

Oracle® Tuxedo

Installing the Oracle Tuxedo System

12c Release 2 (12.1.3)

April 2014

ORACLE®

Installing the Oracle Tuxedo System, 12c Release 2 (12.1.3)

Copyright © 1996, 2014, Oracle and/or its affiliates. All rights reserved.

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 or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on 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. 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.

Contents

1. Preparing to Install the Oracle Tuxedo System

Oracle Universal Installer (OUI)	1-1
Installing Oracle Tuxedo 12cR2 with Oracle Home	1-2
Oracle Home	1-2
Installing Oracle Tuxedo 12cR2	1-2
Removing Oracle Home	1-2
Determining the Default Oracle Home	1-2
Multiple Oracle Homes	1-3
Target Home	1-3
Oracle Universal Installer Inventory	1-3
Oracle Universal Installer Inventory Structure	1-4
Detaching Oracle Homes from the Central Inventory	1-7
Customizing and Creating Response Files	1-9
About Response Files	1-10
Creating a Response File With Record Mode	1-11
Response File Format	1-12
Response File Parameters	1-14
Installing with a Response File	1-21
Modes of Installation	1-24
Installation Media	1-24
Special Instructions for UNIX Users	1-24
Failed to Connect to Server Error	1-25

Providing a UNIX Installer Location with Root Privileges	1-25
Providing a UNIX Group Name	1-26
Deinstalling Oracle Products	1-26
Removing Oracle Products and Oracle Homes	1-26
Deinstalling Top-level Products With Dependents	1-27
Silent Deinstallation	1-27
About Oracle Universal Installer Log Files	1-28
Oracle Internationalization	1-29
Installation Dialogs Language	1-29
Oracle Tuxedo Web Distribution	1-29
Oracle Tuxedo Software Components	1-29
Tuxedo .NET Workstation Client	1-32
LLE and SSL Levels of Encryption	1-32
Hardware and Software Prerequisites	1-32
System Requirements	1-32
Temporary Storage Space Requirements	1-33
Interprocess Communication Resources Configuration	1-34
Oracle Installation Program	1-35
Previous Tuxedo Versions	1-35
Installation Methods	1-36
Cancelling Installation	1-36
Install Types	1-36
Oracle Home Directory	1-37
Choosing an Oracle Home Directory	1-37
Multiple Oracle Homes	1-38
Installation Road Map	1-38

2. Installing Oracle Tuxedo Using GUI-Mode Installation

What Is GUI-Mode Installation?	2-2
Before You Start	2-2
Starting GUI-Mode Installation on a Windows System	2-2
Starting GUI-Mode Installation on a UNIX System	2-3
Running GUI-Mode Installation	2-4
Assigning File Ownership on a UNIX System	2-10
Understanding the Oracle Tuxedo Shortcuts on a Windows System	2-10
Understanding the Oracle Administration Program on a Windows System	2-10
Reviewing the Windows Server Registry Content	2-11
Windows Service Change	2-11
Windows Control Panel Change	2-12
Earlier Version Detected	2-12
Developer Key	2-13
Environment Key	2-13
Security Key	2-13
What Do I Do Next?	2-14

3. Installing Oracle Tuxedo Using Silent Installation

What Is Silent Installation?	3-1
Before You Start	3-1
Using Silent Installation: Main Steps	3-2
Creating a Response File	3-2
Using Record Mode	3-2
Required Variables in Tuxedo Response File	3-3
Installing with a Response File	3-5
Specifying a Response File	3-5
Optional Parameters When Specifying a Response File	3-6

UNIX Template File	3-6
What Do I Do Next?	3-27

4. Upgrading the Oracle Tuxedo System to 12c Release 2 (12.1.3)

Preparing Your Machine for an Upgrade	4-1
Selecting an Upgrade Procedure	4-2
Backing Up Files	4-3
Performing a Simple Upgrade	4-3
Performing Additional Upgrade Steps to Windows Server	4-4
Upgrading from Tuxedo Previous Releases to Tuxedo 12c Release 2 (12.1.3).	4-5
Performing a Hot Upgrade	4-6
Pre-Upgrade Notes	4-6
Hot-Upgrade Procedure	4-7
Hot-Upgrade Example	4-8
Rebuilding an Application	4-9
Installing Oracle Jolt 12c Release 2 (12.1.3) with Oracle WebLogic Server	4-10

5. Performing Post-Installation Tasks

Understanding the Oracle Tuxedo Directory Structure	5-2
Understanding the Oracle Service Architecture Leveraging Tuxedo Directory Structure	5-6
Post Oracle TSAM Plus Installation Directory Structure	5-7
Oracle TSAM Plus Agent Directory Structure (For Linux)	5-8
Oracle TSAM Plus Manager Directory Structure (For Linux).	5-9
Understanding the Oracle Tuxedo Architecture.	5-11
UBBCONFIG File	5-12
MASTER Machine	5-12
TUXCONFIG File	5-12

TUXCONFIG Environment Variable	5-13
TUXDIR Environment Variable	5-13
Using the Registry to Further Restrict Access on a Windows System.	5-13
Setting Up Your Environment	5-14
Setting Environment Variables on a Windows System	5-16
Setting Environment Variables on a UNIX System.	5-17
Starting the tlisten Process	5-18
tlisten Invocation	5-19
tlisten Password	5-19
Editing a UBBCONFIG File	5-20
Using the TYPE Parameter in UBBCONFIG.	5-23
Checking IPC Requirements.	5-23
Creating the Universal Device List and the Transaction Log.	5-24
Defining the TLOG	5-25
Creating the UDL and UDL Entries for TLOG.	5-25
Running simpapp to Verify Your Installation	5-26
Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation	5-27
Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation on a Windows System.	5-27
Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation on a UNIX System	5-30
Running simpapp to Verify the Oracle Tuxedo CORBA C++ Software Installation	5-32
Running simpapp to Verify the Oracle Tuxedo CORBA C++ Software Installation on a Windows System.	5-33
Running simpapp to Verify the Oracle Tuxedo CORBA C++ Software Installation on a UNIX System	5-34
Running buildtms for Oracle Tuxedo Applications That Use XA Resource Managers	5-36

Uninstalling Oracle Tuxedo	5-36
Reinstalling Oracle Tuxedo	5-36

A. Oracle Tuxedo 12c Release 2 (12.1.3) Platform Data Sheets

Supported Platforms	A-3
Platform Support Policy	A-6
Tunable Parameters	A-7
Supported Platform Data Sheets.	A-7
HP-UX 11i v3 (32-bit) on Itanium 64-bit	A-7
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for HP-UX 11i v3 (32-bit) on Itanium 64-bit	A-7
Hardware Requirements for HP-UX 11i v3 (32-bit) on Itanium 64-bit	A-7
Software Requirements for HP-UX 11i v3 (32-bit) on Itanium 64-bit	A-8
Network Requirements for HP-UX 11i v3 (32-bit) on Itanium 64-bit	A-9
Disk Space Requirements for HP-UX 11i v3 (32-bit) on Itanium 64-bit	A-9
Tunable Parameters for HP-UX 11i v3 (32-bit) on Itanium 64-bit	A-9
HP-UX 11i v3 (64-bit) on Itanium	A-9
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for HP-UX 11i v3 (64-bit) on Itanium	A-9
Hardware Requirements for HP-UX 11i v3 (64-bit) on Itanium	A-10
Software Requirements for HP-UX 11i v3 (64-bit) on Itanium	A-10
Network Requirements for HP-UX 11i v3 (64-bit) on Itanium.	A-11
Disk Space Requirements for HP-UX 11i v3 (64-bit) on Itanium.	A-11
Tunable Parameters for HP-UX 11i v3 (64-bit) on Itanium	A-11
IBM AIX 6.1 (32-bit) on IBM PowerPC	A-11
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for IBM AIX 6.1 (64-bit)	A-11
Hardware Requirements for IBM AIX 6.1 (32-bit).	A-12

Software Requirements for IBM AIX 6.1 (32-bit)	A-12
Network Requirements for IBM AIX 6.1 (32-bit)	A-13
Disk Space Requirements for IBM AIX 6.1 (32-bit)	A-14
Tunable Parameters for IBM AIX 6.1 (32-bit)	A-14
IBM AIX 6.1 (64-bit) on IBM PowerPC	A-14
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for IBM AIX 6.1 (64-bit) on IBM PowerPC	A-14
Hardware Requirements for IBM AIX 6.1 (64-bit)	A-14
Software Requirements for IBM AIX 6.1 (64-bit)	A-15
Network Requirements for IBM AIX 6.1 (64-bit)	A-16
Disk Space Requirements for IBM AIX 6.1 (64-bit)	A-16
Temporary Storage Space Requirements for IBM AIX 6.1 (64-bit)	A-16
Tunable Parameters for IBM AIX 6.1 (64-bit)	A-17
IBM AIX 7.1 (32-bit) on IBM PowerPC	A-17
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for IBM AIX 7.1 (32-bit)	A-17
Hardware Requirements for IBM AIX 7.1 (32-bit)	A-17
Software Requirements for IBM AIX 7.1 (32-bit)	A-17
Network Requirements for IBM AIX 7.1 (32-bit)	A-18
Disk Space Requirements for IBM AIX 7.1 (32-bit)	A-19
Temporary Storage Space Requirements for IBM AIX 7.1 (32-bit)	A-19
Tunable Parameters for IBM AIX 7.1 (32-bit)	A-19
IBM AIX 7.1 (64-bit) on IBM PowerPC	A-19
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for IBM AIX 7.1 (64-bit)	A-19
Hardware Requirements for IBM AIX 7.1 (64-bit)	A-19
Software Requirements for IBM AIX 7.1 (64-bit)	A-20
Network Requirements for IBM AIX 7.1 (64-bit)	A-21

Disk Space Requirements for IBM AIX 7.1 (64-bit)	A-21
Temporary Storage Space Requirements for IBM AIX 7.1 (64-bit)	A-21
Tunable Parameters for IBM AIX 7.1 (64-bit)	A-22
Novell SUSE Linux Enterprise Server 10 (32-bit) on x86	A-22
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Novell SUSE Linux Enterprise Server 10 (32-bit)	A-22
Hardware Requirements for Novell SUSE Linux Enterprise Server 10 (32-bit) . . A-22	
Software Requirements for Novell SUSE Linux Enterprise Server 10 (32-bit) . . A-22	
Network Requirements for Novell SUSE Linux Enterprise Server 10 (32-bit) . . A-23	
Disk Space Requirements for Novell SUSE Linux Enterprise Server 10 (32-bit) A-24	
Tunable Parameters for Novell SUSE Linux Enterprise Server 10 (32-bit) .	A-24
Novell SUSE Linux Enterprise Server 11 (32-bit) on x86-64	A-24
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Novell SUSE Linux Enterprise Server 11 (32-bit)	A-24
Hardware Requirements for Novell SUSE Linux Enterprise Server 11 (32-bit) . . A-24	
Software Requirements for Novell SUSE Linux Enterprise Server 11 (32-bit) . . A-24	
Network Requirements for Novell SUSE Linux Enterprise Server 11 (32-bit) . . A-25	
Disk Space Requirements for Novell SUSE Linux Enterprise Server 11 (32-bit) A-25	
Mounting and Unmounting the CD for Novell SUSE Linux Enterprise Server 11 (32-bit)	A-26
Tunable Parameters for Novell SUSE Linux Enterprise Server 11 (32-bit) .	A-26

Novell SUSE Linux Enterprise Server 12 (64-bit) on x86-64	A-26
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Novell SUSE Linux Enterprise Server 12 (64-bit)	A-26
Hardware Requirements for Novell SUSE Linux Enterprise Server 12 (64-bit) . . A-26	
Software Requirements for Novell SUSE Linux Enterprise Server 12 (64-bit) . . A-26	
Network Requirements for Novell SUSE Linux Enterprise Server 12 (64-bit) . . . A-27	
Disk Space Requirements for Novell SUSE Linux Enterprise Server 12 (64-bit). A-27	
Mounting and Unmounting the CD for Novell SUSE Linux Enterprise Server 12 (64-bit).	A-28
Oracle Linux 5.6 (64-bit) on x86-64	A-28
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 5.6 (64-bit).	A-28
Hardware Requirements for Oracle Linux 5.6 (64-bit)	A-28
Software Requirements for Oracle Linux 5.6 (64-bit).	A-28
Network Requirements for Oracle Linux 5.6 (64-bit)	A-30
Disk Space Requirements for Oracle Linux 5.6 (64-bit)	A-30
Mounting and Unmounting the CD for Oracle Linux 5.6 (64-bit).	A-30
Tunable Parameters for Oracle Linux 5.6 (64-bit)	A-30
Oracle Linux 5.8 (32-bit) on x86-64	A-30
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 5.8 (32-bit).	A-30
Hardware Requirements for Oracle Linux 5.8 (32-bit)	A-30
Software Requirements for Oracle Linux 5.8 (32-bit).	A-31
Network Requirements for Oracle Linux 5.8 (32-bit)	A-32

Disk Space Requirements for Oracle Linux 5.8 (32-bit)	A-32
Mounting and Unmounting the CD for Oracle Linux 5.8 (32-bit)	A-32
Tunable Parameters for Oracle Linux 5.8 (32-bit).	A-32
Oracle Linux 6.0 (64-bit) on x86-64	A-32
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 6.0 (64-bit)	A-32
Hardware Requirements for Oracle Linux 6.0 (64-bit)	A-32
Software Requirements for Oracle Linux 6.0 (64-bit).	A-33
Network Requirements for Oracle Linux 6.0 (64-bit).	A-34
Disk Space Requirements for Oracle Linux 6.0 (64-bit).	A-34
Tunable Parameters for Oracle Linux 6.0 (64-bit).	A-34
Oracle Linux 6.5 (32-bit) on x86-64	A-34
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 6.5 (32-bit)	A-34
Hardware Requirements for Oracle Linux 6.5 (32-bit)	A-34
Software Requirements for Oracle Linux 6.5 (32-bit).	A-35
Network Requirements for Oracle Linux 6.5 (32-bit).	A-36
Disk Space Requirements for Oracle Linux 6.5 (32-bit).	A-36
Mounting and Unmounting the CD for Oracle Linux 6.5 (32-bit)	A-36
Tunable Parameters for Oracle Linux 6.5 (32-bit).	A-36
Oracle Linux 7.0 (32-bit) on x86-64	A-36
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 7.0 (32-bit)	A-36
Hardware Requirements for Oracle Linux 7.0 (32-bit)	A-36
Software Requirements for Oracle Linux 7.0 (32-bit).	A-37
Network Requirements for Oracle Linux 7.0 (32-bit).	A-38
Disk Space Requirements for Oracle Linux 7.0 (32-bit).	A-38
Temporary Storage Space Requirements for Oracle Linux 7.0 (32-bit)	A-38

Mounting and Unmounting the CD for Oracle Linux 7.0 (32-bit)	A-38
Tunable Parameters for Oracle Linux 7.0 (32-bit)	A-38
Oracle Linux 7.0 (64-bit) on x86-64	A-38
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 7.0 (64-bit)	A-38
Hardware Requirements for Oracle Linux 7.0 (64-bit)	A-39
Software Requirements for Oracle Linux 7.0 (64-bit)	A-39
Network Requirements for Oracle Linux 7.0 (64-bit)	A-40
Disk Space Requirements for Oracle Linux 7.0 (64-bit)	A-40
Mounting and Unmounting the CD for Oracle Linux 7.0 (64-bit)	A-40
Tunable Parameters for Oracle Linux 7.0 (64-bit)	A-40
Oracle Linux 5.6 (64-bit) on Exalogic 2.0	A-40
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 5.6 (64-bit)	A-40
Hardware Requirements for Oracle Linux 5.6 (64-bit)	A-41
Software Requirements for Oracle Linux 5.6 (64-bit)	A-41
Network Requirements for Oracle Linux 5.6 (64-bit)	A-42
Disk Space Requirements for Oracle Linux 5.6 (64-bit)	A-42
Tunable Parameters for Oracle Linux 5.6 (64-bit)	A-42
Oracle Linux 5.8 (64-bit) on Exalogic 2.0	A-42
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 5.8 (64-bit)	A-42
Hardware Requirements for Oracle Linux 5.8 (64-bit)	A-43
Software Requirements for Oracle Linux 5.8 (64-bit)	A-43
Network Requirements for Oracle Linux 5.8 (64-bit)	A-44
Disk Space Requirements for Oracle Linux 5.8 (64-bit)	A-44
Tunable Parameters for Oracle Linux 5.8 (64-bit)	A-44
Oracle Linux 6.0 (64-bit) on Exalogic 2.0	A-44

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 6.0 (64-bit)	A-44
Hardware Requirements for Oracle Linux 6.0 (64-bit)	A-45
Software Requirements for Oracle Linux 6.0 (64-bit)	A-45
Network Requirements for Oracle Linux 6.0 (64-bit)	A-46
Disk Space Requirements for Oracle Linux 6.0 (64-bit)	A-46
Tunable Parameters for Oracle Linux 6.0 (64-bit)	A-46
Oracle Solaris 10 (32-bit) on SPARC	A-46
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Solaris 10 (32-bit) on SPARC	A-46
Hardware Requirements for Oracle Solaris 10 (32-bit) on SPARC	A-46
Software Requirements for Oracle Solaris 10 (32-bit) on SPARC	A-47
Network Requirements for Oracle Solaris 10 (32-bit) on SPARC	A-48
Disk Space Requirements for Oracle Solaris 10 (32-bit) on SPARC	A-48
Tunable Parameters for Oracle Solaris 10 (32-bit) on SPARC	A-48
Oracle Solaris 10 (64-bit) on SPARC	A-48
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Solaris 10 (64-bit) on SPARC	A-49
Hardware Requirements for Oracle Solaris 10 (64-bit) on SPARC	A-49
Software Requirements for Oracle Solaris 10 (64-bit) on SPARC	A-49
Network Requirements for Oracle Solaris 10 (64-bit) on SPARC	A-50
Disk Space Requirements for Oracle Solaris 10 (64-bit) on SPARC	A-50
Tunable Parameters for Oracle Solaris 10 (64-bit) on SPARC	A-50
Oracle Solaris 11 (32-bit) on SPARC	A-51
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Solaris 11 (32-bit) on SPARC	A-51
Hardware Requirements for Oracle Solaris 11 (32-bit) on SPARC	A-51
Software Requirements for Oracle Solaris 11 (32-bit) on SPARC	A-51

Network Requirements for Oracle Solaris 11 (32-bit) on SPARC	A-53
Disk Space Requirements for Oracle Solaris 11 (32-bit) on SPARC	A-53
Tunable Parameters for Oracle Solaris 11 (32-bit) on SPARC	A-53
Oracle Solaris 11 (64-bit) on SPARC	A-53
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Solaris 11 (64-bit)	A-53
Hardware Requirements for Oracle Solaris 11 (64-bit) on SPARC	A-53
Software Requirements for Oracle Solaris 11 (64-bit) on SPARC	A-54
Network Requirements for Oracle Solaris 11 (64-bit) on SPARC	A-55
Disk Space Requirements for Oracle Solaris 11 (64-bit) on SPARC	A-55
Tunable Parameters for Oracle Solaris 11 (64-bit) on SPARC	A-55
Red Hat Enterprise Linux 5.6 (64-bit) on x86-64	A-55
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat Enterprise Linux 5.6 (64-bit)	A-55
Hardware Requirements for Red Hat Enterprise Linux 5.6 (64-bit)	A-55
Software Requirements for Red Hat Enterprise Linux 5.6 (64-bit)	A-56
Network Requirements for Red Hat Enterprise Linux 5.6 (64-bit)	A-57
Disk Space Requirements for Red Hat Enterprise Linux 5.6 (64-bit)	A-57
Tunable Parameters for Red Hat Enterprise Linux 5.6 (64-bit)	A-57
Red Hat Enterprise Linux 6.0 (64-bit) on x86-64	A-57
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat Enterprise Linux 6.0 (64-bit)	A-57
Hardware Requirements for Red Hat Enterprise Linux 6.0 (64-bit)	A-57
Software Requirements for Red Hat Enterprise Linux 6.0 (64-bit)	A-58
Network Requirements for Red Hat Enterprise Linux 6.0 (64-bit)	A-59
Disk Space Requirements for Red Hat Enterprise Linux 6.0 (64-bit)	A-59
Tunable Parameters for Red Hat Enterprise Linux 6.0 (64-bit)	A-59
Red Hat Enterprise Linux 6.3 (64-bit) on s390x	A-59

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat	
Enterprise Linux 6.3 (64-bit)	A-59
Hardware Requirements for Red Hat Enterprise Linux 6.3 (64-bit)	A-59
Software Requirements for Red Hat Enterprise Linux 6.3 (64-bit)	A-60
Network Requirements for Red Hat Enterprise Linux 6.3 (64-bit)	A-60
Disk Space Requirements for Red Hat Enterprise Linux 6.3 (64-bit)	A-61
Tunable Parameters for Red Hat Enterprise Linux 6.3 (64-bit)	A-61
Red Hat Enterprise Linux 6.5 (32-bit) on x86-64.	A-61
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat	
Enterprise Linux 6.5 (32-bit)	A-61
Hardware Requirements for Red Hat Enterprise Linux 6.5 (32-bit)	A-61
Software Requirements for Red Hat Enterprise Linux 6.5 (32-bit)	A-62
Network Requirements for Red Hat Enterprise Linux 6.5 (32-bit)	A-62
Disk Space Requirements for Red Hat Enterprise Linux 6.5 (32-bit)	A-63
Mounting and Unmounting the CD for Red Hat Enterprise Linux 6.5 (32-bit) . .	
A-63	
Tunable Parameters for Red Hat Enterprise Linux 6.5 (32-bit)	A-63
Red Hat Enterprise Linux 7.0 (32-bit) on x86-64.	A-63
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat	
Enterprise Linux 7.0 (32-bit)	A-63
Hardware Requirements for Red Hat Enterprise Linux 7.0 (32-bit)	A-63
Software Requirements for Red Hat Enterprise Linux 7.0 (32-bit)	A-64
Network Requirements for Red Hat Enterprise Linux 7.0 (32-bit)	A-65
Disk Space Requirements for Red Hat Enterprise Linux 7.0 (32-bit)	A-65
Temporary Storage Space Requirements for Red Hat Enterprise Linux 7.0 (32-bit)	
A-65	
Mounting and Unmounting the CD for Red Hat Enterprise Linux 7.0 (32-bit) . .	
A-65	

Tunable Parameters for Red Hat Enterprise Linux 7.0 (32-bit)	A-65
Red Hat Enterprise Linux 7.0 (64-bit) on x86-64	A-65
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat Enterprise Linux 7.0 (64-bit).	A-65
Hardware Requirements for Red Hat Enterprise Linux 7.0 (64-bit)	A-66
Software Requirements for Red Hat Enterprise Linux 7.0 (64-bit)	A-66
Network Requirements for Red Hat Enterprise Linux 7.0 (64-bit)	A-67
Disk Space Requirements for Red Hat Enterprise Linux 7.0 (64-bit)	A-67
Mounting and Unmounting the CD for Red Hat Enterprise Linux 7.0 (64-bit). . . A-67	
Tunable Parameters for Red Hat Enterprise Linux 7.0 (64-bit)	A-67
NeoKylin Linux 6.5 (64-bit) on x86-64	A-67
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for NeoKylin Linux 6.5	A-67
Hardware Requirements for NeoKylin Linux 6.5 (64-bit).	A-68
Software Requirements for NeoKylin Linux 6.5 (64-bit)	A-68
Network Requirements for NeoKylin Linux 6.5 (64-bit).	A-69
Disk Space Requirements for NeoKylin Linux 6.5 (64-bit)	A-69
Mounting and Unmounting the CD for NeoKylin Linux 6.5 (64-bit)	A-69
Tunable Parameters for NeoKylin Linux 6.5 (64-bit)	A-69
NeoKylin Linux 7.2 (64-bit) on x86-64	A-69
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for NeoKylin Linux 7.2	A-69
Hardware Requirements for NeoKylin Linux 7.2 (64-bit).	A-69
Software Requirements for NeoKylin Linux 7.2 (64-bit)	A-70
Network Requirements for NeoKylin Linux 7.2 (64-bit).	A-70
Disk Space Requirements for NeoKylin Linux 7.2 (64-bit)	A-71
Mounting and Unmounting the CD for NeoKylin Linux 7.2 (64-bit)	A-71

Tunable Parameters for NeoKylin Linux 7.2 (64-bit)	A-71
Microsoft Windows Server 2008 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010	A-71
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2008 R2 (32-bit)	A-71
Hardware Requirements for Windows Server 2008 R2 (32-bit)	A-71
Software Requirements for Windows Server 2008 R2 (32-bit)	A-72
Network Requirements for Windows Server 2008 R2 (32-bit)	A-72
Disk Space Requirements for Windows Server 2008 R2 (32-bit)	A-73
Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2010	A-73
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2008 R2 (64-bit)	A-73
Hardware Requirements for Windows Server 2008 R2 (64-bit)	A-73
Software Requirements for Windows Server 2008 R2 (64-bit)	A-73
Network Requirements for Windows Server 2008 R2 (64-bit)	A-75
Disk Space Requirements for Windows Server 2008 R2 (64-bit)	A-75
Tunable Parameters for Microsoft Windows Server 2008 R2 (64-bit)	A-75
Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2012	A-76
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2008 R2 (64-bit)	A-76
Hardware Requirements for Windows Server 2008 R2 (64-bit)	A-76
Software Requirements for Windows Server 2008 R2 (64-bit)	A-76
Network Requirements for Windows Server 2008 R2 (64-bit)	A-78
Disk Space Requirements for Windows Server 2008 R2 (64-bit)	A-78
Tunable Parameters for Microsoft Windows Server 2008 R2 (64-bit)	A-78
Microsoft Windows Server 2012 (32-bit) on x86-64 with MS Visual Studio 2013A-78	

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2012 (32-bit)	A-78
Hardware Requirements for Windows Server 2012 (32-bit)	A-79
Software Requirements for Windows Server 2012 (32-bit)	A-79
Network Requirements for Windows Server 2012 (32-bit)	A-80
Disk Space Requirements for Windows Server 2012 (32-bit)	A-80
Tunable Parameters for Microsoft Windows Server 2012 (32-bit)	A-81
Microsoft Windows Server 2012 (64-bit) on x86-64 with MS Visual Studio 2012A-81	
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Install Types Server 2012 (64-bit)	A-81
Hardware Requirements for Windows Server 2012 (64-bit)	A-81
Software Requirements for Windows Server 2012 (64-bit)	A-82
Network Requirements for Windows Server 2012 (64-bit)	A-83
Disk Space Requirements for Windows Server 2012 (64-bit)	A-83
Tunable Parameters for Microsoft Windows Server 2012 (64-bit)	A-83
Microsoft Windows Server 2012 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010.	A-84
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2012 R2 (32-bit)	A-84
Hardware Requirements for Windows Server 2012 R2 (32-bit)	A-84
Software Requirements for Windows Server 2012 R2 (32-bit)	A-84
Network Requirements for Windows Server 2012 R2 (32-bit)	A-85
Disk Space Requirements for Windows Server 2012 R2 (32-bit)	A-85
Mounting and Unmounting the CD for Windows Server 2012 R2 (32-bit). .	A-86
Microsoft Windows 7 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012. A-86	
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 7 (64-bit).	A-86

Hardware Requirements for Windows 7 (64-bit)	A-86
Software Requirements for Windows 7 (64-bit)	A-86
Network Requirements for Windows 7 (64-bit)	A-88
Disk Space Requirements for Windows 7 (64-bit)	A-88
Microsoft Windows 8 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012 A-88	
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 8 (64-bit)	A-88
Hardware Requirements for Windows 8 (64-bit)	A-89
Software Requirements for Windows 8(64-bit)	A-89
Network Requirements for Windows 8 (64-bit)	A-90
Disk Space Requirements for Windows 8 (64-bit)	A-91
Microsoft Windows 8.1 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010 A-91	
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 8.1 (32-bit)	A-91
Hardware Requirements for Windows 8.1 (32-bit)	A-91
Software Requirements for Windows 8.1 (32-bit)	A-91
Network Requirements for Windows 8.1 (32-bit)	A-92
Disk Space Requirements for Windows 8.1 (32-bit)	A-92
Microsoft Windows 8.1 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012 A-93	
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 8.1 (64-bit)	A-93
Hardware Requirements for Windows 8.1 (64-bit)	A-93
Software Requirements for Windows 8.1 (64-bit)	A-93
Network Requirements for Windows 8.1 (64-bit)	A-94
Disk Space Requirements for Windows 8.1 (64-bit)	A-94

Microsoft Windows 10 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012	
A-95	
Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 10	
(64-bit)	A-95
Hardware Requirements for Windows 10 (64-bit)	A-95
Software Requirements for Windows 10 (64-bit)	A-95
Network Requirements for Windows 10 (64-bit)	A-96
Disk Space Requirements for Windows 10 (64-bit)	A-96

B. Starting the Oracle Tuxedo Administration Console

What is the Oracle Tuxedo Administration Console?	B-1
Administration Console File Tree	B-2
Server Requirements	B-3
Browser Requirements	B-3
Hardware Requirements	B-4
Setting Up Your Environment for the Oracle Tuxedo Administration Console	B-4
Starting tuxwsvr	B-5
Starting wlisten	B-6
Starting the Oracle Tuxedo Administration Console	B-6
Limitation(s)	B-8
Exiting the Oracle Tuxedo Administration Console	B-8

C. Oracle Jolt 12c Release 2 (12.1.3) Overview and Installation Information

About Oracle Jolt	C-1
Oracle Jolt Components	C-3
Jolt Server Listener	C-4
Jolt Server Handler	C-5

Jolt Repository Server	C-5
Jolt Repository	C-5
Jolt Internet Relay	C-5
Jolt Class Library	C-6
JoltBeans	C-8
Oracle Jolt Client Personalities	C-8
Oracle Jolt Capabilities	C-9
Oracle Jolt Client Support	C-10
Oracle Jolt Client Requirements	C-11
Oracle Jolt Client Class Library	C-11
Oracle Jolt Release Interoperability	C-12
Oracle Jolt Pre-Installation Checklist	C-12
Oracle Jolt Documentation	C-12

D. File and Database Management and Disk Space Allocation

Introduction	D-1
How the Oracle Tuxedo System Manages Files	D-1
Arranging for Raw Disk Space	D-2
How the Oracle Tuxedo Filesystem Is Organized	D-2
Space for Queue Spaces	D-4
Space for Application Servers	D-4

E. IPC Resource Configuration on a UNIX System

Parameter Sets Controlling IPC Resources	E-1
Shared Memory	E-2
Semaphores	E-3
Message Queues and Messages	E-4
What Is Application Deadlock?	E-5

Performance Implications of Blocking Conditions	E-5
Tunable Message Parameters	E-5
Other Kernel Tunables	E-7

Preparing to Install the Oracle Tuxedo System

The following sections provide information that you need to know before installing the Oracle Tuxedo 12c Release 2 (12.1.3) product software:

- [Oracle Universal Installer \(OUI\)](#)
- [Oracle Tuxedo Web Distribution](#)
- [Oracle Tuxedo Software Components](#)
- [LLE and SSL Levels of Encryption](#)
- [Hardware and Software Prerequisites](#)
- [Interprocess Communication Resources Configuration](#)
- [Oracle Installation Program](#)
- [Oracle Home Directory](#)
- [Installation Road Map](#)

Oracle Universal Installer (OUI)

The Oracle Tuxedo 12c Release 2 (12.1.3) product software installer is based on the Oracle Universal Installer (OUI). To install the Oracle Tuxedo 12c Release 2 (12.1.3) product software on your host, you must know how to use OUI to install Oracle products. For more information, see [Introduction to Oracle Universal Installer](#).

