

# **Oracle® Tuxedo**

Oracle Tuxedo/Oracle Exalogic Users Guide

12c Release 1 (12.1.1)

June 2012

**ORACLE®**

Copyright © 1996, 2012, 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.

# Oracle Tuxedo/Oracle Exalagic Users Guide

Overview . . . . .	-1
About this Guide . . . . .	-2
About Oracle Exalagic . . . . .	-2
About Oracle Tuxedo optimizations for Oracle Exalagic . . . . .	-2
Tuxedo Optimizations on Exalagic . . . . .	-3
Direct Cross Node Communication Leveraging RDMA . . . . .	-3
Self-Tuning Lock Mechanism . . . . .	-4
Oracle Tuxedo SDP Support . . . . .	-4
Shared Memory Interprocess Communication . . . . .	-5
Partial One Phase Read-Only Optimization for RAC . . . . .	-5
Shared Applications Staging . . . . .	-5
Tightly Coupled Transactions Spanning Domains . . . . .	-5
XA Transaction Affinity . . . . .	-6
Common XID . . . . .	-6
Single Group Multiple Branches (SGMB) . . . . .	-6
Failover/Failback across Database Instances . . . . .	-7
Load Balancing across RAC Instances . . . . .	-7
Concurrent Global Transaction Table Lock . . . . .	-7
Oracle Tuxedo Configuration . . . . .	-8
Direct Cross Node Communication Leveraging RDMA . . . . .	-8
UBBCONFIG File . . . . .	-9
Setting Shell Limit for Memory Lock . . . . .	-12
Setting Default Directory Name for File Transfer . . . . .	-13
Self-Tuning Lock Mechanism . . . . .	-13
Oracle Tuxedo SDP Support . . . . .	-13

MP .....	-14
GWTDOMAIN .....	-15
WSL .....	-17
/WS client .....	-18
JSL .....	-18
WTC .....	-19
Shared Memory Interprocess Communication .....	-20
Partial One Phase Read-Only Optimization for RAC .....	-20
Database .....	-20
Choosing APPDIR .....	-21
Tightly Coupled Transactions Spanning Domains .....	-22
XA Transaction Affinity .....	-22
Common XID .....	-22
Single Group Multiple Branches (SGMB) .....	-23
Failover/Failback across Database Instances .....	-23
Load Balancing across RAC Instances .....	-23
Concurrent Global Transaction Table Lock .....	-23
Best Practices to Optimize Performance .....	-23
Direct Cross Node Communication Leveraging RDMA .....	-24
Scenarios recommended .....	-24
Setting EXALOGIC_MSGQ_CACHE_SIZE .....	-24
Calculating Shared Memory Size for Msgq_daemon .....	-24
Using tmloadcf .....	-24
Adjusting Shared memory size: .....	-27
Self-Tuning Lock Mechanism .....	-28
Shared Memory Interprocess Communication .....	-28
Partial One Phase Read-Only Optimization for RAC .....	-28
XA Transaction Affinity .....	-28

Common XID .....	-28
Failover/Failback across Database Instances .....	-28
Load Balancing across RAC Instances .....	-29
Single Group Multiple Branches (SGMB) .....	-29
Oracle Tuxedo SDP Support .....	-29
Running Oracle Tuxedo .....	-29
Start/Stop tux_msgq_monitor .....	-29
Assistant Tools .....	-30
Shell Scripts for Start/Stop Oracle Tuxedo .....	-30
Running Oracle Tuxedo on OVM .....	-31
Upgrade .....	-31
Appendix .....	-32
Terminology .....	-32
Oracle Tuxedo installation .....	-33
Configuration for Exalogic .....	-33
Platform Requirements .....	-33
Start Installation .....	-33



# Oracle Tuxedo/Oracle Exalogic Users Guide

This chapter contains the following topics:

- [Overview](#)
- [Oracle Tuxedo Configuration](#)
- [Best Practices to Optimize Performance](#)
- [Running Oracle Tuxedo](#)
- [Running Oracle Tuxedo on OVM](#)
- [Appendix](#)

## Overview

This section contains the following topics.

- [About this Guide](#)
- [About Oracle Exalogic](#)
- [About Oracle Tuxedo optimizations for Oracle Exalogic](#)
- [Tuxedo Optimizations on Exalogic](#)

## About this Guide

This document introduces all Oracle Tuxedo optimizations for Exalogic. With this document, you can easily install, configure, and run Tuxedo on Exalogic.

## About Oracle Exalogic

Oracle Exalogic is an Engineered Systems that integrates compute, networking and storage hardware with virtualization, operating system and management software. It provides breakthrough performance, reliability, availability, scalability and investment protection for the widest possible range of business application workloads.

## About Oracle Tuxedo optimizations for Oracle Exalogic

From 11.1.1.3.0, Oracle Tuxedo provides many optimizations for Oracle Exalogic platforms; [Table 1](#) lists the Exalogic supported features:

**Table 1 Exalogic Supported Oracle Tuxedo Features**

Feature Name	Oracle Tuxedo Version
Direct Cross Node Communication Leveraging RDMA	Oracle Tuxedo 11gR1 (11.1.1.3.0) or above
Self-Tuning Lock Mechanism	Oracle Tuxedo 11gR1 (11.1.1.3.0) or above
Oracle Tuxedo SDP Support	Oracle Tuxedo 11gR1 (11.1.1.3.0) or above
Shared Memory Interprocess Communication	Oracle Tuxedo 12cR1 (12.1.1) or above
Partial One Phase Read-Only Optimization for RAC	Oracle Tuxedo 12cR1 (12.1.1) or above
Shared Applications Staging	Oracle Tuxedo 12cR1 (12.1.1) or above
Tightly Coupled Transactions Spanning Domains	Oracle Tuxedo 12cR1 (12.1.1) or above
XA Transaction Affinity	Oracle Tuxedo 12cR1 (12.1.1) with Rolling Patch 073 or above
Common XID	Oracle Tuxedo 12cR1 (12.1.1) with Rolling Patch 073 or above

**Table 1 Exalogic Supported Oracle Tuxedo Features**

Single Group Multiple Branches (SGMB)	Oracle Tuxedo 12cR1 (12.1.1) with Rolling Patch 073 or above
Failover/Failback across Database Instances	Oracle Tuxedo 12cR1 (12.1.1) with Rolling Patch 073 or above
Load Balancing across RAC Instances	Oracle Tuxedo 12cR1 (12.1.1) with Rolling Patch 073 or above
Concurrent Global Transaction Table Lock	Oracle Tuxedo 12cR1 (12.1.1) with Rolling Patch 073 or above

**Note:** From Oracle Tuxedo 12cR1 (12.1.1) Rolling Patch 073, all optimizations support both Exalogic Linux 64bit and SPARC 64 bit, except for "Direct Cross Node Communication Leveraging RDMA".

