following table provides a sample of the parameters that must be defined in every configuration file.

**Table 6-9 Parameter Sample**

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 Windows 2003 Server system, see your system administrator. To obtain the node name on a UNIX system, run the <code>uname -n</code> command.

Table 6-9 (Cont.) Parameter Sample

Parameter	Specifies
APPDIR = string	A list of one or more directories in which application and administrative servers are booted on this machine. For Windows, the value of a string is the absolute pathname of one directory, optionally followed by a semicolon-separated list of pathnames for other directories on the machine being defined. 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 <code>tmloadcf(1)</code> 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 *Oracle Tuxedo 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 fails with syntax errors.

## 6.15 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.

## 6.16 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. The following is an output report based on the values in `ubbsimple`.

```
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 22c Release Platform Data Sheets](#).

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 an 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 22c Release Platform Data Sheets](#).

## 6.17 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 uses global transactions. The TLOG is a log file in which information about transactions is kept until the transaction is completed.

## 6.18 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 uses global transactions. The following table lists the parameters:

**Table 6-10 Parameter**

Parameter	Description
<code>TLOGDEVICE = string</code>	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.
<code>TLOGOFFSET = offset</code>	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.
<code>TLOGNAME = string</code>	The name of the DTP transaction log for this machine. If not specified, the default is "TLOG."
<code>TLOGSIZE = size</code>	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.

## 6.19 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 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 resides), 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 uses 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.

## 6.20 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.

## 6.21 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](#)

## 6.21.1 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/tuxedo12cR1` directory, set and export `TUXCONFIG` as follows:

```
prompt> TUXDIR=/home/oracle/tuxedo12cR1
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`

## 6.22 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 the directory `$TUXDIR/bin` on UNIX systems.

If you run the CORBA C++ University sample applications, each sample's `make` file 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.

- [Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation](#)

### 6.22.1 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 Oracle Tuxedo 22c Release 1 (22.1.0.0.0) 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](#). *Tutorials for Developing Oracle Tuxedo ATMI Applications*.

- [Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation on a UNIX System](#)

### 6.22.1.1 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](#). 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 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/tuxedo12cR1"`).
  - 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. The following listing shows the output produced by `tmboot -y`:

```
Booting all admin and server processes in /home/me/atmi/tuxconfig
INFO: Oracle Tuxedo , Version 22.1.0.0.0
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 makes 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 22c Release Platform Data Sheets](#).

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
```

## 6.23 Uninstalling Oracle Tuxedo

Uninstalling Oracle Tuxedo 22c Release 1 (22.1.0.0.0) 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.

After launching OUI, the Welcome screen appears. Click **Deinstall Products**. Select the product(s) you want to remove from the Contents tab of the Inventory panel and click **Remove**.

You can also remove Oracle homes in the same manner. After you have removed an Oracle home, you can reuse its name and location to install other products. Then confirm the deinstallation. After the selected products are deinstalled, the Inventory panel appears.

## 6.24 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 displays a warning that the selected ORACLE\_HOME already installed Tuxedo. If you want to continue the installation, the components which not exist in the selected ORACLE\_HOME are installed. Existing Tuxedo component are not reinstalled by default.

# 7

## Oracle Tuxedo 22c Release Platform Data Sheets

The following sections provide platform-specific information for the platforms on which the Oracle Tuxedo 22c Release system software is supported:

For more information about the list of all the platforms supported by Oracle Tuxedo 22c, see [Platform Data Sheets](#)

- [Platform Support Policy](#)
- [Database Support Policy](#)
- [Supported Platforms](#)
- [Oracle Linux 7 \(64-bit\) on x86-64](#)
- [Oracle Linux 8 \(64-bit\) on x86-64](#)
- [Oracle Linux 9 \(64-bit\) on x86-64](#)
- [Oracle Solaris 11 \(64-bit\) on SPARC](#)
- [Red Hat Enterprise Linux 7 \(64-bit\) on x86-64](#)
- [Red Hat Enterprise Linux 8 \(64-bit\) on x86-64](#)
- [Red Hat Enterprise Linux 9 \(64-bit\) on x86-64](#)
- [SUSE Linux Enterprise Server 12.5 \(64-bit\) on x86-64](#)
- [AlmaLinux 8 \(64-bit\) on x86-64](#)
- [IBM AIX 7.x \(64-bit\)](#)
- [Microsoft Windows 10 \(32-bit, Client-only\) on x86-64](#)
- [Microsoft Windows 11 \(32-bit, Client-only\) on x86-64](#)
- [Microsoft Windows Server 2019 \(32-bit, Client-only\) on x86-64](#)
- [Microsoft Windows Server 2019 \(64-bit\) on x86-64](#)
- [Microsoft Windows Server 2022 \(64-bit\) on x86-64](#)
- [HP-UX 11i v3 \(64-bit\) on Itanium](#)

### 7.1 Platform Support Policy

The Oracle policy 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.
- The virtualization software environments provided by Oracle including OVM, OVM for SPARC (aka LDOMS) and VirtualBox are supported.

Oracle Tuxedo is also considered supported in virtualization software environments provided by the operating system vendor (such as 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.
- All minor releases higher than the certified OS version are supported with the assumption that OS vendor maintains binary compatibility. For example, if Tuxedo 12.2.2 is certified on Red Hat Enterprise Linux 6.2, Red Hat Enterprise Linux 6.x (x>2) are supported with the assumption that Red Hat will maintain binary compatibility among all minor versions of release 6.
- Oracle Tuxedo supports both Standard and Enterprise editions for a Windows server if a certification has been done on one of the platforms. It's based on the assumption that Microsoft maintains binary compatibility between the editions.
- An operating system platform, which is binary-compatible with an Oracle Tuxedo-supported platform, is also considered an Oracle Tuxedo-supported platform. AlmaLinux 8 is binary-compatible with Red Hat Enterprise Linux 8, which is an Oracle Tuxedo-supported platform. So, AlmaLinux 8 too is considered an Oracle Tuxedo-supported platform.

## 7.2 Database Support Policy

- Tuxedo applications not making use of XA interface can make use of any database and version of their choice. Tuxedo applications are agnostic of database used.
- Tuxedo does not certify each and every relational database and various versions available. Instead, Tuxedo is considered supported for a relational database that supports the XA protocol (that is, XA-compliant). Check with the database vendor if your database is XA-compliant; if it is, Tuxedo is considered supported with it.
- Note that Tuxedo applications leveraging performance optimizations in Tuxedo Advanced Performance Pack (TAPP) may require a specific version of Oracle Database. Please see such requirement in the TAPP documentation. The TAPP option is supported since 12c Release 1 (12.1.1). Here is the TAPP link for this release: [Using Oracle Tuxedo Advanced Performance Pack](#).
- The Oracle policy is that when the database vendor discontinues support for a specific version, Oracle support is also discontinued.
- Oracle will not be able to notify customers and users when support for a specific database version is discontinued. As a user of the database version, we expect you to find out from the database vendor of discontinued support and plan accordingly.

## 7.3 Supported Platforms

The following are the supported platforms for the *Oracle Tuxedo 22c Release 1*:

**Table 7-1 List of Supported Platforms**

Vendor	Operating System	Release/Version
HP	HP-UX	HP-UX 11i v3 (64-bit) on Itanium
Oracle	Linux	Oracle Linux 7 (64-bit) on x86-64.
		Oracle Linux 8 (64-bit) on x86-64.
		Oracle Linux 9 (64-bit) on x86-64.
Oracle	Solaris	Oracle Solaris 11 (64-bit) on SPARC
SUSE	Linux	SUSE Linux 12.5 (64-bit) on x86-64.
Red Hat	Linux	Red Hat Enterprise Linux 7 (64-bit) on x86-64.
		Red Hat Enterprise Linux 8 (64-bit) on x86-64.
		Red Hat Enterprise Linux 9 (64-bit) on x86-64.
AlmaLinux	Linux	AlmaLinux 8.6 (64-bit OS) on x86-64
IBM	AIX	IBM AIX 7.2 (64-bit)
		IBM AIX 7.3 (64-bit)
Microsoft	Windows	Microsoft Windows 10 (32-bit, Client-only) on x86-64
		Microsoft Windows 11 (32-bit, Client-only) on x86-64
		Microsoft Windows Server 2019 (32-bit, Client-only) on x86-64
		Microsoft Windows Server 2019 (64-bit) on x86-64
		Microsoft Windows Server 2022 (64-bit) on x86-64



### Note:

Oracle Linux 8, RHEL 8 and AlmaLinux 8 no longer includes **libnsl**. So, you must install **libnsl** before you install Oracle Tuxedo 22c.