OUI is a Java-based installer that enables you to install Oracle components.

Installing Oracle Tuxedo 12cR2 with Oracle Home

When you install Oracle Tuxedo 12cR2, an Oracle home is created. Oracle home is managed by OUI.

Oracle Home

An Oracle home is the system context where Oracle products run. It consists of the following:

- Directory location where the products are installed
- Corresponding system path setup
- Program groups associated with the products installed in that home (where applicable)
- Services running from that home

Installing Oracle Tuxedo 12cR2

To install Oracle Tuxedo 12cR2 and create the Oracle home, do the following steps:

1. Run OUI.
2. In the Specify Home Details page, enter the Oracle home settings for the installation session.
3. Continue installation.

Removing Oracle Home

To remove an existing Oracle home, run OUI and click Deinstall Products. In the inventory dialog, select the home(s) you want to delete and click Remove. You can also use the `REMOVE_HOMES` variable in the command line or in a response file.

The syntax is as follows:

```
[On UNIX] ./runInstaller -deinstall -silent  
ORACLE_HOME=<LOCATION_OF_ORACLE_HOME>  
"REMOVE_HOMES={<LOCATION_OF_ORACLE_HOME_TO_BE_REMOVED>}"
```

Determining the Default Oracle Home

By default, when you start OUI, it searches your system to determine the default Oracle home where Oracle software should be installed.

In each case, the `ORACLE_HOME` name is taken first from the command line if it is specified, or from the response file if specified. If not specified, the following convention is used for the name:

```
Ora<Oracle Home Name>
```

The `ORACLE_HOME` path is taken first from the command line if specified, or from the response file if specified. If not specified, the `ORACLE_HOME` environment variable is used.

If `ORACLE_BASE` is specified in the environment, the default `ORACLE_HOME` starts with `$ORACLE_BASE`.

Multiple Oracle Homes

OUI supports the installation of several active Oracle homes on the same host as long as the products support this at run-time. Multiple versions of the same product or different products can run from different Oracle homes concurrently. Products installed in one home do not conflict or interact with products installed on another home. You can update software on any home at any time, assuming all Oracle applications, services, and processes installed on the target home are shut down. Processes from other homes may still be running.

Target Home

The Oracle home currently accessed by OUI for installation or deinstallation is the target home. To upgrade or remove products from the target home(s), these products must be shut down or stopped.

If the selected `ORACLE_HOME` has already installed Tuxedo of the same version, the installer will show a warning.

It is not allowed to install Tuxedo of the same version or install related products on different `ORACLE_HOME` for windows platforms; otherwise, you will get an error box.

Oracle Universal Installer Inventory

The OUI inventory stores information about all Oracle software products installed in all Oracle homes on a host (provided the product installed using Oracle Universal Installer).

Inventory information is stored in Extensible Markup Language (XML) format. The XML format allows for easier problems diagnosis and faster data loading. Secure information is not stored directly in the inventory. As a result, during removal of some products, you may be prompted to enter the required credentials for validation.

Oracle Universal Installer Inventory Structure

The OUI inventory has the following hierarchical structure:

- Central Inventory Pointer File
- Central Inventory
- Oracle Home Inventory

Central Inventory Pointer File

Every Oracle software installation has an associated Central Inventory where details of all Oracle products installed on a host are registered. The Central Inventory is located in the directory that the inventory pointer file specifies. Each Oracle software installation has its own Central Inventory pointer file that is unknown to another Oracle software installation.

For Oracle homes sharing the same Central Inventory, OUI performs all read and write operations on the inventory. The operations on the Central Inventory are performed through a locking mechanism. This implies that when an installation, upgrade, or patching operation occurs on an Oracle home, these operations become blocked on other Oracle homes that share the same Central Inventory.

Below table shows the location of the default inventory pointer file for various platforms:

Linux, LinuxPPC	/etc/oraInst.loc
AIX	/etc/oraInst.loc
Solaris SPARC, HP-UX-IA64	/var/opt/oracle/oraInst.loc
Windows	HKEY_LOCAL_MACHINE/Software/Oracle/inst.loc

In UNIX, if you do not want to use the Central Inventory located in the directory specified by the inventory pointer file, you can use the `-invPtrLoc` flag to specify another inventory pointer file. The syntax is as follows:

```
./runInstaller -silent -invPtrLoc <Location_of_oraInst.loc>  
ORACLE_HOME="<Location_of_Oracle_Home>"  
ORACLE_HOME_NAME="<Name_of_Oracle_Home>"
```

Note: If the content of the `oraInst.loc` file is empty, OUI prompts you to create a new inventory.

Central Inventory

The Central Inventory contains information relating to all Oracle products installed on a host. It contains the following files and folders:

- Inventory File
- Logs Directory

Inventory File

This file lists all Oracle homes installed on the node. For each Oracle home, it also lists the Oracle home name, home index, and nodes where home is installed. It also mentions if the home is an Oracle Clusterware home or a removed Oracle home. It can only detect removed Oracle homes created using OUI version 11.1 and later.

This file is located at:

```
<central inventory location>/ContentsXML/inventory.xml
```

Note: It is recommended that you do not remove or manually edit this file as it could affect installation and patching.

Logs Directory

The Central Inventory contains installation logs at the following location:

```
<central inventory location>/logs
```

The logs directory contains the logs corresponding to all installations performed on a particular node. You can also find a copy of the installation log in the `$ORACLE_HOME/cfgtoollogs` directory.

The installation logs for an installation are identified by the timestamp associated with the log files. These files are generally saved in the following format:

```
<Name_of_Action><YYYY-MM-DD_HH-MM-SS{AM/PM}>.log
```

For example, consider an `attachHome` operation performed on 17th, May, 2007 at 6.45AM. The associated log file would be created as follows:

```
AttachHome2007-05-17_06-45-00AM.log
```

Note: The installation logs do not contain any errors or failures.

Oracle Home Inventory

The Oracle home inventory (or local inventory), is present inside each Oracle home. It contains information relevant to a particular Oracle home. This file is located at:

```
$ORACLE_HOME/inventory
```

It contains the following files and folders:

- Components File

This file contains details about third-party applications (like Java Runtime Environment (JRE)), required by different Java-based Oracle tools and components. In addition, it also contains details of all the components as well as patchsets or interim patches installed in the Oracle home. This file is located at:

```
ORACLE_HOME/inventory/ContentsXML/comps.xml
```

- Home Properties File

This file contains node list details, the local node name, and the CRS flag for the Oracle home. In a shared Oracle home, the local node information is not presented. This file also contains the following information:

- GUID — Unique global ID for the Oracle home
- ARU ID — Unique platform ID. The patching and patchset application depends on this ID.
- ARU ID DESCRIPTION — Platform description

The information in `oraclehomeproperties.xml` overrides the information in `inventory.xml`. This file is located at:

```
$ORACLE_HOME/inventory/ContentsXML/oraclehomeproperties.xml
```

- Other Folders

The following table lists the other folders you can find in the Oracle home inventory:

Folder Name	Description
Scripts	Contains the scripts used for the cloning operation.
ContentsXML	Contains the details of the components and libraries installed.

Templates	Contains the template files used for cloning.
oneoffs	Contains the details of the one-off patches applied.

Creating the Central Inventory

OUI enables you to set up the Central Inventory on a clean host or register an existing Oracle home with the Central Inventory when it is lost or corrupted. If the Central Inventory does not exist, OUI creates the Central Inventory in the location specified by the `oraInst.loc` file.

You can set up the Central Inventory by using the `-attachHome` flag of Oracle Universal Installer. The syntax is as follows:

```
./runInstaller -silent -attachHome -invPtrLoc ./oraInst.loc
ORACLE_HOME="<<Oracle_Home_Location>" ORACLE_HOME_NAME="<<Oracle_Home_Name>"
"CLUSTER_NODES={<node1,node2>}" LOCAL_NODE="<<node_name>"
```

Using the Session Variables

You can use the following session variables:

- ORACLE_HOME
- ORACLE_HOME_NAME

Verifying the Operation

After attaching the Oracle home, you can verify the success of the operation by verifying the contents of the log file located in the `<central_inventory>/logs` directory. You can also view the contents of the `inventory.xml` file under the `<central-inventory>/ContentsXML` directory to verify if the Oracle home is registered.

Note: Cloning on Windows operating systems is recommended to create the Central Inventory.

Detaching Oracle Homes from the Central Inventory

You can detach an Oracle home from the Central Inventory. When you pass the flag, it updates the `inventory.xml` file presented in the Central Inventory. The syntax is as follows:

```
./runInstaller -silent -detachHome -invPtrLoc ./oraInst.loc
ORACLE_HOME="<<Oracle_Home_Location>" ORACLE_HOME_NAME="<<Oracle_Home_Name>"
```

Using Optional Flags

If you are using a shared Oracle home, use the `-cfs` flag. This ensures that the local node information is not populated inside a shared Oracle home.

```
./runInstaller -silent -cfs -detachHome -invPtrLoc ./oraInst.loc  
ORACLE_HOME="<<Oracle_Home_Location>" ORACLE_HOME_NAME="<<Oracle_Home_Name>"  
./runInstaller -silent -detachHome -invPtrLoc ./oraInst.loc  
ORACLE_HOME="<<Oracle_Home_Location>" ORACLE_HOME_NAME="<<Oracle_Home_Name>"
```

You can completely clean the Oracle home and remove the home directory using the `-removeallfiles` flag. The syntax is as follows:

```
./runInstaller -silent -deinstall -removeallfiles -invPtrLoc ./oraInst.loc  
ORACLE_HOME="<<Oracle_Home_Location>" ORACLE_HOME_NAME="<<Oracle_Home_Name>"
```

If you want to disable the warning message that appears when you use the `-removeallfiles` flag, use the `-nowarningonremovefiles` flag. The syntax is as follows:

```
./runInstaller -silent -deinstall -nowarningonremovefiles -invPtrLoc  
./oraInst.loc  
ORACLE_HOME="<<Oracle_Home_Location>" ORACLE_HOME_NAME="<<Oracle_Home_Name>"
```

Removing the Central Inventory

Even after all Oracle homes on a host are removed, you will find traces of the inventory with certain log files. If you do not want to maintain these files and want to remove the Central Inventory, do the following:

Removing the Central Inventory on UNIX Platforms

You can remove the Central Inventory on UNIX. Do the following steps:

1. Locate the `oraInst.loc` file and get the Central Inventory location (`inventory_loc` parameter) from this file.
 - For Solaris, it is located in the `/va/opt/oracle` folder.
 - For Linux, it is located in the `/etc` folder.

2. Remove the Central Inventory by executing the following command:

```
rm -rf <central_inventory_location>
```


3. Remove the `oraInst.loc` file by executing the following command with root privileges:

- Solaris: `rm /va/opt/oracle/oraInst.loc`
- Linux: `rm /etc/oraInst.loc`

Removing the Central Inventory on Windows Platforms

You can remove the Central Inventory on Windows. Do the following steps:

1. Locate the registry key:

```
\\HKEY_LOCAL_MACHINE\Software\Oracle\inst_loc
```

2. Get the Central Inventory location from this key.

3. Delete the Central Inventory directory and all its contents.

4. Delete the registry key:

```
\\HKEY_LOCAL_MACHINE\Software\Oracle\inst_loc
```

Backing up the Inventory

You can back up the Oracle home using your preferred method. You can use any method (such as zip, tar, and cpio), to compress the Oracle home.

It is highly recommended to back up the Oracle home before any upgrade or patch operation. You should also back up the Central Inventory when Oracle home is installed or deinstalled.

Recovering from Inventory Corruption

The inventory (Central and the Oracle home inventory) is critically important in Oracle software life-cycle management. The following sections explain what you need to do in case of inventory corruption.

Customizing and Creating Response Files

This section introduces you OUI response file availability for silent and suppressed installations. This section also describes how to modify or create a response file so you can customize and standardize Oracle products installation in your organization.

About Response Files

What Is a Silent Installation?

A silent installation runs in the console and does not use the GUI. The interactive dialogs you normally see are not displayed during silent installation. Instead of prompting you to select a series of installation options, OUI installs the software using a predefined set of options stored in the response file or entered in the command line. You can view silent installation progress in the console.

What Is a Response File?

A response file is a specification file containing information you normally enter through the OUI user interface during an interactive installation session. Each answer is stored as a value for a variable identified in the response file. For example, values for Oracle home or installation type can be set automatically within the response file.

Why Perform a Silent Installation?

Silent installation can be useful if you have to install an Oracle product multiple times on multiple computers. If the options you select while installing on each computer are always the same, you save the time of reviewing each installation screen and selecting the various installation options.

Silent installations can also ensure that multiple users in your organization use the same installation options when they install your Oracle products. This makes supporting those users easier because you already know what components and options have been installed on each computer.

Before you perform silent installation, you should review the settings in the response file template provided with your Oracle product.

Notes:

- If you attempt to perform silent installation on a UNIX computer where no Oracle products have been installed, OUI uses the default inventory location, and then prompts you to run the `oraInstRoot.sh` script with root privileges upon successful installation.
- The script is saved in the `_*Central Inventory - oraInventory *_` directory. This script sets up the Central Inventory on a clean host. You can override the default location by setting it in `INVENTORY_LOCATION`.

- This location is ignored if a Central Inventory already exists and is pointed to `/var/opt/oracle/oraInst.loc` (or `/etc/oraInst.loc` for Linux, AIX, and Sequent).

For more information, see [“Oracle Universal Installer Inventory”](#) on page 1-3.

Creating a Response File With Record Mode

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

When you use record mode, OUI records the installation session into a response file. You specify the name of the response file in the command line. The recorded response file is generated immediately after the Summary page, so 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 will be 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 those cases, the variable values specified in the original source response file will be recorded into the new response file.

The following sections describe how to use record mode on Windows and UNIX systems.

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 Oracle Universal Installer executable file (`setup.exe` or `runInstaller.sh`) for your installation.
2. Enter the following command:

```
setup -record -destinationFile <response_file_name> (on Windows)
```

```
./runInstaller -record -destinationFile <response_file_name> (on UNIX)
```

Replace the `<response_file_name>`

with the complete path for the new response file. For example:

On Windows:

```
setup -record -destinationFile C:\response_files\install_oracle11g.rsp
```

On UNIX:

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

3. Use the OUI user interface to select your installation options. These will be recorded.

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.

Response File Format

The following sections describe the organization and content of an OUI response file.

Note: It is recommended that you use an absolute path for the file name. However, if the file name is a relative path, it should be relative to the directory where the `oraparam.ini` file is presented.

Variable Values

Variable Lookup Order

All variable values within a response file are in the name-value format. If two components have a variable with the same name, the expression should be written as follows to preserve the uniqueness of each variable:

```
<component>:<variable>=<value>
```

Component Nomenclature and Version Ambiguity

If two components have the same internal name, but are different versions, resolve the ambiguity by specifying the expression as follows:

```
<component>:<version>:<variable>=<value>
```

Order of Variables

OUI looks for command line variables in the following order:

```
<component>:<version>:<variable>=<value>
```

```
<component>:<variable>=<value>
```

```
<variable>=<value>
```

Similarly, if command line variables are not found, OUI looks for variables in the response file in the same order as shown above.

Note: OUI treats incorrect context, format, or type values within a response file as if no value were specified.

Response File Entries Order

There are no restrictions on where you place entries in the response file. You can insert entries in any order desired. You can also subsequently move existing entries to any other position within the file. Be aware, however, that although the order of the entries is unimportant, naming conventions require that the variable names must be unique regardless of where they appear in the file.

Setting the Recommendation Value

Values for variables are specified as:

```
<variable> = <recommendation> : <value>
```

The values that are given as `<value_required>` must be specified for silent installation to be successful.

For values given as `<value_unspecified>`, you can optionally specify a value, where `<value>` can be one of the following types listed in below table.

Type	Representation
Number	10
Boolean	TRUE or FALSE (case insensitive)
String	"Value"
String List	{"value1", "value2"}

The `<recommendation>` parameter can be specified as `Forced` or `Default`.

- If you specify `Forced`, no dialog appears during installation. The value is automatically used. You cannot change the value.
- If you specify `Default`, the dialog appears during installation with the value as the default. You can choose another value if desired.
- If nothing is specified for `<recommendation>`, `Forced` is used as the default.

Comments

Comments begin with a "#" (hash or pound) symbol. They contain information about the type of the variable, state whether the variable appears in dialog, and describe the function of the variable. A variable and a value are associated with a comment.

For example:

```
#TopLevelComponent:StringList;Used in Dialog
TopLevelComponent={"Demo", "1.0" }
```

Headers

The header is a comment that describes the different formats and comment conventions used in a response file.

For example:

```
#Parameter : UNIX_GROUP_NAME
#Type : String
#Description : UNIX group to be set for the inventory directory.
#Valid only on UNIX platforms.
#Example : UNIX_GROUP_NAME = "install"
UNIX_GROUP_NAME=<Value Unspecified>
```

Response File Parameters

The following list describes OUI parameters. All products installed using OUI have these parameters in addition to product-specific variables.

Note: The parameters provided below are OUI-specific and not necessarily the parameters applicable for your platform and version. Always check your specific installation guide for the list of valid response file parameters.

For Tuxedo specific response file parameters, see [“Required Variables in Tuxedo Response File” on page 3-3](#).

INCLUDE

Specifies the list of response files to be included in a particular response file. If you want to include other response files in your main response file, you must use this parameter. Note that the values in the main response file have priority over the included response files.

```
INCLUDE={"file1.rsp","file2.rsp",....."filen.rsp"}
```

You should specify the absolute path in the `INCLUDE` statement for each response file to be included. If you wish to specify a relative path, note that the location is relative to the current working directory (the location of the `oraparam.ini` file).

RESPONSEFILE_VERSION

Specifies the version number of the response file.

```
RESPONSEFILE_VERSION = <version_number>
```

FROM_LOCATION

The location of the source of the products to be installed. The default generated value for this parameter is a path relative to the location of the `runInstaller` file `oraparam.ini`. Relative paths are necessary for shared response files used by multiple users over a network. Since people may be mapping to different drives, absolute paths will not work for shared response files.

Notes:

- You must enter a value for `FROM_LOCATION` for a complete silent installation. You may want to use the command line to set this parameter. If the location is a relative path, remember that the path should be relative to the location of the `oraparam.ini` file.
- Make sure that `/var/opt/oracle/oraInst.loc` (or `/etc/oraInst.loc` for Linux, AIX, and Sequent) exists. The response file for UNIX has a public variable, `setunixinstallgroup`, that only becomes effective when it is the first OUI installation and the `oraInst.loc` file is not already present at the location mentioned above.

FROM_LOCATION_CD_LABEL

Used in multiple-CD installations. It includes the label of the compact disk where the file `products.jar` exists. You can find the label in the `disk.label` file in the same directory as `products.jar`.

LOCATION_FOR_DISK2

This parameter is for the complete path to other disks.

```
LOCATION_FOR_DISK2="F:\teststage\cd\Disk2"
```

If there are more than two disks, more variables will be added as `LOCATION_FOR_DISK3`, and so on.

The CD location for a silent installation is located using two mechanisms:

- OUI looks for the `Location_For_Disk{DiskNumber}` variable in the response file and uses that location.
- If the variable does not have a value or does not have the required files, it looks for the components under `.././Disk{DiskNumber}/stage` (from `products.jar`).

Therefore, you can either specify the location to look for it in the response file or you can copy them into the disk. For example:

```
<Product_Shiphome_Location>/stage
```

RESTART_SYSTEM

Set this boolean variable to `TRUE` if you want to restart the system without user confirmation. This is the force value for restarting the system.

NEXT_SESSION

Set this boolean variable to `TRUE` if the installer needs to go to the File Locations page for another installation or to another response file if you are performing a silent installation.

```
NEXT_SESSION = TRUE
```

NEXT_SESSION_ON_FAIL

Set this boolean variable to `TRUE` to allow users to invoke another installation session even if the current installation session fails. This variable is used only if `NEXT_SESSION` variable is set to `TRUE`.

```
NEXT_SESSION_ON_FAIL = TRUE
```

NEXT_SESSION_RESPONSE

Use this string variable to enter the complete path of the response file for the next session if you want to automatically begin another silent installation when the current installation is complete.

If you specify only a file name, the Installer looks for the response file in the `<TEMP>/orainstall` directory. This variable is only used if `NEXT_SESSION` is set to `TRUE`; otherwise, OUI ignores the value of this variable.

```
NEXT_SESSION_RESPONSE="/private/usr2/nextinstall.rsp"
```


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.

SHOW_COMPONENT_LOCATIONS_PAGE

The location page, which appears in a custom installation type, can be suppressed by setting this value to `FALSE`. If you set the value to `FALSE`, you are prevented from specifying alternate directories. If there are products with installed directories which can be changed, you may want to set the value to `TRUE`.

SHOW_CUSTOM_TREE_PAGE

Set `SHOW_CUSTOM_TREE_PAGE` to `TRUE` if the custom tree page in the installer must be shown. In the Custom Tree page, dependencies can be selected or deselected. This page appears only during custom installation type.

SHOW_END_SESSION_PAGE

The installation success/failure page, which appears at the end of an installation, can be suppressed by setting this value to `FALSE`.

SHOW_EXIT_CONFIRMATION

Set to `TRUE` if you need to show the confirmation when exiting the installer.

```
SHOW_EXIT_CONFIRMATION = TRUE
```

SHOW_INSTALL_PROGRESS_PAGE

The installation progress page, which appears during the installation phase, can be suppressed by setting the value to `FALSE`.

SHOW_OPTIONAL_CONFIG_TOOL_PAGE

Set to `TRUE` if you must show the Optional Configuration Tools page in the installer. The Optional Configuration Tools page shows a list of optional configuration tools that are part of this installation and the status of each tool, including detailed information on why the tool has failed.

SHOW_ROOTSH_CONFIRMATION

Set to `TRUE` if you need to show the Confirmation dialog asking to run the `root.sh` script in the installer. This variable is valid only on UNIX platforms.

```
SHOW_ROOTSH_CONFIRMATION = TRUE
```

SHOW_SPLASH_SCREEN

Set to `TRUE` if the initial splash screen in the installer needs to be shown.

```
SHOW_SPLASH_SCREEN = TRUE
```

SHOW_SUMMARY_PAGE

The Summary page can be suppressed by setting this value to `FALSE`.

SHOW_WELCOME_PAGE

Set to `TRUE` if you need to show the Welcome page on the installer.

```
SHOW_WELCOME_PAGE = FALSE
```

SHOW_RELEASE_NOTES

Set this parameter to `TRUE` if you want the release notes for this installation to be shown at the end of the installation. A dialog box lists the available release notes. Note that the

`SHOW_END_SESSION` parameter must be set to `TRUE` before you can use this parameter.

TOPLEVEL_COMPONENT

The name of the component (products), and the version as a string list. You must enter a value for `TOPLEVEL_COMPONENT`.

Usually, the components are represented with a pair of strings: the first one representing the internal name, the second representing the version.

For example, RDBMS 11.1. may be represented as `{ "oracle.rdbms" , "11.1.0.0.0" }`.

UNIX_GROUP_NAME

The UNIX group name to be set for the inventory on UNIX platforms.

Note: The UNIX group name is used for first-time installations only.

REMOVE_HOMES

Use to identify the Oracle home(s) you want to remove from the inventory during a deinstallation session. For each home, specify the home name using the full path information.

```
REMOVE_HOMES={ "/home/oracle/ora9i" , "/home/oracle/ora8i" }
```

DEINSTALL_LIST

Use to enter a list of components to be removed during a silent deinstall session. For each component, specify the internal component name and version.

```
DEINSTALL_LIST={"sample1" , "1.0.0.0.0" }
```

SHOW_DEINSTALL_CONFIRMATION

Set to `FALSE` if you want to hide the deinstall confirmation dialog box during silent deinstallation.

SHOW_DEINSTALL_PROGRESS

Set this parameter to `FALSE` if you want to hide the deinstallation progress dialog box during silent deinstallation.

DEPENDENCY_LIST

List of dependents on which the component depends. These dependent components represent the list of components that appear as "selected" during installation. Following is a list of a few parameter characteristics:

- The `DEPENDENCY_LIST` variable is only generated when dependencies are present and if the dependency is not a required one.
- You cannot list components to appear as deselected items in a dialog.
- Specifying required dependents is redundant since they will be selected anyway. The dependents selection can have two other types of dependents: optional and one-or-more.
- The list of components is specified by an internal name and version number.

A typical example is listed here:

```
DEPENDENCY_LIST={"oracle.netclt" , "9.2.0.4.0" , "oracle.netmgr" , "9.2.0.4.0" }
```

CLUSTER_NODES

Lists the nodes on the cluster where Oracle home is installed or would be installed. For new installations, this would be the node list where the installation needs to be done (including the local node). For patching or upgrades, this is the node list where Oracle home is already installed.

```
"CLUSTER_NODES={alpha-1, alpha-2}"
```

REMOTE_NODES

Lists the remote nodes (apart from the local node), on the install cluster during a silent installation. OUI installs on all named nodes.

```
"REMOTE_NODES={alpha-1, alpha-2}"
```

LOCAL_NODE

This parameter specifies the current node where the installation occurs.

```
LOCAL_NODE={}
```

OPTIONAL_CONFIG_TOOLS

While all the required configuration tools are launched by the installer, you can control the optional configuration tools you would want to launch by specifying the tool internal names in the `OPTIONAL_CONFIG_TOOLS` section.

Note: The `OPTIONAL_CONFIG_TOOLS` variable is only generated when at least one optional configuration tool is available.

You can also call the Net Configuration Assistant or the Database Configuration Assistant at the end of a database installation in silent mode. For example, to launch the Net Configuration Assistant in silent mode, you can pass the parameter `s_responseFileName="netca.rsp"`.

You can specify both the Auto-launch optional tools and User-launch optional tools in a string list.

For Example:

```
OPTIONAL_CONFIG_TOOLS = {"configtool2 ", "configtool3"}
```

```
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>
```

If no value is specified for this variable, all the tools are run by default. If there is a value specified, only those optional tools are run while the rest of the tools are ignored.

Suppress the configuration tool by mentioning only the tools that you want to run as part of the `OPTIONAL_CONFIG_TOOLS` variable added for each component. You should use the configuration tool internal names.

The response file generator generates these internal names also as part of the options provided for the variable.

For example, if `oracle.server` has `Tool1` and `Tool2` and you want to run only `Tool1` in the response file, you can specify it as follows:

```
oracle.server:11.1.0.0.0:OPTIONAL_CONFIG_TOOLS={"Tool1" }
```

INSTALL_TYPE

You can set the installation type variable to determine the installation type of the currently selected top-level component.

The installation type variable is only generated for top-level components and only when there is more than one installation type available.

Note: You must enter a value for `INSTALL_TYPE`.