Since Oracle Tuxedo 12cR1 (12.1.1) Rolling Patch 073 on Oracle Linux 32-bit platforms, all optimizations support Exalogic Linux 32bit, except for "Direct Cross Node Communication Leveraging RDMA", and "Oracle Tuxedo SDP Support".

For more information about these features, please see [Tuxedo Optimizations on Exalogic](#).

## Tuxedo Optimizations on Exalogic

### Direct Cross Node Communication Leveraging RDMA

This is a new feature in Tuxedo 11.1.1.3.0, which can significantly improve the performance of Tuxedo application under MP mode.

In previous releases, messages between local client and remote server must go through bridge. For example, first the message will be sent to local bridge through IPC queue, next the local bridge sends it to remote bridge through network, then the remote bridge sends the message to server's IPC queue, finally the server retrieves the message from its IPC queue, so the Bridge will become a bottleneck under high concurrency. By utilizing the RDMA capabilities of Infiniband, Tuxedo 11.1.1.3.0 introduced a new feature of "Direct Cross Node Communication Leveraging RDMA", it provide the ability for local client to transfer message to remote server directly.

For more information about configuration, see [Oracle Tuxedo Configuration](#).

## Self-Tuning Lock Mechanism

This feature can adjust the value of SPINCOUNT dynamically for the best use of CPU cycle.

The Tuxedo bulletin board (BB) is a memory segment in which all the application configuration and dynamic processing information is held at run time. For some Tuxedo system operations (such as service name lookups and transactions), the BB must be locked for exclusive access: that is, it must be accessible by only one process. If a process or thread finds that the BB is locked by another process or thread, it retries, or spins on the lock for SPINCOUNT number of times (user level method via spin) before giving up and going to sleep on a waiting queue (system level method via system semaphore). Because sleeping is a costly operation, it is efficient to do some amount of spinning before sleeping.

Because the value of the SPINCOUNT parameter is application- and system-dependent, the administrator has to tune the SPINCOUT to be a proper value manually by observing the application throughput under different values of SPINCOUNT.

Self-Tuning Lock Mechanism takes the job of tuning automatically. It is designed to figure out a proper value of SPINCOUNT so that most requests to lock BB are completed by spinning instead of sleeping on a waiting queue.

The algorithm of Self-Tuning Lock Mechanism is improved in this release to help the tuning more accurate than before.

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

## Oracle Tuxedo SDP Support

One of the benefits of using InfiniBand based network hardware is the ability to utilize the socket direct protocol, or SDP. This protocol allows applications to communicate with each other via the normal socket interface but bypass the network processing associated with TCP/IP which includes things like ordering, fragmentation, timeouts, retries, and the like because the InfiniBand hardware takes care of those concerns. As well SDP can support zero copy transfers as the InfiniBand hardware is capable of directly transferring buffers from the caller's address space.

By utilizing SDP, Tuxedo applications can reduce the amount of CPU consumed for networking operations as well as increase the overall throughput of network operations. SDP can be used on all Tuxedo network connections including BRIDGE to BRIDGE communication, the domain gateway GWTDOMAIN for communication with other Tuxedo domains, for workstation and Jolt clients, and as well for communication with WebLogic Server via the WebLogic Tuxedo Connector.

For more information about configuration, see [Oracle Tuxedo Configuration](#).

## Shared Memory Interprocess Communication

This release significantly enhances performance of Tuxedo applications on Exalogic with use of shared memory queues instead of IPC Message Queues for inter process communication on the same Tuxedo node. With the use of shared memory queues, the sender and receiver processes can exchange pre-allocated messages in shared memory, thus eliminating the need to copy messages several times before message reaches its intended target and resulting in much better throughput and lower latency.

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

## Partial One Phase Read-Only Optimization for RAC

This feature takes advantage of the read-only optimization of resource manager for XA. Given two phase commit scenario, the prepare requests are synchronized to the participated groups except the reserved one. If all transaction branches in those groups are read-only, Tuxedo will do one-phase commit on the reserved one directly. It means one prepare request (to the reserved one) is saved and writing TLOG is ignored.

Transactions either within or across domains are supported, including global transaction across Tuxedo domain and WLS via WTC (in WLS 12.1.1 - Contact Oracle Support for a patch, or higher release of WLS).

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

## Shared Applications Staging

With Oracle Tuxedo 12c, one can share application directory (APPDIR) among many compute nodes of the storage appliance on an Exalogic system, making it easier to manage application deployment.

For more information about configuration, see [Oracle Tuxedo Configuration](#).

## Tightly Coupled Transactions Spanning Domains

In Oracle Tuxedo 11.1.1.3.0 or earlier, the transaction crossing domain is loosely coupled even if the branches of the transaction running on same database due to different global transaction identifiers (GTRIDs) are used in different domains. Since Oracle Tuxedo 12.1.1, common GTRID has been introduced in default to make branches within a global transaction crossing domains using common GTRID. The branches would be tightly coupled if they are running on same database (if the database allows).

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

## XA Transaction Affinity

XA Transaction Affinity provides the ability to route all Oracle database requests within one global transaction to the same Oracle RAC instance when possible; no matter if the requests come from an Oracle Tuxedo application server or Oracle WebLogic Server. This feature can reduce the cost of redirecting database requests to a new Oracle RAC instance, and thus can improve overall application performance.

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

## Common XID

In previous releases, for global transactions, each participating group has its own transaction branch, and a distinguished transaction branch identifier (XID) identifies each branch. If a global transaction involves multiple groups, Tuxedo adopts two-phase commit on each branch, taking the first participating group as the coordinator.

With the common XID (transaction branch identifier) feature in this release, Tuxedo shares the XID of the coordinator group with all other groups within the same global transaction. This is as opposed to each group having its own XID and thus requiring two-phase commit in earlier releases if multiple groups are participating.

Common XID eliminates the need to XA commit operations for groups that connect to the same Oracle RAC instance through the same service by using the coordinator branch directly.