For more information about the list of all the platforms supported by Oracle Tuxedo 22c, see [Platform Data Sheets](#).

## 7.4 Oracle Linux 7 (64-bit) on x86-64

The following sections list Oracle Tuxedo 22c Release requirements for Oracle Linux 7 (64-bit) on x86-64.

- [Available Oracle Tuxedo 22c Install Types for Oracle Linux 7 \(64-bit\)](#)
- [Hardware Requirements for Oracle Linux 7 \(64-bit\)](#)
- [Software Requirements for Oracle Linux 7 \(64-bit\)](#)
- [Network Requirements for Oracle Linux 7 \(64-bit\)](#)



- [Disk Space Requirements for Oracle Linux 7 \(64-bit\)](#)
- [Temporary Storage Space Requirements for Oracle Linux 7 \(64-bit\)](#)
- [Tunable Parameters for Oracle Linux 7 \(64-bit\)](#)

## 7.4.1 Available Oracle Tuxedo 22c Install Types for Oracle Linux 7 (64-bit)

- Full install type
- Server install type
- Client install type
- Custom install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

## 7.4.2 Hardware Requirements for Oracle Linux 7 (64-bit)

- X86-64
- 1 GB of RAM minimum
- At least 4MB RAM for each Oracle Tuxedo system server

## 7.4.3 Software Requirements for Oracle Linux 7 (64-bit)

**Table 7-2 Software Requirements for Oracle Linux 7 (64-bit)**

Component	Requirement
OS Version (Patches)	Oracle Linux 7 x86_64 (64-bit)
C/C++ Compilers	C/C++: gcc version 4.8.5 20150623 (Red Hat 4.8.5-4) (GCC)
COBOL Compiler	Visual COBOL 6.0
Java Runtime Environment (JRE)	Tested with JRE: 1.8.0_341, 11.0.15.1, 17.0.4 and OpenJDK17
Java Development Kit (SDK)	Tested with JDK 1.8.0_341, 11.0.15.1, 17.0.4 and OpenJDK17
Database	Tested with Oracle Client 12c, 19c and 21c

 **Note:**

- 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, compiles with the newer version of compiler without requiring any changes.
- Executables and libraries:  
Libraries and executables built using the tested version, runs with libraries and executables that are built with the newer version of compilers.

## 7.4.4 Network Requirements for Oracle Linux 7 (64-bit)

- TCP/IP, using the Sockets network interface

## 7.4.5 Disk Space Requirements for Oracle Linux 7 (64-bit)

For all install types and components, 578MB is the minimum disk space requirement for installation on Oracle Linux 7 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.4.6 Temporary Storage Space Requirements for Oracle Linux 7 (64-bit)

For all install types and components, 107MB is the minimum temporary storage space requirement for installation on Oracle Linux 7 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.4.7 Tunable Parameters for Oracle Linux 7 (64-bit)

See `sysctl` manpage.

# 7.5 Oracle Linux 8 (64-bit) on x86-64

The following sections list Oracle Tuxedo 22c Release requirements for Oracle Linux 8 (64-bit) on x86-64.

- [Available Oracle Tuxedo 22c Release Install Types for Oracle Linux 8 \(64-bit\)](#)
- [Hardware Requirements for Oracle Linux 8 \(64-bit\)](#)
- [Software Requirements for Oracle Linux 8 \(64-bit\)](#)
- [Network Requirements for Oracle Linux 8 \(64-bit\)](#)
- [Disk Space Requirements for Red Hat Enterprise Linux 8 \(64-bit\)](#)
- [Temporary Storage Space Requirements for Oracle Linux 8 \(64-bit\)](#)
- [Tunable Parameters for Oracle Linux 8 \(64-bit\)](#)

## 7.5.1 Available Oracle Tuxedo 22c Release Install Types for Oracle Linux 8 (64-bit)

- Full install type

- Server install type
- Client install type
- Custom install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

## 7.5.2 Hardware Requirements for Oracle Linux 8 (64-bit)

- X86-64
- 1 GB of RAM minimum
- At least 4MB RAM for each Oracle Tuxedo system server

## 7.5.3 Software Requirements for Oracle Linux 8 (64-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 8 x86_64 (64-bit)
C/C++ Compilers	C/C++: gcc version 8.5.0 20210514 (Red Hat 8.5.0-4.0.1)
COBOL Compiler	Visual COBOL 6.0
Java Runtime Environment (JRE)	Tested with JRE 1.8.0_341, 11.0.15.1, and 17.0.4 and OpenJDK 17.
Java Development Kit (SDK)	Tested with JDK 1.8.0_341, 11.0.15.1, and 17.0.4 and OpenJDK 17.
Database	Tested with Oracle Client 12c

### Note:

- `libnsl` is no longer available in Oracle Linux 8. Thus, you must install `libnsl` before you install Oracle Tuxedo 22c.
- 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, compiles with the newer version of compiler without requiring any changes.
  - Executables and libraries:  
Libraries and executables built using the tested version, run with libraries and executables that are built with the newer version of compilers.

## 7.5.4 Network Requirements for Oracle Linux 8 (64-bit)

- TCP/IP, using the Sockets network interface

## 7.5.5 Disk Space Requirements for Red Hat Enterprise Linux 8 (64-bit)

For all install types and components, 578MB is the minimum disk space requirement for installation on Red Hat Enterprise Linux 8 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.5.6 Temporary Storage Space Requirements for Oracle Linux 8 (64-bit)

For all install types and components, 107MB is the minimum temporary storage space requirement for installation on Oracle Linux 8 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.5.7 Tunable Parameters for Oracle Linux 8 (64-bit)

See `sysctl` manpage.

## 7.6 Oracle Linux 9 (64-bit) on x86-64

Oracle Tuxedo 22c now supports Oracle Linux 9 (64 bit).

The following sections list Oracle Tuxedo 22c Release requirements for Oracle Linux 9 (64-bit).

- [Software Requirements for Oracle Linux 9 \(64-bit\)](#)

### 7.6.1 Software Requirements for Oracle Linux 9 (64-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 9 x86_64 (64-bit)
C/C++ Compilers	C/C++: gcc (GCC) 11.2.1 20220127 (Red Hat 11.2.1-9.4.0.2)
COBOL Compiler	Visual COBOL 6.0
Java Runtime Environment (JRE)	Tested with JRE: 1.8.0_341, and 17.0.4
Java Development Kit (SDK)	Tested with JRE: 1.8.0_341, and 17.0.4
Database	Tested with Oracle Client 19c

 **Note:**

- `libnsl` is no longer available in Oracle Linux 9. Thus, you must install `libnsl` before you install Oracle Tuxedo 22c.
- 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, compiles with the newer version of compiler without requiring any changes.
  - Executables and libraries:  
Libraries and executables built using the tested version, run with libraries and executables that are built with the newer version of compilers.

## 7.7 Oracle Solaris 11 (64-bit) on SPARC

The following sections list Oracle Tuxedo 22c Release requirements for Oracle Solaris 11 (64-bit) on SPARC:

- [Available Oracle Tuxedo 22c Release Install Types for Oracle Solaris 11 \(64-bit\)](#)
- [Hardware Requirements for Oracle Solaris 11 \(64-bit\)](#)
- [Software Requirements for Oracle Solaris 11 \(64-bit\)](#)
- [Network Requirements for Oracle Solaris 11 \(64-bit\)](#)
- [Disk Space Requirements for Oracle Solaris 11 \(64-bit\)](#)
- [Temporary Storage Space Requirements for Oracle Solaris 11 \(64-bit\)](#)
- [Tunable Parameters for Oracle Solaris 11 \(64-bit\)](#)

### 7.7.1 Available Oracle Tuxedo 22c Release Install Types for Oracle Solaris 11 (64-bit)

- Full install type
- Server install type
- Client install type
- Custom install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

## 7.7.2 Hardware Requirements for Oracle Solaris 11 (64-bit)

- SPARC
- 1GB of RAM minimum
- At least 19MB RAM for each Oracle Tuxedo system server

## 7.7.3 Software Requirements for Oracle Solaris 11 (64-bit)

Component	Requirement
OS Version	SunOS 5.11 (11.4)
C/C++ Compilers	Oracle Developer Studio 12.6
COBOL	Micro Focus Visual COBOL 6.0
Java Runtime Environment (JRE)	Tested with Oracle JRE 1.8.0_341 and 11.0.19
Java Development Kit (SDK)	Tested with Oracle JDK 1.8.0_341 and 11.0.19
Database	Tested with Oracle Client 19c

### Note:

- Oracle Developer Studio runtime libraries are typically located in `/usr/lib/sparcv9` and it must precede `/lib/64` in `LD_LIBRARY_PATH`
- 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 compiler 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.