SELECTED_LANGUAGES

You can set the languages for component installation. You must use the internal name while specifying the value:

```
en, : English
```

```
ja, : Japanese
```

For example, to specify Japanese:

```
SELECTED_LANGUAGES = {"ja" }
```

Note: The `SELECTED_LANGUAGES` variable is only generated when more than one language is available.

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.

These are called silent installations. 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, you will receive an error message. Before you can perform a silent installation on such a computer, you must first run the `oraInstRoot.sh` script, which is saved 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.

For more information, see [“Oracle Universal Installer Inventory” on page 1-3](#).

Specifying a Response File

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

On Windows: `setup.exe -responseFile <filename> <optional_parameters>`

On UNIX: `./runInstaller -responseFile <filename> <optional_parameters>`

Note: You must specify the complete responseFile path. If you do not, OUI assumes the location to be 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 Windows:

```
setup -help
```

On UNIX:

```
./runInstaller -help
```

For Windows, when you execute `setup -help`, a new command window appears, with the "Preparing to launch..." message. A moment later, help information appears in that window.

Optional Parameters When Specifying a Response File

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

- `-nowelcome` — Used flag with the `-responseFile` flag to suppress the Welcome dialog that appears during installation.
- `-silent` — Used with the `-responseFile` flag to run OUI in complete silent mode. Note that the Welcome dialog is suppressed automatically.
- In a file named `silentInstall<timestamp>.log` for hosts without an Oracle inventory. This file is generated in the `/tmp` directory (UNIX) and the directory specified by the `TEMP` variable (Windows).

- 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.

Setting Response File Variables From the Command Line

With OUI 2.1 and higher, you can specify the value of certain variables when you start OUI from the command line. Specifically, you can specify session and component variables.

For more information, see [“Response File Format” on page 1-12](#).

Note: When you specify the value of a variable on the command line, that value overrides the value of the variable if it is defined in the response file.

Specifying the Value of a Session Variable

To specify the value of a session variable, use the following command syntax:

On UNIX:

```
./runInstaller session:<variable_name>=<value>
```

On Windows:

```
setup.exe session:<variable_name>=<value>
```

For example, to prevent the Universal Welcome page from displaying:

On UNIX:

```
./runInstaller session:SHOW_WELCOME_PAGE=false
```

On Windows:

```
setup.exe session:SHOW_WELCOME_PAGE=false
```

Note: The `"session:"` tag is optional and is used mainly to remove any possible ambiguity.

Specifying the Value of a Component Variable

To specify the value of a component variable, use the following command syntax:

On UNIX:

```
./runInstaller
<component_name>:<component_version>:<variable_name>=<value>
```

On Windows:

```
setup.exe <component_name>:<component_version>:<variable_name>=<value>
```

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.

- **Suppressed:**

Use OUI suppressed mode to supply the necessary information by using a combination of a response file or command line entries with certain interactive dialogs. You can choose which dialogs to suppress by supplying the information on the command line when you invoke OUI. This method is most useful when an installation has a common set of parameters that can be captured in a response file, in addition to custom information that must be input by hand.

- **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.

Note: You can use the `-noConsole` flag on Windows to suppress the display of console messages.

Installation Media

Note: On Windows, when you start the installer from a shared drive, you need to map the shared drive and then invoke the installer from the shared drive. When you invoke `runInstaller` (UNIX) or `setup.exe` (Windows), you should invoke it from the directory where this command is present, or you must specify the complete path to `runInstaller` (UNIX) or `setup.exe` (Windows).

Special Instructions for UNIX Users

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

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 will display OUI.
3. On the computer that will display OUI, enter the following command, which allows other computers to display information on the computer monitor: `%xhost +`
4. Rerun the `runInstaller` 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.

Providing a UNIX Installer Location with Root Privileges

You must have root privileges to perform various UNIX installation operations. For example, you must have root privileges to be able to create the OUI inventory.

If you are installing OUI for the first time, you are prompted to run a shell script from another terminal window before proceeding with the installation. OUI prompts you to run `root.sh` after installation completes only if the script is required to run as root before configuration assistants are run. Otherwise, you are prompted to run `root.sh` as root later.

Note: When running OUI in silent mode, if `root.sh` is required prior to configuration assistants, OUI skips configuration assistants during the installation. You must run `root.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 `./root.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 the first time you install.

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.

Deinstalling Oracle Products

Removing Oracle Products and Oracle Homes

You can deinstall Oracle products before selecting products to install, or after a successful installation.

To remove an Oracle product or Oracle home using interactive mode, do the following steps:

1. Start OUI from a CD-ROM or:
 - For Windows platforms, launch OUI from the Start menu by selecting **Start, Installation Products, Oracle Universal Installer**.
 - For UNIX platforms, from the command line, run the script called `runInstaller` from the directory where it is stored by default at the same level as the first Oracle home created on that host.
2. Click **Deinstall Products** on the Welcome screen.

The Inventory panel appears.
3. 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.
4. The Remove Confirmation Dialog appears; you are prompted if you want to remove the products and their dependent components. Click **Yes**.

OUI warns you of any product dependencies that might cause problems if particular products are removed, and prompts you to confirm the deinstallation.

Pay special attention to the full list of products being removed before proceeding. OUI computes this list based on the dependencies of each component.

Note: You can also remove products by using the OUI **Installed Products** button as long as you perform this action before selecting products to install.

Deinstalling Top-level Products With Dependents

A top-level component is the most important component of an installation. It is the installable product you see on the first installation screen. You can only install one top-level component for each installation session.

When you select a specific component for removal, OUI analyzes the dependency information to determine if other components should be removed along with it. Generally, if a component is selected for removal, the following components are removed with it:

All components that have a required dependency on the selected component.

Dependents of the selected component that have no other dependents. A dependent is a component on which the top-level component (dependent) has a dependency.

Silent Deinstallation

Not only can you perform command line installations, you can also perform command line deinstallations. A command line deinstallation enables you to remove Oracle products or Oracle homes from your system without using the OUI graphical user interface.

You can choose to display no dialog boxes or prompts, or you can selectively avoid displaying certain dialog boxes that are normally used during a deinstallation.

Immediately Displaying the Inventory Dialog Box

Use the following commands to immediately display the Inventory dialog box, which allows you to select items for removal without navigating the OUI startup screen:

```
setup.exe -deinstall -silent (on Windows)
```

```
./runInstaller -deinstall -silent (on UNIX)
```

Hiding the Inventory Dialog Box

If you want to hide the inventory dialog box during a deinstallation, you can specify the products to be removed in the response file `DEINSTALL_LIST` parameter; specify Oracle homes to be removed with the `REMOVE_HOMES` variable.

As with other response file parameters, you can also specify the `DEINSTALL_LIST` parameter on the OUI command line. For example, on a UNIX machine, if you use private inventory specified in `~/oraInst.loc` and `ORACLE_HOME="/home/user/OraHome"`, enter:

```
./runInstaller -invPtrLoc ~/oraInst.loc -deinstall -silent
DEINSTALL_LIST={"Tuxedo,12.1.3.0.0"} ORACLE_HOME="/home/user/OraHome"
```

To remove Oracle homes from the inventory, use the `REMOVE_HOMES` variable.

```
./runInstaller -invPtrLoc ~/oraInst.loc -deinstall -silent
DEINSTALL_LIST={"Tuxedo,12.1.3.0.0"} ORACLE_HOME="/home/user/OraHome"
REMOVE_HOMES={"/home/user/OraHome" }
```

Hiding the Deinstallation Confirmation and Progress Dialog Boxes

Use the following commands to hide the deinstallation confirmation and progress dialog boxes during a command line deinstallation:

On a Windows system:

```
setup.exe -deinstall -silent session:SHOW_DEINSTALL_PROGRESS=false
session:SHOW_DESINSTALL_CONFIRMATION=false
```

On a UNIX system:

```
./runInstaller -deinstall -silent session:SHOW_DEINSTALL_PROGRESS=false
session:SHOW_DESINSTALL_CONFIRMATION=false
```

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 on your Windows or UNIX computer. 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 the `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.

Oracle Internationalization

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. OUI dialogs are displayed in this language if available. If specific 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.

Oracle Tuxedo Web Distribution

An evaluation copy of Oracle Tuxedo is available for download from the Oracle corporate Web site at <http://www.oracle.com/technology/software/index.html>.

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

Oracle Tuxedo Software Components

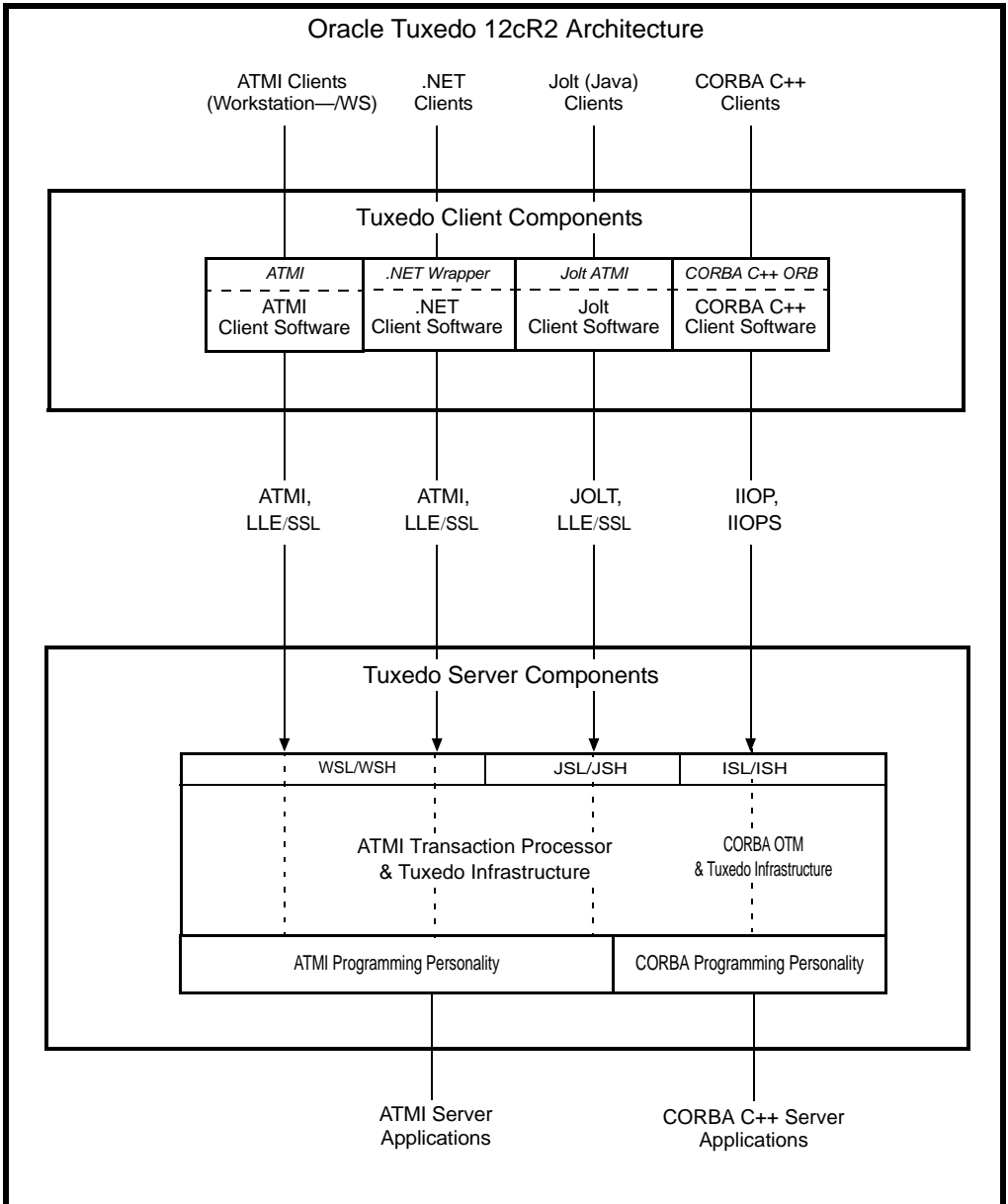
Included with the Oracle Tuxedo 12c Release 2 (12.1.3) software distribution is the Oracle Jolt product software, which enables Java clients to call Oracle Tuxedo services, and the Oracle SNMP Agent product software, which enables Oracle Tuxedo applications to be managed from a centralized enterprise management console, and Oracle Tuxedo System and Applications Monitor Plus Agent, which enables Oracle Tuxedo to be managed from Tuxedo System and Applications Monitor Plus manager (Oracle TSAM), and Oracle Service Architecture Leveraging Tuxedo (Oracle SALT), which provides native Tuxedo Web service.

The Oracle Tuxedo 12c Release 2 (12.1.3) 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 12c Release 2 (12.1.3) server software
 - Oracle SNMP Agent 12c Release 2 (12.1.3) software
 - Oracle Tuxedo Administration Console software
 - Link-Level Encryption (LLE) and Secure Sockets Layer (SSL) encryption 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 12c Release 2 (12.1.3) client software
 - Tuxedo.NET Workstation client software
 - LLE and SSL encryption software
- Oracle Tuxedo System and Applications Monitor Plus Agent:
 - Oracle Tuxedo System and Applications Monitor Plus Agent software
- Oracle Service Architecture Leveraging Tuxedo (Oracle SALT):
 - Oracle Service Architecture Leveraging Tuxedo (Oracle SALT) software

Figure 1-1 illustrates a high-level view of the Oracle Tuxedo client and server components.

Figure 1-1 Oracle Tuxedo Client and Server Components



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

Tuxedo .NET Workstation Client

The Tuxedo .NET Workstation Client requires:

- the ATMI Client to be automatically installed when you select the .NET Client component.
- Microsoft’s .NET Framework 4.0 installed on your system. If .NET Framework is not detected, you can continue the .NET Client installation and install .NET Framework later. For more information on the Tuxedo.Net Workstation Client, see [Creating Tuxedo.NET Workstation Client Applications](#).

LLE and SSL Levels of Encryption

LLE and SSL are available at least 112 bit.

Hardware and Software Prerequisites

The Oracle Tuxedo software must be installed on each server machine that will participate 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 in reference page [UBBCONFIG\(5\)](#) in [Oracle Tuxedo File Formats, Data Descriptions, MIBs, and System Processes Reference](#).

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

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

System Requirements

The system requirements for Oracle Tuxedo 12c Release 2 (12.1.3) are given in [Table 1-1](#).

Table 1-1 The System Requirements for Oracle Tuxedo 12c Release 2 (12.1.3)

Component	Requirement
Platform*	Any platform identified in “Oracle Tuxedo 12c Release 2 (12.1.3) Platform Data Sheets” on page A-1.
Hard disk drive	As stated in the data sheet for the target platform in “Oracle Tuxedo 12c Release 2 (12.1.3) Platform Data Sheets” on page A-1. For additional information about disk space requirements, see “File and Database Management and Disk Space Allocation” on page D-1.
Memory	As stated in the data sheet for the target platform in “Oracle Tuxedo 12c Release 2 (12.1.3) Platform Data Sheets” on page A-1.
* Microsoft Windows XP systems support only the Oracle Tuxedo client components.	

For Oracle Jolt installation preparatory information, see [“Oracle Jolt 12c Release 2 \(12.1.3\) Overview and Installation Information”](#) on page C-1. No installation preparation is required for Oracle SNMP Agent.

Note: Before installing Oracle Tuxedo software on a UNIX system, Oracle recommends that you repartition your hard disk device in accordance to [“File and Database Management and Disk Space Allocation”](#) on page D-1.

Temporary Storage Space Requirements

The Oracle Installation program uses a temporary directory in which it extracts the files from the archive that are needed to install Oracle Tuxedo on the target system. During the installation process, your temporary directory must contain sufficient space to accommodate the compressed Java Runtime Environment (JRE) bundled with the installer and an uncompressed copy of the JRE that is expanded into the temporary directory. The installation program moves the JRE from the temporary directory to the *Oracle Home directory* at the end of the installation process. For information about the Oracle Home directory, see [“Oracle Home Directory”](#) on page 1-37.

The amount of temporary storage space needed depends upon the target platform, as stated in the data sheets in [“Oracle Tuxedo 12c Release 2 \(12.1.3\) Platform Data Sheets”](#) on page A-1.

When you start OUI, it automatically copies some executable files and link files into the default /tmp directory (C:\Documents and Settings\\Local Settings\Temp on Microsoft Windows) on the machine. If the machine is set to run cron jobs periodically (along

with many other processes that may be running), these jobs attempt to clean up the default temporary directory, thereby deleting some files and causing OUI to fail.

To ensure there is adequate temporary space, you may want to allocate an alternate directory for use as a temporary directory for the installation. If there are any cron jobs or processes that are automatically run on the machines to clean up the temporary directories, ensure you set the TMP or TEMP environment variable to a different location (other than the default location) that is secure on the hard drive (meaning a location on which the cleanup jobs are not run). Also ensure that you have write permissions on this alternative TEMP directory. This must be done before you execute `runInstaller (setup.exe` on Microsoft Windows).

Note: Specifying an alternative temporary directory location is not mandatory, and is required only if any cron jobs are set on the computers to clean up the `/tmp` directory.

Interprocess Communication Resources Configuration

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

On a Windows 2003 Server system, the Oracle Tuxedo system provides an IPC service called the Oracle ProcMGR (Process Manager), which facilitates interprocess communications. You use the Oracle ProcMGR service to adjust the IPC parameters and maximize the performance of an Oracle Tuxedo application.

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 12c Release 2 \(12.1.3\) Platform Data Sheets”](#) on page A-1. For the recommended IPC values, see [“IPC Resource Configuration on a UNIX System”](#) on page E-1.

After installing the Oracle Tuxedo software and creating an application configuration file (UBBCONFIG file), you use the `tmloadcf(1)` command to calculate and print a list of the minimum IPC resources needed to support the application. If your Oracle Tuxedo application is distributed, the minimum IPC resources must be available on every machine participating in the application. For more information about calculating IPC resources using the `tmloadcf(1)` command, see [“Performing Post-Installation Tasks”](#) on page 5-1.

Note: Before installing Oracle Tuxedo software on a UNIX system, Oracle recommends that you adjust the IPC parameters on the target machine in accordance to [“IPC Resource Configuration on a UNIX System”](#) on page E-1.

Oracle Installation Program

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

Note: On Windows platforms you must re-login to the system after Tuxedo 12c Release 2 (12.1.3) installation has completed.

Previous Tuxedo Versions

If a previous version of the Tuxedo software is detected during GUI mode and silent installation on a Windows system, the Tuxedo installer will automatically stop the following previous Tuxedo version ProcMGR and Listener services:

- Oracle ProcMGR V2.2
- Oracle ProcMGR V8.0
- Oracle ProcMGR V8.1
- Oracle ProcMGR V9.0
- Oracle ProcMGR V9.1
- Oracle ProcMGR V9.1with VS2005 v2
- Oracle ProcMGR V10.0 with VS2005
- Tuxedo 12cR1 with VS2010 Listener on Port 3050
- Tuxedo 11gR1PS1 with VS2008 Listener on Port 3050
- Tuxedo 10.3 with VS2005 Listener on Port 3050
- Tuxedo 10.0 with VS2005 Listener on Port 3050
- Tuxedo 9.1 Listener on Port 3050
- Tuxedo 9.1with VS2005 Listener on Port 3050
- Tuxedo 9.0 Listener on Port 3050
- Tuxedo 8.1 Listener on Port 3050
- Tuxedo 8.0 Listener on Port 3050

- Tuxedo 7.1 Listener on Port 3050
- Tuxedo 6.5 Tuxedo IPC Helper

Note: During GUI installation, users will be prompted with the following message:
“Stopping Previous Version Tuxedo Services”

Listener services configured on ports other than 3050 must be stopped *manually*.

The current Tuxedo distribution requires these services to be stopped in order to run multiple Tuxedo versions on the same machine. The time required for stopping the ProcMGR services is minimal; however, the time required to stop the listener services may take a few minutes.

Installation Methods

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

- Graphical user interface (GUI) installation, described in [“Installing Oracle Tuxedo Using GUI-Mode Installation”](#) on page 2-1.
- Silent installation, described in [“Installing Oracle Tuxedo Using Silent Installation”](#) on page 3-1.

Cancelling Installation

Clicking the “Cancel” or the “close” window button in GUI mode creates an incomplete Tuxedo 12c Release 2 (12.1.3) installation. You will have to re-install Tuxedo 12c Release 2 (12.1.3).

If you installed Tuxedo 12c Release 2 (12.1.3) 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 Tuxedo 12c Release 2 (12.1.3) to a new Oracle Home product directory

Install Types

An install type is a bundle of product software components related by function. Oracle Tuxedo 12c Release 2 (12.1.3) offers the following install types:

- Full Install - consists of all the Oracle Tuxedo 12c Release 2 (12.1.3) server and client components of Tuxedo, SALT, and TSAM Plus agent.
- Server Install - consists of all the Oracle Tuxedo 12c Release 2 (12.1.3) server components of Tuxedo, SALT, and TSAM Plus agent.

- Client Install - consists of all the Oracle Tuxedo 12c Release 2 (12.1.3) 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 12c Release 2 (12.1.3) client software
 - Oracle Tuxedo.NET Workstation client software (Windows)
 - LLE and SSL encryption software

In addition to selecting an install type during an Oracle Tuxedo 12c Release 2 (12.1.3) 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” on page 2-1](#).

Oracle Home Directory

When you install Oracle Tuxedo, you are prompted to specify an Oracle Home directory. The Oracle Home directory is a repository for common files that are used by multiple Oracle products installed on the same machine.

An Oracle home is the system context where Oracle products run. This context consists of the following:

- Directory location where the products are installed
- Corresponding system path setup
- Program groups associated with the products installed in that home (where applicable)
- Services running from that home

Choosing an Oracle Home Directory

During the installation of Oracle Tuxedo 12c Release 2 (12.1.3), you are prompted to choose an existing Oracle Home directory or specify a path for a new Oracle Home directory. If you choose to create a new directory, the Oracle Tuxedo installer program automatically creates the directory for you.

Upon choosing an Oracle Home directory, you are prompted to choose an Oracle product directory for your Oracle Tuxedo installation.

Multiple Oracle Homes

OUI supports the installation of several active Oracle homes on the same host as long as the products support this at run-time. Multiple versions of the same product or different products can run from different Oracle homes concurrently. Products installed on one home do not conflict or interact with products installed on another home. You can update software on any home at any time, assuming all Oracle applications, services, and processes installed on the target home are shut down. Processes from other homes may still be running.

Installation Road Map

You are now ready to begin your installation. To install Oracle Tuxedo 12c Release 2 (12.1.3), see one of the following sections:

- [“Installing Oracle Tuxedo Using GUI-Mode Installation” on page 2-1](#)
- [“Installing Oracle Tuxedo Using Silent Installation” on page 3-1](#)

If you are upgrading from Oracle WebLogic Enterprise or Oracle Tuxedo 12cR1 or earlier software, see [“Upgrading the Oracle Tuxedo System to 12c Release 2 \(12.1.3\)” on page 4-1](#).

If you want to uninstall your Oracle Tuxedo software, see [“Performing Post-Installation Tasks” on page 5-1](#).

Installing Oracle Tuxedo Using GUI-Mode Installation

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

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

What Is GUI-Mode Installation?

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

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

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

Before You Start

If you are upgrading from Oracle WebLogic Enterprise or Oracle Tuxedo 12cR1 or earlier software, see [“Upgrading the Oracle Tuxedo System to 12c Release 2 \(12.1.3\)” on page 4-1](#) and follow the instructions given there. Then return here to continue your Oracle Tuxedo installation.

On a Windows machine that has the Oracle WebLogic Enterprise software or an older version of the Oracle Tuxedo software installed, a window named “Previous Version Detected!” will display at the very beginning of the Oracle Tuxedo installation. This window contains a message that encourages you to exit the installation program and remove the older software before installing Oracle Tuxedo 12c Release 2 (12.1.3).

If you choose to remove the older software, refer to [“Upgrading the Oracle Tuxedo System to 12c Release 2 \(12.1.3\)” on page 4-1](#), follow the instructions to remove the older software from your machine, and then restart the installation.

If you choose *not* to remove the older software, install the Oracle Tuxedo 12c Release 2 (12.1.3) software to a different Oracle Home directory than the Oracle Home directory where the Oracle WebLogic Enterprise software or the older version of the Oracle Tuxedo software is installed. See also [“Previous Tuxedo Versions” on page 1-35](#)

Starting GUI-Mode Installation on a Windows System

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

1. Select a Windows system that meets the hardware and software requirements described in [“Oracle Tuxedo 12c Release 2 \(12.1.3\) Platform Data Sheets” on page A-1](#).
2. Log in to the Windows system as the Administrator or as a member of the Administrator group.

You need administrative privileges to install Oracle Tuxedo server components on a Windows system. If you are going to install only Oracle Tuxedo client components, you do not need administrative privileges.

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

For disk space requirements, see [“Oracle Tuxedo 12c Release 2 \(12.1.3\) Platform Data Sheets” on page A-1](#). For important disk space allocation information, see [“File and Database Management and Disk Space Allocation” on page D-1](#).

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 12c Release 2 (12.1.3) 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 `setup.exe` to start the installation.
5. Proceed to [“Running GUI-Mode Installation” on page 2-4](#).
6. Re-login to the system after Tuxedo 12c Release 2 (12.1.3) installation has completed.

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 12c Release 2 \(12.1.3\) Platform Data Sheets” on page A-1](#)
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 12c Release 2 \(12.1.3\) Platform Data Sheets” on page A-1](#). For important disk space allocation information, see [“File and Database Management and Disk Space Allocation” on page D-1](#)

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 12c Release 2 (12.1.3) 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 `runInstaller` to invoke the installation procedure.

Note: GUI mode is the default for Tuxedo 12c Release 2 (12.1.3) installation. If a GUI interface is not available on your Unix platform, you will receive an error message and the installation process is aborted.

If a GUI interface is not available on your Unix platform, you must use silent mode installation. For more information, see [What Is Silent Installation?](#)

5. Proceed to [“Running GUI-Mode Installation” on page 2-4.](#)

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 and (2) the language-specific message catalog installed during the installation. If the language-related environment variable is not set or is set to a value unknown to the installer program, both the language displayed and the message catalog installed default to English.

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

For GUI-Mode installation screens, see [“Oracle Universal Installer Installation Screens” on page F-9.](#)

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
Welcome	Click Next 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. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location. This screen does not appear on Windows operating systems.

Table 2-1 Instructions

In This Window . . .	Perform the Following 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"> • Full Install (the default) - all the Oracle Tuxedo server and Tuxedo, SALT, and TSAM Plus agent client components. • Server Install - all Tuxedo, SALT, and TSAM Plus agent server components of • Client Install - Tuxedo client software components only • 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. <p>For a detailed list of software components for each install type, see “Install Types” on page 1-36.</p> <p>Select an install type and then click Next to launch the Choose Oracle Home Directory window-your selection includes the appropriate encryption software for Link-Level Encryption (LLE) and/or Secure Sockets Layer (SSL). Or select an install type and then click Customize to launch the Customize Install window.</p>

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
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, you will be warned and asked if you want 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<timestamp>.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/tuxedo12.1.3.0.0</code>" by default (UNIX) and cannot be changed after the <code>ORACLE_HOME</code> is determined. Tuxedo installer doesn't allow to install two same version tuxedo product into the same <code>ORACLE_HOME</code>.</p> <p>If the selected <code>ORACLE_HOME</code> already installed the same version tuxedo, the installer will show a warning.</p>

Table 2-1 Instructions

In This Window . . .	Perform the Following 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 “Jolt Internet Relay” on page C-5.</p> <p>Observe the following software component mappings:</p> <ul style="list-style-type: none"> • <i>Tuxedo Server</i>—contains ATMI server software; CORBA C++ server software; Oracle Jolt server software; Oracle SNMP Agent software, and Oracle Tuxedo Administration Console software • <i>Tuxedo ATMI Client</i>—contains Oracle ATMI Workstation (/WS) client software • <i>Tuxedo CORBA Client</i>—contains Oracle CORBA C++ client software (C++ client ORB) including environmental objects • <i>Tuxedo Jolt JRLY</i>—contains Oracle Jolt Relay software • <i>Tuxedo Jolt Client</i>—contains Oracle Jolt client software • <i>Tuxedo .NET Client</i>—contains Oracle .Net client software • <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 • <i>Oracle Services Architecture Leveraging Tuxedo (Oracle SALT)</i>—contains Oracle Services Architecture Leveraging Tuxedo (Oracle SALT) software <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 .NET Client Install (Windows Platforms only)	<p>The Confirm to install .NET v4.0 Client panel displays if Microsoft .NET Framework v4.0 is not detected and install type contains ".NET Client".</p> <p>Click Next to proceed with the installation.</p>
Confirm enable TSAM plus agent	<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>

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
SSL Installation Choice	<p>This window appears only if you have installed Server, ATMI and/or CORBA Client install type components.</p> <p>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">• LDAP Service Name: Fully qualified domain name of the LDAP server (for example, pcwiz.mydomain.com)• LDAP PortID: Port number through which the local machine communicates with the LDAP server (for example, 389)• LDAP BaseObject: Distinguished name of the base object for search in the LDAP server (for example, o=beasys.com)
LDAP filter file choose	<p>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 completed)</p>
Select samples installation	<p>If the selected install feature sets include server component, this screen appears to confirm install the samples.</p>
Configure tlisten Choice	<p>tlisten is needed for MP/clustered mode configuration and Oracle TSAM Plus. If you need to configure tlisten, select "Yes", and then input the parameters for tlisten configuration on the screens that follow.</p> <p>This window appears only if you have installed the Server Install.</p>

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
Configure tlisten password	<p>This window appears only if you select to configure tlisten.</p> <p>Enter a tlisten password of your choice. Your password must be a string of alphanumeric characters in clear-text format that is no more than 80 characters in length. You use this password to log in to the Oracle Tuxedo Administration Console.</p> <p>Note: Machines on the same domain must have the same tlisten password defined.</p> <p>Oracle Tuxedo uses the tlisten 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> or <code>wlisten(1)</code> gateway processes, Oracle Tuxedo authenticates using the tlisten password.</p> <p>Enter and re-enter a tlisten password and then click Next to continue.</p>
Configure tlisten port (Windows Platforms only)	<p>Set the tlisten port value</p> <p>Note: For Windows platforms, the default value is 3050. There is no default value for UNIX platforms</p>
Tlisten Encryption Method (Windows Platforms only)	<p>Specify the tlisten encryption method: LLE or SSL.</p> <p>The default is LLE.</p>
Choose Encryption Min Bit (Windows Platforms only)	<p>Specify tlisten minimum encryption bit.</p> <p>The default minimum value is 0.</p>
Choose Encryption max Bit (Windows Platforms only)	<p>Specify tlisten maximum encryption bit.</p> <p>The default maximum value is 256.</p>
SSL Parameter Input	<p>If the SSL encryption method is chosen, you must enter Security Principal name, location, and password.</p>
Summary	Click Install to start the installation
Installing	Nothing to do
Install End	Click Exit to exit the installation program.