In the cases where all groups in a global transaction use the coordinator branch directly, one-phase commit protocol (instead of two-phase commit protocol) is used, and thus avoids writing `TLOG`.

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

## Single Group Multiple Branches (SGMB)

In previous releases, servers in the same participated group use the same transaction branch in a global transaction; if these servers connect to different instances on the same RAC, the transaction branch may fail and an XA error, `XAER_AFFINITY`, will be reported, meaning one branch cannot go through different instances. For this reason, Tuxedo groups can only use singleton RAC services. A DTP service (if the DTP option, `-x` in `srvctl`, is specified) or a service offered by only one instance could be a singleton RAC service.

In this release, this feature eliminates the need to use singleton RAC service when multiple servers in a server group participate in the same global transaction. If servers in the same server group and same global transaction happen to connect to different RAC instances, a different

transaction branch is used. This enables such applications to perform load balancing across available RAC instances.

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

**Note:** The transaction still fails if more than 16 instances are involved in a single group.

## Failover/Failback across Database Instances

Fast Application Notification (FAN) is a facility provided by Oracle Database to allow database clients to know about changes in the state of the database. These notifications let an application respond proactively to events such as a planned outage of a RAC node or an imbalance in database load. Tuxedo provides support for FAN notifications by a new system server `TMFAN` that can monitor Oracle RAC instance and notify Tuxedo application server to establish a new Database connection in case of Database instance up or down.

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

## Load Balancing across RAC Instances

Based on FAN notification, Tuxedo `TMFAN` server can receive load balancing advisories that include the load information of each RAC instance. If the change in advised load exceeds the threshold specified in the `TMFAN` command line switches, then Tuxedo request will be routed to the Tuxedo application server that has a lower Database load.

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

## Concurrent Global Transaction Table Lock

Oracle Tuxedo manages global transactions by maintaining a table of active global transactions and their participants in the Oracle Tuxedo bulletin board called the Global Transaction Table or GTT. As this table is accessed by multiple concurrent processes it must be protected with a semaphore. In the normal Oracle Tuxedo case, the bulletin board lock is used to serialize access to this table. However, under heavy transaction load, the contention for this lock can become substantial resulting in an artificial performance bottleneck.

The XPP moves the serialization of access to the GTT from the bulletin board lock to a number of other locks, one for accessing the GTT, and one for each entry in the GTT. This allows a much greater level of concurrency when accessing the GTT and eliminates this bottleneck.

For more information, see [Using Oracle Tuxedo Advanced Performance Pack](#).

## Oracle Tuxedo Configuration

This section introduces the basic Oracle Tuxedo feature configuration on Exalogic. Some new `UBBCONFIG` `OPTIONS` are introduced by Tuxedo Advanced Performance Pack. For example, all of the Tuxedo Advanced Performance Pack features are enabled if the `OPTIONS` parameter in `RESOURCES` in `UBBCONFIG` is set to `EECS`. An alternative method is to continue setting `EXALOGIC` and feature specific `OPTIONS` in `UBBCONFIG`. However, all new features that Tuxedo Advanced Performance Pack introduces are not supported by `OPTION EXALOGIC`.

- [Direct Cross Node Communication Leveraging RDMA](#)
- [Self-Tuning Lock Mechanism](#)
- [Oracle Tuxedo SDP Support](#)
- [Shared Memory Interprocess Communication](#)
- [Partial One Phase Read-Only Optimization for RAC](#)
- [Database](#)
- [Choosing APPDIR](#)
- [Tightly Coupled Transactions Spanning Domains](#)
- [XA Transaction Affinity \(New\)](#)
- [Common XID \(New\)](#)
- [Single Group Multiple Branches \(SGMB\) \(New\)](#)
- [Failover/Failback across Database Instances \(New\)](#)
- [Load Balancing across RAC Instances \(New\)](#)
- [Concurrent Global Transaction Table Lock \(New\)](#)

### Direct Cross Node Communication Leveraging RDMA

The configuration for "Direct Cross Node Communication Leveraging RDMA" includes.

- [UBBCONFIG File](#)
- [Setting Shell Limit for Memory Lock](#)
- [Setting Default Directory Name for File Transfer](#)

## UBBCONFIG File

Direct Cross Node Communication Leveraging RDMA is only supported under MP mode. To enable this feature, you must specify `ECS` in `OPTIONS`, otherwise the message goes through the Bridge.

There is one attribute for Direct Cross Node Communication Leveraging RDMA in the `*RESOURCE` section.

### EXALOGIC\_SHARED\_PATH

The directory name of Oracle Tuxedo file transfer. The function of `EXALOGIC_SHARED_PATH` here is the same as that of environment variable `EXALOGIC_SHARED_PATH`; however, at Tuxedo runtime, such environment variable has higher priority. `EXALOGIC_SHARED_PATH` must be a shared directory with read/write permissions for all Tuxedo nodes and can be specified in `*RESOURCE` section only if RDMA is enabled.

There are five attributes for Direct Cross Node Communication Leveraging RDMA in the `*MACHINES` section.

### RDMADAEMONIP

The IP address where the `Msgq_daemon` is bound. It must be configured, and must be an IPoIB address (not an Ethernet based IP address). You should configure one `Msgq_daemon` for one logic machine.

### RDMADAEMONPORT

The port number where `Msgq_daemon` listens on. It must be configured.

### RDMAQSIZE

The EMSQ queue size. The default value is 65536 bytes if not defined in the UBBCONFIG file.

### RDMAQENTRIES

The EMSQ queue entry number, that is the maximum number of messages allowed in this queue. The default value is 64 if not defined in the UBBCONFIG file.

### EXALOGIC\_MSGQ\_CACHE\_SIZE

The entry number for Oracle Tuxedo EMSQ cache. The function of `EXALOGIC_MSGQ_CACHE_SIZE` here is the same as that of the environment variable `EXALOGIC_MSGQ_CACHE_SIZE`; however, the environment variable has the higher priority. The value must be between 32 and 2048 inclusive. `EXALOGIC_MSGQ_CACHE_SIZE` can be specified in `*MACHINES` only when RDMA is enabled. The default value is 32 if it is not defined in UBBCONFIG.

In some scenarios, Tuxedo performance can be improved when increasing this number. For more details, please see [Setting EXALOGIC\\_MSGQ\\_CACHE\\_SIZE](#).

After enable RDMA option in the \*RESOURCES section. Attribute "TYPE" of \*MACHINES section cannot be set, since by default, any machines in MP mode should be Exalogic machine (with the same type) to support RDMA feature.