## 7.7.4 Network Requirements for Oracle Solaris 11 (64-bit)

TCP/IP provided by the operating system.

## 7.7.5 Disk Space Requirements for Oracle Solaris 11 (64-bit)

For all install types and components, 654MB is the minimum disk space requirement for installation on Oracle Solaris 11 (64-bit) SPARC systems. This requirement assumes installation of the default components for the selected install type.

## 7.7.6 Temporary Storage Space Requirements for Oracle Solaris 11 (64-bit)

For all install types and components, 45MB is the minimum temporary storage space requirement for installation on Oracle Solaris 11 (64-bit) SPARC systems. This requirement assumes installation of the default components for the selected install type.

## 7.7.7 Tunable Parameters for Oracle Solaris 11 (64-bit)

- 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*).

## 7.8 Red Hat Enterprise Linux 7 (64-bit) on x86-64

The following sections list Oracle Tuxedo 22c Release requirements for Red Hat Enterprise Linux 7 (64-bit) on x86-64.

- [Available Oracle Tuxedo 22c Release Install Types for Red Hat Enterprise Linux 7 \(64-bit\)](#)
- [Hardware Requirements for Red Hat Enterprise Linux 7 \(64-bit\)](#)
- [Software Requirements for Red Hat Enterprise Linux 7 \(64-bit\)](#)
- [Network Requirements for Red Hat Enterprise Linux 7 \(64-bit\)](#)
- [Disk Space Requirements for Red Hat Enterprise Linux 7 \(64-bit\)](#)
- [Temporary Storage Space Requirements for Red Hat Enterprise Linux 7 \(64-bit\)](#)
- [Tunable Parameters for Red Hat Enterprise Linux 7 \(64-bit\)](#)

### 7.8.1 Available Oracle Tuxedo 22c Release Install Types for Red Hat Enterprise Linux 7 (64-bit)

- Full install type
- Server install type
- Client install type
- Custom install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

### 7.8.2 Hardware Requirements for Red Hat Enterprise Linux 7 (64-bit)

- X86-64
- 1 GB of RAM minimum
- At least 4MB RAM for each Oracle Tuxedo system server

## 7.8.3 Software Requirements for Red Hat Enterprise Linux 7 (64-bit)

**Table 7-3 Software Requirements for Red Hat Enterprise Linux 7 (64-bit)**

Component	Requirement
OS Version (Patches)	Red Hat Enterprise Linux 7 x86_64 (64-bit)
C/C++ Compilers	C/C++: gcc version 4.8.5 20150623 (Red Hat 4.8.5-39.0.3)
COBOL Compiler	Visual COBOL 6.0
Java Runtime Environment (JRE)	Tested with JRE 1.8.0_341, 11.0.15.1, and 17.0.4 and OpenJDK 17
Java Development Kit (SDK)	Tested with JDK 1.8.0_341, 11.0.15.1, and 17.0.4 and OpenJDK 17
Database	Tested with Oracle Client 12c, 19c and 21c

## 7.8.4 Network Requirements for Red Hat Enterprise Linux 7 (64-bit)

TCP/IP, using the Sockets network interface.

## 7.8.5 Disk Space Requirements for Red Hat Enterprise Linux 7 (64-bit)

For all install types and components, 578MB is the minimum disk space requirement for installation on Red Hat Enterprise Linux 7 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.8.6 Temporary Storage Space Requirements for Red Hat Enterprise Linux 7 (64-bit)

For all install types and components, 107MB is the minimum temporary storage space requirement for installation on Red Hat Enterprise Linux 7 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.8.7 Tunable Parameters for Red Hat Enterprise Linux 7 (64-bit)

See `sysctl` manpage.

## 7.9 Red Hat Enterprise Linux 8 (64-bit) on x86-64

The following sections list Oracle Tuxedo 22c Release requirements for Red Hat Enterprise Linux 8 (64-bit) on x86-64.

- [Hardware Requirements for Red Hat Enterprise Linux 8 \(64-bit\)](#)
- [Available Oracle Tuxedo 22c Release Install Types for Red Hat Enterprise Linux 8 \(64-bit\)](#)
- [Software Requirements for Red Hat Enterprise Linux 8 \(64-bit\)](#)
- [Disk Space Requirements for Red Hat Enterprise Linux 8 \(64-bit\)](#)



- [Temporary Storage Space Requirements for Red Hat Enterprise Linux 8 \(64-bit\)](#)
- [Tunable Parameters for Red Hat Enterprise Linux 8 \(64-bit\)](#)

## 7.9.1 Hardware Requirements for Red Hat Enterprise Linux 8 (64-bit)

- x86-64
- 1 GB of RAM minimum
- At least 4MB RAM for each Oracle Tuxedo system server

## 7.9.2 Available Oracle Tuxedo 22c Release Install Types for Red Hat Enterprise Linux 8 (64-bit)

- Full install type
- Server install type
- Client install type
- Custom install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

## 7.9.3 Software Requirements for Red Hat Enterprise Linux 8 (64-bit)

Component	Requirement
OS Version (Patches)	Red Hat Enterprise Linux 8 x86_64 (64-bit)
C/C++ Compilers	C/C++: gcc version 8.5.0 20210514 (Red Hat 8.5.0-4.0.1)
COBOL compiler	Visual COBOL 6.0
Java Runtime Environment (JRE)	Tested with JRE: 1.8.0_341, 11.0.15.1, 17.0.4 and OpenJDK 17
Java Development Kit (SDK)	Tested with JDK 1.8.0_341, 11.0.15.1, and 17.0.4 and OpenJDK 17
Database	Tested with Oracle Client 12c, 19c and 21c

 **Note:**

- RHEL 8 no longer includes `libnsl`. Thus, you must install `libnsl` before you install Oracle Tuxedo 22c.
- 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 compiles with the newer version of compiler without requiring any changes.
  - Executables and libraries:  
Libraries and executables built using the tested version runs with libraries and executables that are built with the newer version of compilers.

## 7.9.4 Disk Space Requirements for Red Hat Enterprise Linux 8 (64-bit)

For all install types and components, 578MB is the minimum disk space requirement for installation on Red Hat Enterprise Linux 8 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.9.5 Temporary Storage Space Requirements for Red Hat Enterprise Linux 8 (64-bit)

For all install types and components, 107MB is the minimum temporary storage space requirement for installation on Red Hat Enterprise Linux 8 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.9.6 Tunable Parameters for Red Hat Enterprise Linux 8 (64-bit)

See `sysctl` manpage.

## 7.10 Red Hat Enterprise Linux 9 (64-bit) on x86-64

Oracle Tuxedo 22c now supports Red Hat Enterprise Linux 9.

The following sections list Oracle Tuxedo 22c Release requirements for Red Hat Enterprise Linux 9.

- [Software Requirements for Red Hat Enterprise Linux 9 \(64-bit\)](#)

## 7.10.1 Software Requirements for Red Hat Enterprise Linux 9 (64-bit)

Component	Requirement
OS Version (Patches)	Red Hat Enterprise Linux 9 x86_64 (64-bit)
C/C++ Compilers	C/C++: gcc (GCC) 11.2.1 20220127 (Red Hat 11.2.1-9.4.0.2)
COBOL Compiler	Visual COBOL 6.0
Java Runtime Environment (JRE)	Tested with JRE: 1.8.0_341, and 17.0.4
Java Development Kit (SDK)	Tested with JDK 1.8.0_341, and 17.0.4
Database	Tested with Oracle Client 19c

## 7.11 SUSE Linux Enterprise Server 12.5 (64-bit) on x86-64

The following sections list Oracle Tuxedo 22c Release requirements for SUSE Linux Enterprise Server 12.5 (64-bit) on x86-64.