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

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.

Understanding the Oracle Tuxedo Shortcuts on a Windows System

When installing the Oracle Tuxedo software on a Windows system, the installer program automatically uses Oracle WebLogic E-Business Platform as the parent folder for Oracle Tuxedo. As additional Oracle Home directories are added to the system, the installer program continues to create new Oracle WebLogic E-Business Platform folders using the convention.

Each Oracle Home directory has an associated Oracle WebLogic E-Business Platform folder. Each Oracle WebLogic E-Business Platform folder contains one or more Oracle product folders (see note) and two files: OracleHOME Location and Install History.

Note: Because an Oracle Home directory may be the home directory for an Oracle Tuxedo installation, an Oracle WebLogic Server installation, or other Oracle product installations, an Oracle WebLogic E-Business Platform folder may contain multiple Oracle product folders.

Understanding the Oracle Administration Program on a Windows System

In addition to the Oracle Tuxedo Administration Console, the Oracle Tuxedo 12c Release 2 (12.1.3) software for Windows provides an Oracle Administration program and two Windows services (Tlisten and Oracle procMGR) for configuring the Oracle Tuxedo system on a Windows

2003 Server system. Only if the installation included Oracle Tuxedo server components will these additional administration tools be installed on the Windows 2003 Server system.

After you have installed or upgraded the Oracle Tuxedo software on your Windows 2003 Server system, you can use the Oracle Administration control panel (Start → Settings → Control Panel → Oracle Tuxedo 12c Release 2 (12.1.3) with VS2008 Administration) to perform the following tasks:

- Access other machines on the network on which the Oracle Tuxedo system is installed
- Set and modify environment variables for the Oracle Tuxedo system
- Direct Oracle Tuxedo system messages to the Windows 2003 Server Event Log
- Configure Oracle Tuxedo `tlisten(1)` processes to start automatically
- Tune interprocess communication (IPC) resources to maximize Oracle Tuxedo system performance

For instructions on performing these tasks, see “[Configuring Oracle Tuxedo ATMI for Windows Server 2003](#)” in *Using Oracle Tuxedo ATMI on Windows*.

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 `regedt32`, and click OK to launch the Registry Editor window.

Windows Service Change

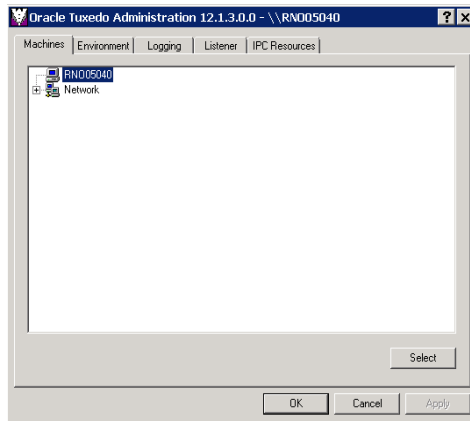
If Oracle Tuxedo 12c Release 2 (12.1.3) is installed successfully, Windows Service "ORACLE ProcMGR V12.1.3.0.0_VS2010" and "TUXEDO 12.1.3.0.0_VS2010 (Port:<tlisten_portid>)" are installed.

Name	Description	Status	Startup Type	Log On As
Oracle Fusion Middleware 11.1.1.4.0 NodeManager...			Manual	Local System
ORACLE ProcMGR V12.1.3.0.0_V\$2010		Started	Automatic	Local System

Name	Description	Status	Startup Type	Log On As
Telephony	Provides T...		Manual	Network S...
Thread Ordering Server	Provides or...		Manual	Local Service
TListen 12.1.3.0.0_V\$2010(Port: 3050)		Started	Automatic	Local System
TSM Data Exchange	Exchange s...		Manual	Local Service

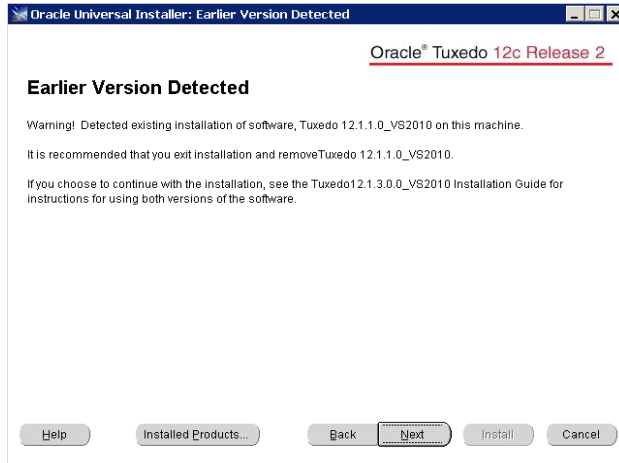
Windows Control Panel Change

If Oracle Tuxedo 12c Release 2 (12.1.3) is installed successfully, Oracle Tuxedo 12.1.3.0.0 Control Panel is installed as below.



Earlier Version Detected

The following warning information occurs if an earlier version of Tuxedo is detected.



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.

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 [“Performing Post-Installation Tasks” on page 5-1](#).

Installing Oracle Tuxedo Using Silent Installation

The following sections describe how to install Oracle Tuxedo using silent installation on both Windows and UNIX systems:

- [What Is Silent Installation?](#)
- [Before You Start](#)
- [Using Silent Installation: Main Steps](#)
- [What Do I Do Next?](#)

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 both Windows and 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.

Before You Start

If you are upgrading from Oracle WebLogic Enterprise or Oracle Tuxedo 12cR1 or earlier software, see [“Upgrading the Oracle Tuxedo System to 12c Release 2 \(12.1.3\)”](#) on page 4-1 and follow the instructions given there. Then return here to continue your Oracle Tuxedo installation.

If you are planning to use Oracle Tuxedo 12c Release 2 (12.1.3) in conjunction with a previous Tuxedo version on a Windows system, see [“Oracle Installation Program” on page 1-35](#).

Using Silent Installation: Main Steps

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 will be recorded to the new response file.

The following sections describe how to use record mode on Windows and UNIX systems.

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.exe` or `runInstaller.sh`) for your installation.
2. Enter the following command:

On Windows:

```
setup -record -destinationFile <response_file_name>
```

On UNIX:

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

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

For example:

On Windows:

```
setup -record -destinationFile C:\response_files\install_oracle11g.rsp
```

On UNIX:

```
./runInstaller -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.

Required Variables in Tuxedo Response File

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

Note: For String type variables, you must contain 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` can be selected from below table:

Install type	Description
Full Install	This install type will install all of the Tuxedo Server and Client files
Server Install	This install type will install the Tuxedo server files only.
Client Install	This install type will install the Tuxedo Client files only.
Custom Install	This install type will install the components selected by customer

For more information, see [“Install Types” on page 1-36](#).

CONFIG_TLISTEN

Configures the `tlisten` parameters during installation. Set this variable to true.

`ENCRYPT_CHOICE` (Windows)

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

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

TLISTEN_PASSWORD

Configures `tlisten` password.

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

`TLISTEN_PORT` (Windows)

Configures `tlisten` port.

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

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

SSL_PARAMETERS (Windows)

Configures `tlisten` SSL Parameters Security Principal (Name, Location, Password).

If `ENCRYPT_CHOICE=1`, `SSL_PARAMETERS` must be set.

MIN_ENCRYPT_BITS (Windows)

Configures Min Encryption Bits.

`MIN_ENCRYPT_BITS` must be set.

MAX_ENCRYPT_BITS (Windows)

Configures Max Encryption Bits.

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

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.

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, you will receive an error message. Before you can perform a silent installation in this situation, you must first run the `oraInstRoot.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.

For more information, see [“Oracle Universal Installer Inventory” on page 1-3](#).

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 Windows:

```
setup.exe -responseFile <filename> <optional_parameters>
```

On UNIX:

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

Notes:

- To install Oracle Tuxedo in complete silent mode, you must specify `<optional_parameters>` to `-silent`.

- You must specify the complete `responseFile` path. 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 Windows:

```
setup -help
```

On UNIX:

```
./runInstaller -help
```

In Windows, when you execute `setup -help`, a new command window appears displaying the "Preparing to launch..." message. A moment later, the help information appears in that window.

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 and the directory specified by the `TEMP` variable on Windows platforms.
- 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.

UNIX Template File

This sample UNIX response file (`installer.properties`) in [Listing 3-1](#) applies to a silent installation of Oracle Tuxedo 12c Release 2 (12.1.3).

Listing 3-1 UNIX response File

```

#####
## Copyright (c) 1999, 2013 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", "12.1.3.0.0"}

#-----

DEINSTALL_LIST={"Tuxedo", "12.1.3.0.0"}

#-----

#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", "12.1.3.0.0"}
#-----
-----
TOPLEVEL_COMPONENT={"Tuxedo", "12.1.3.0.0"}

#-----
-----

#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/tuxedo12.1.3.0.0/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
```

What Do I Do Next?

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

- [“Assigning File Ownership on a UNIX System” on page 2-10](#)
- [“Understanding the Oracle Tuxedo Shortcuts on a Windows System” on page 2-10](#)
- [“Understanding the Oracle Administration Program on a Windows System” on page 2-10](#)
- [“Reviewing the Windows Server Registry Content” on page 2-11](#)

To configure your Oracle Tuxedo software, verify that your software is installed correctly, or to uninstall Oracle Tuxedo software, see [“Performing Post-Installation Tasks” on page 5-1](#).

Upgrading the Oracle Tuxedo System to 12c Release 2 (12.1.3)

The following sections provide procedures for upgrading your Oracle Tuxedo or Oracle WebLogic Enterprise application to Oracle Tuxedo 12c Release 2 (12.1.3) using simple and hot upgrade procedures:

- [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 Tuxedo 12c Release 2 \(12.1.3\)](#)
- [Performing a Hot Upgrade](#)
- [Rebuilding an Application](#)
- [Installing Oracle Jolt 12c Release 2 \(12.1.3\) with Oracle WebLogic Server](#)

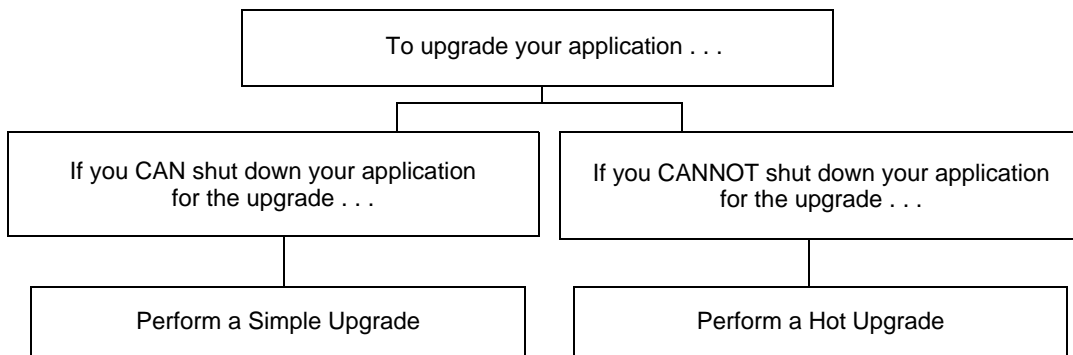
Preparing Your Machine for an Upgrade

Before you install the Oracle Tuxedo 12c Release 2 (12.1.3) 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 12c Release 2 \(12.1.3\) Platform Data Sheets”](#) on page A-1.

Selecting an Upgrade Procedure

If you are installing Oracle Tuxedo 12c Release 2 (12.1.3) on a platform that contains an earlier release of Oracle Tuxedo software or the Oracle WebLogic Enterprise software, be sure to read the following procedures carefully.



If you can shut down your application for the upgrade, then shut down your application and perform a simple upgrade. For instructions on performing a simple upgrade, see [“Performing a Simple Upgrade”](#) on page 4-3.

If you cannot shut down your application for the upgrade, you can perform a hot upgrade for any of the following software products to Oracle Tuxedo 12c Release 2 (12.1.3) without having to shut down your enterprise:

- Versions 8.1, 9.0, 9.1, 10.0, 10.3, 11.1.1.1.0, 11.1.1.2.0, 11.1.1.3.0, and 12.1.1.0.0 of Oracle Tuxedo
- Version 5.1 of WebLogic Enterprise

You can add the Oracle Tuxedo 12c Release 2 (12.1.3) software to existing Oracle Tuxedo or Oracle WebLogic Enterprise domains without shutting down the existing applications, and you can add new Oracle Tuxedo 12c Release 2 (12.1.3) application servers to those domains without shutting down the existing applications. For instructions on performing a hot upgrade, see [“Performing a Hot Upgrade”](#) on page 4-6.

Backing Up Files

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

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

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

Performing a Simple Upgrade

A simple upgrade consists of the following basic steps:

1. If you already have the `DMTLOG` and `TLOG`, make sure that no pending transaction logs are in them.
2. Shut down the application.
3. Back up files as described in [“Backing Up Files” on page 4-3](#).
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” on page 5-36](#).
7. If upgrading a Windows machine running Oracle WebLogic Enterprise or Oracle Tuxedo 10.0 or earlier software, perform the additional steps specified in [“Performing Additional Upgrade Steps to Windows Server” on page 4-4](#).
8. Install the Oracle Tuxedo 12c Release 2 (12.1.3) software on at least the MASTER and backup MASTER machines.
9. Reload the text version of the configuration file by running `tmloadcf(1)` on the MASTER machine.
10. Recompile and relink your application programs.
11. If you already have `DMTLOG` and `TLOG`, remove them.
12. Reboot the application.
13. As time permits, shut down and upgrade the other machines in your configuration.

Performing Additional Upgrade Steps to Windows Server

For a Windows machine running Oracle WebLogic Enterprise or Oracle Tuxedo 12c Release 2 (12.1.3) or earlier software, perform the following additional steps before installing the Oracle Tuxedo 12c Release 2 (12.1.3) 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.
Note: The Oracle Tuxedo `procMGR` service is called IPC Helper in Oracle Tuxedo 6.5 or Oracle WebLogic Enterprise 5.1.
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. Click the Close button.
3. Run the Windows Add/Remove program to remove the prior Oracle Tuxedo or Oracle WebLogic Enterprise 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 or Oracle WebLogic Enterprise software, select it, and click the Add/Remove button.
Note: If you have both the C++ and Java components of WebLogic Enterprise 5.1 installed on your machine, you must remove the Java software *before* removing the C++ software.
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.

Upgrading from Tuxedo Previous Releases to Tuxedo 12c Release 2 (12.1.3)

If you are upgrading from Tuxedo previous releases to Tuxedo 12c Release 2 (12.1.3), please note the following:

- In order to allow a Tuxedo 12c Release 2 (12.1.3) server to interoperate with Tuxedo 6.5 software, the Tuxedo 12c Release 2 (12.1.3) server `servopts -t` option must be specified. The server may be a workstation listener (WSL) process (where the `servopts -t` option allows interoperability for all of its workstation handler—WSH—processes), a domain gateway (GWTDOMAIN) process, or a system or application server process. For more information, see [GWTDOMAIN\(5\)](#) in the *Oracle Tuxedo File Formats, Data Descriptions, MIBs, And System Processes Reference*.
- In order to allow a domain gateway (GWTDOMAIN) in Tuxedo 12c Release 2 (12.1.3) 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 Tuxedo 12c Release 2 (12.1.3). 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, `tpreturn()` API method throws an exception when an error is encountered. You need to change your existing code to invoke `tpreturn()` API method in a try-catch block; otherwise, you will fail to successfully compile the code.

Performing a Hot Upgrade

Before performing a hot upgrade, back up files as described in [“Backing Up Files” on page 4-3](#).

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

Pre-Upgrade Notes

Once you have finished migrating the `MASTER` to Oracle Tuxedo 12c Release 2 (12.1.3) (via a hot upgrade), you will not be able to migrate it back to the acting backup until you have upgraded the backup to Oracle Tuxedo 12c Release 2 (12.1.3). In other words, migrating the `MASTER` is an irreversible procedure; once you have completed it, you cannot change the `MASTER` back to the release of the Oracle Tuxedo software on which it was running previously.

When you run `tmadmin(1)` on an Oracle Tuxedo 12c Release 2 (12.1.3) machine with an old-release `MASTER`, the `UPGRADE` file shown in [“Hot-Upgrade Example” on page 4-8](#) must reside in the current directory. To make sure that the files are in the current directory, you might, for example, run `tmadmin` from the Tuxedo application directory: `%APPDIR% (Windows)/ $APPDIR (UNIX) directory`.

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

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

To minimize downtime, Oracle recommends installing Oracle Tuxedo 12c Release 2 (12.1.3) on the `MASTER` machine and backup `MASTER`, and rebuilding all clients and servers in the directory

on each machine that is defined by `%APPDIR%/ $APPDIR`, in parallel with the directory in which the old-release clients and servers were created. In other words, both an old-release application and an Oracle Tuxedo 12c Release 2 (12.1.3) application should be available on each machine. *This advice is not a requirement, but a recommendation.*

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

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

Hot-Upgrade Procedure

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

1. Make sure your environment is set up as follows:cR2
 - 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” on page 5-36](#).
3. If upgrading a Windows machine running Oracle WebLogic Enterprise or Oracle Tuxedo 9.1 or earlier software, perform the additional steps specified in [“Performing Additional Upgrade Steps to Windows Server” on page 4-4](#).
4. Shut down the backup `MASTER` machine.

5. Install the Oracle Tuxedo 12c Release 2 (12.1.3) software on the backup MASTER machine and reboot the backup machine.
 6. Migrate the MASTER machine running the old release to the backup MASTER machine running Oracle Tuxedo 12c Release 2 (12.1.3).
 7. Shut down the machine that is now acting as the backup MASTER.
 8. Install Oracle Tuxedo 12c Release 2 (12.1.3) on the old MASTER.
 9. Using Oracle Tuxedo 12c Release 2 (12.1.3), reboot the machine that you shut down in step 7.
- For more information about hot upgrades, refer to the following section, [“Hot-Upgrade Example.”](#)

Hot-Upgrade Example

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

Suppose you have the following on a UNIX platform:

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

Refer to the following instruction sequence in [Listing 4-1](#) to perform a hot upgrade for your application.

Listing 4-1 Typical Hot Upgrade

```
MACH1> tmsshutdown -B MACH2 -l MACH2
MACH1> Run tmconfig to reset the APPDIR, TUXCONFIG, TUXDIR for MACH2
MACH2> Kill tlisten process
MACH2> Load Tuxedo 12cR2 if it is not already loaded
MACH2> cd $APPDIR2
MACH2> rm $TUXCONFIG2
MACH2> >UPGRADE      # Indicator that upgrade being done
```



```

MACH2> Reset PATH, LD_LIBRARY_PATH, TUXCONFIG, APPDIR
MACH2> Rebuild clients and servers, create TLOG
MACH2> Start Tuxedo 12cR2 tlisten
MACH1> >UPGRADE # Indicator that upgrade being done
MACH1> tmbot -B MACH2 -l MACH2
MACH2> tmadmin
master
y
psr
q
MACH2> tmsshutdown -B MACH1 -l MACH1
MACH2> Run tmconfig to reset the APPDIR, TUXCONFIG, TUXDIR for MACH1
MACH1> Kill tlisten process
MACH1> Load Tuxedo 12cR2 if it is not already loaded
MACH1> cd $APPDIR1
MACH1> rm $TUXCONFIG1
MACH1> Reset PATH, LD_LIBRARY_PATH, TUXCONFIG, APPDIR
MACH1> Rebuild clients and servers, create TLOG
MACH1> Start Tuxedo 12cR2 tlisten
MACH1> rm UPGRADE # Remove indicator that upgrade being done
MACH2> rm UPGRADE # Remove indicator that upgrade being done
MACH2> tmbot -B MACH1 -l MACH1

```

Note: If you are performing the upgrade illustrated in [Listing 4-1](#) from a Windows platform, be sure to stop the Oracle Tuxedo `procMGR` (called IPC Helper in Oracle Tuxedo 6.5 or Oracle WebLogic Enterprise 5.1) service after stopping the `tlisten` process and to start the Oracle Tuxedo `procMGR` service *before* starting the `tlisten` process.

Rebuilding an Application

Now that you have successfully installed the Oracle Tuxedo software, you need to rebuild all application clients and servers that you want to execute on a system upgraded to Oracle Tuxedo 12c Release 2 (12.1.3).

Notes: Rebuild is mandatory when upgrading from a previous Oracle Tuxedo version to Oracle Tuxedo 12c Release 2 (12.1.3).

The TMS server must be rebuilt when upgrading a previous Oracle Tuxedo version to Oracle Tuxedo 12c Release 2 (12.1.3).

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, may continue to use executables from an earlier release. For example, even if your `MASTER` and backup `MASTER` machines are running Oracle Tuxedo 12c Release 2 (12.1.3), 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 (12.1.3) only on Workstation clients, and you do not want to upgrade any of your Oracle Tuxedo system server machines to Oracle Tuxedo 12c Release 2 (12.1.3), you may do so. In other words, with any release level beginning with Oracle Tuxedo 6.5, Workstation clients may be intermixed freely in a configuration in which the `MASTER` machines is running Oracle Tuxedo 6.5 or later.

Installing Oracle Jolt 12c Release 2 (12.1.3) with Oracle WebLogic Server

This section provides supplementary installation instructions for customers using Oracle Jolt 12c Release 2 (12.1.3) with Oracle WebLogic Server 9.x, 10.x, or 12.1.x and Oracle Tuxedo 9.0, 9.1, 10.0, 10.3, 11.1.1.x, or 12.1.1. You can install the Jolt client personality software known as “Oracle Jolt for Oracle WebLogic Server” on an Oracle WebLogic Server 9.x, 10.x, or 12.1.x machine by copying the Jolt client personality software from an Oracle Tuxedo 12c Release 2 (12.1.3) machine to the target Oracle WebLogic Server machine. Of course, if the Oracle Tuxedo 12c Release 2 (12.1.3) machine and the target Oracle WebLogic Server machine are the same machine, you need only include the Jolt client library class files in your `CLASSPATH` variable setting.

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

For instructions on configuring the Jolt connection pool on Oracle WebLogic Server, see [Using Oracle Jolt with Oracle WebLogic Server](#).

Performing Post-Installation Tasks

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

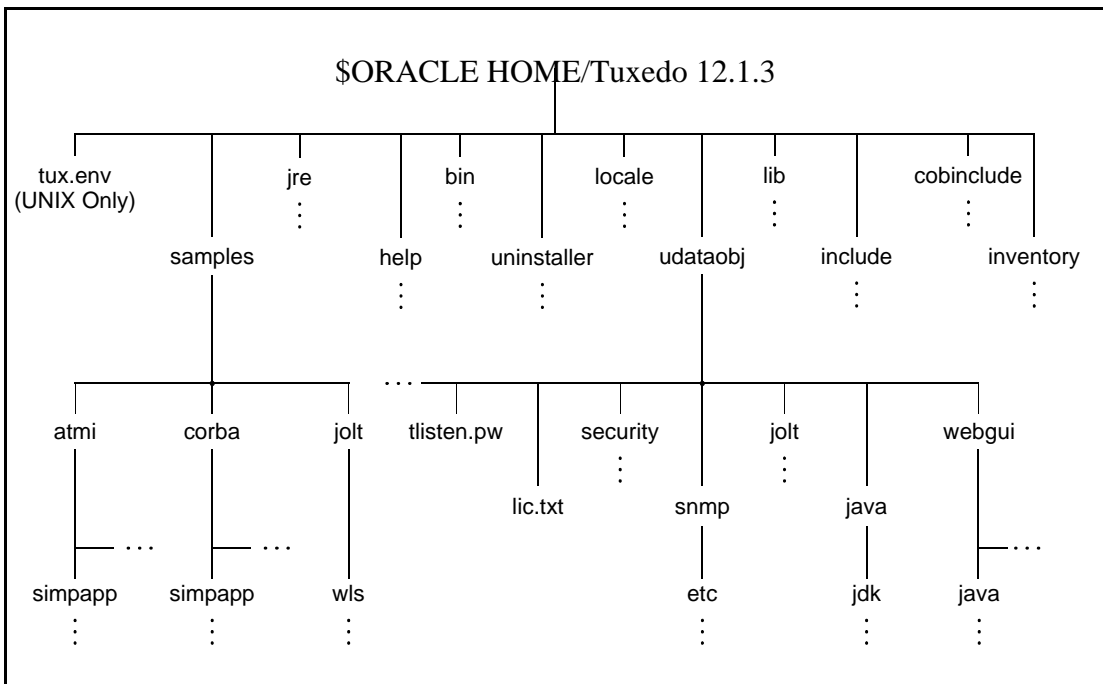
- [Understanding the Oracle Tuxedo Directory Structure](#)
- [Understanding the Oracle Service Architecture Leveraging Tuxedo Directory Structure](#)
- [Post Oracle TSAM Plus Installation Directory Structure](#)
- [Understanding the Oracle Tuxedo Architecture](#)
- [Using the Registry to Further Restrict Access on a Windows System](#)
- [Setting Up Your Environment](#)
- [Starting the tlisten Process](#)
- [Editing a UBBCONFIG File](#)
- [Using the TYPE Parameter in UBBCONFIG](#)
- [Checking IPC Requirements](#)
- [Creating the Universal Device List and the Transaction Log](#)
- [Running simpapp to Verify Your Installation](#)
- [Running buildtms for Oracle Tuxedo Applications That Use XA Resource Managers](#)
- [Uninstalling Oracle Tuxedo](#)
- [Reinstalling Oracle Tuxedo](#)

Understanding the Oracle Tuxedo Directory Structure

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

Oracle Tuxedo 12c Release 2 (12.1.3) Directory Structure is shown in [Figure 5-1](#).

Figure 5-1 Oracle Tuxedo 12c Release 2 (12.1.3) Directory Structure



The product directory shown here, `tuxedo12c` Release 2 (12.1.3), is the default for Oracle Tuxedo 12c Release 2 (12.1.3). The default name can be changed during installation.

The top-level directories and files of the Oracle Tuxedo directory structure are briefly described in [Table 5-1](#).

Table 5-1 Top-level Directories and Files of the Oracle Tuxedo Directory Structure

Directory Name	Description
<p>samples</p> <p>Note: Samples are not installed unless they are selected during Tuxedo 12c Release 2 (12.1.3) installation.</p>	<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 subdirectories:</p> <ul style="list-style-type: none"> • atmi A collection of simple applications that demonstrate the many features of the Oracle Tuxedo Application-to-Transaction Monitor Interface (ATMI) server software. • corba A collection of simple applications that demonstrate the many features of the Oracle Tuxedo Common Object Request Broker Architecture (CORBA) C++ server software. • jolt A collection of simple applications that demonstrate the many features of the Oracle Jolt server software.
help	<p>Contains online help files for the Oracle Tuxedo Administration Console.</p>
jre directory	<p>A directory containing the 1.5x version of the Java Runtime Environment (JRE). JRE 1.5.x provides the Java Virtual Machine, or JVM, required by the Oracle Tuxedo installation program. This version of the JRE is included in the Oracle Tuxedo 12c Release 2 (12.1.3) distribution. It is automatically installed in the Oracle Home directory when you install Oracle Tuxedo.</p> <p>Note: The JRE cannot be used for development. To develop Oracle Tuxedo Java client applications, you must ensure that the Java Development Kit (JDK) is installed on your system.</p>

Table 5-1 Top-level Directories and Files of the Oracle Tuxedo Directory Structure

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

Table 5-1 Top-level Directories and Files of the Oracle Tuxedo Directory Structure

Directory Name	Description
lib	Contains compiled object files, including dynamic shared libraries (for platforms on which Oracle Tuxedo uses dynamic shared libraries) and other object files needed to build Oracle Tuxedo clients and servers.
include	Contains C and C++ language header files, as well as OMG IDL files. Includes subdirectories <code>unicode</code> and <code>xercesc</code> ; may include subdirectory <code>rpc</code> , depending on the platform.
udataobj	<p>Contains other directories and files required by Oracle Tuxedo. The <code>udataobj</code> directory contains the following subdirectories and files:</p> <ul style="list-style-type: none"> • <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. • <code>jolt</code> Contains the files for the Oracle Jolt software components that you selected to install. • <code>snmp</code> Contains the <code>etc</code> directory, which in turn contains the SNMP Agent configuration files, MIB files, and other miscellaneous files. • <code>java</code> Contains the classes and Java archive files needed to run Java applications. • <code>webgui</code> Contains the Java and image files for the Oracle Tuxedo Administration Console. • <code>tlisten.pw</code> (file) Contains the <code>tlisten</code> administrative password that you entered during the installation.
tux.env (file)	UNIX systems only: Contains Oracle Tuxedo environment variables for UNIX installations and serves as a model for setting those variables.

Understanding the Oracle 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.

[Table 5-2](#) lists brief descriptions of SALT directories and files.

Table 5-2 SALT Directory and File Structure

Directory Name	Description
<code>bin</code>	<p>Contains SALT executable programs, including:</p> <ul style="list-style-type: none"> • <code>GWWS</code> • <code>tmwsdlgen</code> • <code>wsadmin</code> • <code>wsdlcvt</code> • <code>wsloadcf</code> • <code>tmwsdlflt</code> (internally used)
<code>include</code>	<p>Contains C language header files for SALT plug-in development and administrative development, including:</p> <ul style="list-style-type: none"> • <code>auth_pi_ex.h</code> • <code>credmap_pi_ex.h</code> • <code>custtype_pi_ex.h</code> • <code>wsadmflds.h</code> • <code>wsoapflds.h</code>
<code>lib</code>	<p>Contains dynamic shared libraries needed to run SALT programs, including:</p> <ul style="list-style-type: none"> • <code>libwsbase.*</code> (shared library suffix is variant on different platforms) • <code>libwsconf.*</code> • <code>libsoapeng.*</code>
<code>locale</code>	<p>Contains the localization of system messages used by SALT.</p> <p><code>C</code> subdirectory contains the following SALT specific message catalogs for the default locale (U.S. English): <code>GWWS</code>, <code>WSBASE</code>, <code>WSCFG</code>, <code>WSCMD</code>.</p> <p>file <code>CATNAMES</code> is the Oracle Tuxedo catalog name definition file that includes the above message catalog names used by SALT.</p>

Table 5-2 SALT Directory and File Structure

Directory Name	Description
samples/salt	A collection of simple applications that demonstrate the SALT features. Note: samples/salt is installed only if requested during installation
udataobj	Contains the following information files for SALT: <ul style="list-style-type: none"> 3rd_party_licenses_salt.txt License information for 3rd party software used by SALT wsadmflds SALT administrative FML32 field definitions wsoapflds SALT pre-defined FML32 fields for outbound Web Service call
inventory	Contains information about installed products for use with the Oracle Universal Installer.
udataobj/salt	Contains sub-directories and files required by SALT, including: <ul style="list-style-type: none"> policy subdirectory Contains SALT pre-defined WS-* policy files. ws subdirectory Contains various external XML Schema files required by SALT. xalan subdirectory Contains Apache Xalan Java 2.7 class library files and corresponding license information. xslt files wsdl*.xsl files are used by SALT WSDL converter (wsdlcvt). SALT XML Schema files *.xsd files are XML Schema files of SALT configuration file format.
uninst_salt_3	Contains files required to uninstall the SALT 12c Release 2 (12.1.3) software.

Post Oracle TSAM Plus Installation Directory Structure

After Oracle TSAM Plus installation has completed, the following directories are placed in the Oracle TSAM Plus installation directory:

- [Oracle TSAM Plus Agent Directory Structure \(For Linux\)](#)

- [Oracle TSAM Plus Manager Directory Structure \(For Linux\)](#)

Oracle TSAM Plus Agent Directory Structure (For Linux)

[Listing 5-1](#) shows the post-installation Oracle TSAM Plus Agent directories and files added to \$TUXDIR:

Listing 5-1 Oracle TSAM Plus Agent Directory Structure (For Linux)