You can also get/change the configuration via TM\_MIB. For more information, see [File Formats, Data Descriptions, MIBs, and System Processes Reference](#).

[Listing 1](#) shows an example UBBCONFIG File with Direct Cross Node Communication Leveraging RDMA enabled.

---

**Listing 1 UBBCONFIG File Example with Direct Cross Node Communication Leveraging RDMA Enabled**

---

```
*RESOURCES
IPCKEY          87654

MASTER         site1,site2
MAXACCESSERS   40
MAXSERVERS     40
MAXSERVICES    40
MODEL          MP
OPTIONS        LAN,EECS
LDBAL          Y

*MACHINES
slce04cn01     LMID=site1
                APPDIR="/home/oracle/tuxedo12.1.1.0/samples/atmi/simpapp"

TUXCONFIG="/home/oracle/tuxedo12.1.1.0/samples/atmi/simpapp/tuxconfig"
                TUXDIR="/home/oracle/tuxedo12.1.1.0"
                UID=601
```

```
GID=601
    RDMADAEMONIP=192.168.10.1
    RDMADAEMONPORT=9800
RDMAQSIZE=65536
RDMAQENTRIES=64

slce04cn02    LMID=site2

APPDIR="/home/oracle/tuxedo12.1.1.0/samples/atmi/simpapp/slave"

TUXCONFIG="/home/oracle/tuxedo12.1.1.0/samples/atmi/simpapp/slave/tuxconfi
g"

    TUXDIR="/home/oracle/tuxedo12.1.1.0"
    UID=601
    GID=601
    RDMADAEMONIP=192.168.10.2
    RDMADAEMONPORT=9800
    RDMAQSIZE=65536
    RDMAQENTRIES=64

*GROUPS
GROUP1
    LMID=site1    GRPNO=1    OPENINFO=NONE

GROUP2
    LMID=site2    GRPNO=2    OPENINFO=NONE

*NETWORK
site1    NADDR="//slce04cn01:5432"
```

```
NLSADDR="//slce04cn01:5442"

site1 NADDR="//slce04cn02:5432"
      NLSADDR="//slce04cn02:5442"

*SERVERS

DEFAULT:

      CLOPT="-A"

simpserv      SRVGRP=GROUP2 SRVID=3

*SERVICES

TOUPPER
```

---

## Setting Shell Limit for Memory Lock

The shared memory used by `Msgq_Daemon` will be locked into physical memory to avoid being paged to swap area, so it is necessary to set a proper value to memlock in `/etc/securitylimits.conf`.

Please use the following formula to get the minimum value for memlock:

```
[Msgq_daemon shared memory size]*2 + MAXACCESSERS *14 000 kb
```

`Msgq_daemon shared memory size`: The size of shared memory allocated by `Msgq_daemon`. For more information, see "[Calculating Shared Memory Size for Msgq\\_daemon](#)".

`MAXACCESSERS`: An attribute in the `UBBCONFIG` file.

For example:

```
Msgq_daemon shared memory size: 200*1024 kb
```

```
MAXACCESSERS: 100
```

```
200*1024*2 + 100 * 14000 = 1809600
```

Specify it in `/etc/securitylimits.conf` like that:

```
* hard memlock 1809600
* soft memlock 1809600
```

## Setting Default Directory Name for File Transfer

Before starting Oracle Tuxedo, ensure that there is a shared directory for all Exalogic nodes when Direct Cross Node Communication Leveraging RDMA is enabled. Make sure that access permissions are properly set.

The default name is `/u01/common/patches/tuxttmpfile`, you can also set your own directory using the `EXALOGIC_SHARED_PATH` environment variable. It is used for Oracle Tuxedo file transfer. When the EMSQ is full, or the message size exceeds the queue size, Oracle Tuxedo puts this message into a temporary file under the `/u01/common/patches/tuxttmpfile` directory, and sends notification directly to the remote process queue. The remote process can then get the file as long as it receives the notification.

## Self-Tuning Lock Mechanism

For configuration information, see "[Self-Tuning Lock Mechanism](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Oracle Tuxedo SDP Support

To enable Oracle Tuxedo SDP Support, you must specify `EECS` for `OPTIONS` in `*RESOURCES` section, and set the relevant configuration in `UBBCONFIG` file or `DMCONFIG` file.

You can also get or change the configuration via `TM_MIB`. For more information, see [File Formats, Data Descriptions, MIBs, and System Processes Reference](#).

This section covers the following configurations:

- MP
- GWTDOMAIN
- WSL
- /WS Client
- JSL
- WTC

## MP

According to the requirements, MP should work inside IB clusters, i.e., both master and slave machines are inside IB cluster, so only consider SDP and IPoIB are used inside IB cluster, and in bootstrap phase, tmboot, tlisten, bsbridge and bridge are using Socket API to communicate with each other.

- Configure SDP in MP deployment model.

Prefixed "sdp:" to the network address, and the network address must be an IPoIB address.

[Listing 2](#) shows a UBBCONFIG file SDP example.

### Listing 2 UBBCONFIG File SDP Configuration Example

---

```
*NETWORK
SITE1
    NADDR="sdp://IB_IP: 9003 "
    NLSADDR="sdp://IB_IP: 3050
SITE2
    NADDR="sdp://IB_IP: 9003 "
    NLSADDR="sdp://IB_IP: 3050"

To start tlisten:
tlisten "Cd /dev/tcp "Cl sdp://IB_IP: 3050
```

---

- Configure IPoIB in MP deployment model

Using the IPoIB address as the network address.

[Listing 3](#) shows a UBBCONFIG file IPoIB configuration example.

### Listing 3 UBBCONFIG File IPoIB Configuration Example

---

```
*NETWORK
```

```

SITE1

      NADDR="//IB_IP: 9003 "
      NLSADDR="//IB_IP: 3050

SITE2

      NADDR="//IB_IP: 9003 "
      NLSADDR="//IB_IP: 3050"

```

To start tlisten:

```
tlisten -d /dev/tcp -l //IB_IP: 3050
```

---

## GWTDOMAIN

If the node running `GWTDOMAIN` has multiple network interfaces (multi-homed) with multiple IP addresses, it is better to use explicit IP address when configuring `GWTDOMAIN` in `DMCONFIG` file instead of host name. Typically, every Exalogic node has at least two types of network interface, i.e., IB interface and Ethernet interface, in order to facilitate to demonstrate how to configure `GWTDOMAIN`, just presume IB interface is bound to IP address `IB_IP`, and Ethernet interface with IP address `ETH_IP`.