- [Available Oracle Tuxedo 22c Release Install Types for SUSE Linux Enterprise Server 12.5 \(64-bit\)](#)
- [Hardware Requirements for SUSE Linux Enterprise Server 12.5 \(64-bit\)](#)
- [Software Requirements for SUSE Linux Enterprise Server 12.5 \(64-bit\)](#)
- [Network Requirements for SUSE Linux Enterprise Server 12.5 \(64-bit\)](#)
- [Disk Space Requirements for SUSE Linux Enterprise Server 12.5 \(64-bit\)](#)
- [Temporary Storage Space Requirements for SUSE Linux Enterprise Server 12.5 \(64-bit\)](#)
- [Tunable Parameters for SUSE Linux Enterprise Server 12.5 \(64-bit\)](#)

### 7.11.1 Available Oracle Tuxedo 22c Release Install Types for SUSE Linux Enterprise Server 12.5 (64-bit)

- Full install type
- Server install type
- Client install type
- Custom install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

### 7.11.2 Hardware Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)

- X86-64
- 1 GB of RAM minimum
- At least 4MB RAM for each Oracle Tuxedo system server

### 7.11.3 Software Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)

**Table 7-4 Software Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)**

Component	Requirement
OS Version (Patches)	SUSE Linux Enterprise Server 12.5 x86_64 (64-bit)
C/C++ Compilers	C/C++: gcc version 4.8.5 (SUSE Linux)
Java Runtime Environment (JRE)	Tested with JRE 1.8.0_341, 11.0.15.1, and 17.0.4 and OpenJDK 17
Java Development Kit (SDK)	Tested with JDK 1.8.0_341, 11.0.15.1, and 17.0.4 and OpenJDK 17
Database	Tested with Oracle Client 12c, 19c and 21c

### 7.11.4 Network Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)

- TCP/IP, using the Sockets network interface

### 7.11.5 Disk Space Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)

For all install types and components, 578MB is the minimum disk space requirement for installation on SUSE Linux Enterprise Server 12.5 (64-bit). This requirement assumes installation of the default components for the selected install type.

### 7.11.6 Temporary Storage Space Requirements for SUSE Linux Enterprise Server 12.5 (64-bit)

For all install types and components, 107MB is the minimum temporary storage space requirement for installation on SUSE Linux Enterprise Server 12.5 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

### 7.11.7 Tunable Parameters for SUSE Linux Enterprise Server 12.5 (64-bit)

See `sysctl` manpage.

## 7.12 AlmaLinux 8 (64-bit) on x86-64

The following sections list Oracle Tuxedo 22c Release requirements for AlmaLinux 8 (64-bit) on x86-64.

- [Hardware Requirements for AlmaLinux 8 \(64-bit\)](#)
- [Available Oracle Tuxedo 22c Release Install Types for AlmaLinux 8 \(64-bit\)](#)

- [Software Requirements for AlmaLinux 8 \(64-bit\)](#)
- [Disk Space Requirements for AlmaLinux 8 \(64-bit\)](#)
- [Temporary Storage Space Requirements for AlmaLinux 8 \(64-bit\)](#)
- [Tunable Parameters for AlmaLinux 8 \(64-bit\)](#)

## 7.12.1 Hardware Requirements for AlmaLinux 8 (64-bit)

- x86-64
- 1 GB of RAM minimum
- At least 4MB RAM for each Oracle Tuxedo system server

## 7.12.2 Available Oracle Tuxedo 22c Release Install Types for AlmaLinux 8 (64-bit)

- Full install type
- Server install type
- Client install type
- Custom install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

## 7.12.3 Software Requirements for AlmaLinux 8 (64-bit)

Component	Requirement
OS Version (Patches)	AlmaLinux 8.6 x86_64 (64-bit)
C/C++ Compilers	C/C++: gcc version 8.5.0 20210514
Java Runtime Environment (JRE)	Tested with JRE: 1.8.0_341, 11.0.15.1, 17.0.4 and OpenJDK 17
Java Development Kit (SDK)	Tested with JDK 1.8.0_341, 11.0.15.1, and 17.0.4 and OpenJDK 17
Database	Tested with Oracle Client 12c, 19c and 21c

 **Note:**

- AlmaLinux 8 does not include `libnsl`. Thus, you must install `libnsl` before you install Oracle Tuxedo 22c.
- 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:
  - Source code:  
C/C++ source code that was compiled and linked using the tested version compiles with the newer version of compiler without requiring any changes.
  - Executable and libraries:  
Libraries and executable built using the tested version runs with libraries and executable that are built with the newer version of compilers.

## 7.12.4 Disk Space Requirements for AlmaLinux 8 (64-bit)

For all install types and components, 578MB is the minimum disk space requirement for installation on AlmaLinux 8 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.12.5 Temporary Storage Space Requirements for AlmaLinux 8 (64-bit)

For all install types and components, 107MB is the minimum temporary storage space requirement for installation on AlmaLinux 8 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

## 7.12.6 Tunable Parameters for AlmaLinux 8 (64-bit)

See `sysctl` manpage.

## 7.13 IBM AIX 7.x (64-bit)

The following sections list Oracle Tuxedo 22c Release requirements for IBM AIX 7.x (64-bit)

 **Note:**

The information for Tuxedo 22c on IBM AIX 7.x is applicable for IBM AIX 7.2, and 7.3.

**Topics:**

- [Available Oracle Tuxedo 22c Install Types for IBM AIX 7.x \(64-bit\)](#)
- [Hardware Requirements for IBM AIX 7.x \(64-bit\)](#)

- [Software Requirements for IBM AIX 7.x \(64-bit\)](#)
- [Network Requirements for IBM AIX 7.x \(64-bit\)](#)
- [Disk Space Requirements for IBM AIX 7.x \(64-bit\)](#)
- [Temporary Storage Space Requirements for IBM AIX 7.x \(64-bit\)](#)
- [Tunable Parameters for IBM AIX 7.x \(64-bit\)](#)

### 7.13.1 Available Oracle Tuxedo 22c Install Types for IBM AIX 7.x (64-bit)

- Full install type
- Server install type
- Client install type
- Custom install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

### 7.13.2 Hardware Requirements for IBM AIX 7.x (64-bit)

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

### 7.13.3 Software Requirements for IBM AIX 7.x (64-bit)

Component	Requirement
OS Version (Patches)	IBM AIX 7.2 or 7.3, 64-bit
C/C++ Compilers	IBM OPEN XL C/C++ 17.1.1
Java Runtime Environment (JRE)	Tested with IBM JRE 1.8.0_331, 11.0.17.0 and 17.0.5.0
Java Software Development Kit (SDK)	Tested with IBM JDK 1.8.0_331, 11.0.17.0 and 17.0.5.0
Database	Tested with Oracle Client 19c

 **Note:**

- Add the bin directory of XL C/C++ 17.1.1 to the very beginning of PATH and the lib of XL C/C++ 17.1.1 to the very beginning of LIBPATH
- Open XL C/C++ compiler 17.1.1 supports C standard C99 by default. If you want to use C standard C90, compile option "-std=c90" should be specified
- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker
  - Oracle Tuxedo 22c on AIX uses new style of std C++ header files because Open XL C/C++ 17.1.1 only supports new style of std C++ header files.
- If you are using the Oracle Tuxedo IPv6 feature with Oracle JOLT, you must enable `java.net.preferIPv6Addresses` and disable `java.net.preferIPv4Stack` using the following java arguments at the same time:
  - `-Djava.net.preferIPv6Addresses=true`
  - `-Djava.net.preferIPv4Stack=false`
- 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 compiler 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.

### 7.13.4 Network Requirements for IBM AIX 7.x (64-bit)

- TCP/IP, using the Sockets network interface

### 7.13.5 Disk Space Requirements for IBM AIX 7.x (64-bit)

For all install types and components, 563MB is the minimum disk space requirement for installation on IBM AIX 7.x (64-bit) IBM PowerPC systems. This requirement assumes installation of the default components for the selected install type.

### 7.13.6 Temporary Storage Space Requirements for IBM AIX 7.x (64-bit)

For all install types and components, 113MB is the minimum temporary storage space requirement for installation on IBM AIX 7.x (64-bit) IBM PowerPC systems. This requirement assumes installation of the default components for the selected install type.



## 7.13.7 Tunable Parameters for IBM AIX 7.x (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 `chdev -l sys0 -a maxuproc=",,,,"` command, or enter `smit chgsys` and select maximum number of processes allowed per user.

## 7.14 Microsoft Windows 10 (32-bit, Client-only) on x86-64

Oracle Tuxedo 22c Release 1 (22.1.0.0.0) 32-bit client-only supports Microsoft Windows 10 on x86-64. The requirements to install 32-bit client-only on Microsoft Windows 10 on x86-64 are the same as Microsoft Windows Server 2019 on x86-64 (32-bit, client-only).



### Note:

Microsoft Windows 10 operating system must be installed in 64-bit mode.



### See Also:

[Microsoft Windows Server 2019 \(32-bit, Client-only\) on x86-64](#)

## 7.15 Microsoft Windows 11 (32-bit, Client-only) on x86-64

Oracle Tuxedo 22c Release 1 (22.1.0.0.0) 32-bit client-only supports Microsoft Windows 11 on x86-64. The requirements to install 32-bit, client-only on Microsoft Windows 11 on x86-64 are the same as Microsoft Windows Server 2019 on x86-64 (32-bit, client-only).