```
./udataobj/fmb.def
./udataobj/monflds
./bin/accrpt
./bin/unregtsam.sh
./bin/LMS
./bin/tmping
./bin/regtsam.sh
./lib/libtmagent.a
./lib/libtsamjni.a
./lib/tsam_ext.o
./lib/libtsam.so
./lib/libtmjmx.a
./lib/libtmjmx.so
./lib/libtmagent.so
./lib/libtsam.a
./lib/libtsamjni.so
./include/e_perf_mon.h
./include/tsam_ext.h
./include/tsam_snmptrap.h
./include/monflds.h
./include/fmb.def.h
```

```
./locale/C/TSAM.text  
./locale/C/TSAM_CAT  
./jmx/jmxaccess.properties  
./jmx/tmjmx_tux.jar  
./jmx/tuxubb.jar  
./jmx/java.policy  
./jmx/com.bea.core.jatmi_1.0.0.0.jar  
./jmx/tmjmx_metadata.jar
```

Oracle TSAM Plus Manager Directory Structure (For Linux)

[Listing 5-2](#) shows the post-installation Oracle TSAM Plus Manager directories, sub-directories, and files added to `$ORACLE_HOME/tsam`:

Listing 5-2 Oracle TSAM Plus Manager Directory Structure (For Linux)

```
/apache-tomcat (bundled Tomcat installed only)  
...  
/TSAM (bundled Derby installed only)  
...  
/db-derby-bin (bundled Derby installed only)  
...  
/bin  
  
/3rd_party_licenses.txt  
  
/startup.sh
```

/shutdown.sh

/em

/12.1.0.2.0_oracle.fmw.txdo_2000_0.opar

/deploy

/tsam.ear

/AppServerDeployer.sh

...

/jdk

...

/logs (bundled Tomcat installed only)

...

/inventory

/ContentsXML

/comps.xml

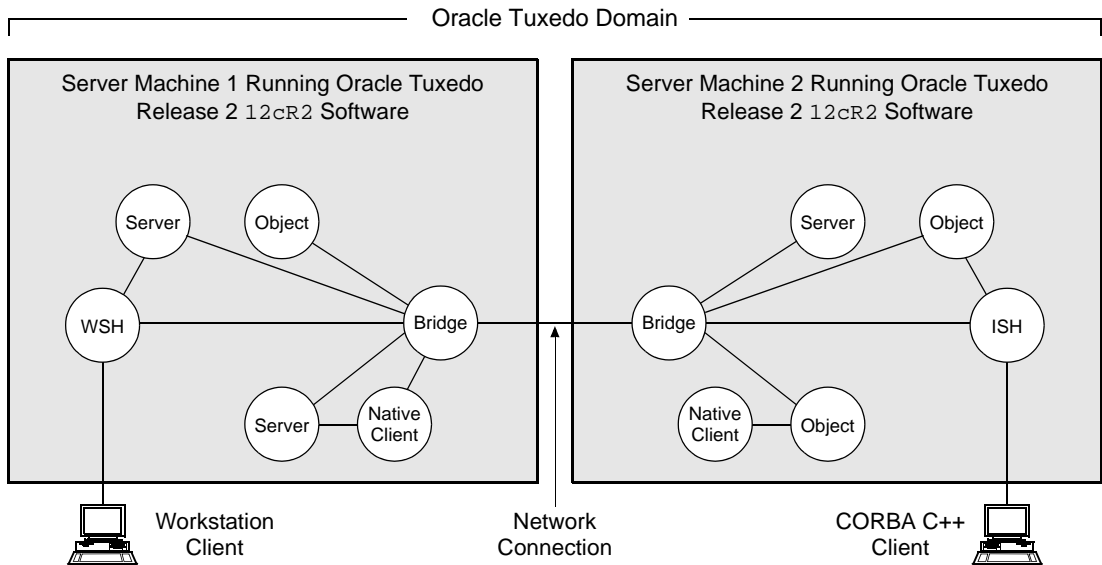
/ uninstall

...

Understanding the Oracle Tuxedo Architecture

Figure 5-2 shows an Oracle Tuxedo *domain*, which is the basis of the Oracle Tuxedo architecture.

Figure 5-2 Simplified View of Oracle Tuxedo Architecture



A Tuxedo domain, also known as a Tuxedo application, is a business software program, built upon the Tuxedo system, that is defined and controlled by a single configuration file—the `UBBCONFIG` file. A Tuxedo domain consists of many system processes, one or more application client processes, one or more application server processes, and one or more computer machines connected over a network.

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

- [UBBCONFIG File](#)
- [MASTER Machine](#)
- [TUXCONFIG File](#)
- [TUXCONFIG Environment Variable](#)

- [TUXDIR Environment Variable](#)

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

UBBCONFIG File

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

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

MASTER Machine

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

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

TUXCONFIG File

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

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

TUXCONFIG Environment Variable

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

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

TUXDIR Environment Variable

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

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

Using the Registry to Further Restrict Access on a Windows System

Oracle Tuxedo-provided client programs are run directly by users with the users' own permissions. In addition, users running native clients (that is, clients running on the same machine on which the server program is running) have access to the UBBCONFIG file and interprocess communication (IPC) mechanisms such as the *bulletin board* (a reserved piece of shared memory in which parameters governing the application and statistics about the application are stored).

To gain access to the Oracle Tuxedo functionality, native clients join an Oracle Tuxedo application using the identity of the application administrator (`tpsadm`). However, because `tpsadm` is a trusted user, this setting causes the Oracle Tuxedo system to bypass the user authentication process.

To prevent this lapse in security on your Windows 2003 server machine, follow these steps:

1. Choose Start → Run to launch the Run dialog box, enter `regedt32`, and click OK to launch the Registry Editor window.

2. Select `HKEY_LOCAL_MACHINE` → `Software` → `Oracle Systems` → `Tuxedo` → `10.3.0.0_VS2005`.
3. Select `SECURITY` → `Permissions`.
4. Disable control for `Everyone` and allow access only to users with administrative privileges.

Setting Up Your Environment

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

Table 5-3 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 will be 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 <code>UBBCONFIG</code> configuration file.
WEBJAVADIR	Absolute pathname of the Java and image files for the Oracle Tuxedo Administration Console on this server machine.

Table 5-4 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 5-5 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.

Table 5-6 Java Environment Variables

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

* The Oracle Tuxedo 12c Release 2 (12.1.3) distribution does not include a JDK.

Setting Environment Variables on a Windows System

On a Windows 2003 server machine, you need to set the following environment variables shown in [Table 5-7](#) to set up your environment:

Table 5-7 Environment Variables

```
set TUXDIR=pathname_of_Oracle_Tuxedo_product_directory
```

```
set APPDIR=pathname_of_Oracle_Tuxedo_application_directory
```

```
set TUXCONFIG=pathname_of_TUXCONFIG_file
```

```
set WEBJAVADIR=%TUXDIR%\udataobj\webgui\java
```

```
set PATH=%APPDIR%;%TUXDIR%\bin;\bin;%PATH%
```

Examples of TUXDIR, APPDIR, and TUXCONFIG are:

```
TUXDIR=C:\Oracle\tuxedo12cR1
APPDIR=C:\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. As an alternative to setting environment variables from a command-line shell, use the Environment page of the Oracle Administration program, described in [“Setting and Modifying Environment Variables”](#) in *Using Oracle Tuxedo ATMI on Windows*.

Windows 2003 Server accesses the required dynamically loadable library files through its PATH variable setting. Specifically, Windows 2000 searches for dynamically loadable library files in the following order:

1. The directory from which the Oracle Tuxedo application was loaded
2. The current directory
3. The Windows system directory (for example, C:\WINDOWS\System32)
4. The Windows directory (for example, C:\WINDOWS)
5. The directories listed in the PATH environment variable

For more information about setting environment variables, see [“Setting and Modifying Environment Variables”](#) in *Using Oracle Tuxedo ATMI on Windows*.

Setting Environment Variables on a UNIX System

On a UNIX server machine, set and export the following environment variables shown in [Table 5-8](#) to set up your environment:

Table 5-8 Environment Variables

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

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

```
TUXDIR=/home/Oracle/tuxedo12cR1
APPDIR=/home/me/simpapp
TUXCONFIG=$APPDIR/tuxconfig
```

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

Starting the tlisten Process

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

In addition to the installer program starting a `tlisten` process on port 3050 during the installation of Oracle Tuxedo, a `tlisten` process may be started as shown in [tlisten Process Table 5-9](#).

Table 5-9 tlisten Process

On This Machine . . .	By This Administrator . . .	Using This Method . . .
Windows 2003 server	Oracle Tuxedo application administrator	Listener page of the Oracle Administration program; for details, see “Configuring tlisten Processes to Start Automatically” in <i>Using Oracle Tuxedo ATMI on Windows</i>
		Manually starting a <code>tlisten</code> process from a command-line shell
UNIX server	UNIX system administrator	As part of a UNIX initialization (boot) script
	Oracle Tuxedo application administrator	As a <code>cron</code> job
		Manually starting a <code>tlisten</code> process from a command-line shell

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 [UBBCONFIG\(5\)](#) in *OracleTuxedo File Formats, Data Descriptions, MIBs, and System Processes Reference*.

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

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

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

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

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

tlisten Password

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

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

A common password is required for two machines in 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.

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`.

[Listing 5-3](#) shows an example `ubbsimple` file. The example has been modified from the `ubbsimple` file delivered on a Windows or UNIX system to include example pathname values for both Windows and UNIX systems.

Listing 5-3 ubbsimple for the ATMI-based simpapp Application

```
#ident "@(#)apps:simpapp/ubbsimple $Revision: 1.3 $  
  
#Skeleton UBBCONFIG file for the Tuxedo Simple Application.  
#Replace the <bracketed> items with the appropriate values.  
  
*RESOURCES  
IPCKEY <Replace with a valid IPC Key>
```

```

#Example:
#IPCKEY          123456

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

*MACHINES
DEFAULT:
APPDIR="<Replace with the current directory pathname>"
TUXCONFIG="<Replace with your TUXCONFIG Pathname>"
TUXDIR="<Directory where Tuxedo is installed>"

#Windows
#Example:
# APPDIR="C:\home\me\simpapp"
# TUXCONFIG="C:\home\me\simpapp\tuxconfig"
# TUXDIR="C:\oracle\tuxedo12cR1"

#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"
simplserv       SRVGRP=GROUP1 SRVID=1

*SERVICES
TOUPPER

```

In the configuration file for your application, you must replace the strings enclosed in angle brackets with values specific to your application. [Listing 5-10](#) provides a sample of the parameters that must be defined in every configuration file.

Table 5-10 Parameter Sample

This Parameter . . .	Specifies . . .
IPCKEY	A numeric key that identifies the shared memory segment where the structures used by your application are located. The value must be greater than 32,768 and less than 262,143.
<i>machine_name</i>	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.
APPDIR = <i>string</i>	A list of one or more directories in which application and administrative servers will be booted on this machine. For Windows, the value of <i>string</i> 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 <i>string</i> 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 = <i>string</i>	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 = <i>string</i>	The absolute pathname of the product directory of the Oracle Tuxedo software on this machine.

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

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

Using the TYPE Parameter in UBBCONFIG

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

Checking IPC Requirements

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

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

```
tmloadcf -c UBBCONFIG
```

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

[Listing 5-4](#) is an output report based on the values in `ubbsimple`.

Listing 5-4 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 $1 \leq A \leq 8$.

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 12c Release 2 \(12.1.3\) Platform Data Sheets”](#) on page A-1. To map the traditional names to the names specific to the Windows 2003 Server platform, see the table titled [“IPC Resource Name Mappings Between Windows and UNIX Systems”](#) in *Using Oracle Tuxedo ATMI on Windows*.

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 `SEMMP` 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 Windows 2003 Server system, see [“Configuring IPC Resources to Maximize System Performance”](#) in *Using Oracle Tuxedo ATMI on Windows*. For instructions on changing the current IPC values for a UNIX system, see the data sheet for your platform in [“Oracle Tuxedo 12c Release 2 \(12.1.3\) Platform Data Sheets”](#) on page A-1.

Creating the Universal Device List and the Transaction Log

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

Defining the TLOG

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

Table 5-11 lists the parameters.

Table 5-11 Parameter

This Parameter . . .	Specifies . . .
<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.

Creating the UDL and UDL Entries for TLOG

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

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

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

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

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

Here `-z config` specifies the full pathname of the device on which the UDL should be created (that is, where the TLOG will reside), and `-b blocks` specifies the number of blocks to be allocated on the device. The value of `config` should match the value of the `TLOGDEVICE` parameter in the `MACHINES` section of the `UBBCONFIG` file. The blocks must be larger than the value of `TLOGSIZE`. If `-z` is not specified, the value of `config` defaults to the value of the `FSCONFIG` environment variable.

3. Log in as the application administrator on each remaining non-`MASTER` machine that will use global transactions and repeat step 2.

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

Running `simpapp` to Verify Your Installation

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

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

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

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

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

For example, the invocation

```
simpcl "hello world"
```

results in the output

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

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

Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation

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

Note: Samples are not installed unless they are selected to be installed during Tuxedo 12c Release 2 (12.1.3) installation.

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

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

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

1. Log in to the target machine as the Administrator or as a member of the Administrator group and open a command-line shell.
2. Create a working directory for your sample application and change to it:

```
cd C:\home\me
mkdir atmi
cd atmi
```

3. Set the environment variables used by the Oracle Tuxedo system, as explained in “[Setting Environment Variables on a Windows System](#)” on page 5-16. Set `APPDIR` and `TUXCONFIG` as follows:

```
set APPDIR=C:\home\me\atmi
set TUXCONFIG=%APPDIR%\tuxconfig
```

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

4. Copy the `simpapp` files to your working directory. You will need to edit one of them: the configuration file, `ubbsimple`. Check the permissions on all the files in your working directory and, if necessary, change the permissions to allow full access. For example:

```
copy %TUXDIR%\samples\atmi\simpapp\*. * *.*
attrib -R /S *.*
```

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 `"C:\home\me\atmi"`.
- Set `TUXCONFIG` to the literal pathname corresponding to `$APPDIR/tuxconfig` (in our example, `"C:\home\me\atmi\tuxconfig"`).
- Set `TUXDIR` to the absolute pathname of the product directory of the Oracle Tuxedo software on this machine (for example, `"C:\oracle\tuxedo12cR1"`).
- Set `MACHINE-NAME` to the name of your system. `MACHINE-NAME` must be entered in uppercase letters on a Windows system.

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

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

```
tmloadcf -y ubbsimple
```

8. Boot `simpapp` by typing the following command:

```
tmboot -y
```

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

[Listing 5-5](#) shows the output produced by `tboot -y`.

Listing 5-5 Output Produced by `tboot -y`

```

Booting all admin and server processes in C:\home\me\atmi\tuxconfig
INFO: Oracle Tuxedo, Version 12.1.3.0.0
INFO: Serial #: 000102-9125503751, Maxusers 25
Booting admin processes ...
exec BBL -A:
    process id=24180 ... Started.
Booting server processes ...
exec simpserv -A :
    process id=24181 ... Started.
2 processes started.

```

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

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

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

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

```
tmloadcf -c %APPDIR%\ubbsimple
```

If the current value of any IPC parameter configured in your operating system is less than a minimum (either variable or fixed) listed in the `tmloadcf` output, you must increase the value of that parameter. For instructions on determining and changing the current IPC values for your platform, see [“Configuring IPC Resources to Maximize System Performance”](#) in *Using Oracle Tuxedo ATMI on Windows*.

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

```
simpcl "hello world"
```

The following is displayed:

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

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

```
tmshutdown -y
```

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

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

1. Log in to the target machine as the Oracle Tuxedo application administrator and open a command-line shell.
2. Create a working directory for your sample application and change to it:

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

3. Set and export the environment variables used by the Oracle Tuxedo system, as explained in [“Setting Environment Variables on a UNIX System” on page 5-17](#). Set `APPDIR` and `TUXCONFIG` as follows:

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

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

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

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

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

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

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

- Set `IPCKEY` to a valid IPC key. This value must be greater than 32,768 and less than 262,143.
- Set `APPDIR` to `"/home/me/atmi"`.
- Set `TUXCONFIG` to the literal pathname corresponding to `$APPDIR/tuxconfig` (in our example, `"/home/me/atmi/tuxconfig"`).
- Set `TUXDIR` to the absolute pathname of the product directory of the Oracle Tuxedo software on this machine (for example, `"/home/oracle/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.

[Listing 5-6](#) shows the output produced by `tmboot -y`.

Listing 5-6 Output Produced by `tmboot -y`

```
Booting all admin and server processes in /home/me/atmi/tuxconfig
INFO: Oracle Tuxedo , Version 12.1.3.0.0
INFO: Serial #: 000102-9125503751, Maxusers 25
Booting admin processes ...
exec BBL -A:
    process id=24180 ... Started.
Booting server processes ...
exec simpserv -A :
```

```
process id=24181 ... Started.  
2 processes started.
```

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

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

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

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

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

If the current value of any IPC parameter configured in your operating system is less than a minimum (either variable or fixed) listed in the `tmloadcf` output, you must increase the value of that parameter. For instructions on determining and changing the current IPC values for your platform, see the data sheet for your platform in [“Oracle Tuxedo 12c Release 2 \(12.1.3\) Platform Data Sheets” on page A-1](#).

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

```
simpcl "hello world"
```

The following is displayed:

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

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

```
tmshutdown -y
```

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

To verify that you have successfully installed the Oracle Tuxedo CORBA C++ software on your system, run the CORBA version of the `simpapp` application, which on a Windows system can be found in the directory `%TUXDIR%\samples\corba\simpapp`, and 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 Windows System

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

1. Log in to the target machine as the Administrator or as a member of the Administrator group and open a command-line shell.

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

```
cd C:\home\me
mkdir corba
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 `C:\oracle\tuxedo12cR1` directory, set `TUXCONFIG` as follows:

```
set TUXDIR=C:\oracle\tuxedo12cR1
```

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

```
copy %TUXDIR%\samples\corba\simpapp\*.* *.*
attrib -R /S *.*
```

5. Ensure that `nmake` is in your path.
6. To run `simpapp` automatically, enter `runme`. 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 the sample manually to observe the `simpapp` processes starting and stopping, follow these steps:
 - a. Enter `results\setenv`.
 - b. Enter `tmboot -y`. The application starts several processes.
 - c. Enter `simple_client`. The prompt `String?` appears.
 - d. Enter a word in lowercase letters. The application converts the word to uppercase and then to lowercase letters.
 - e. Enter `tmshutdown -y`. The application shuts down the processes.
8. To restore the directory to its original state, perform these steps:
 - a. `results\setenv`
 - b. `nmake -f makefile.nt clean`

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> tmbboot -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> tmsshutdown -y`

g. The application shuts down the processes.

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

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

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

Running buildtms for Oracle Tuxedo Applications That Use XA Resource Managers

For Oracle Tuxedo applications that use distributed transactions and XA-compliant resource managers, you must use the `buildtms` command to construct a transaction manager server load module. This requirement exists on Windows 2003 Server and UNIX systems. When the module has been created, it must reside in `%TUXDIR%\bin` on Windows 2003 Server systems, or `$TUXDIR/bin` on UNIX systems.

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

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

Uninstalling Oracle Tuxedo

Uninstalling Oracle Tuxedo 12c Release 2 (12.1.3), 10.0, 9.1, 9.0, 8.1 or 8.0 does not remove the Oracle Home directory associated with the installation but does remove all Oracle Tuxedo components installed by the installer program. The uninstallation also removes the product directory associated with the installation.

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.

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.

Oracle Tuxedo 12c Release 2 (12.1.3) Platform Data Sheets

The following sections provide platform-specific information for the platforms on which the Oracle Tuxedo 12c Release 2 (12.1.3) system software is supported:

- [Supported Platforms](#)
 - [Platform Support Policy](#)
- [Supported Platform Data Sheets](#)
 - [HP-UX 11i v3 \(32-bit\) on Itanium 64-bit](#)
 - [HP-UX 11i v3 \(64-bit\) on Itanium](#)
 - [IBM AIX 6.1 \(32-bit\) on IBM PowerPC](#)
 - [IBM AIX 6.1 \(64-bit\) on IBM PowerPC](#)
 - [IBM AIX 7.1 \(32-bit\) on IBM PowerPC](#)
 - [IBM AIX 7.1 \(64-bit\) on IBM PowerPC](#)
 - [Novell SUSE Linux Enterprise Server 10 \(32-bit\) on x86](#)
 - [Novell SUSE Linux Enterprise Server 11 \(32-bit\) on x86-64](#)
 - [Novell SUSE Linux Enterprise Server 12 \(64-bit\) on x86-64](#)
 - [Oracle Linux 5.6 \(64-bit\) on x86-64](#)
 - [Oracle Linux 5.8 \(32-bit\) on x86-64](#)
 - [Oracle Linux 6.0 \(64-bit\) on x86-64](#)

- Oracle Linux 6.5 (32-bit) on x86-64
- Oracle Linux 7.0 (32-bit) on x86-64
- Oracle Linux 7.0 (64-bit) on x86-64
- Oracle Linux 5.6 (64-bit) on Exalogic 2.0
- Oracle Linux 5.8 (64-bit) on Exalogic 2.0
- Oracle Linux 6.0 (64-bit) on Exalogic 2.0
- Oracle Solaris 10 (32-bit) on SPARC
- Oracle Solaris 10 (64-bit) on SPARC
- Oracle Solaris 11 (32-bit) on SPARC
- Oracle Solaris 11 (64-bit) on SPARC
- Red Hat Enterprise Linux 5.6 (64-bit) on x86-64
- Red Hat Enterprise Linux 6.0 (64-bit) on x86-64
- Red Hat Enterprise Linux 6.3 (64-bit) on s390x
- Red Hat Enterprise Linux 6.5 (32-bit) on x86-64
- Red Hat Enterprise Linux 7.0 (32-bit) on x86-64
- Red Hat Enterprise Linux 7.0 (64-bit) on x86-64
- NeoKylin Linux 6.5 (64-bit) on x86-64
- NeoKylin Linux 7.2 (64-bit) on x86-64
- Microsoft Windows Server 2008 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010
- Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2010
- Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2012
- Microsoft Windows Server 2012 (32-bit) on x86-64 with MS Visual Studio 2013
- Microsoft Windows Server 2012 (64-bit) on x86-64 with MS Visual Studio 2012
- Microsoft Windows Server 2012 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010
- Microsoft Windows 7 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012
- Microsoft Windows 8 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012

- Microsoft Windows 8.1 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010
- Microsoft Windows 8.1 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012
- Microsoft Windows 10 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012

Supported Platforms

Vendor	Operating System	Release/Version
HP	HP-UX	HP-UX 11i v3 (32-bit) on Itanium 64-bit HP-UX 11i v3 (64-bit) on Itanium
IBM	AIX	IBM AIX 6.1 (32-bit) on IBM PowerPC IBM AIX 6.1 (64-bit) on IBM PowerPC IBM AIX 7.1 (32-bit) on IBM PowerPC IBM AIX 7.1 (64-bit) on IBM PowerPC
Novell	Linux	Novell SUSE Linux Enterprise Server 10 (32-bit) on x86 Novell SUSE Linux Enterprise Server 11 (32-bit) on x86-64 Novell SUSE Linux Enterprise Server 12 (64-bit) on x86-64

Vendor	Operating System	Release/Version
Oracle	Linux	Oracle Linux 5.6 (64-bit) on x86-64
		Oracle Linux 5.8 (32-bit) on x86-64
		Oracle Linux 6.0 (64-bit) on x86-64
		Oracle Linux 6.5 (32-bit) on x86-64
		Oracle Linux 7.0 (32-bit) on x86-64
		Oracle Linux 7.0 (64-bit) on x86-64
		Oracle Linux 5.6 (64-bit) on Exalogic 2.0
		Oracle Linux 5.8 (64-bit) on Exalogic 2.0
	Oracle Linux 6.0 (64-bit) on Exalogic 2.0	
	Solaris	Oracle Solaris 10 (32-bit) on SPARC
Oracle Solaris 10 (64-bit) on SPARC		
Oracle Solaris 11 (32-bit) on SPARC		
Oracle Solaris 11 (64-bit) on SPARC		
Red Hat	Linux	Red Hat Enterprise Linux 5.6 (64-bit) on x86-64
		Red Hat Enterprise Linux 6.0 (64-bit) on x86-64
		Red Hat Enterprise Linux 6.3 (64-bit) on s390x
		Red Hat Enterprise Linux 6.5 (32-bit) on x86-64
		Red Hat Enterprise Linux 7.0 (32-bit) on x86-64
		Red Hat Enterprise Linux 7.0 (64-bit) on x86-64

Vendor	Operating System	Release/Version
NeoKylin	Linux	NeoKylin Linux 6.5 (64-bit) on x86-64 NeoKylin Linux 7.2 (64-bit) on x86-64
Microsoft	Windows	Microsoft Windows Server 2008 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010 Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2010 Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2012 Microsoft Windows Server 2012 (32-bit) on x86-64 with MS Visual Studio 2013 Microsoft Windows Server 2012 (64-bit) on x86-64 with MS Visual Studio 2012 Microsoft Windows Server 2012 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010 Microsoft Windows 7 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012 Microsoft Windows 8 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012 Microsoft Windows 8.1 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010 Microsoft Windows 8.1 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012 Microsoft Windows 10 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012

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

- A list of available Oracle Tuxedo Install Types
- Hardware, software, network, and disk space requirements

- Instructions for mounting and unmounting the Oracle Tuxedo software DVD
- Tunable parameters

Platform Support Policy

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

- x86 or x86-64 denotes various CPUs based on x86 or x86-64 architecture.
- Oracle Tuxedo 32-bit binaries are supported on 64-bit operating systems, such as on Solaris, AIX and Windows. This support is based on OS vendor compatibility assurance that 32-bit applications can run on 64-bit OS without any changes. For more 32-bit application support on 64-bit OS information, please refer to respective OS vendor documentation.
- 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 12.1.3 is certified on Red Hat Enterprise Linux 6.0, Red Hat Enterprise Linux 6.x (x>0) are supported with

the assumption that Red Hat will maintain binary compatibility among all minor versions of release 6.

- 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.

Tunable Parameters

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

To adjust the tunable parameters, follow these steps:

1. Determine whether the current values are adequate.

For instructions about determining whether the current tunable parameter values are adequate, see "[IPC Resource Configuration on a UNIX System](#)" and "[Checking IPC Requirements](#)".

2. Reset the tunable parameters as necessary.

Supported Platform Data Sheets

HP-UX 11i v3 (32-bit) on Itanium 64-bit

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for HP-UX 11i v3 (32-bit) system on Itanium 64-bit.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for HP-UX 11i v3 (32-bit) on Itanium 64-bit

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

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

Hardware Requirements for HP-UX 11i v3 (32-bit) on Itanium 64-bit

- Intel Itanium (64-bit)

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

Software Requirements for HP-UX 11i v3 (32-bit) on Itanium 64-bit

Component	Requirement
OS Version (Patches)	HP-UX B.11.31 ia64
C/C++ and COBOL compilers	C/C++: HP C/aC++ B3910B A.06.12 [Nov. 03 2006] COBOL: Micro Focus 5.1
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0.07, JDK 1.7.0.11, and JDK 1.8.0.00-hp-ux
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with Java 2 JRE 1.6.0.07, JDK 1.7.0.11, and JDK 1.8.0.00-hp-ux
Database for CORBA C++ applications	Tested with Oracle Database 11gR2 and Oracle Database 12c
IBM WebSphere MQ for Oracle Tuxedo MQ Adapter	Tested with WebSphere MQ 7.0.1.3
SSL Certificate Authorities	Verisign
Lightweight Directory Access Protocol (LDAP) Directory Servers	iPlanet Directory Server; needed to retrieve X.509v3 digital certificates for SSL

Additional Notes

- If you 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`
- 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.

Network Requirements for HP-UX 11i v3 (32-bit) on Itanium 64-bit

- TCP/IP provided by the operating system

Disk Space Requirements for HP-UX 11i v3 (32-bit) on Itanium 64-bit

For all install types and components, 1.05GB is the minimum disk space requirement for installation on HP-UX 11i v3 (32-bit) on Itanium 64-bit systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for HP-UX 11i v3 (32-bit) on Itanium 64-bit

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

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

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for HP-UX 11i v3 (64-bit) system on Itanium.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for HP-UX 11i v3 (64-bit) on Itanium

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

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

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

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++ and COBOL compilers	C/C++: HP C/aC++ B3910B A.06.12 [Nov. 03 2006] COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0.07
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with Java 2 JRE 1.6.0.07
Database for CORBA C++ applications	Tested with Oracle Database 11gR1 and Oracle Database 12c
IBM WebSphere MQ for Oracle Tuxedo MQ Adapter	Tested with WebSphere MQ 7.0.1.3

Additional Notes

- 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`
- 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.

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

- TCP/IP, using the Sockets network interface

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

For all install types and components, 1GB 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.

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.

IBM AIX 6.1 (32-bit) on IBM PowerPC

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for AIX 6.1 (32-bit) systems on IBM PowerPC.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for IBM AIX 6.1 (64-bit)

- Full install type
- Server install type

- Client install type

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

Hardware Requirements for IBM AIX 6.1 (32-bit)

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

Software Requirements for IBM AIX 6.1 (32-bit)

Component	Requirement
OS Version (Patches)	IBM AIX 6.1 64-bit
C/C++ and COBOL compilers	C/C++: IBM XL C/C++ Enterprise Edition V10, V11, or V12 for AIX or later compatible versions COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0 and JRE 7
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0 and JRE 7
Non-Oracle CORBA Java clients	Tested with Java 2 SDK 1.6.0 (or higher) Interface Definition Language (IDL) ORB (run time)
Database for CORBA C++ applications	Tested with Oracle Database 11gR1 and Oracle Database 12c

Additional Notes

- If you use GAUTHSVR with JDK1.6 or later, please specify the following:
`JAVA_OPTS=-Djavax.xml.stream.XMLInputFactory=com.bea.xml.stream.MXParse
rFactory` in your environment before booting GAUTHSVR. If you use GAUTHSVR with Java 1.6, please add the 64-bit JRE library path to LIBPATH. The default library is located in