Functionally, `GWTDOMAIN` acts as both server and client in role, as server, it will listen on a configured IP address and port number in the `DMCONFIG` file to accept connection request from other `GWTDOMAIN`, as client, it will initiate connection request to other `GWTDOMAIN` by policy configured in the `DMCONFIG` file.

- Configure `GWTDOMAIN` to listen on SDP.
  - Prefix "sdp:" to the network address, and the network address must be an IPoIB address, as shown in [Listing 4](#).

### Listing 4 GWTDOMAIN Listening on SDPD DMCONFIG File Configuration Example

---

```

*DM_LOCAL
SCLCU03          GWGRP=DOMGRP
                  TYPE=TDOMAIN

```

```
*DM_TDOMAIN  
SCLCU03 NWADDR="sdp://IB_IP: 27610"
```

---

- Configure `GWTDOMAIN` to connect using SDP.

Prefixed "sdp:" to the network address, and the network address must be an IPoIB address as shown in [Listing 5](#).

### Listing 5 GWTDOMAIN Connection Using SDP DMCONFIG File Configuration Example

---

```
*DM_LOCAL  
SCLCU03 GWGRP=DOMGRP  
TYPE=TDOMAIN  
  
*DM_REMOTE  
SCLCU05 TYPE=TDOMAIN  
DOMAINID="EXALOGIC_SCLCU05"  
  
*DM_TDOMAIN  
SCLCU05 NWADDR="sdp://IB_IP: 27610"
```

---

- Configure `GWTDOMAIN` to listen on IPoIB.

Using the IPoIB address as the network address, the following is the `DMCONFIG` file configuration example for IPoIB:

### Listing 6 GWTDOMAIN Listen on IPoIB DMCONFIG File Configuration Example

---

```
*DM_LOCAL  
SCLCU03 GWGRP=DOMGRP  
TYPE=TDOMAIN  
  
*DM_TDOMAIN
```

```
SCLCU03 NWADDR="//IB_IP: 27610"
```

---

- Configure `GWTDOMAIN` to connect using IPoIB.

Using the IPoIB address as the network address, the following is the `DMCONFIG` file configuration example for IPoIB:

### Listing 7 GWTDOMAIN Connection Using IPoIB DMCONFIG File Configuration Example

---

```
*DM_LOCAL
SCLCU03          GWGRP=DOMGRP
                  TYPE=TDOMAIN

*DM_REMOTE
SCLCU05 TYPE=TDOMAIN
                  DOMAINID="EXALOGIC_SCLCU05"

*DM_TDOMAIN
SCLCU05 NWADDR="//IB_IP: 27610"
```

---

## WSL

- Configure WSL listen on SDP.

Prefixed "sdp:" to the network address, and the network address must be an IPoIB address as shown in [Listing 8](#).

### Listing 8 WSL Listening on SDP DMCONFIG File Configuration Example

---

```
*SERVERS
DEFAULT:          CLOPT="-A"
WSL               SRVGRP=WSGRP SRVID=1001
                  CLOPT="-A -- -nsdp://IB_IP: 11101 -m1 -M10 -x1"
```

- To configure WSL listen on IPoIB.

Using the IPoIB address as the network address, as shown in WSL Listening on IPoIB DM.

---

### Listing 9 WSL Listening on IPoIB DMCONFIG File Configuration Example

---

```
*SERVERS
DEFAULT:      CLOPT="-A"
WSL           SRVGRP=WSGRP SRVID=1001
              CLOPT="-A -- -n //IB_IP: 11101 -m1 -M10 -x1"
```

---

## /WS client

- Configure /WS client to use SDP

Prefixed "sdp:" to network address for WSNADDR, and the network address must be an IPoIB address, for example:

```
export WSNADDR=sdp://IB_IP:1001
```

- Configure /WS client to use IPoIB

Using the IPoIB address as the network address for WSNADDR, for example:

```
export WSNADDR="//IB_IP:1001
```

## JSL

Configure JSL listen on SDP

Prefixed "sdp:" to the network address, and the network address must be an IPoIB address as shown in [Listing 10](#).

---

### Listing 10 JSL Listening on SDP UBBCONFIG File Configuration Example

---

```
*SERVERS
```

```

DEFAULT:      CLOPT="-A"
JSL           SRVGRP=WSGRP SRVID=1001
              CLOPT="-A -- -nsdp: //IB_IP: 11101 -m1 -M10 -x1"

```

---

- To configure JSL listen on IPoIB.

Using the IPoIB address as the network address, the following is the `UBBCONFIG` file configuration example for JSL with IPoIB enabled:

#### Listing 11 JSL Listening on IPoIB UBBCONFIG File Configuration Example

---

```

*SERVERS
DEFAULT:      CLOPT="-A"
JSL           SRVGRP=WSGRP SRVID=1001
              CLOPT="-A -- -n //IB_IP: 11101 -m1 -M10 -x1"

```

---

## WTC

To enable SDP connection between WTC and Oracle Tuxedo, do the following steps:

1. Specify the `NWAddr` of the WTC service Local/Remote Access Points as follows:

```
sdp://IB_IP:port
```

It is the same as the `GWTDOMAIN NWADDR` configuration in the `DMCONFIG` file.

2. Add additional Java Option “`-Djava.net.preferIPv4Stack=true`” to the java command line to start up WLS server.

**Notes:** If the WTC access point has SSL enabled, after configuring for the SDP, the SSL configuration is ignored.

Only Weblogic Server 12c (12.1.1) and higher can connect to Oracle Tuxedo via SDP. For more information, see [Enable IPv4 for SDP transport](#), [NWAddr attribute for WTC local Tuxedo Domain configuration](#), and [NWAddr attribute for WTC remote Tuxedo Domain configuration](#).

## Shared Memory Interprocess Communication

For configuration information, see "[Shared Memory Interprocess Communication](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Partial One Phase Read-Only Optimization for RAC

For configuration information, see "[Partial One Phase Read-Only Optimization for RAC](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Database

The Oracle Tuxedo system uses the X/Open XA interface for communicating with the various resource managers. The XA Standard is widely supported in all the major database vendor products.

You can use SDP (Sockets Direct Protocol) for Oracle Database invocations. There is no special requirement for Oracle Tuxedo application.