### See Also:

[Microsoft Windows Server 2019 \(32-bit, Client-only\) on x86-64](#)

## 7.16 Microsoft Windows Server 2019 (32-bit, Client-only) on x86-64

The following sections list Oracle Tuxedo 22c Release 1 (22.1.0.0.0) requirements for 32-bit client-only on Microsoft Windows Server 2019 on x86-64.

- [Available Oracle Tuxedo 22c Release Install Types for Microsoft Windows Server 2019 \(32-bit\)](#)
- [Hardware Requirements for Microsoft Windows Server 2019 \(32-bit\)](#)
- [Software Requirements for Microsoft Windows Server 2019 \(32-bit\)](#)
- [Disk Space Requirements for Microsoft Windows Server 2019 \(32-bit\)](#)
- [Network Requirements for Microsoft Windows Server 2019 \(32-bit\)](#)

## 7.16.1 Available Oracle Tuxedo 22c Release Install Types for Microsoft Windows Server 2019 (32-bit)

Oracle Tuxedo 22c Release 1 (22.1.0.0.0) supports only the Client install type for Microsoft Windows Server 2019 (32-bit client-only).



### Note:

Even though Oracle Tuxedo installer allows other install types, only Client install type is supported.

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

## 7.16.2 Hardware Requirements for Microsoft Windows Server 2019 (32-bit)

- X86-64
- 1 GB of RAM minimum

## 7.16.3 Software Requirements for Microsoft Windows Server 2019 (32-bit)

Component	Requirement
OS Version (Patches)	Windows Server 2019 (64-bit)
C/C++ Compilers	Microsoft Visual Studio 2015
COBOL Compilers	Micro Focus Visual COBOL 7.0
.Net Framework	Tested with .Net Framework 4.8
Java/JRE for the Java run-time environment	Tested with 1.8.0_341, 11.0.17 and 17.0.5
Java Development Kit (SDK)	Tested with 1.8.0_341, 11.0.17 and 17.0.5
Database	Tested with Oracle Client 19c

 **Note:**

- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- When using `buildnetclient` to build 32-bit .net client, the following flags are needed: `-csflag "/platform:x86"`
- If Micro Focus Visual COBOL 7.0 compiler and Microsoft 2022 C/C++ compiler are used, the `-l -link /SUBSYSTEM:CONSOLE /ENTRY:<entry>` link option should be used when compiling a COBOL client. For example, `buildclient -C -o CSIMPCL -f CSIMPCL.obj -f cbllds.obj -l "-link /SUBSYSTEM:CONSOLE /ENTRY:mainCRTStartup`
- 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 compiler 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.

## 7.16.4 Disk Space Requirements for Microsoft Windows Server 2019 (32-bit)

491MB is the minimum disk space requirement for client install type.

## 7.16.5 Network Requirements for Microsoft Windows Server 2019 (32-bit)

TCP/IP provided by the operating system.

## 7.17 Microsoft Windows Server 2019 (64-bit) on x86-64

The following sections list Oracle Tuxedo 22c Release 1 (22.1.0.0.0) requirements for Microsoft Windows Server 2019 (64-bit) on x86-64 with Intel/AMD 64-bit Processor.

- [Available Oracle Tuxedo 22c Release Install Types for Microsoft Windows Server 2019 \(64-bit\)](#)
- [Hardware Requirements for Microsoft Windows Server 2019 \(64-bit\)](#)
- [Software Requirements for Microsoft Windows Server 2019 \(64-bit\)](#)
- [Network Requirements for Microsoft Windows Server 2019 \(64-bit\)](#)

- [Disk Space Requirements for Microsoft Windows Server 2019 \(64-bit\)](#)
- [Tunable Parameters for Microsoft Windows Server 2019 \(64-bit\)](#)

### 7.17.1 Available Oracle Tuxedo 22c Release Install Types for Microsoft Windows Server 2019 (64-bit)

- Full install type
- Server install type
- Client install type
- Custom install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

### 7.17.2 Hardware Requirements for Microsoft Windows Server 2019 (64-bit)

- X86-64
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

### 7.17.3 Software Requirements for Microsoft Windows Server 2019 (64-bit)

Component	Requirement
OS Version (Patches)	Windows Server 2019 64bit
C/C++ Compilers	Microsoft Visual Studio 2022
COBOL Compilers	Micro Focus Visual COBOL 7.0
.Net Framework	Tested with .Net Framework 4.8
Java/JRE for the Java run-time environment	Tested with JDK/JRE 1.8.0_341, 11.0.17 and 17.0.5
Java Software Development Kit (SDK) for the Java development environment	Tested with JDK1.8.0_341, 11.0.17 and 17.0.5
Database	Tested with Oracle Client 19c

 **Note:**

- 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.
- If Micro Focus Visual COBOL 7.0 compiler and Microsoft 2022 C/C++ compiler are used, the `-l -link /SUBSYSTEM:CONSOLE /ENTRY:<entry>` link option should be used when compiling a COBOL client. For example, `buildclient -C -o CSIMPCL -f CSIMPCL.obj -f cbllds.obj -l "-link /SUBSYSTEM:CONSOLE /ENTRY:mainCRTStartup`
- 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 compiler 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.

## 7.17.4 Network Requirements for Microsoft Windows Server 2019 (64-bit)

- TCP/IP provided by Microsoft Windows Server 2019 (64-bit) on x86-64 with Intel/AMD 64-bit processor.

## 7.17.5 Disk Space Requirements for Microsoft Windows Server 2019 (64-bit)

For all install types and components, 672MB is the minimum disk space requirement for installation on Microsoft Windows Server 2019 (64-bit) on x86-64 with Intel/AMD64-bit processor. This requirement assumes installation of the default components for the selected install set.

## 7.17.6 Tunable Parameters for Microsoft Windows Server 2019 (64-bit)

Before running the Oracle Tuxedo software, you may need to reconfigure the interprocess communication (IPC) parameters shown in the figure titled [Oracle Administration Window with IPC Resources Page Displayed](#) in *Using Oracle Tuxedo ATMI on Windows*. For instructions about reconfiguring the IPC parameters, see

[Configuring IPC Resources to Maximize System Performance](#) in *Using Oracle Tuxedo ATMI on Windows* and [Checking IPC Requirements](#).

## 7.18 Microsoft Windows Server 2022 (64-bit) on x86-64

Oracle Tuxedo 22c Release 1 (22.1.0.0.0) supports Microsoft Windows Server 2022 (64-bit) on x86-64. The requirements to install Microsoft Windows Server 2022 (64-bit) on x86-64 with Intel/AMD 64-bit Processor are same as Microsoft Windows Server 2019 (64-bit) on x86-64.



### See Also:

[Microsoft Windows Server 2019 \(64-bit\) on x86-64](#)

## 7.19 HP-UX 11i v3 (64-bit) on Itanium

The following sections list Oracle Tuxedo 22c Release requirements for HP-UX 11i v3 (64-bit) system on Itanium.

- [Available Oracle Tuxedo 22c Install Types for HP-UX 11i v3 \(64-bit\)](#)
- [Hardware Requirements for HP-UX 11i v3 \(64-bit\) on Itanium](#)
- [Software Requirements for HP-UX 11i v3 \(64-bit\) on Itanium](#)
- [Network Requirements for HP-UX 11i v3 \(64-bit\) on Itanium](#)
- [Disk Space Requirements for HP-UX 11i v3 \(64-bit\) on Itanium](#)
- [Temporary Storage Space Requirements for HP-UX 11i v3 \(64-bit\) on Itanium](#)
- [Tunable Parameters for HP-UX 11i v3 \(64-bit\) on Itanium](#)

### 7.19.1 Available Oracle Tuxedo 22c Install Types for HP-UX 11i v3 (64-bit)

- Full install type
- Server install type
- Client install type

For descriptions of Oracle Tuxedo Install Types, see [Install Types](#).

### 7.19.2 Hardware Requirements for HP-UX 11i v3 (64-bit) on Itanium

- Intel Itanium (64-bit)
- 1 GB of RAM minimum
- At least 4MB RAM for each Oracle Tuxedo system server

### 7.19.3 Software Requirements for HP-UX 11i v3 (64-bit) on Itanium

Component	Requirement
OS Version (Patches)	HP-UX B.11.31 ia64 (64-bit)
C/C++ Compilers	HP C/aC++ B3910B A.06.29 [Oct 18 2016]
Java 2 Runtime Environment (JRE)	Tested with Java 1.8.0.26-hp-ux
Java 2 Software Development Kit (SDK)	Tested with Java 1.8.0.26-hp-ux
Database for CORBA C++ applications	Tested with Oracle Database 19c

 **Note:**

- The aCC C/C++ compiler supports C standard C99 by default. To use C++11, you need to specify "+std=c++11" in the compile option.
- When using certain Oracle Tuxedo features related to parsing XML, it is crucial to set the `LANG` or `LC_All` variables to `en_US.utf8`. Failure to do so may result in the display of an error message stating "Could not load a transcoding service." Additionally, some features may cease to function as intended.