```
/usr/java6_64/jre/lib/ppc64. Run export
LIBPATH=/usr/java6_64/jre/lib/ppc64:$LIBPATH to set the correct LIBPATH.
```

- When using the Oracle MQ Adapter for Oracle Tuxedo in an AIX version 6.1 64-bit environment, if the `dltmqlnk` command is not executed to remove the symbolic link to the 32-bit WebSphere MQ libraries, you must export the `TUXEDO_LIBPATH_PREPEND` environment variable and set its value to the 64-bit WebSphere MQ libraries path.
- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.3.0 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- When developing CORBA C++ client/server applications, you must use `ifdef` directives to include standard C++ headers or old C headers, otherwise compiler will fail.

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

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

- For 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.

Network Requirements for IBM AIX 6.1 (32-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for IBM AIX 6.1 (32-bit)

For all install types and components, 449,800KB is the minimum disk space requirement for installation on IBM AIX 6.1 (32-bit) IBM PowerPC systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for IBM AIX 6.1 (32-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".

IBM AIX 6.1 (64-bit) on IBM PowerPC

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for IBM AIX 6.1 (64-bit) on IBM PowerPC systems.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for IBM AIX 6.1 (64-bit) on IBM PowerPC

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

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

Hardware Requirements for IBM AIX 6.1 (64-bit)

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

Software Requirements for IBM AIX 6.1 (64-bit)

Component	Requirement
OS Version (Patches)	IBM AIX 6.1 64-bit
C/C++ and COBOL compilers	C/C++: IBM XL C/C++ Enterprise Edition V10, V11, or V12 for AIX or later compatible versions COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0 and JRE 7
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0 and JRE 7
Non-Oracle CORBA Java clients	Tested with Java 2 SDK 1.6.0 (or higher) Interface Definition Language (IDL) ORB (run time)
Database for CORBA C++ applications	Tested with Oracle Database 11gR1 and Oracle Database 12c

Additional Notes

- If you use GAUTHSVR with JDK1.6 or later, please specify the following:
`JAVA_OPTS=-Djavax.xml.stream.XMLInputFactory=com.bea.xml.stream.MXParserFactory` in your environment before booting GAUTHSVR. If you use GAUTHSVR with Java 1.6, please add the 64-bit JRE library path to LIBPATH. The default library is located in `/usr/java6_64/jre/lib/ppc64`. Run export `LIBPATH=/usr/java6_64/jre/lib/ppc64:$LIBPATH` to set the correct LIBPATH.
- When using the Oracle MQ Adapter for Oracle Tuxedo in an AIX version 6.1 64-bit environment, if the `dlmqlnk` command is not executed to remove the symbolic link to the 32-bit WebSphere MQ libraries, you must export the `TUXEDO_LIBPATH_PREPEND` environment variable and set its value to the 64-bit WebSphere MQ libraries path.
- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.3.0 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.

- CORBA C++ users need a C++ compiler and linker.
- When developing CORBA C++ client/server applications, you must use `ifdef` directives to include standard C++ headers or old C headers, otherwise compiler will fail.

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

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

Network Requirements for IBM AIX 6.1 (64-bit)

- TCP/IP, using the Sockets network interface

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

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

Temporary Storage Space Requirements for IBM AIX 6.1 (64-bit)

For all install types and components, 105MB is the minimum temporary storage space requirement for installation on IBM AIX 6.1 (64-bit) IBM PowerPC systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for IBM AIX 6.1 (64-bit)

No IPC configuration is required for AIX. To check the value of a kernel tuning parameter (maxuproc only), enter `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".

IBM AIX 7.1 (32-bit) on IBM PowerPC

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for AIX 7.1 (32-bit) systems on IBM PowerPC.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for IBM AIX 7.1 (32-bit)

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

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

Hardware Requirements for IBM AIX 7.1 (32-bit)

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

Software Requirements for IBM AIX 7.1 (32-bit)

Component	Requirement
OS Version (Patches)	IBM AIX 7.1 64-bit
C/C++ compilers	C/C++: IBM XL C/C++ for AIX, V11.1
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0

Component	Requirement
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0
Non-Oracle CORBA Java clients	Tested with Java 2 SDK 1.6.0 (or higher) Interface Definition Language (IDL) ORB (run time)
Database for CORBA C++ applications	Tested with Oracle Database 11gR2

Additional Notes

- ATMI users need a C, C++, or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- When developing CORBA C++ client/server applications, you must use `ifdef` directives to include standard C++ headers or old C headers, otherwise compiler will fail.

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

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

- For 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.

Network Requirements for IBM AIX 7.1 (32-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for IBM AIX 7.1 (32-bit)

For all install types and components, 603MB is the minimum disk space requirement for installation on IBM AIX 7.1 (32-bit) IBM PowerPC systems. This requirement assumes installation of the default components for the selected install type.

Temporary Storage Space Requirements for IBM AIX 7.1 (32-bit)

For all install types and components, 104MB is the minimum temporary storage space requirement for installation on IBM AIX 7.1 (32-bit) IBM PowerPC systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for IBM AIX 7.1 (32-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".

IBM AIX 7.1 (64-bit) on IBM PowerPC

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for AIX 7.1 (64-bit) systems on IBM PowerPC.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for IBM AIX 7.1 (64-bit)

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

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

Hardware Requirements for IBM AIX 7.1 (64-bit)

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

Software Requirements for IBM AIX 7.1 (64-bit)

Component	Requirement
OS Version (Patches)	IBM AIX 7.1 64-bit
C/C++ and COBOL compilers	C/C++: IBM XL C/C++ for AIX, V10 or V11 COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0 and JRE 1.7.0
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0 and JDK 1.7.0
Non-Oracle CORBA Java clients	Tested with Java 2 SDK 1.6.0 (or higher) Interface Definition Language (IDL) ORB (run time)
Database for CORBA C++ applications	Tested with Oracle Database 11gR2 and Oracle Database 12c

Additional Notes

- If you use GAUTHSVR with JDK1.6 or later, please specify `JAVA_OPTS=-Djavax.xml.stream.XMLInputFactory=com.bea.xml.stream.MXParserFactory` in your environment before booting GAUTHSVR. If you use GAUTHSVR with Java 1.6, please add the 64-bit JRE library path to `LIBPATH`. The default library is located in `/usr/java6_64/jre/lib/ppc64`. Run `export LIBPATH=/usr/java6_64/jre/lib/ppc64:$LIBPATH` to set the correct `LIBPATH`.
- When using the Oracle MQ Adapter for Oracle Tuxedo in an AIX version 7.1 64-bit environment, if the `dltmqlnk` command is not executed to remove the symbolic link to the 32-bit WebSphere MQ libraries, you must export the `TUXEDO_LIBPATH_PREPEND` environment variable and set its value to the 64-bit WebSphere MQ libraries path.
- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.3.0 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.

- CORBA C++ users need a C++ compiler and linker.
- When developing CORBA C++ client/server applications, you must use `ifdef` directives to include standard C++ headers or old C headers, otherwise compiler will fail.

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

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

Network Requirements for IBM AIX 7.1 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for IBM AIX 7.1 (64-bit)

For all install types and components, 430, 448KB is the minimum disk space requirement for installation on IBM AIX 7.1 (64-bit) IBM PowerPC systems. This requirement assumes installation of the default components for the selected install type.

Temporary Storage Space Requirements for IBM AIX 7.1 (64-bit)

For all install types and components, 105MB is the minimum temporary storage space requirement for installation on IBM AIX 7.1 (64-bit) IBM PowerPC systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for IBM AIX 7.1 (64-bit)

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

To change the value of a kernel tuning parameter (maxuproc only), enter `chdev -l sys0 -a maxuproc=" , , , "` command, or enter `smit chgsys` and select "maximum number of processes allowed per user".

Novell SUSE Linux Enterprise Server 10 (32-bit) on x86

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Novell SUSE Linux Enterprise Server 10 (32-bit) systems on x86.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Novell SUSE Linux Enterprise Server 10 (32-bit)

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

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

Hardware Requirements for Novell SUSE Linux Enterprise Server 10 (32-bit)

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

Software Requirements for Novell SUSE Linux Enterprise Server 10 (32-bit)

Component	Requirement
OS Version (Patches)	Novell SUSE Linux Enterprise Server 10 x86 (32-bit)
C/C++ and COBOL compilers	C/C++: gcc/g++ 4.1.0 (SUSE Linux) COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler

Component	Requirement
Java 2 JRE for the Java run-time environment	Tested with Java 2 Jdk 1.6.0_13
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with Jdk 1.6.0_13
Database for CORBA C++ applications	Tested with Oracle Database 11gR2 and Oracle Database 12c
IBM WebSphere MQ for Oracle Tuxedo MQ Adapter	Tested with WebSphere MQ 7.0.1.1

Additional Notes

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

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

- Source code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Novell SUSE Linux Enterprise Server 10 (32-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Novell SUSE Linux Enterprise Server 10 (32-bit)

For all install types and components, 192, 824 KB is the minimum disk space requirement for installation on Novell SUSE Linux Enterprise Server 10 (32-bit) x86 systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Novell SUSE Linux Enterprise Server 10 (32-bit)

See `sysctl` manpage.

Novell SUSE Linux Enterprise Server 11 (32-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Novell SUSE Linux Enterprise Server 11 (32-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Novell SUSE Linux Enterprise Server 11 (32-bit)

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

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

Hardware Requirements for Novell SUSE Linux Enterprise Server 11 (32-bit)

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

Software Requirements for Novell SUSE Linux Enterprise Server 11 (32-bit)

Component	Requirement
OS Version (Patches)	Novell SUSE Linux Enterprise Server 11 x86_64 (64-bit)
C/C++ compilers	C/C++: gcc version 4.3.2 [gcc-4_3-branch revision 141291] (SUSE Linux)

Component	Requirement
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.7.0_51
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.7.0_51
Database for CORBA C++ applications	Tested with Oracle Client 12.1.0.2

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Novell SUSE Linux Enterprise Server 11 (32-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Novell SUSE Linux Enterprise Server 11 (32-bit)

For all install types and components, 558 MB is the minimum disk space requirement for installation on Novell SUSE Linux Enterprise Server 11 (32-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Novell SUSE Linux Enterprise Server 11 (32-bit)

Mount CD-ROM automatically.

Tunable Parameters for Novell SUSE Linux Enterprise Server 11 (32-bit)

See `sysctl` manpage.

Novell SUSE Linux Enterprise Server 12 (64-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Novell SUSE Linux Enterprise Server 12 (64-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Novell SUSE Linux Enterprise Server 12 (64-bit)

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

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

Hardware Requirements for Novell SUSE Linux Enterprise Server 12 (64-bit)

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

Software Requirements for Novell SUSE Linux Enterprise Server 12 (64-bit)

Component	Requirement
OS Version (Patches)	Novell SUSE Linux Enterprise Server 12 x86_64 (64-bit)
C/C++ compilers	C/C++: gcc version 4.8.3 20140627 [gcc-4_8-branch revision 212064] (SUSE Linux)

Component	Requirement
Java 2 JRE for the Java run-time environment	Tested with JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database for CORBA C++ applications	Tested with Oracle Client 12c

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Novell SUSE Linux Enterprise Server 12 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Novell SUSE Linux Enterprise Server 12 (64-bit)

For all install types and components, 198,319 KB is the minimum disk space requirement for installation on Novell SUSE Linux Enterprise Server 12 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Novell SUSE Linux Enterprise Server 12 (64-bit)

Mount CD-ROM automatically.

Tunable Parameters for Novell SUSE Linux Enterprise Server 12 (64-bit)

See `sysctl` manpage.

Oracle Linux 5.6 (64-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Linux 5.6 (64-bit) on x86-64 systems.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 5.6 (64-bit)

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

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

Hardware Requirements for Oracle Linux 5.6 (64-bit)

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

Software Requirements for Oracle Linux 5.6 (64-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 5.6 x86 (64-bit), kernel 2.6.18-238.el5xen
C/C++ and COBOL compilers	C/C++: gcc/g++ 4.1.2 COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler.

Component	Requirement
Java 2 JRE for the Java run-time environment	Tested with Jrockit1.6.0_31, JDK1.6.0_91, JDK1.7.0_75, JDK1.8.0_31, and JRE 1.7.0_04
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with Jrockit1.6.0_31, JDK1.6.0_91, JDK1.7.0_75, JDK1.8.0_31, and JRE 1.7.0_04
Database for CORBA C++ applications	Tested with Oracle Database 11gR2 and Oracle Database 12c

Additional Notes

- The Direct Cross Node Communication Leveraging RDMA, Self-tuning Lock Mechanism, and SDP and Oracle Tuxedo SDP Support features are only supported on Exalogic. The TLOG Information To Oracle Database feature is supported on Exalogic and normal platforms.

For more information, see the [Oracle Tuxedo 12c Release 2 \(12.1.3\) Release Notes](#).

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For COBOL-IT, you must create the softlink cob to the `$(COBOL-IT_HOME)/bin/cobmf` utility.
- 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.

Network Requirements for Oracle Linux 5.6 (64-bit)

- TCP/IP, using the Sockets network interface

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

For all Install Types and components, 193,464 KB is the minimum disk space requirement for installation on Oracle Linux 5.6 (64-bit) systems. This requirement assumes installation of the default components for the selected install type.

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

Mount CD-ROM automatically.

Tunable Parameters for Oracle Linux 5.6 (64-bit)

See `sysctl` manpage.

Oracle Linux 5.8 (32-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Linux 5.8 (32-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 5.8 (32-bit)

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

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

Hardware Requirements for Oracle Linux 5.8 (32-bit)

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

Software Requirements for Oracle Linux 5.8 (32-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 5.8 x86_64 (64-bit)
C/C++ and COBOL compilers	C/C++: cc (GCC) 4.1.2 20080704 (Red Hat 4.1.2-52) COBOL: Micro Focus 5.1, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database for CORBA C++ applications	Tested with Oracle Client 11gR1

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Oracle Linux 5.8 (32-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Oracle Linux 5.8 (32-bit)

For all install types and components, 452 MB is the minimum disk space requirement for installation on Oracle Linux 5.8 (32-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Oracle Linux 5.8 (32-bit)

Mount CD-ROM automatically.

Tunable Parameters for Oracle Linux 5.8 (32-bit)

See `sysctl` manpage.

Oracle Linux 6.0 (64-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Linux 6.0 (64-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 6.0 (64-bit)

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

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

Hardware Requirements for Oracle Linux 6.0 (64-bit)

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

Software Requirements for Oracle Linux 6.0 (64-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 6.0 x86 64-bit
C/C++ and COBOL compilers	C/C++: cc (GCC) 4.4.4 20100726 (Red Hat 4.4.4-13) COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_13
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_13
Database for CORBA C++ applications	Tested with Oracle Client 11gR2 and Oracle Database 12c

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Oracle Linux 6.0 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Oracle Linux 6.0 (64-bit)

For all install types and components, 371,319 KB is the minimum disk space requirement for installation on Oracle Linux 6.0 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Oracle Linux 6.0 (64-bit)

See `sysctl` manpage.

Oracle Linux 6.5 (32-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Linux 6.5 (32-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 6.5 (32-bit)

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

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

Hardware Requirements for Oracle Linux 6.5 (32-bit)

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

Software Requirements for Oracle Linux 6.5 (32-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 6.5 x86_64 (64-bit)
C/C++ and COBOL compilers	C/C++: cc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-4) COBOL: Micro Focus 5.1, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_13
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_13
Database for CORBA C++ applications	Tested with Oracle Client 11gR1

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Oracle Linux 6.5 (32-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Oracle Linux 6.5 (32-bit)

For all install types and components, 452 MB is the minimum disk space requirement for installation on Oracle Linux 6.5 (32-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Oracle Linux 6.5 (32-bit)

Mount CD-ROM automatically.

Tunable Parameters for Oracle Linux 6.5 (32-bit)

See `sysctl` manpage.

Oracle Linux 7.0 (32-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Linux 7.0 (32-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 7.0 (32-bit)

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

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

Hardware Requirements for Oracle Linux 7.0 (32-bit)

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

Software Requirements for Oracle Linux 7.0 (32-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 7.0 x86_64 (64-bit)
C/C++ compilers	C/C++: gcc version 4.8.2 20140120 (Red Hat 4.8.2-16) (GCC)
Java 2 JRE for the Java run-time environment	Tested with JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database for CORBA C++ applications	Tested with Oracle Client 12c

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- If you want to run Oracle Tuxedo with Oracle Database 12c 32-bit on RHEL/Oracle Linux 7 64-bit, see [32-Bit Oracle Database Client Installation Fails on Linux x86](#) in Oracle Database Release Notes. The required glibc package must be installed.
- 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.

Network Requirements for Oracle Linux 7.0 (32-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Oracle Linux 7.0 (32-bit)

For all install types and components, 558MB is the minimum disk space requirement for installation on Oracle Linux 7.0 (32-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Temporary Storage Space Requirements for Oracle Linux 7.0 (32-bit)

For all install types and components, 98MB is the minimum temporary storage space requirement for installation on Oracle Linux 7.0 (32-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Oracle Linux 7.0 (32-bit)

Mount CD-ROM automatically.

Tunable Parameters for Oracle Linux 7.0 (32-bit)

See `sysctl` manpage.

Oracle Linux 7.0 (64-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Linux 7.0 (64-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 7.0 (64-bit)

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

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

Hardware Requirements for Oracle Linux 7.0 (64-bit)

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

Software Requirements for Oracle Linux 7.0 (64-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 7.0 x86_64 (64-bit)
C/C++ and COBOL compilers	C/C++: cc (GCC) 4.8.2 20140120 (Red Hat 4.8.2-16) COBOL: Micro Focus 5.1, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.8.0_40
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.8.0_40
Database for CORBA C++ applications	Tested with Oracle Client 11gR2

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For Jolt application, users need to install Oracle Tuxedo 12c Release 2 (12.1.3) Rolling Patch 07 or later.
- 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.

Network Requirements for Oracle Linux 7.0 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Oracle Linux 7.0 (64-bit)

For all install types and components, 451 MB is the minimum disk space requirement for installation on Oracle Linux 7.0 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Oracle Linux 7.0 (64-bit)

Mount CD-ROM automatically.

Tunable Parameters for Oracle Linux 7.0 (64-bit)

See `sysctl` manpage.

Oracle Linux 5.6 (64-bit) on Exalogic 2.0

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Linux 5.6 (64-bit) on Exalogic 2.0 systems.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 5.6 (64-bit)

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

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

Hardware Requirements for Oracle Linux 5.6 (64-bit)

- x86-64
- At least 4MB RAM for each Oracle Tuxedo system server

Software Requirements for Oracle Linux 5.6 (64-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 5.6 (64-bit), kernel 2.6.32-200.21.1.el5uek
C/C++ and COBOL compilers	C/C++: gcc/g++ 4.1.2 20080704 COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with JRockit 1.6 (1.6.0_20-20101014-1350-linux-x86_64) Tested with JRE 1.7 (1.7.0_04-b20)
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JRockit 1.6 (1.6.0_20-20101014-1350-linux-x86_64) Tested with JDK 1.7 (1.7.0_04-b20)
Database for CORBA C++ applications	Tested with Oracle Database 11.2.0.1.0 and Oracle Database 12c

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For COBOL-IT, you must create the softlink cob to the `$COBOL-IT_HOME/bin/cobmf` utility.
- 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.

Network Requirements for Oracle Linux 5.6 (64-bit)

- TCP/IP, using the Sockets network interface
- InfiniBand network, using 40Gb

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

For all install types and components, 198,319 KB is the minimum disk space requirement for installation on Oracle Linux 5.6 (64-bit) systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Oracle Linux 5.6 (64-bit)

See `sysctl` manpage.

Oracle Linux 5.8 (64-bit) on Exalogic 2.0

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Linux 5.8 (64-bit) on Exalogic 2.0 systems.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 5.8 (64-bit)

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

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

Hardware Requirements for Oracle Linux 5.8 (64-bit)

- x86-64
- At least 4MB RAM for each Oracle Tuxedo system server

Software Requirements for Oracle Linux 5.8 (64-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 5.8 (64-bit), kernel 2.6.32-200.21.1.el5uek
C/C++ and COBOL compilers	C/C++: gcc/g++ 4.1.2 20080704 COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with JRockit 1.6 (1.6.0_20-20101014-1350-linux-x86_64) Tested with JRE 1.7 (1.7.0_04-b20)
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JRockit 1.6 (1.6.0_20-20101014-1350-linux-x86_64) Tested with JDK 1.7 (1.7.0_04-b20)
Database for CORBA C++ applications	Tested with Oracle Database 11.2.0.1.0 and Oracle Database 12c

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For COBOL-IT, you must create the softlink cob to the `$COBOL-IT_HOME/bin/cobmf` utility .
- 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.

Network Requirements for Oracle Linux 5.8 (64-bit)

- TCP/IP, using the Sockets network interface
- InfiniBand network, using 40Gb

Disk Space Requirements for Oracle Linux 5.8 (64-bit)

For all install types and components, 198,319 KB is the minimum disk space requirement for installation on Oracle Linux 5.8 (64-bit) systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Oracle Linux 5.8 (64-bit)

See `sysctl` manpage.

Oracle Linux 6.0 (64-bit) on Exalogic 2.0

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Linux 6.0 (64-bit) on Exalogic 2.0 systems.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Linux 6.0 (64-bit)

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

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

Hardware Requirements for Oracle Linux 6.0 (64-bit)

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

Software Requirements for Oracle Linux 6.0 (64-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 6.0 Exalogenic 2.0
C/C++ and COBOL compilers	C/C++: cc (GCC) 4.4.4 20100726 (Red Hat 4.4.4-13) COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_13
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_13
Database for CORBA C++ applications	Tested with Oracle Client 11gR2 and Oracle Database 12c

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Oracle Linux 6.0 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Oracle Linux 6.0 (64-bit)

For all install types and components, 371,319 KB is the minimum disk space requirement for installation on Oracle Linux 6.0 (64-bit) on Exalogic 2.0 systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Oracle Linux 6.0 (64-bit)

See `sysctl` manpage.

Oracle Solaris 10 (32-bit) on SPARC

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Solaris 10 (32-bit) on SPARC.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Solaris 10 (32-bit) on SPARC

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

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

Hardware Requirements for Oracle Solaris 10 (32-bit) on SPARC

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

Software Requirements for Oracle Solaris 10 (32-bit) on SPARC

Component	Requirement
OS Version (Patches)	Solaris 10 (64-bit) on SPARC
C/C++ and COBOL compilers	C/C++: Sun Studio 12 COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_32 and JRE 1.7.0_04
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_32 and JRE 1.7.0_04
Database applications	Tested with Oracle Client 11gR2_sol_sparc
SSL certificate authorities	Verisign
Lightweight Directory Access Protocol (LDAP) directory servers	iPlanet Directory Server; needed to retrieve X.509v3 digital certificates for SSL
IBM WebSphere MQ for Oracle Tuxedo MQ Adapter	Tested with WebSphere MQ 7.0.1.0

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:
 - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
LLE and SSL support connections between machines and domains.
 - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.

- Oracle Tuxedo CORBA C++ client supports SSL 3.0. SSL connectivity between these clients, and the Oracle Tuxedo ISL/ISH has been certified.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Oracle Solaris 10 (32-bit) on SPARC

- TCP/IP provided by the operating system

Disk Space Requirements for Oracle Solaris 10 (32-bit) on SPARC

For all install types and components, 595 MB is the minimum disk space requirement for installation on Oracle Solaris 10 (32-bit) on SPARC systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Oracle Solaris 10 (32-bit) on SPARC

See `sysctl` manpage.

Oracle Solaris 10 (64-bit) on SPARC

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Solaris 10 (64-bit) on SPARC systems.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Solaris 10 (64-bit) on SPARC

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

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

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

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

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

Component	Requirement
OS Version (Patches)	SunOS 5.10 (64-bit)
C/C++ and COBOL compilers	C/C++: Sun Studio 12 COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_32 and JRE 1.7.0_04
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_32 and JRE 1.7.0_04
Non-Oracle CORBA Java clients	Tested with Java 2 SDK 1.6.0_10 Interface Definition Language (IDL) ORB (run time)
Database for CORBA C++ applications	Tested with Oracle Database 10gR2 and Oracle Database 12c
IBM WebSphere MQ for Oracle Tuxedo MQ Adapter	Tested with WebSphere MQ 7.0.0.1

Additional Notes

- When Oracle is used, Programmer/2000 Pro*C/C++ version 10.1.0.3.0 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.
 - Note:** To use Micro Focus Server Express 5.1, C/C++ compilers should use Sun Studio 12 or higher.
- 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.

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

- TCP/IP using the Sockets network interface.

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

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

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

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

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

Oracle Solaris 11 (32-bit) on SPARC

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Solaris 11 (32-bit) on SPARC.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Solaris 11 (32-bit) on SPARC

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

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

Hardware Requirements for Oracle Solaris 11 (32-bit) on SPARC

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

Software Requirements for Oracle Solaris 11 (32-bit) on SPARC

Component	Requirement
OS Version (Patches)	Solaris 11 (64-bit) on SPARC
C/C++ and COBOL compilers	C/C++: Sun Studio 12 COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_32 and JRE 1.7.0_04
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_32 and JRE 1.7.0_04
Database applications	Tested with Oracle Client 11gR2_sol_sparc

Component	Requirement
SSL certificate authorities	Verisign
Lightweight Directory Access Protocol (LDAP) directory servers	iPlanet Directory Server; needed to retrieve X.509v3 digital certificates for SSL
IBM WebSphere MQ for Oracle Tuxedo MQ Adapter	Tested with WebSphere MQ 7.0.1.0

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:
 - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
LLE and SSL support connections between machines and domains.
 - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
 - Oracle Tuxedo CORBA C++ client supports SSL 3.0. SSL connectivity between these clients, and the Oracle Tuxedo ISL/ISH has been certified.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

 - Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Oracle Solaris 11 (32-bit) on SPARC

- TCP/IP provided by the operating system

Disk Space Requirements for Oracle Solaris 11 (32-bit) on SPARC

For all install types and components, 595 MB is the minimum disk space requirement for installation on Oracle Solaris 11 (32-bit) on SPARC systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Oracle Solaris 11 (32-bit) on SPARC

See `sysctl` manpage.

Oracle Solaris 11 (64-bit) on SPARC

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Oracle Solaris 11 (64-bit) systems on SPARC.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Oracle Solaris 11 (64-bit)

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

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

Hardware Requirements for Oracle Solaris 11 (64-bit) on SPARC

- SPARC
- 1 GB of RAM minimum
- At least 6MB RAM for each Oracle Tuxedo system server

Software Requirements for Oracle Solaris 11 (64-bit) on SPARC

Component	Requirement
OS Version (Patches)	SunOS 5.11
C/C++ and COBOL compilers	C/C++: Sun Studio 12 COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_32 and JRE 1.7.0_04
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_32 and JRE 1.7.0_04
Database for CORBA C++ applications	Tested with Oracle Database 11gR2 and Oracle Database 12c

Additional Notes

- When Oracle is used, Programmer/2000 Pro*C/C++ version 10.1.0.3.0 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.
 - Note:** To use Micro Focus Server Express 5.1, C/C++ compilers should use Sun Studio 12 or higher.
- 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.

Network Requirements for Oracle Solaris 11 (64-bit) on SPARC

- TCP/IP using the Sockets network interface.

Disk Space Requirements for Oracle Solaris 11 (64-bit) on SPARC

For all install types and components, 224,355 KB 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.

Tunable Parameters for Oracle Solaris 11 (64-bit) on SPARC

- Use the `mdb` tool to change the kernel parameters.
- The tunable parameters currently set on your system are in the kernel configuration file located in the `/etc` directory (`/etc/system`).

Red Hat Enterprise Linux 5.6 (64-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Red Hat Enterprise Linux 5.6 (64-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat Enterprise Linux 5.6 (64-bit)

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

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

Hardware Requirements for Red Hat Enterprise Linux 5.6 (64-bit)

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

Software Requirements for Red Hat Enterprise Linux 5.6 (64-bit)

Component	Requirement
OS Version (Patches)	Red Hat Enterprise Linux 5.6 x86 64-bit
C/C++ and COBOL compilers	C/C++: gcc 4.1.2 COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Jrockit 2 JRE 1.6.0_31 and JRE1.7.0_04
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with Jrockit 1.6.0_31 and JDK1.7.0_04
Database for CORBA C++ applications	Tested with Oracle Database 11gR2 and Oracle Database 12c
IBM WebSphere MQ for Oracle Tuxedo MQ Adapter	Tested with WebSphere MQ 7.0.0.1

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Red Hat Enterprise Linux 5.6 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Red Hat Enterprise Linux 5.6 (64-bit)

For all install types and components, 200,840 KB is the minimum disk space requirement for installation on Red Hat Enterprise Linux 5.6 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Red Hat Enterprise Linux 5.6 (64-bit)

See `sysctl` manpage.

Red Hat Enterprise Linux 6.0 (64-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Red Hat Enterprise Linux 6.0 (64-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat Enterprise Linux 6.0 (64-bit)

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

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

Hardware Requirements for Red Hat Enterprise Linux 6.0 (64-bit)

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

Software Requirements for Red Hat Enterprise Linux 6.0 (64-bit)

Component	Requirement
OS Version (Patches)	Red Hat Enterprise Linux 6.0 x86 64-bit
C/C++ and COBOL compilers	C/C++: cc (GCC) 4.4.4 20100726 (Red Hat 4.4.4-13) COBOL: Micro Focus Server Express 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_13
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_13
Database for CORBA C++ applications	Tested with Oracle Client 11gR2 and Oracle Database 12c

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Red Hat Enterprise Linux 6.0 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Red Hat Enterprise Linux 6.0 (64-bit)

For all install types and components, 371,319 KB is the minimum disk space requirement for installation on Red Hat Enterprise Linux 6.0 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Red Hat Enterprise Linux 6.0 (64-bit)

See `sysctl` manpage.

Red Hat Enterprise Linux 6.3 (64-bit) on s390x

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Red Hat Enterprise Linux 6.3 (64-bit) on s390x.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat Enterprise Linux 6.3 (64-bit)

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

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

Hardware Requirements for Red Hat Enterprise Linux 6.3 (64-bit)

- s390x
- 1 GB of RAM minimum
- At least 4MB RAM for each Oracle Tuxedo system server

Software Requirements for Red Hat Enterprise Linux 6.3 (64-bit)

Component	Requirement
OS Version (Patches)	Red Hat Enterprise Linux 6.3 (s390x)
C/C++ compilers	C/C++: gcc/g++ 4.4.6
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0
Database for CORBA C++ applications	Tested with Oracle Client 11gR2

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C or C++ compiler.
- CORBA, SCA, and Xalan are not supported.
- 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.

Network Requirements for Red Hat Enterprise Linux 6.3 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Red Hat Enterprise Linux 6.3 (64-bit)

For all install types and components, 315,000 KB is the minimum disk space requirement for installation on Red Hat Enterprise Linux 6.3 (64-bit) on s390x systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Red Hat Enterprise Linux 6.3 (64-bit)

See `sysctl` manpage.

Red Hat Enterprise Linux 6.5 (32-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Red Hat Enterprise Linux 6.5 (32-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat Enterprise Linux 6.5 (32-bit)

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

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

Hardware Requirements for Red Hat Enterprise Linux 6.5 (32-bit)

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

Software Requirements for Red Hat Enterprise Linux 6.5 (32-bit)

Component	Requirement
OS Version (Patches)	Red Hat Enterprise Linux 6.5 x86_64 (64-bit)
C/C++ and COBOL compilers	C/C++: cc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-4) COBOL: Micro Focus 5.1, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_13
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_13
Database for CORBA C++ applications	Tested with Oracle Client 11gR1

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Red Hat Enterprise Linux 6.5 (32-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Red Hat Enterprise Linux 6.5 (32-bit)

For all install types and components, 452 MB is the minimum disk space requirement for installation on Red Hat Enterprise Linux 6.5 (32-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Red Hat Enterprise Linux 6.5 (32-bit)

Mount CD-ROM automatically.

Tunable Parameters for Red Hat Enterprise Linux 6.5 (32-bit)

See `sysctl` manpage.

Red Hat Enterprise Linux 7.0 (32-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Red Hat Enterprise Linux 7.0 (32-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat Enterprise Linux 7.0 (32-bit)

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

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

Hardware Requirements for Red Hat Enterprise Linux 7.0 (32-bit)

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

Software Requirements for Red Hat Enterprise Linux 7.0 (32-bit)

Component	Requirement
OS Version (Patches)	Red Hat Enterprise Linux 7.0 x86_64 (64-bit)
C/C++ compilers	C/C++: gcc version 4.8.2 20140120 (Red Hat 4.8.2-16) (GCC)
Java 2 JRE for the Java run-time environment	Tested with JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database for CORBA C++ applications	Tested with Oracle Client 12c

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- If you want to run Oracle Tuxedo with Oracle Database 12c 32-bit on RHEL/Oracle Linux 7 64-bit, see [32-Bit Oracle Database Client Installation Fails on Linux x86](#) in Oracle Database Release Notes. The required glibc package must be installed.
- 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.

Network Requirements for Red Hat Enterprise Linux 7.0 (32-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Red Hat Enterprise Linux 7.0 (32-bit)

For all install types and components, 558MB is the minimum disk space requirement for installation on Red Hat Enterprise Linux 7.0 (32-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Temporary Storage Space Requirements for Red Hat Enterprise Linux 7.0 (32-bit)

For all install types and components, 98MB is the minimum temporary storage space requirement for installation on Red Hat Enterprise Linux 7.0 (32-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Red Hat Enterprise Linux 7.0 (32-bit)

Mount CD-ROM automatically.

Tunable Parameters for Red Hat Enterprise Linux 7.0 (32-bit)

See `sysctl` manpage.

Red Hat Enterprise Linux 7.0 (64-bit) on x86-64

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Red Hat Enterprise Linux 7.0 (64-bit) on x86-64.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Red Hat Enterprise Linux 7.0 (64-bit)

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

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

Hardware Requirements for Red Hat Enterprise Linux 7.0 (64-bit)

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

Software Requirements for Red Hat Enterprise Linux 7.0 (64-bit)

Component	Requirement
OS Version (Patches)	Red Hat Enterprise Linux 7.0 x86_64 (64-bit)
C/C++ and COBOL compilers	C/C++: cc (GCC) 4.8.2 20140120 (Red Hat 4.8.2-16) COBOL: Micro Focus 5.1, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.8.0_40
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.8.0_40
Database for CORBA C++ applications	Tested with Oracle Client 11gR2

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For Jolt application, Oracle Tuxedo Rolling Patch 07 or later should be installed
- 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.

Network Requirements for Red Hat Enterprise Linux 7.0 (64-bit)

- TCP/IP provided by Red Hat Enterprise Linux 7.0 (64-bit) on x86-64

Disk Space Requirements for Red Hat Enterprise Linux 7.0 (64-bit)

For all install types and components, 451 MB is the minimum disk space requirement for installation on Red Hat Enterprise Linux 7.0 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Red Hat Enterprise Linux 7.0 (64-bit)

Mount CD-ROM automatically.

Tunable Parameters for Red Hat Enterprise Linux 7.0 (64-bit)

See `sysctl` manpage.

NeoKylin Linux 6.5 (64-bit) on x86-64

The following sections list requirements for Oracle Tuxedo 12c Release 2 (12.1.3) for NeoKylin Linux 6.5 (64-bit) on x86-64 systems.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for NeoKylin Linux 6.5

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

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

Hardware Requirements for NeoKylin Linux 6.5 (64-bit)

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

Software Requirements for NeoKylin Linux 6.5 (64-bit)

Component	Requirement
OS Version (Patches)	NeoKylin Linux 6.5 x86_64 (64-bit)
C/C++ compilers	C/C++: gcc version 4.4.7 20120313 (NeoKylin 4.4.7-4) (GCC)
Java 2 JRE for the Java run-time environment	Tested with JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database for CORBA C++ applications	Tested with Oracle Client 12c

Additional Notes

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

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

 - Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for NeoKylin Linux 6.5 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for NeoKylin Linux 6.5 (64-bit)

For all install types and components, 198,319 KB is the minimum disk space requirement for installation on NeoKylin Linux 6.5 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for NeoKylin Linux 6.5 (64-bit)

Mount CD-ROM automatically.

Tunable Parameters for NeoKylin Linux 6.5 (64-bit)

See `sysctl` manpage.

NeoKylin Linux 7.2 (64-bit) on x86-64

The following sections list requirements for Oracle Tuxedo 12c Release 2 (12.1.3) for NeoKylin Linux 7.2 (64-bit) on x86-64 systems.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for NeoKylin Linux 7.2

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

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

Hardware Requirements for NeoKylin Linux 7.2 (64-bit)

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

Software Requirements for NeoKylin Linux 7.2 (64-bit)

Component	Requirement
OS Version (Patches)	NeoKylin Linux Advanced Server release 7.2 x86_64 (64-bit)
C/C++ compilers	C/C++: cc (GCC) 4.8.5 20150623 (NeoKylin 4.8.5-4)
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.8.0_65
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.8.0_65

Additional Notes

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

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for NeoKylin Linux 7.2 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for NeoKylin Linux 7.2 (64-bit)

For all install types and components, 198,319 KB is the minimum disk space requirement for installation on NeoKylin Linux 7.2 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for NeoKylin Linux 7.2 (64-bit)

Mount CD-ROM automatically.

Tunable Parameters for NeoKylin Linux 7.2 (64-bit)

See `sysctl` manpage.

Microsoft Windows Server 2008 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010

The following sections list requirements for Oracle Tuxedo 12c Release 2 (12.1.3) Microsoft Windows Server 2008 R2 (32-bit) systems on x86-64 with MS Visual Studio 2010.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2008 R2 (32-bit)

- Full client install type
- Jolt client install type
- ATMI client install type
- CORBA client install type
- .NET client install type

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

Note: Microsoft .NET Framework 4.0 must be installed in .NET client install type.

Hardware Requirements for Windows Server 2008 R2 (32-bit)

- x86_64 processor
- 1 GB of RAM minimum

Software Requirements for Windows Server 2008 R2 (32-bit)

Component	Requirement
OS Version (Patches)	Windows server 2008 R2 Enterprise 64-bit
C/C++ compilers	C/C++: MS Visual Studio 2010
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_45
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_45

Additional Notes

- ATMI users need a C or C++ compiler.
- CORBA C++ users need a C++ compiler and linker.
- When using buildnetclient to build 32-bit .net client on Windows Server 2008 R2 64-bit, the following flags is needed: -csflag "/platform:x86".
- 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.

Network Requirements for Windows Server 2008 R2 (32-bit)

- TCP/IP provided by Microsoft Windows Server 2008 R2 (32-bit Winsock).

Disk Space Requirements for Windows Server 2008 R2 (32-bit)

For all install types and components, 394,021 KB is the minimum disk space requirement for installation on Microsoft Windows Server 2008 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010 systems. This requirement assumes installation of the default components for the selected install type.

Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2010

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2010.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2008 R2 (64-bit)

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

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

Hardware Requirements for Windows Server 2008 R2 (64-bit)

- x86-64
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

Software Requirements for Windows Server 2008 R2 (64-bit)

Component	Requirement
OS Version (Patches)	Windows server 2008 R2 Enterprise 64bit
C/C++ and COBOL compilers	C/C++: Visual Studio 2010 COBOL: Micro Focus 5.1, Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler.

Component	Requirement
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database	Tested with Oracle Client 11gR2

Additional Notes

- If MicroFocus 5.1 COBOL compiler and VC10.0 (or later compatible versions) C/C++ compiler are used, the `-l 'link /SUBSYSTEM:CONSOLE /ENTRY:<COBOL_OBJ_FILE>link` option should be used when compiling a COBOL client.

Note: For Windows LIB and PATH environment variables, the MicroFocus COBOL library path must be specified after the Visual Studio 2010 library path (i.e., Visual Studio 2010 libraries take precedence over MicroFocus COBOL libraries during the linking stage).

The recommended practice is to run the MicroFocus COBOL `createnv.bat` file first, and then run the Visual Studio 2010 `vcvarsall.bat` file to set environment variables.

- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.5 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:
 - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
 - LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
 - LLE and SSL support connections between machines and domains.

- The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Windows Server 2008 R2 (64-bit)

- TCP/IP provided by Microsoft Windows Server 2008 (64-bit Winsock).

Disk Space Requirements for Windows Server 2008 R2 (64-bit)

For all install types and components, 478,020 KB is the minimum disk space requirement for installation on Windows Server 2008 R2 (64-bit) x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Microsoft Windows Server 2008 R2 (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](#)".

Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2012

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2012.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2008 R2 (64-bit)

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

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

Hardware Requirements for Windows Server 2008 R2 (64-bit)

- x86-64
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

Software Requirements for Windows Server 2008 R2 (64-bit)

Component	Requirement
OS Version (Patches)	Windows server 2008 R2 Enterprise 64bit
C/C++ and COBOL compilers	C/C++: Visual Studio 2012 COBOL: Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler.
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database	Tested with Oracle Client 11gR2

Additional Notes

- If Micro Focus Visual COBOL v2.1 compiler and VC11.0 (or later compatible versions) C/C++ compiler are used, the `-l 'link /SUBSYSTEM:CONSOLE /ENTRY:<COBOL_OBJ_FILE>link` option should be used when compiling a COBOL client.

Note: For Windows LIB and PATH environment variables, the Micro Focus COBOL library path must be specified after the Visual Studio 2012 library path (i.e., Visual Studio 2012 libraries take precedence over Micro Focus COBOL libraries during the linking stage).

The recommended practice is to run the Micro Focus COBOL `createnv.bat` file first, and then run the Visual Studio 2012 `vcvarsall.bat` file to set environment variables.
- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.5 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:
 - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
 - LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
 - LLE and SSL support connections between machines and domains.
 - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

 - Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Windows Server 2008 R2 (64-bit)

- TCP/IP provided by Microsoft Windows Server 2008 R2 (64-bit Winsock).

Disk Space Requirements for Windows Server 2008 R2 (64-bit)

For all install types and components, 478,020 KB is the minimum disk space requirement for installation on Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2012 systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Microsoft Windows Server 2008 R2 (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](#)".

Microsoft Windows Server 2012 (32-bit) on x86-64 with MS Visual Studio 2013

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Microsoft Windows Server 2012 (32-bit) on x86-64 with MS Visual Studio 2013.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2012 (32-bit)

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

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

Hardware Requirements for Windows Server 2012 (32-bit)

- x86-64
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

Software Requirements for Windows Server 2012 (32-bit)

Component	Requirement
OS Version (Patches)	Windows server 2012 Enterprise 64-bit
C/C++ and COBOL compilers	C/C++: Visual Studio 2013 COBOL: Micro Focus COBOL Version 2.2.02100
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database	Tested with Oracle Client 11gR2 32-bit

Additional Notes

- If Micro Focus Visual COBOL v2.1 compiler and VC11.0 (or later compatible versions) C/C++ compiler are used, the `-l 'link /SUBSYSTEM:CONSOLE /ENTRY:<COBOL_OBJ_FILE>link` option should be used when compiling a COBOL client.

Notes:

- For Windows LIB and PATH environment variables, the MicroFocus COBOL library path must be specified after the Visual Studio 2013 library path (i.e., Visual Studio 2013 libraries take precedence over Micro Focus COBOL libraries during the linking stage).
- The recommended practice is to run the MicroFocus COBOL `createnv.bat` file first, and then run the Visual Studio 2013 `vcvarsall.bat` file to set environment variables.
- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.5 is required to build the Oracle Tuxedo University sample applications.

- ATMI users need a C, C++, or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:
 - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
 - LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
 - LLE and SSL support connections between machines and domains.
 - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

- Source Code:
 - C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Windows Server 2012 (32-bit)

- TCP/IP provided by Microsoft Windows Server 2012 (32-bit Winsock).

Disk Space Requirements for Windows Server 2012 (32-bit)

For all install sets and components, 482 MB is the minimum disk space requirement for installation on Windows Server 2012 (32-bit) x86-64 systems. This requirement assumes installation of the default components for the selected install set.

Tunable Parameters for Microsoft Windows Server 2012 (32-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](#)".

Microsoft Windows Server 2012 (64-bit) on x86-64 with MS Visual Studio 2012

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Microsoft Windows Server 2012 (64-bit) on x86-64 with MS Visual Studio 2012.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Install Types Server 2012 (64-bit)

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

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

Hardware Requirements for Windows Server 2012 (64-bit)

- x86-64
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

Software Requirements for Windows Server 2012 (64-bit)

Component	Requirement
OS Version (Patches)	Windows server 2012 Standard 64bit
C/C++ and COBOL compilers	C/C++: Visual Studio 2012 COBOL: Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler.
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database	Tested with Oracle Client 11gR2

Additional Notes

- If Micro Focus Visual COBOL v2.1 compiler and VC11.0 (or later compatible versions) C/C++ compiler are used, the `-l 'link /SUBSYSTEM:CONSOLE /ENTRY:<COBOL_OBJ_FILE>link` option should be used when compiling a COBOL client.

Note: For Windows LIB and PATH environment variables, the Micro Focus COBOL library path must be specified after the Visual Studio 2012 library path (i.e., Visual Studio 2012 libraries take precedence over Micro Focus COBOL libraries during the linking stage).

The recommended practice is to run the Micro Focus COBOL `createnv.bat` file first, and then run the Visual Studio 2012 `vcvarsall.bat` file to set environment variables.

- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.5 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:

- LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
- LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
- LLE and SSL support connections between machines and domains.
- The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

- Source Code:
 - C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Windows Server 2012 (64-bit)

- TCP/IP provided by Microsoft Windows Server 2012 (64-bit Winsock).

Disk Space Requirements for Windows Server 2012 (64-bit)

For all install types and components, 478,020 KB is the minimum disk space requirement for installation on Microsoft Windows Server 2012 (64-bit) on x86-64 with MS Visual Studio 2012 systems. This requirement assumes installation of the default components for the selected install type.

Tunable Parameters for Microsoft Windows Server 2012 (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](#)".

Microsoft Windows Server 2012 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Microsoft Windows Server 2012 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows Server 2012 R2 (32-bit)

- Full client install type
- ATMI client install type
- CORBA client install type
- Jolt client install type
- .NET client install type

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

Hardware Requirements for Windows Server 2012 R2 (32-bit)

- x86-64
- 1 GB of RAM minimum

Software Requirements for Windows Server 2012 R2 (32-bit)

Component	Requirement
OS Version (Patches)	Windows Server 2012 R2 x86-64
C/C++ compilers	C/C++: MS Visual Studio 2010
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.7.0_21

Component	Requirement
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.7.0_21
Database	Tested with Oracle Client 12.1.0.2.0

Additional Notes

- ATMI users need a C, C++.
- CORBA C++ users need a C++ Compiler and linker.
- When using buildnetclient to build 32-bit .net client on Windows Server 2012 R2 64-bit, the following flags is needed: `-csflag "/platform:x86"`.
- 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.

Network Requirements for Windows Server 2012 R2 (32-bit)

- TCP/IP provided by Microsoft Windows Server 2012 R2 (32-bit Winsock).

Disk Space Requirements for Windows Server 2012 R2 (32-bit)

For all install types and components, 434 MB is the minimum disk space requirement for installation on Microsoft Windows Server 2012 R2 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010 systems. This requirement assumes installation of the default components for the selected install type.

Mounting and Unmounting the CD for Windows Server 2012 R2 (32-bit)

Mount CD-ROM automatically.

Microsoft Windows 7 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Microsoft Windows 7 (64-bit) on x86-64 with MS Visual Studio 2012.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 7 (64-bit)

- Client install type

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

Hardware Requirements for Windows 7 (64-bit)

- x86-64
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

Software Requirements for Windows 7 (64-bit)

Component	Requirement
OS Version (Patches)	Windows 7 64bit
C/C++ and COBOL compilers	C/C++: Visual Studio 2012 COBOL: Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler.
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_26

Component	Requirement
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database	Tested with Oracle Client 11gR2

Additional Notes

- If Micro Focus Visual COBOL v2.1 compiler and VC11.0 (or later compatible versions) C/C++ compiler are used, the `-l 'link /SUBSYSTEM:CONSOLE /ENTRY:<COBOL_OBJ_FILE>link` option should be used when compiling a COBOL client.

Note: For Windows LIB and PATH environment variables, the MicroFocus COBOL library path must be specified after the Visual Studio 2012 library path (i.e., Visual Studio 2012 libraries take precedence over MicroFocus COBOL libraries during the linking stage).

The recommended practice is to run the MicroFocus COBOL `createnv.bat` file first, and then run the Visual Studio 2012 `vcvarsall.bat` file to set environment variables.

- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.5 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:
 - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
 - LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
 - LLE and SSL support connections between machines and domains.
 - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.

- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

- Source Code:

C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Windows 7 (64-bit)

- TCP/IP provided by Microsoft Windows 7 (64-bit Winsock).

Disk Space Requirements for Windows 7 (64-bit)

For all install types and components, 478,020 KB is the minimum disk space requirement for installation on Windows 7 (64-bit) x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Microsoft Windows 8 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012

The following sections list Oracle Tuxedo 12c Release 2 (12.1.3) requirements for Microsoft Windows 8 (64-bit) on x86-64 with MS Visual Studio 2012.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 8 (64-bit)

- Client install type

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

Hardware Requirements for Windows 8 (64-bit)

- x86-64
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

Software Requirements for Windows 8(64-bit)

Component	Requirement
OS Version (Patches)	Windows 8 64bit
C/C++ and COBOL compilers	C/C++: Visual Studio 2012 COBOL: Micro Focus Visual COBOL v2.1 or later, or other compatible COBOL compiler.
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database	Tested with Oracle Client 11gR2

Additional Notes

- If Micro Focus Visual COBOL v2.1 compiler and VC11.0 (or later compatible versions) C/C++ compiler are used, the `-l '-link /SUBSYSTEM:CONSOLE /ENTRY:<COBOL_OBJ_FILE>link` option should be used when compiling a COBOL client.

Note: For Windows LIB and PATH environment variables, the MicroFocus COBOL library path must be specified after the Visual Studio 2012 library path (i.e., Visual Studio 2012 libraries take precedence over MicroFocus COBOL libraries during the linking stage).

The recommended practice is to run the MicroFocus COBOL `createnv.bat` file first, and then run the Visual Studio 2012 `vcvarsall.bat` file to set environment variables.

- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.5 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:
 - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
 - LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
 - LLE and SSL support connections between machines and domains.
 - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

- Source Code:
 - C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Windows 8 (64-bit)

- TCP/IP provided by Microsoft Windows 8 (64-bit Winsock).

Disk Space Requirements for Windows 8 (64-bit)

For all install types and components, 478,020 KB is the minimum disk space requirement for installation on Windows 8 (64-bit) x86-64 systems. This requirement assumes installation of the default components for the selected install type.

Microsoft Windows 8.1 (32-bit) (Client Only) on x86-64 with MS Visual Studio 2010

The following sections list requirements for Oracle Tuxedo 12c Release 2 (12.1.3) Microsoft Windows 8.1 (32-bit) on x86-64 with MS Visual Studio 2010.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 8.1 (32-bit)

- Full client install set
- ATMI client install set
- CORBA client install set
- Jolt client install set
- .NET client install set

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

Hardware Requirements for Windows 8.1 (32-bit)

- x86-64
- 1 GB of RAM minimum

Software Requirements for Windows 8.1 (32-bit)

Component	Requirement
OS Version (Patches)	Windows 8.1 64bit
C/C++ compilers	C/C++: MS Visual Studio 2010
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_26

Component	Requirement
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database	Tested with Oracle Client 12.1.0

Additional Notes

- ATMI users need a C, C++.
- CORBA C++ users need a C++ Compiler and linker.
- When using buildnetclient to build 32-bit .net client on Windows 8.1 64-bit, the following flags is needed: `-csflag "/platform:x86"`.
- 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.

Network Requirements for Windows 8.1 (32-bit)

- TCP/IP provided by Microsoft Windows 8.1 (32-bit Winsock).

Disk Space Requirements for Windows 8.1 (32-bit)

For all install sets and components, 363,655 KB is the minimum disk space requirement for installation on Microsoft Windows 8.1 (Client Only) on x86-64 with MS Visual Studio 2010 systems. This requirement assumes installation of the default components for the selected install set.

Microsoft Windows 8.1 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012

The following sections list requirements for Oracle Tuxedo 12c Release 2 (12.1.3) Microsoft Windows 8.1 (64-bit) on x86-64 with MS Visual Studio 2012.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 8.1 (64-bit)

- Client install type

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

Hardware Requirements for Windows 8.1 (64-bit)

- x86-64
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

Software Requirements for Windows 8.1 (64-bit)

Component	Requirement
OS Version (Patches)	Windows 8.1 64bit
C/C++ compilers	C/C++: MS Visual Studio 2012
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.7.0_21
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.7.0_21
Database	Tested with Oracle Client 12.1.0

Additional Notes

- ATMI users need a C, C++.

- CORBA C++ users need a C++ Compiler and linker.
- When using `buildnetclient` to build 32-bit .net client on Windows 8.1 64-bit, the following flags is needed: `-csflag "/platform:x86"`.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:
 - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
 - LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
 - LLE and SSL support connections between machines and domains.
 - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

- Source Code:
 - C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Windows 8.1 (64-bit)

- TCP/IP provided by Microsoft Windows 8.1 (64-bit Winsock).

Disk Space Requirements for Windows 8.1 (64-bit)

For all install sets and components, 418MB is the minimum disk space requirement for installation on Microsoft Windows 8.1 (64-bit) (Client Only) on x86-64 with MS Visual Studio

2012 systems. This requirement assumes installation of the default components for the selected install set.

Microsoft Windows 10 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012

The following sections list requirements for Oracle Tuxedo 12c Release 2 (12.1.3) Microsoft Windows 10 (64-bit) on x86-64 with MS Visual Studio 2012.

Available Oracle Tuxedo 12c Release 2 (12.1.3) Install Types for Windows 10 (64-bit)

- Client install type

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

Hardware Requirements for Windows 10 (64-bit)

- x86-64
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

Software Requirements for Windows 10 (64-bit)

Component	Requirement
OS Version (Patches)	Windows 10 64bit
C/C++ compilers	C/C++: MS Visual Studio 2012
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_26
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_26
Database	Tested with Oracle Client 11.2.0

Additional Notes

- ATMI users need a C, C++.

- CORBA C++ users need a C++ Compiler and linker.
- For Oracle Tuxedo 112-bit, 128-bit, and 256-bit encryption:
 - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).
 - LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).
 - LLE and SSL support connections between machines and domains.
 - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

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

- Source Code:
 - C/C++ source code that was compiled and linked using the tested version will compile with the newer version of 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.

Network Requirements for Windows 10 (64-bit)

- TCP/IP provided by Microsoft Windows 10 (64-bit Winsock).

Disk Space Requirements for Windows 10 (64-bit)

For all install sets and components, 418MB is the minimum disk space requirement for installation on Microsoft Windows 10 (64-bit) (Client Only) on x86-64 with MS Visual Studio 2012 systems. This requirement assumes installation of the default components for the selected install set.

Starting the Oracle Tuxedo Administration Console

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

- [What is the Oracle Tuxedo Administration Console?](#)
- [Administration Console File Tree](#)
- [Server Requirements](#)
- [Browser Requirements](#)
- [Hardware Requirements](#)
- [Setting Up Your Environment for the Oracle Tuxedo Administration Console](#)
- [Starting the Oracle Tuxedo Administration Console](#)
- [Exiting the Oracle Tuxedo Administration Console](#)

What is the Oracle Tuxedo Administration Console?

The Oracle Tuxedo Administration Console is a graphical user interface that enables administrators to perform most administration and configuration tasks for Oracle Tuxedo

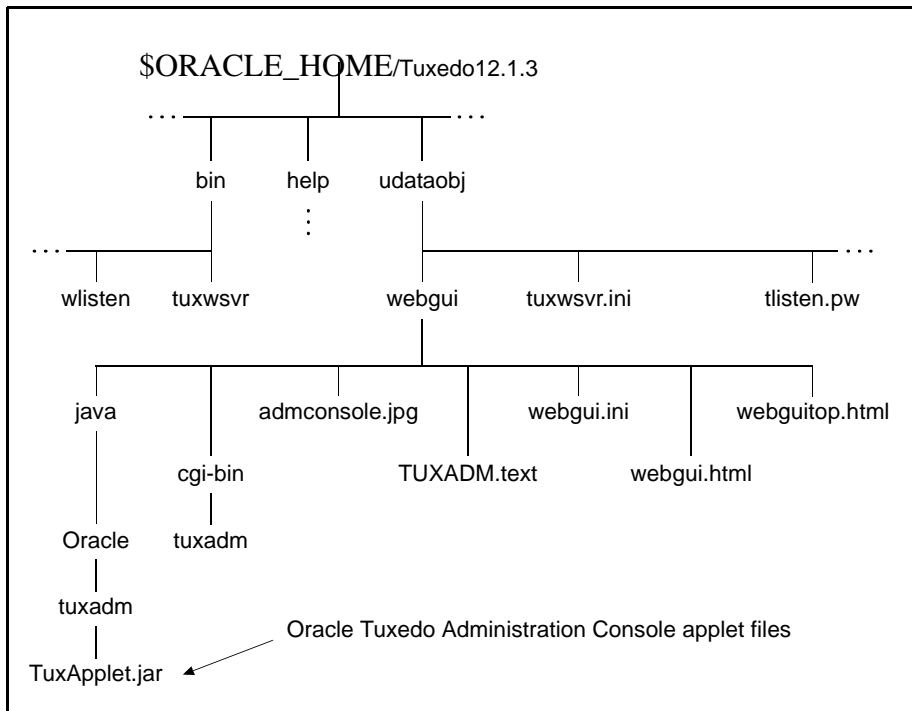
applications. It is implemented as a set of Java applets, which can run on most platforms that support a Java-capable Web browser.

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

Administration Console File Tree

During the Oracle Tuxedo software installation, the installer program places most of the directories and files for the Oracle Tuxedo Administration Console in the `webgui` directory, as shown in [Figure B-1](#).

Figure B-1 Oracle Tuxedo Administration Console File Tree



The installer program installs the following HTML files:

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

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

Server Requirements

The Oracle Tuxedo Administration Console server for Oracle Tuxedo 12c Release 2 (12.1.3) is supported on the platforms on which Oracle Tuxedo 12c Release 2 (12.1.3) system software is supported. For more information, see [Oracle Tuxedo 12c Release 2 \(12.1.3\) Platform Data Sheets](#).

Browser Requirements

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