Please configure the database to support InfiniBand, as described in [Configuring SDP Protocol Support for Infiniband Network Communication to the Database Server](#) in the Oracle Database Net Services Administrators Guide.

**Note:** The following SDP parameters affect performance when sending large data blocks to the database.

- `recv_poll`

The receiving poll time before going to sleep waiting for data. You can improve the write performance by decreasing the `recv_poll` value. It is recommended to set it to zero.

- `sdp_zcopy_thresh`

The threshold for zero copy sending data. If the messages are longer than the `sdp_zcopy_thresh` value, it causes the user buffer to be pinned and the data sent directly from the original buffer. This results in decreased CPU use. The default value is 64K; it is recommended to set it to zero.

For example, you can set the two parameters in `/etc/modprobe.conf` on the server node as follows:

```
options ib_sdp sdp_zcopy_thresh=0 recv_poll=0
```

## Choosing APPDIR

You can deploy your Oracle Tuxedo application to a shared directory on Exalogic in MP environment (named Shared Applications Staging) with the requirements that both `EECS` option and MP mode are set and enabled. Before booting the Oracle Tuxedo application, ensure the following parameters are set correctly in the `UBBCONFIG` file:

### TUXCONFIG

The `TUXCONFIG` must be different for each node.

### TLOGDEVICE

The `TLOGDEVICE` must be different for each node.

### ULOGPFX

Set different path for `ULOGPFX` if you want to have a separate `ULOG`.

### Access Permission for shared APPDIR

Users from different Exalogic nodes must have the same uid and gid of OS.

Besides above, each node had better use distinctive `TMIFRSVR repository_file`, standard output/error file, `AUDITLOG` file, and `ALOGPFX` to have a clear logging system. All applications should be set distinctive names to use the Shared Applications Staging feature better.

[Listing 12](#) shows a `UBBCONFIG` file shared `APPDIR` example.

#### Listing 12 UBBCONFIG File Shared APPDIR

---

```
...
*MACHINES
slce04cn01 LMID=site1
    APPDIR="/home/oracle/tuxapp"
    TUXCONFIG="/home/oracle/tuxapp/tuxconfig_cn01"
    TUXDIR="/home/oracle/tuxedo11gR1"
    TLOGDEVICE=/home/oracle/tuxapp/TLOG1
    ULOGPFX="/ home/oracle/tuxapp /ULOG_cn01"
    RDMADAEMONIP="192.168.10.1"
    RDMADAEMONPORT=9800
```

```
RDMAQSIZE=1048576
```

```
RDMAQENTRIES=1024
```

```
slce04cn02 LMID=site2
```

```
APPDIR=" home/oracle/tuxapp"
```

```
TUXCONFIG=" home/oracle/tuxapp/tuxconfig_cn02"
```

```
TUXDIR="/home/oracle/tuxedo11gR1"
```

```
TLOGDEVICE=/home/oracle/tuxapp/TLOG2
```

```
ULOGPFX="/home/oracle/tuxapp /ULOG_cn02"
```

```
RDMADAEMONIP="192.168.10.2"
```

```
RDMADAEMONPORT=9800
```

```
RDMAQSIZE=1048576
```

```
RDMAQENTRIES=1024
```

---

If SECURITY is set in the UBBCONFIG file, only MP domains with EECS enabled can share a common APPDIR.

## Tightly Coupled Transactions Spanning Domains

For configuration information, see "[Tightly Coupled Transactions Spanning Domains](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## XA Transaction Affinity

For configuration information, see "[XA Transaction Affinity](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Common XID

For configuration information, see "[Common XID](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Single Group Multiple Branches (SGMB)

For configuration information, see "[Single Group Multiple Branches \(SGMB\)](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Failover/Failback across Database Instances

For configuration information, see "[Failover/Failback across Database Instances](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Load Balancing across RAC Instances

For configuration information, see "[Load Balancing across RAC Instances](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Concurrent Global Transaction Table Lock

For configuration information, see "[Concurrent Global Transaction Table Lock](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

# Best Practices to Optimize Performance

This section contains the following topics:

- [Direct Cross Node Communication Leveraging RDMA](#)
- [Self-Tuning Lock Mechanism](#)
- [Shared Memory Interprocess Communication](#)
- [Partial One Phase Read-Only Optimization for RAC](#)
- [XA Transaction Affinity](#)
- [Common XID](#)
- [Failover/Failback across Database Instances](#)
- [Load Balancing across RAC Instances](#)
- [Single Group Multiple Branches \(SGMB\)](#)
- [Oracle Tuxedo SDP Support](#)

## Direct Cross Node Communication Leveraging RDMA

### Scenarios recommended

The feature provides the ability for client to directly access remote server, it eliminates the bottleneck on `BRIDGE`. When Tuxedo is under high concurrent remote access in MP mode, the throughput will have significant improvement if this feature is enabled in `UBBCONFIG`.

**Note:** The following scenario is not recommended for this feature:

The client connects with the remote server through `BRIDGE`, and work on it for a relatively short duration. For example, `tpinit()` followed several `tpcall()`, then `tpterm()`. The overhead of creating/opening/closing for RDMA connection is much higher than it on Unix IPC queue. So it cannot have obvious performance improvement under this scenario.

### Setting `EXALOGIC_MSGQ_CACHE_SIZE`

Each Oracle Tuxedo thread has an `EMSQ` runtime cache; the default entry number is 32. You can change it between 32 and 2048 using the environment `EXALOGIC_MSGQ_CACHE_SIZE` variable before the Oracle Tuxedo application starts, or setting it in `UBBCONFIG`. In some scenarios, increasing the number can improve Oracle Tuxedo performance, for example:

- If there are 40 remote Oracle Tuxedo servers providing the same service and clients call the service repeatedly, setting `EXALOGIC_MSGQ_CACHE_SIZE` to a value equal to or greater than 40 on the client can improve performance.
- If there are 50 WSHs, and each WSH receives response messages from the same remote server, setting `EXALOGIC_MSGQ_CACHE_SIZE` to a value equal to or greater than 50 on the server environment can improve performance.

## Calculating Shared Memory Size for `Msgq_daemon`

### Using `tmloadcf`

To get the recommended value, please run `tmloadcf -c ubb` as shown in [Listing 13](#).