### 7.19.4 Network Requirements for HP-UX 11i v3 (64-bit) on Itanium

TCP/IP, using the Sockets network interface.

### 7.19.5 Disk Space Requirements for HP-UX 11i v3 (64-bit) on Itanium

For all install types and components, 884MB is the minimum disk space requirement for installation on HP-UX 11i v3 (64-bit) Itanium systems. This requirement assumes installation of the default components for the selected install type.

### 7.19.6 Temporary Storage Space Requirements for HP-UX 11i v3 (64-bit) on Itanium

For all install types and components, 138MB is the minimum temporary storage space requirement for installation on HP-UX 11i v3 (64-bit) Itanium systems. This requirement assumes installation of the default components for the selected install type.

### 7.19.7 Tunable Parameters for HP-UX 11i v3 (64-bit) on Itanium

For HP-UX configuration instructions, see the `kcweb` manpage or the kernel configuration section in the `sam` command.

# 8

## Oracle Jolt 22c Release Overview and Installation Information

The following sections provide an overview of Oracle Jolt 22c Release and present the prerequisites and preparatory information for installing the Oracle Jolt 22c Release software components:

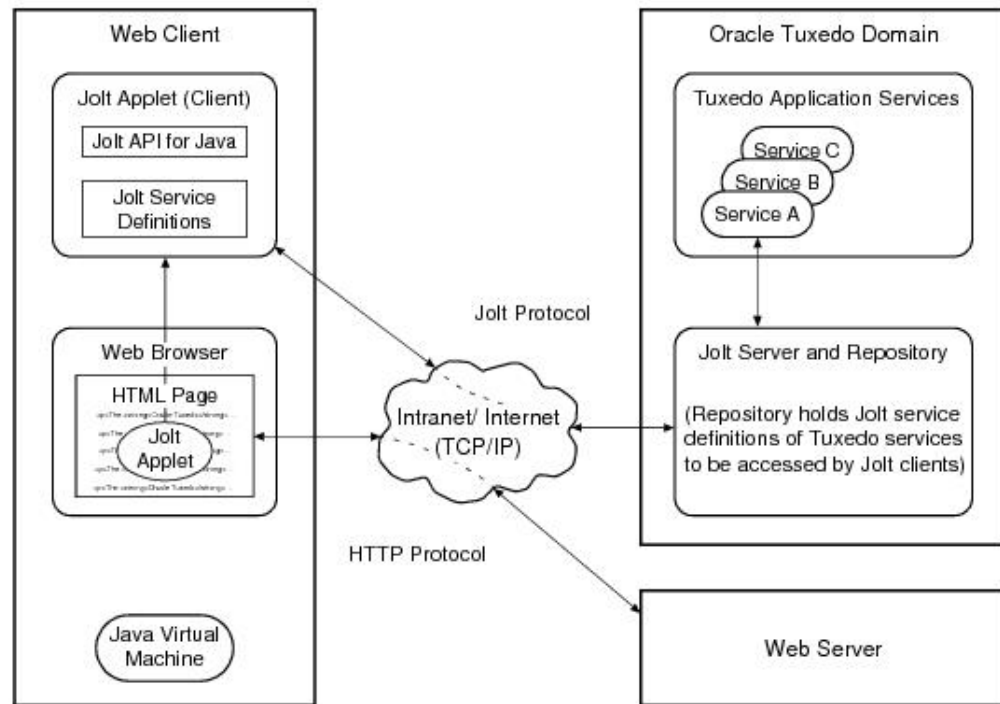
- [About Oracle Jolt](#)
- [Oracle Jolt Components](#)
- [Jolt Server Listener](#)
- [Jolt Server Handler](#)
- [Metadata Repository Server](#)
- [Metadata Repository](#)
- [Jolt Internet Relay](#)
- [Jolt Class Library](#)
- [JoltBeans](#)
- [Jolt ECID](#)
- [Oracle Jolt Client Personalities](#)
- [Oracle Jolt Capabilities](#)
- [Oracle Jolt Client Support](#)
- [Oracle Jolt Client Requirements](#)
- [Oracle Jolt Client Class Library](#)
- [Oracle Jolt Release Interoperability](#)
- [Oracle Jolt Documentation](#)

### 8.1 About Oracle Jolt

Oracle Jolt is a Java class library and API that enables remote Java clients to access existing Oracle Tuxedo ATMI services. As shown in the following figure, Oracle Jolt extends the functionality of existing Tuxedo ATMI applications to include intranet- and Internet-wide availability.



**Figure 8-1 Oracle Jolt Communication Architecture Using the Jolt Applet**



Note: Web server and Jolt server must be running on the same computer machine.

The preceding figure shows one of five types of Jolt client personalities supported by Oracle Jolt, all of which are briefly described in [Oracle Jolt Client Personalities](#).

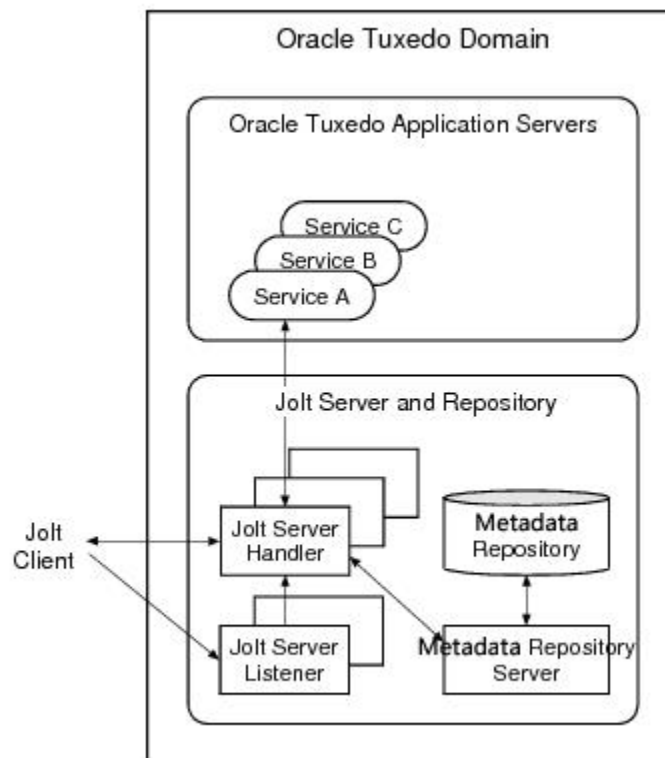
## 8.2 Oracle Jolt Components

Oracle Jolt consists of the following components for enabling secure, reliable access to servers inside corporate firewalls, and for creating Java-based client programs that access Tuxedo ATMI services:

- [Jolt Server Listener](#)
- [Jolt Server Handler](#)
- [Metadata Repository Server](#)
- [Metadata Repository](#)
- [Jolt Internet Relay](#)
- [Jolt Class Library](#)
- [JoltBeans](#)

As shown in the following figure, the Jolt server implementation consists of one or more Jolt Server Handlers, one or more Jolt Server Listeners, and one and only one Metadata Repository Server, all running on the same Oracle Tuxedo server machine.

Figure 8-2 Oracle Jolt Server Implementation



A Jolt server listens for network connections from Jolt clients, translates Jolt messages, multiplexes multiple Jolt clients into a single process, and submits and retrieves requests to and from a Tuxedo ATMI application. As with all Tuxedo system executables, the Jolt server components reside in the `tux_prod_dir/bin` directory, where `tux_prod_dir` represents the directory in which the Oracle Tuxedo 22c Release 1 (22.1.0.0) Oracle distribution is installed.

## 8.3 Jolt Server Listener

A Jolt Server Listener (JSL) is a listening process, running on the Tuxedo server, that accepts connection requests from Jolt clients and assigns connections to a Jolt Server Handler also running on the Tuxedo server. It also manages the pool of Jolt Server Handler processes, starting them in response to load demands.

## 8.4 Jolt Server Handler

A Jolt Server Handler (JSH) is a gateway process, running on the Tuxedo server, that handles communications between Jolt clients and the Tuxedo ATMI server application. A JSH process resides within the administrative domain of the application and is registered in the local Tuxedo bulletin board as a client. Each JSH process can manage multiple Jolt clients. A JSH multiplexes all requests and replies with a particular Jolt client over a single connection.

## 8.5 Metadata Repository Server

TMMETADATA server retrieves Jolt service definitions from the Tuxedo Service Metadata Repository and returns the service definitions to the JSH. The TMMETADATA server also updates or adds Jolt service definitions.