```
# Web GUI initialization file.
# Created Sep 12, 2002 2:02:07 PM by Oracle software installation program.
#
TUXDIR=C:\oracle\tuxedo12cR1
INIFILE=C:\oracle\tuxedo12cR1\udataobj\webgui\webgui.ini
NADDR=//foo5:4003
DEVICE=/dev/tcp
CODEBASE=/java
DOCBASE=/doc
SNAPDIR=C:\oracle\tuxedo12cR1\udataobj\webgui\java\snapshot
SNAPBASE=/java/snapshot
ENCRYPTBITS=56
```

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

[Table B-1](#) lists the platforms and browsers supported for Oracle Tuxedo 12c Release 2 (12.1.3) when the Oracle Tuxedo Administration Console is configured for 56-bit or 128-bit encryption.

Table B-1 Platforms and Browsers

On this platform . . .	Oracle tested these browsers . . .	Using Java plug-in . . .
Microsoft Windows 2008 Server (32-bit) on x86	Internet Explorer 8	1.4.x
Microsoft Windows 7 on x86	Internet Explorer 8	1.4.x

If the Oracle Tuxedo Administration Console is configured with no encryption (0-bit encryption), Internet Explorer 8 is supported on any Windows platform regardless of the Java plug-in version.

Hardware Requirements

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

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

Setting Up Your Environment for the Oracle Tuxedo Administration Console

To run the Oracle Tuxedo Administration Console, first set up your environment as described in [“Setting Up Your Environment” on page 5-14](#). Ensure that your `TUXDIR`, `WEBJAVADIR`, and `PATH` environment variables are set correctly. Then set up the following two server processes:

- `tuxwsvr(1)`

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

- `wlisten(1)`

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

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

Starting `tuxwsvr`

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

Table B-2 lists the commands for starting `tuxwsvr`.

Table B-2 Starting `tuxsvr`

To Start Tuxsvr on This Platform ...	Enter the Following Command ...
Windows 2003 Server system	<code>tuxwsvr -l //machine:port -i %TUXDIR%\udataobj\tuxwsvr.ini</code>
UNIX system	<code>tuxwsvr -l //machine:port -i \$TUXDIR/udataobj/tuxwsvr.ini</code>

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