#### Listing 13 `UBBCONFIG` File `*MACHINES` Section

---

```
*MACHINES
ex03
```

```
LMID=site1
...
RDMADAEMONIP="192.168.10.1"
RDMADAEMONPORT=9800
RDMAQSIZE=100000
RDMAQENTRIES=100
MAXACCESSERS=100
...
ex03_1 LMID=site2
...
RDMADAEMONIP="192.168.10.2"
RDMADAEMONPORT=9800
RDMAQENTRIES=1000
MAXACCESSERS=200
...
ex04 LMID=site3
...
RDMADAEMONIP="192.168.10.3"
RDMADAEMONPORT=9800
RDMAQSIZE=100000
RDMAQENTRIES=100
MAXACCESSERS=200
MAXSERVERS=100
...
ex04_1 LMID=site4
...
RDMADAEMONIP="192.168.10.4"
```

```

RDMADAEMONPORT=9800
RDMAQSIZE=1000000
RDMAQENTRIES=1000
MAXACCESSERS=100
...

```

---

Run command `tmloadcf -c ubb`, get the output shown in [Listing 14](#).

**Listing 14 tmloadcf -c ubb Output Example**

---

Ipc sizing (minimum /T values only) ...

Fixed Minimums Per Node

```

SHMMIN: 1
SHMALL: 1
SEMMAP: SEMMNI

```

Variable Minimums Per Node

```

SEMUME,          A                      SHMMAX
                SEMMNU,          *                      *

```

Node	SEMMNS	SEMMSL	SEMMSL	SEMMNI	MSGMNI	MSGMAP	SHMSEG	RCDMSZ
-----	-----	-----	-----	-----	-----	-----	-----	-----
ex03	126	15	120	A + 2	26	52	1178K	220M
ex04	221	28	220	A + 1	26	52	1340K	340M
ex04_1	121	15	120	A + 1	26	52	1178K	1300M
ex03_1	221	28	220	A + 1	25	50	1340K	2500M

---

RCDMSZ increases linearly when any of following items configured in UBB increases:

- MAXACCESSERS
- RDMAQSIZE
- The number of logic machines in the UBBCONFIG file

### Adjusting Shared memory size:

After getting the RCDMSZ by `tmloadcf`, you can adjust the actual size according to the following runtime factors:

- Frequency of start and exit of processes

For performance reasons, shared memory resource are not immediately freed after the queue is destroyed. The more frequently processes start/stop, the more frequently queues are created/destroyed, the more resource s are occupied by stale queues. At most, you should allocate more shared memory using the following formula:

$$3 * EXALOGIC\_MSGQ\_CACHE\_SIZE * MAXSERVERS * RDMAQSIZE$$

- EXALOGIC\_MSGQ\_CACHE\_SIZE: Specify this value in UBBCONFIG file or environment variable, the default value is 32.
- MAXSERVERS: A UBBCONFIG file attribute.
- RDMAQSIZE: A UBBCONFIG file attribute.

For example, if MAXSERVERS is 100, RDMAQSIZE is 100000 and processes on machine ex04 start/stop frequently, you should configure the shared memory size with at most 1300M (340M+960M) instead of 340M.

- The actual number of machines in the domain

The number of machines in the UBBCONFIG file is considered in calculating RCDMSZ. If new machines are added using MIB, you should increase the size linearly.

- The entry number of queue cache

In some scenarios, `Msgq_daemon` needs to allocate more shared memory when increasing queue cache entry number using EXALOGIC\_MSGQ\_CACHE\_SIZE. For example, if Oracle Tuxedo process start/stop or `tpinit/tpterm` frequently. Then you should consider adjusting the initial `Msgq_daemon` shared memory size, otherwise the message transfer may fail.

The default value is 32 for each Oracle Tuxedo thread, and `tmloadcf -c` always assumes the value of 32. If you change the default value, adjust the `Msgq_daemon` shared memory size using the following method:

$\text{MEMSZ} * \text{CACHENUM} / 32$

- MEMSZ: The current shared memory size calculated by above method.
- CACHENUM: The maximum entry queue cache number for each Oracle Tuxedo thread.

**Note:** You can adjust the size using this method only when setting EXALOGIC\_MSGQ\_CACHE\_SIZE at the machine level.

**Note:** For detailed information about configuration, please see Direct Cross Node Communication Leveraging RDMA in [Oracle Tuxedo Configuration](#).

## Self-Tuning Lock Mechanism

For best practices, see "[Self-Tuning Lock Mechanism](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Shared Memory Interprocess Communication

For best practices, see "[Shared Memory Interprocess Communication](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Partial One Phase Read-Only Optimization for RAC

For best practices, see "[Partial One Phase Read-Only Optimization for RAC](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## XA Transaction Affinity

For best practices, see "[XA Transaction Affinity](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Common XID

For best practices, see "[Common XID](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Failover/Failback across Database Instances

For best practices, see "[Failover/Failback across Database Instances](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Load Balancing across RAC Instances

For best practices, see "[Load Balancing across RAC Instances](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Single Group Multiple Branches (SGMB)

For best practices, see "[Single Group Multiple Branches \(SGMB\)](#)" in *Using Oracle Tuxedo Advanced Performance Pack*.

## Oracle Tuxedo SDP Support

SDP can be used on all Tuxedo network communications, but it is not recommended for the scenario that is applicable to Direct Cross Node Communication Leveraging RDMA or Direct Cross Domain Communication Leveraging RDMA.

## Running Oracle Tuxedo

There is a difference running Oracle Tuxedo on a non-Exalogic platform if Direct Cross Node Communication Leveraging RDMA is enabled. The `tux_msgq_monitor` must be started before booting an Oracle Tuxedo application. This section includes the following topics:

- [Start/Stop tux\\_msgq\\_monitor](#)
- [Assistant Tools](#)

### Start/Stop tux\_msgq\_monitor

- Start `tux_msgq_monitor`

`tux_msgq_monitor` functions as the `Msgq_daemon` default monitor. It is responsible for starting `Msgq_daemon` and checking it is running status. When `Msgq_daemon` terminates abnormally, it restarts using `tux_msgq_monitor`.

`tux_msgq_monitor` should be started before booting the Oracle Tuxedo application. One monitor can only serve one Oracle Tuxedo application on one logical machine.

Before starting `tux_msgq_monitor`, ensure the environment variables `TUXCONFIG`, `LD_LIBRARY_PATH` and `TUXDIR` are properly set. Run `tux_msgq_monitor` as follows:

```
tux_msgq_monitor -i <IPoIB address> -d <port number> -M <shared memory size> -K <shared memory key to be used by Msgq_daemon>
```

`tux_msgq_monitor`: Options lists `tux_msgq_monitor` options.