## 8.6 Metadata Repository

The Metadata Repository, located on the Tuxedo server, is a central repository that contains definitions of Tuxedo ATMI services. These repository definitions are used by Jolt at run time to access Tuxedo services. You can export services to a Jolt client application or unexport services by hiding the definitions from the Jolt client. Using the Oracle Tuxedo Services Console, you can test new and existing Tuxedo services independently of the client applications.

## 8.7 Jolt Internet Relay

Jolt Internet Relay routes messages from a Jolt client to a Jolt Server Listener (JSL) or Jolt Server Handler (JSH). It eliminates the need for the JSL, JSH, and Tuxedo application to run on the same machine as the Web server. The Jolt Internet Relay consists of the following components:

### **Jolt Relay (JRLY)**

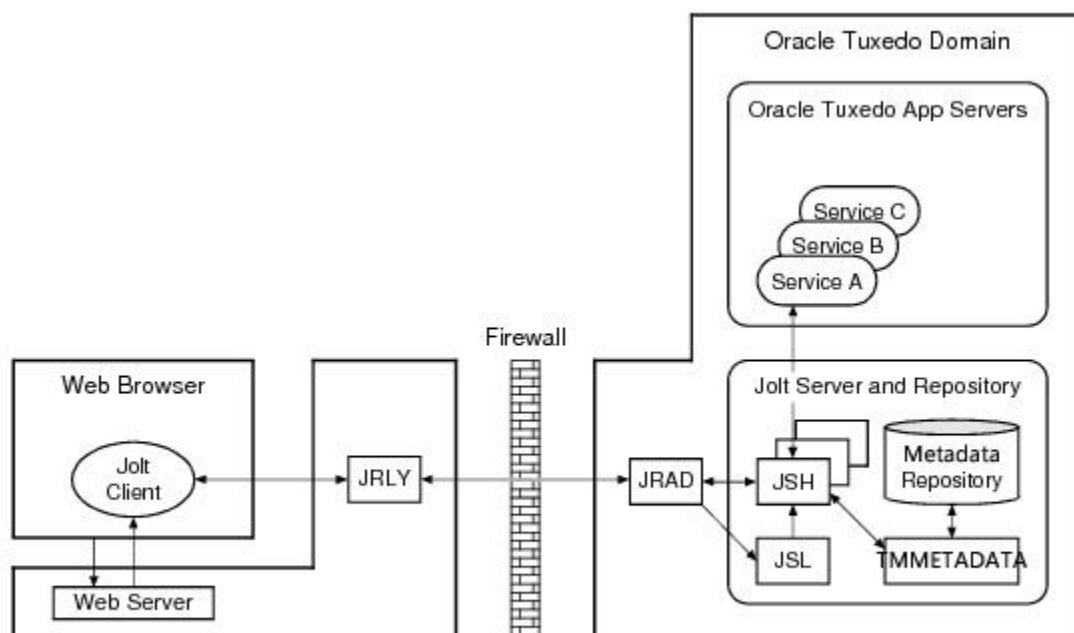
A standalone program that routes Jolt messages from Jolt clients to the JSL or JSH via the Jolt Relay Adapter (JRAD). Jolt Relay is not a Tuxedo system server or client process, and it is not dependent on the Oracle Tuxedo system software version.

### **Jolt Relay Adapter (JRAD)**

A Tuxedo system server that operates as a back-end relay for Jolt Relay. The JRAD may or may not be located on the same Tuxedo host machine and server group to which the JSL server is connected.

The following image illustrates the Jolt Internet Relay connection path

Figure 8-3 Oracle Jolt Internet Relay Connection Path



Note: Web server software and JRLY process must be running on the same computer machine.

A Jolt server can connect directly to intranet Jolt clients and can connect indirectly to Internet Jolt clients through the Jolt Internet Relay, all at the same time. Jolt Internet Relay is transparent to Jolt servers and Jolt clients.

The JRLY executable (`jrly`) resides in `tux_prod_dir/udataobj/jolt/relay` and `tux_prod_dir/bin` directories.

## 8.8 Jolt Class Library

The Jolt class library consists of Java class files that implement the Jolt API. These classes enable Java clients to invoke Oracle Tuxedo ATMI services. The Jolt class library provides functions to set, retrieve, manage, and invoke communication attributes, notifications, network connections, transactions, and services.

The Jolt class library files reside in the `tux_prod_dir/udataobj/jolt` directory and are stored in the following JAR files:

- `jolt.jar`
- `joltadmin.jar`
- `joltatm.jar`
- `JoltBeanDev.jar`
- `JoltBeanRt.jar`
- `JoltBeanDevAwt.jar`
- `JoltBeanRtAwt.jar`

- JoltBeanDevSwing.jar
- JoltBeanRtSwing.jar
- JoltBeanDevSwing11.jar
- JoltBeanRtSwing11.jar
- joltjse.jar
- joltwls.jar

To view the contents of a Jolt JAR file, ensure that the path to the JDK 1.8.x (or higher) software is included at the front of your PATH variable, go to the `tux_prod_dir/udataobj/jolt` directory, and enter the following command:

```
prompt> jar -tvf filename.jar
```

### Example

```
prompt> jar -tvf jolt.jar
 7472 Fri Sep 09 17:16:40 EDT 2022 META-INF/MANIFEST.MF
 7518 Fri Sep 09 17:16:42 EDT 2022 META-INF/ORACLE_C.SF
11546 Fri Sep 09 17:16:42 EDT 2022 META-INF/ORACLE_C.RSA
    0 Sat Sep 10 00:16:22 EDT 2022 META-INF/
  547 Sat Sep 10 00:16:22 EDT 2022 bea/jolt/ApplicationException.class
  786 Sat Sep 10 00:16:22 EDT 2022 bea/jolt/BData.class
  991 Sat Sep 10 00:16:22 EDT 2022 bea/jolt/ByteArrayUtil.class
.
.
.
```

## 8.9 JoltBeans

JoltBeans provides a JavaBeans-compliant interface to Oracle Jolt. JoltBeans are Bean components that can be used in JavaBeans-enabled integrated development environments (IDEs) to construct Jolt clients.

JoltBeans consists of two sets of Java Beans: JoltBeans toolkit (a JavaBeans-compliant interface to Oracle Jolt that includes the JoltServiceBean, JoltSessionBean, and JoltUserEventBean) and Jolt GUI beans, which consist of Jolt-aware Abstract Window Toolkit (AWT) and Swing-based beans. The separation of Oracle Jolt into these components permits the transactional and Internet components of client/server applications to be implemented separately with the security and scalability required for large-scale Internet and intranet services.

## 8.10 Jolt ECID

Jolt call process is as follows: JOLT client --> JSL/JSH --> tuxedo server --> service

1. Jolt Connection Pool supports ECID propagation between Weblogic and Oracle Tuxedo. ECID is propagated and inserted into a request message if java option `-Dtuxedo.ECID_ENABLE` or environment variable `ECID_ENABLE` is set to `y`. Java option `-Dtuxedo.ECID_ENABLE` has higher priority than environment variable

*ECID\_ENABLE*. ECID also can be sent back within a reply message from Oracle Tuxedo.

2. ECID relies on DMS (Dynamic Monitoring Service), ECID is not supported for standalone jolt client by default. If you want ECID to be created, DMS should be installed first.
3. Jolt ECID does not impact JRJLY and JRAD. The format for ECID in jolt client trace is as follows:

```
000915:30475@slc05are:1: ECID <0000KqPMc659XbHpIsT4if1LPfwR000001>:  
atmi: { JoltSession.send(len 132)}
```

## 8.11 Oracle Jolt Client Personalities

In addition to using Oracle Jolt to build client applets and applications that remotely invoke existing and new Tuxedo applications, Java programmers can use Oracle Jolt to build HTTP servlets to perform server-side Java tasks in response to HTTP requests. This latter type of Jolt connectivity enables simple Web clients to access Tuxedo application services through any Web application server that supports generic servlets.