```
# tuxwsvr initialization file.
# Created Sep 12, 2008 2:02:07 PM by Oracle software installation program.
#
CGI      /cgi-bin      C:\oracle\tuxedo12cR1\udataobj\webgui/cgi-bin
HTML     /java         C:\oracle\tuxedo12cR1\udataobj\webgui/java
HTML     /doc         C:\oracle\tuxedo12cR1\help
HTML     /          C:\oracle\tuxedo12cR1\udataobj\webgui
```

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

Starting wlisten

The `wlisten` process is a listener process that receives incoming connections from Web GUI applets and starts a Web GUI gateway process (`wgated`). All `wlisten` options are taken from an initialization file that is specified by the `-i` option. If the `-i` option is not given, then `%TUXDIR%\udataobj\webgui\webgui.ini` is used as the default initialization file on a Windows 2003 Server system, and `$TUXDIR/udataobj/webgui/webgui.ini` is used as the default initialization file on a UNIX system.

To start `wlisten`, follow these steps:

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

For example, on a machine called `f005`, the default port assigned to `wlisten` is 4003. To run `wlisten` with port 6060, edit the `NADDR` parameter as follows:

```
NADDR=//f005:6060
```

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

2. Enter the following command:

```
wlisten
```

Starting the Oracle Tuxedo Administration Console

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

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

1. Start the browser.
2. Enter the following URL:

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

For example: `http://f005:4003/webguitop.html`

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

The Oracle Tuxedo Administration Console entry page-displays.

3. To start the Oracle Tuxedo Administration Console, select the Click Here to Run the Oracle Tuxedo Administration Console prompt at the bottom of the screen. The Login window displays.
4. Enter your login name and password in the appropriate fields, and select LOGIN. The password must correspond to one of the entries in the system-encrypted `tlisten.pw` file.

The main window of the Oracle Tuxedo Administration Console displays.

[Table B-3](#) contains instructions for accessing additional information about the Oracle Tuxedo Administration Console main window.

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

If . . .	Then . . .
The main window is displayed and you want to start working with the GUI	See “Administration Console Tutorial” in <i>Oracle Tuxedo Administration Console Online Help</i> .

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

If . . .	Then . . .
The main window is displayed and you want to read a description of it	See “A Tour of the Main Window” in <i>Oracle Tuxedo Administration Console Online Help</i> .
The main window is not displayed and the <code>Connect Failed</code> error message is displayed	<ol style="list-style-type: none">1. On the Administration Console <i>server</i> machine, verify that the <code>wlisten</code> process is running. On a UNIX machine, for example, enter the <code>ps</code> command.2. If <code>wlisten</code> is not running, open the <code>webgui.ini</code> file and, in the line <code>NADDR=//f005:4003</code>, replace the port number (4003) with a valid port number.3. Enter <code>wlisten</code> again:<ul style="list-style-type: none">• <code>\$ wlisten -i %TUXDIR%\udataobj\webgui\webgui.ini</code> (on Windows 2003 Server)• <code>\$ wlisten -i \$TUXDIR/udataobj/webgui/webgui.ini</code> (on UNIX)4. Check that the <code>tuxwsvr</code> process is running at the port specified in the URL.5. Verify the password. It must match one of the entries in the <code>tlisten.pw</code> file.6. Return to step 1.

Limitation(s)

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

Exiting the Oracle Tuxedo Administration Console

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

Oracle Jolt 12*c* Release 2 (12.1.3) Overview and Installation Information

The following sections provide an overview of Oracle Jolt 12*c* Release 2 (12.1.3) and present the prerequisites and preparatory information for installing the Oracle Jolt 12*c* Release 2 (12.1.3) software components:

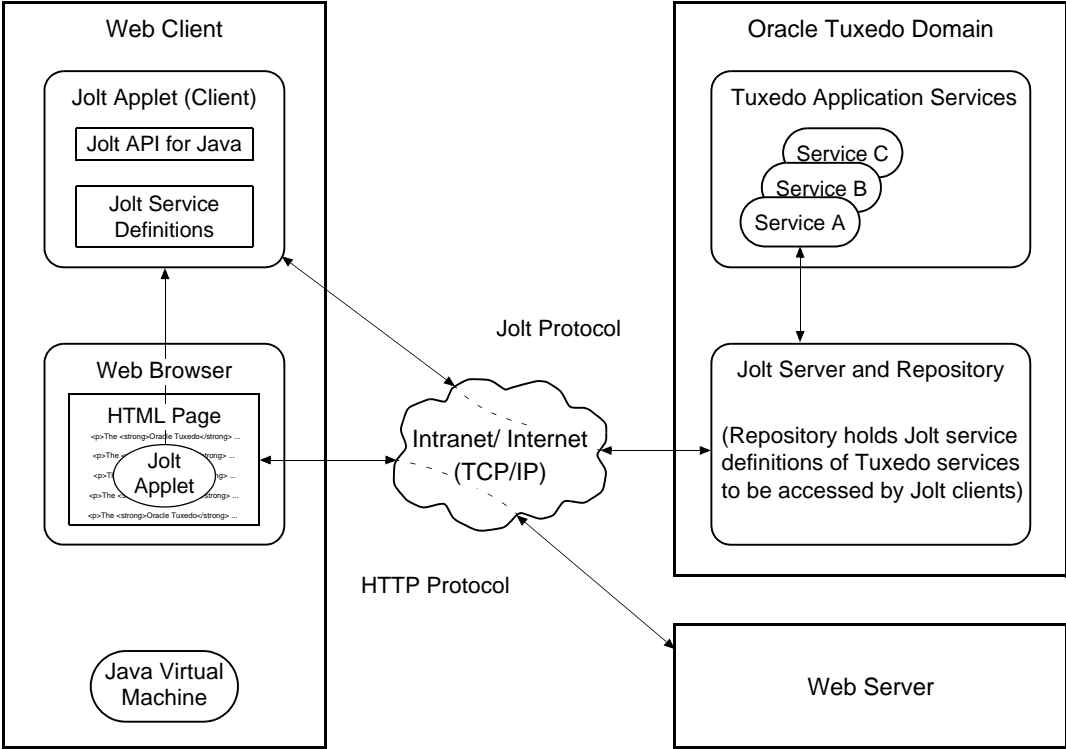
- [About Oracle Jolt](#)
- [Oracle Jolt Components](#)
- [Oracle Jolt Client Personalities](#)
- [Oracle Jolt Capabilities](#)
- [Oracle Jolt Client Support](#)
- [Oracle Jolt Release Interoperability](#)
- [Oracle Jolt Pre-Installation Checklist](#)
- [Oracle Jolt Documentation](#)

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 C-1](#), Oracle Jolt extends the

functionality of existing Tuxedo ATMI applications to include intranet- and Internet-wide availability.

Figure C-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” on page C-8](#).

7

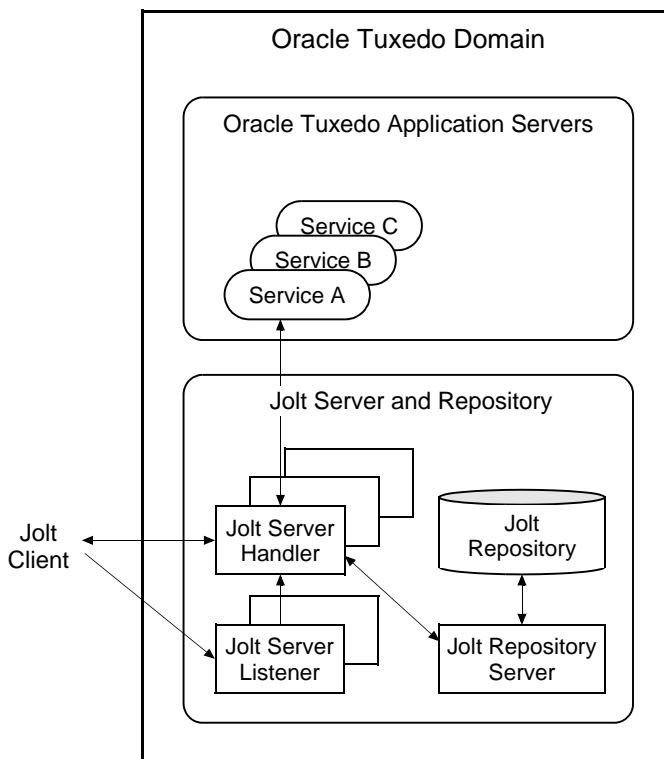
Oracle Jolt Components

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
- Jolt Repository Server
- Jolt Repository
- Jolt Internet Relay
- Jolt Class Library
- JoltBeans

As shown in the following [Figure C-2](#), the Jolt server implementation consists of one or more Jolt Server Handlers, one or more Jolt Server Listeners, and one and only one Jolt Repository Server, all running on the same Oracle Tuxedo server machine.

Figure C-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 12c Release 2 (12.1.3) distribution is installed.

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.

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.

Jolt Repository Server

The Jolt Repository Server (JREPSVR), running on the Tuxedo server, retrieves Jolt service definitions from the Jolt Repository and returns the service definitions to the Jolt Server Handler. The Jolt Repository Server also provides user support for updating or adding Jolt service definitions to the Jolt Repository.

Jolt Repository

The Jolt Repository, located on the Tuxedo server, is a central repository that contains definitions of Tuxedo ATMI services. These Jolt 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 Repository Editor, you can test new and existing Tuxedo services independently of the client applications.

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)

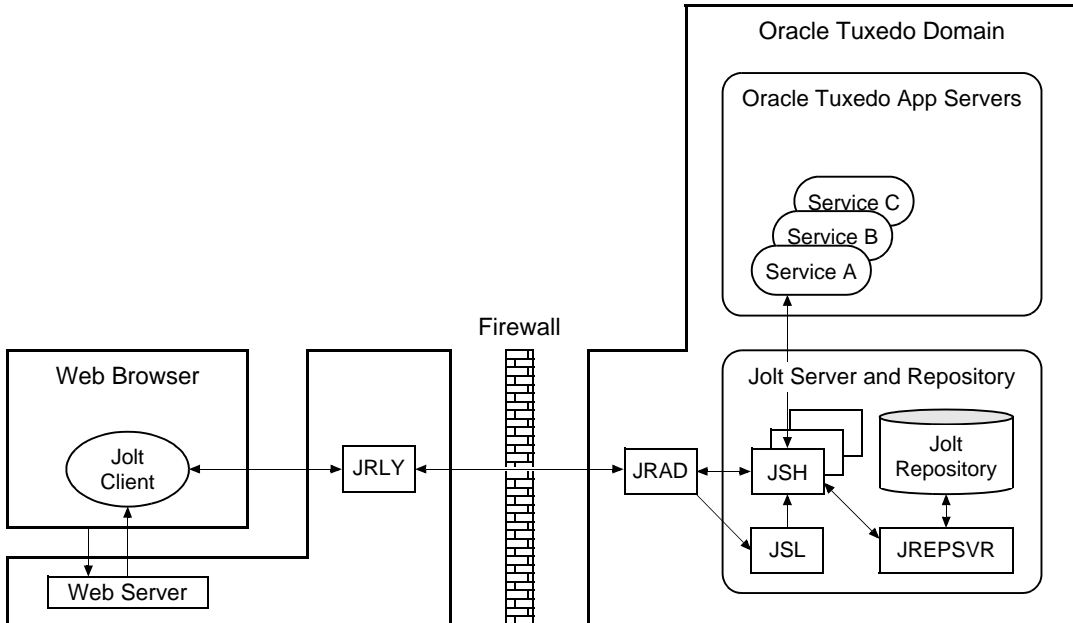
A standalone program that routes Jolt messages from Jolt clients to the JSL or JSH via the Jolt Relay Adapter (JRAD). Jolt Relay is not a Tuxedo system server or client process, and it is not dependent on the Oracle Tuxedo system software version.

- 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.

Figure C-3 illustrates the Jolt Internet Relay connection path.

Figure C-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 the `tux_prod_dir/udataobj/jolt/relay` directory, whereas the JRAD executable resides in the `tux_prod_dir/bin` directory.

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 content of a Jolt JAR file, ensure that the path to the JDK 1.6 (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
```

For example:

```
prompt> jar -tvf jolt.jar
 0 Thu Aug 08 07:19:02 EDT 2002 META-INF/
68 Thu Aug 08 07:19:02 EDT 2002 META-INF/MANIFEST.MF
547 Thu Aug 08 07:19:00 EDT 2002
    bea/jolt/ApplicationException.class
741 Thu Aug 08 07:19:00 EDT 2002 bea/jolt/BData.class
951 Thu Aug 08 07:19:00 EDT 2002 bea/jolt/ByteArrayUtil.class
.
.
.
```

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.

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 *except* `joltjse.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.”

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.

To accommodate the Oracle Jolt 12c Release 2 (12.1.3) server (JSL, JSH, JREPSVR) and Jolt Internet Relay (JRLY, JRAD) components, your environment must provide 2 MB of disk space. For Oracle Jolt 12c Release 2 (12.1.3) system requirements, including supported platforms, see [“Oracle Tuxedo 12c Release 2 \(12.1.3\) Platform Data Sheets” on page A-1](#).

Oracle Jolt Client Support

As stated in “[Oracle Jolt Client Personalities](#)” on page C-8, Oracle Jolt 12c Release 2 (12.1.3) supports the following four client types:

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

[Figure C-1](#) lists the requirements for the Jolt 12c Release 2 (12.1.3) applet and application client types.

Table C-1 Requirements for Oracle Jolt 12c Release 2 (12.1.3) Applet and Application Client Types

This client type . . .	Is supported on . . .
Java applet running in a browser	Microsoft Internet Explorer (IE) 6.0 or higher
Jolt application—standalone application— running in a desktop environment	Java Development Kit (JDK) 1.6.x

[Figure C-2](#) 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 12c Release 2 (12.1.3) are listed in the table.

Table C-2 Requirements for Web-Server based Oracle Jolt 12c Release 2 (12.1.3) 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 1.6.x	JSE Connectivity for Oracle Tuxedo	For a description, see the first note after this table.
Oracle Systems, Inc.	Oracle WebLogic Server 9.0 or higher	Any	WebLogic Connectivity for Oracle Tuxedo (also known as “Oracle Jolt for Oracle WebLogic Server”)	For example installation instructions, see “Installing Oracle Jolt 12c Release 2 (12.1.3) with Oracle WebLogic Server” on page 4-10.

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*.

Oracle Jolt Client Requirements

Oracle Jolt has the following client requirements:

- 574 KB of disk space for client classes
- 1364 KB of disk space for client API documentation
- 190 KB of disk space for client examples
- Java Development Kit (JDK) 1.6.x is certified for Oracle Jolt 12c Release 2 (12.1.3) application development (<http://java.sun.com/j2se>)
- Java-enabled browser (Internet Explorer) or Java Virtual Machine (JVM)

Oracle Jolt Client Class Library

Various implementations of Java tend to show minor differences in characteristics. Oracle Jolt 12c Release 2 (12.1.3) is based on JDK 1.6.

The Oracle Jolt class library is compatible with the browsers and JDK versions shown in [Figure C-3](#).

Table C-3 Oracle Jolt Class Library Compatibility

Vendor	Browser Version	Java Virtual Machine (JVM)	OS Version
Microsoft	Internet Explorer 6.0	JDK 1.6	Windows 7, Windows 2008 Server

Oracle Jolt Release Interoperability

An Oracle Jolt 12c Release 2 (12.1.3) client can interoperate with an Oracle Jolt 9.0, 9.1, 10.0, 10.3, 11.1.1.x, or 12.1.1 server implementation, and the client is able to use the new features available with Oracle Jolt 12c Release 2 (12.1.3).

An Oracle Jolt 9.0, 9.1, 10.0, 10.3, 11.1.1.x or 12.1.1 client can interoperate with an Oracle Jolt 12c Release 2 (12.1.3) server implementation, but only the Oracle Jolt 9.0, 9.1, 10.0, 10.3, 11.1.1.x, or 12.1.1 functionality is available to the Oracle Jolt client, even though new functionality is added to the Oracle Jolt server-side components when they are upgraded to Oracle Jolt 12c Release 2 (12.1.3).

Oracle Jolt Pre-Installation Checklist

On Windows systems, Oracle Jolt 12c Release 2 (12.1.3) automatically installs two Microsoft dynamic link libraries (DLLs), `MSVCRT.DLL` and `MSVCRP71.DLL`.

Oracle Jolt Documentation

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

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

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](#)

Introduction

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

How the Oracle Tuxedo System Manages Files

The Oracle Tuxedo system provides a facility called the Disk Management Interface (DMI), which manages logical files within a single disk device or set of devices. The DMI performs tasks

such as storing binary configuration tables and the transaction log. You can use it to create, initialize, or destroy entries in the Oracle Tuxedo filesystem. To access the DMI, use the `tmadmin(1)` administrative commands described in the *Oracle Tuxedo Command Reference*.

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

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

Oracle Tuxedo files reside on device special files in the designated space and are managed directly by the DMI disk management software. The DMI supports the notion of 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 a `write()` is done. When using raw disk space, however, Oracle Tuxedo has accurate control of the write operation, which is particularly important for entries in the Oracle Tuxedo transaction log. Also, when multiple users are accessing the system, being able to control the write operation is important for assuring database consistency.

Arranging for Raw Disk Space

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

How the Oracle Tuxedo Filesystem Is Organized

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 D-1](#) shows a sample VTOC and UDL diagram for a bankapp (sample application) configuration on a single machine.

Listing D-1 VTOC and UDL Diagram

Output based on setting FSCONFIG=\$TUXCONFIG, and invoking tadmin:

No bulletin board exists. Entering boot mode.

```
> livtoc
```

```
Volume Table of Contents on /usr2/bank/tuxconfig:
```

```
0: VTOC: Device 0 Offset 0 Pages 7
1: UDL: Device 0 Offset 7 Pages 28
2: _RESOURCE_SECT: Device 0 Offset 35 Pages 6
3: _MACHINES_SECT: Device 0 Offset 41 Pages 40
4: _GROUPS_SECT: Device 0 Offset 141 Pages 100
5: _SERVERS_SECT: Device 0 Offset 241 Pages 150
6: _SERVICES_SECT: Device 0 Offset 391 Pages 60
7: _ROUTING_SECT: Device 0 Offset 451 Pages 100
8: _NETWORK_SECT: Device 0 Offset 551 Pages 20
9: _MIBPERMS_SECT: Device 0 Offset 571 Pages 2
10: _NETGROUPS_SECT: Device 0 Offset 573 Pages 2
11: _INTERFACES_SECT: Device 0 Offset 575 Pages 10
```

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

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

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

Table D-1 Size of Oracle Tuxedo System Tables

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

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

Space for Queue Spaces

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

Space for Application Servers

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

IPC Resource Configuration on a UNIX System

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

- [Parameter Sets Controlling IPC Resources](#)
- [Shared Memory](#)
- [Semaphores](#)
- [Message Queues and Messages](#)
- [Other Kernel Tunables](#)

Parameter Sets Controlling IPC Resources

On a UNIX system, the Oracle Tuxedo system uses the IPC resources provided by the UNIX operating system, which are controlled by the tunable parameters listed in [Table E-1](#).

Table E-1 Tunable Parameters

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

The settings for these parameters are application-dependent. Most UNIX systems are shipped with default values that are too low for 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 12c Release 2 \(12.1.3\) Platform Data Sheets” on page A-1](#). If you change a parameter value, you will need to rebuild the kernel and reboot the operating system, using standard administrative tools. Consult your operating system administrator or the system administrator’s guide for your platform for details.

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

Shared Memory

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

The following shared memory parameters may need to be adjusted:

SHMMAX

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

SHMSEG

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

SHMMNI

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

SHMMIN

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

Semaphores

Every process that participates in 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:

SEMMNS

Maximum number of semaphores in the system. The minimum requirement for `SEMMNS` is $\text{MAXACCESSERS} - \text{MAXWSCLIENTS} + 13$

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](#).

SEMMNI

Maximum number of active semaphore sets.

SEMMSL

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

SEMMAP

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

SEMMNU

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

SEMUME

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

Message Queues and Messages

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

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

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

Several message queue parameters are available to define various characteristics of the queue space, as indicated in [Table E-2](#).

Table E-2 Message Que Parameters

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

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

What Is Application Deadlock?

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

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

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

Performance Implications of Blocking Conditions

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

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

Tunable Message Parameters

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

The following message parameters may need to be adjusted:

MSGMNI

Number of unique message queue identifiers. Each process participating in 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:

$$\begin{aligned} \text{MSGMNI} &= \text{MAXACCESSERS} + 7 \\ &+ (\text{number of servers with REPLYQ}) \\ &+ (\text{number of MSSQ sets}) \\ &- (\text{number of servers in MSSQ sets}) \end{aligned}$$

MSGMAX

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

MSGMNB

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

MSGMAP

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

MSGSSZ

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

MSGSEG

Number of message segments in the system.

MSGTQL

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

Other Kernel Tunables

Experience with the Oracle Tuxedo system has shown that some other UNIX system tunables may need to be set to higher values. These parameters are very application dependent and do not apply to all applications. [“Oracle Tuxedo 12c Release 2 \(12.1.3\) Platform Data Sheets” on page A-1](#) includes information on the defaults for each platform and instructions for changing them.

ULIMIT

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

NOFILES

Maximum number of open files per process. 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.