**Table 2 tux\_msgq\_monitor: Options**

Options	Descriptions
-i	IPoIB address, it must be identical to the value of <code>RDMADAEMONIP</code> in <code>UBBCONFIG</code> file.
-d	Port number, it must be identical to the value of <code>RDMADAEMONPORT</code> in <code>UBBCONFIG</code> file.
-M	Shared memory size (in bytes) allocated by <code>Msgq_daemon</code> , you can get it by <code>tmloadcf -c</code> . For more information, see <a href="#">Calculating Shared Memory Size for Msgq_daemon</a> .
-K	Shared memory key used by <code>Msgq_daemon</code> . <code>-K</code> must be great than 0; its maximum value depends on the definition of <code>key_t</code> by OS.

When `tux_msgq_monitor` is running, there will be two active processes named as `tux_msgq_monitor` and `Msgq_daemon`.

After starting `tux_msgq_monitor` successfully, you can boot the Oracle Tuxedo application.

- Shutdown `tux_msgq_monitor`

Execute the following command:

```
kill pid -- tux_msgq_monitor's pid
```

**Note:** If you execute the kill command using `-9`, do the following steps to clear all the resources.

```
kill pid-- Msgq_daemon pid
```

```
ipcrm -m shmid -- the shared memory created by Msgq_daemon.
```

## Assistant Tools

### Shell Scripts for Start/Stop Oracle Tuxedo

There are some shell scripts that simplify the startup/shutdown procedure. Using these tools, you can only run one command to start/stop both `tux_msgq_monitor` and an Oracle Tuxedo

application. Before running these commands, ensure the environment variables `TUXCONFIG`, `LD_LIBRARY_PATH` and `APPDIR` are set properly.

For example, on the master node you can start/stop Oracle Tuxedo as follows:

On master node, there are two shell scripts:

**tmboot.sh -i daemon\_ip -d daemon\_port -M shm\_size -K shm\_key [-l nlsaddr]**

This script starts up `tux_msgq_monitor`, executes `tmboot` to start the Oracle Tuxedo application, and starts `tlisten` if option `"-l"` is specified.

**tmshut.sh**

Stops both the Oracle Tuxedo application and `tux_msgq_monitor`.

On slave node, there are two shell scripts:

**tlisten\_start.sh -l nlsaddr -i daemon\_ip -d daemon\_port -M shm\_size -K shm\_key**

This script starts the `tux_msgq_monitor` and `tlisten`.

**tlisten\_stop.sh**

This script terminates `tlisten` and `tux_msgq_monitor`.

**Note:** In MP mode, for startup, you should run the commands in the following sequence:

- `tmlisten_start.sh` on all the slave nodes
- `tmboot.sh` on the master node

**Note:** For shutdown, you should run commands in the following sequence:

- `tmshut.sh` on the master node
- `tmlisten_stop.sh` on all the slave nodes

## Running Oracle Tuxedo on OVM

There are no special requirements for Oracle Tuxedo running on OVM.

### Upgrade

There are no special requirements if you do not use any Exalogic optimization. For more information, see [Oracle Tuxedo Interoperability Guide](#) and [Upgrading the Oracle Tuxedo System to 12c Release 1 \(12.1.1\)](#).

**Note:** If any Exalogic optimization is specified in `OPTIONS`, you cannot perform a hot upgrade from any previous release.

# Appendix

## Terminology

- **Oracle Home:** An Oracle Home directory serves as the central support directory for all Oracle products installed on the target system.
- **Compute Node:** A physical machine in an Exalogic rack that is meant for running an Oracle Fusion Middleware instance at any given time.
- **Shared Storage:** Shared storage refers to the Sun ZFS Storage 7320 appliance that is accessible by all compute nodes in the Oracle Exalogic Machine. All compute nodes in the Exalogic machine can access this storage appliance simultaneously for both read and write operations.
- **UBBCONFIG:** Each Oracle Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The text version of the configuration file is referred to as UBBCONFIG. The UBBCONFIG file for a Oracle Tuxedo domain contains all the information necessary to boot the application.
- **TUXCONFIG:** 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.
- **XA:** The eXtended Architecture (XA) is the standard-defined interface between the Transaction Manager and the Resource Manager. XA allows program control of Resource Managers that are involved in distributed transactions
- **InfiniBand:** It is a switched fabric communications link used in high-performance computing and enterprise data centers. Its features include high throughput, low latency, quality of service and failover, and it is designed to be scalable.
- **IB:** InfiniBand.
- **IBA:** InfiniBand Architecture
- **EMSQ:** Exalogic Message Queue based on InfiniBand technology.
- **Msgq\_daemon:** A daemon process that manages EMSQ resources.
- **IPoIB:** IP over InfiniBand.
- **EOIB:** Ethernet over InfiniBand.

SDP: Sockets Direct Protocol

## Oracle Tuxedo installation

This section contains the following topics.

- [Configuration for Exalogic](#)
- [Platform Requirements](#)
- [Start Installation](#)

### Configuration for Exalogic

Before the Oracle Tuxedo installation, you should understand the current state of the Exalogic environment.

It is assumed that you have completed all tasks described in the Oracle Exalogic Machine Owner's Guide, which discusses your data center site preparation, Oracle Exalogic machine commissioning, initial networking configuration including IP address assignments, and initial setup of the Sun ZFS Storage 7320 appliance.

### Platform Requirements

Oracle Tuxedo optimizations can run on both Exalogic Linux and SPARC server. For more details, please see [Oracle Tuxedo 12c Release 1 \(12.1.1\) Platform Data Sheets](#).

#### Choosing Oracle Tuxedo Home

We recommend that you can install the Oracle Tuxedo product binaries in one of the shares on Sun ZFS Storage 7320 appliance locations, so you can run Oracle Tuxedo on any Exalogic nodes by one binary copy.

**Note:** The share, which is a shared file system, must be accessible by all compute nodes. You can create a local user account for each node, and ensure it has the same UID and GID (to avoid permission access issues), or create NIS accounts for users.

Oracle Tuxedo must be installed in a different directory if you want to develop the Oracle Tuxedo plug-in interface with different implementation for each Oracle Tuxedo.

### Start Installation

For installation information, see [Installing the Oracle Tuxedo System](#).