Oracle Jolt supports the following types of Java client personalities:

- Jolt applet—a downloadable Java interactive program, running in a Web browser environment, that uses the Jolt classes to invoke Tuxedo services from within a Web page. Accessing Tuxedo ATMI services in this manner requires the downloading and installation of Jolt class package `jolt.jar` (and possibly other Jolt class packages `exceptjoltjse.jar`, and `joltwls.jar` on the machine running the Jolt applet.
- Jolt application—a standalone Java client, running on the Java Virtual Machine and operating system of a client machine, that uses the Jolt classes to invoke Tuxedo services. Accessing Tuxedo ATMI services in this manner requires the installation of Jolt class package `jolt.jar` (and possibly `joltadmin.jar`) on the machine running the Jolt application.
- JSE Connectivity for Oracle Tuxedo—a Jolt HTTP servlet, running in a Java Web application server environment (for example, Oracle WebLogic Server), through which simple Web-browser clients can invoke Tuxedo ATMI services. Accessing Tuxedo ATMI services in this manner requires the installation of Jolt class packages `jolt.jar` and `joltjse.jar` on the machine running the Web application server.  
A Jolt HTTP servlet uses Jolt session pool classes to invoke Tuxedo services on behalf of simple browser clients. Thus, the servlet handles all Jolt transactions on the Web server, which enables simple browser clients to invoke Oracle Tuxedo services without directly connecting to the Jolt server and Oracle Tuxedo.
- WebLogic Connectivity for Oracle Tuxedo—a customized version of Jolt JSE Connectivity for the Oracle WebLogic Server. Accessing Tuxedo ATMI services in this manner requires the installation of Jolt class packages `jolt.jar`, `joltjse.jar`, and `joltwls.jar` on the machine running Oracle WebLogic Server.  
The Jolt client personality WebLogic Connectivity for Oracle Tuxedo is also known as Oracle Jolt for Oracle WebLogic Server.

## 8.12 Oracle Jolt Capabilities

Running as a Java applet or as a standalone Java client application, Oracle Jolt supports the following capabilities:

- Transaction context propagation

- Access to asynchronous Oracle Tuxedo event notifications
- Typed buffer support  
Oracle Jolt automatically converts Java messages to native Oracle Tuxedo data types and buffers, and converts Oracle Tuxedo data types and buffers back to Java messages.
- Jolt session pooling  
Oracle Jolt supports the construction and use of Jolt session (connection) pools to increase efficiency, availability, and reliability.
- Jolt session pooling reset
- Oracle Jolt support an option to reset a Jolt session pool without stopping the Jolt client, in the event of session pool failure. For example, if the Tuxedo server crashes or the Jolt Server Handler shuts down, the Jolt session pool may be reset without stopping the Jolt client.
- Jolt Trace  
If java option `-Dtuxedo.TMTRACE` is set, JOLT client prints out trace as the trace format. JOLT trace format is the same as Oracle Tuxedo. Four trace categories, "atmi", "inet", "trace" and "\*" are supported.  
  
To accommodate the Oracle Jolt 22c Release 1 (22.1.0.0.0) server (JSL, JSH, JREPSVR) and Jolt Internet Relay (JRLY, JRAD) components, your environment must provide 2 MB of disk space. For Oracle Jolt 22c Release 1 (22.1.0.0.0) system requirements, including supported platforms, see [Supported Platforms](#).

## 8.13 Oracle Jolt Client Support

As stated in [Oracle Jolt Client Personalities](#), Oracle Jolt 22c Release 1 (22.1.0.0.0) supports the following four client types:

- Jolt applet
- Jolt application
- JSE Connectivity for Oracle Tuxedo
- WebLogic Connectivity for Oracle Tuxedo

The following table lists the requirements for the Oracle Jolt 22c Release 1 (22.1.0.0.0) applet and application client types:

**Table 8-1 Requirements for Oracle Jolt Oracle Jolt 22c Release 1 (22.1.0.0.0) Applet and Application Client Types**

Client Type	Platform Supported
Jolt application—standalone application—running in a desktop environment	Java Development Kit (JDK) 1.8 In order to use latest Java versions, see <a href="#">Note for A Note on Support for Newer Versions of Java</a> .

The following table lists the requirements for the remaining three Jolt client types, all of which are implemented by HTML-based Jolt client classes running inside a Web server. The Web servers certified with Oracle Jolt 22c Release 1 (22.1.0.0.0) are listed in the table.

**Table 8-2 Requirements for Web-Server based Oracle Jolt 22c Release 1 (22.1.0.0.0) Client Types**

Vendor	Web Server Version	OS Version	Oracle Jolt Client Personality	Comments
Any vendor	Java Servlet Engine (Any version)	Any Supported OS running JDK 1.8	JSE Connectivity for Oracle Tuxedo	For a description, see the first note after this table.
Oracle Systems, Inc.	Oracle WebLogic Server 11g or higher	Any	WebLogic Connectivity for Oracle Tuxedo (also known as "Oracle Jolt for Oracle WebLogic Server")	For example installation instructions, see <a href="#">Installing Oracle Jolt 22c with Oracle WebLogic Server</a> .

 **Note:**

JSE Connectivity for Oracle Tuxedo is the name of the Jolt Web application server that simplifies the handling of servlets in an Oracle Tuxedo application environment. JSE is short for Java Servlet Engine.

**Related Topics**

- [Installing Oracle Jolt 22c with Oracle WebLogic Server](#)

## 8.14 Oracle Jolt Client Requirements

Oracle Jolt has the following client requirements:

- Java Development Kit is certified for Oracle Jolt 22c. For more information, see [Oracle Jolt Client Class Library](#).
- Java-enabled browser (Internet Explorer) or Java Virtual Machine (JVM)

## 8.15 Oracle Jolt Client Class Library

Various implementations of Java tend to show minor differences in characteristics. Oracle Jolt 22c Release 1 (22.1.0.0.0) has been certified with JDK1.8.0\_341, JDK-17.0.4, JDK 11.0.15.1 and OpenJDK 17.

## 8.16 Oracle Jolt Release Interoperability

An Oracle Jolt 22c Release 1 (22.1.0.0.0) client can interoperate with an Oracle Jolt 9.0, 9.1, 10.0, 10.3, 11.1.1.x, 12.1.1, 12.1.3 or 12.2.2 server implementation, and the client is able to use the new features available with Oracle Jolt 22c Release 1 (22.1.0.0.0).

An Oracle Jolt 9.0, 9.1, 10.0, 10.3, 11.1.1.x, 12.1.1, 12.1.3 or 12.2.2 client can interoperate with an Oracle Jolt 22c Release 1 (22.1.0.0.0) server implementation. However, only Oracle Jolt 9.0, 9.1, 10.0, 10.3, and 11.1.1.x, 12.1.1, 12.1.3, or 12.2.2 functionality is available to the



Oracle Jolt client, although the server-side components of Oracle Jolt 22c Release 1 (22.1.0.0.0) are added in the enhanced functionality.

## 8.17 Oracle Jolt Documentation

For more information about Oracle Jolt, see the following documentation:

- [Using Oracle Jolt](#)
- [Using Oracle Jolt with Oracle WebLogic Server](#)

# 9

## 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 an 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](#)

### 9.1 Introduction

The following discussions apply to UNIX system.

### 9.2 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 an 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 `awrite()` 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.

## 9.3 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.

## 9.4 How the Oracle Tuxedo Filesystem is Organized

An 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 an 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 tmdmin:
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. The following table lists the size of each element in the Oracle Tuxedo filesystem.

**Table 9-1 Size of Oracle Tuxedo System Tables**

Element	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.

## 9.5 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`.

## 9.6 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).

# 10

## IPC Resource Configuration on a UNIX System

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

- [Parameter Sets Controlling IPC Resources](#)

### 10.1 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 the following table:

**Table 10-1 Tunable Parameters**

Prefix of Tunable Parameters	Resources Controlled
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 an 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 22c Release Platform Data Sheets](#). If you change a parameter value, then must 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](#)
- [Semaphores](#)
- [Message Queues and Messages](#)
- [What Is Application Deadlock?](#)
- [Performance Implications of Blocking Conditions](#)
- [Tunable Message Parameters](#)
- [Other Kernel Tunables](#)

## 10.1.1 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 IIO 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.

### **SHMNI**

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 IIO listener (ISL) that is running.

### **SHMMIN**

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

## 10.1.2 Semaphores

Every process that participates in an 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:

### **SEMNS**

Maximum number of semaphores in the system. The minimum requirement for `SEMNS` 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](#) or [UBBCONFIG\(5\)](#) in the [File Formats, Data Descriptions, MIBs, and System Processes Reference](#).

### **SEMNI**

Maximum number of active semaphore sets.

**SEMMSL**

Maximum number of semaphores per semaphore set. `SEMMSL` and `SEMMSL` are commonly chosen so that their product equals `SEMMSL`. 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 `SEMMSL`.

**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 `SEMMSL`. (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.

### 10.1.3 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 shown in the table below:

**Table 10-2 Message Queue Parameters**

Parameter	Description
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 file name 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.

## 10.1.4 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.

## 10.1.5 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.

## 10.1.6 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 an 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.

## 10.1.7 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 22c Release Platform Data Sheets](#) 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. An 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.