Oracle® Tuxedo Message Queue (OTMQ)

Installation Guide

12*c* Release 2 (12.1.3)

December 2014

ORACLE

Installing Oracle Tuxedo Message Queue, 12*c* Release 2 (12.1.3)

Contents

1. Preparing to Install the Oracle Tuxedo Message Queue System

CHAPTER 1

Preparing to Install the Oracle Tuxedo Message Queue System

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

- Oracle Universal Installer (OUI)
- Oracle Tuxedo Message Queue Web Distribution
- Oracle Tuxedo Message Queue Software Components
- Hardware and Software Prerequisites
- Interprocess Communication Resources Configuration
- Oracle Installation Program
- Oracle Home Directory
- Installation Road Map

Oracle Universal Installer (OUI)

The Oracle Tuxedo Message Queue 12*c* Release 2 (12.1.3) product software installer is based on the Oracle Universal Installer (OUI). To install the Oracle Tuxedo Message Queue 12*c* 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 Message Queue 12cR2 with Oracle Home

When you install Oracle Tuxedo Message Queue 12*c* Release 2 (12.1.3), 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 Message Queue 12cR2

To install Oracle Tuxedo Message Queue 12*c* Release 2 (12.1.3) 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 Oracle Tuxedo Message Queue of the same version, the installer will show a warning.

It is not allowed to install Oracle Tuxedo Message Queue 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, HPUX-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 Oracle Tuxedo Message Queue specific response file parameters, see "Required Variables in Oracle Tuxedo Message Queue Response File".

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, which 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".

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".

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, enter:

```
./runInstaller -deinstall -silent DEINSTALL_LIST={"component1","1.0.1.2"}
```

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

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 Message Queue Web Distribution

An evaluation copy of Oracle Tuxedo Message Queue 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 Message Queue product software are available for download from the Oracle corporate Web site.

Oracle Tuxedo Message Queue Software Components

The Oracle Tuxedo Message Queue 12c Release 2 (12.1.3) software distribution contains the following components:

- Oracle Tuxedo Message Queue Core
- Oracle Tuxedo Message Queue (OTMQ)
- Oracle Tuxedo Message Queue (OTMQ) Standalone
- Tuxedo .NET Client/Tuxedo Server Cores/Tuxedo ATMI Client/Tuxedo Client Core/Tuxedo ATMI Client Core
- Oracle Universal Installer (oracle.swd.oui)/Oracle One-Off Patch Installer (oracle.swd.opatch)/Installer SDK Component (oracle.swd.oui.core)/Java Runtime Environment (oracle.swd.jre)/oracle.swd.oui.core.min

Notes:

- Installing on top of existing Oracle Tuxedo will only install the component of "Oracle Tuxedo Message Queue Core" and "Oracle Tuxedo Message Queue (OTMQ)".
- Standalone installation mode will install all components.

Hardware and Software Prerequisites

The Oracle Tuxedo Message Queue software must be installed on each server machine that will participate in an Oracle Tuxedo Message Queue application, also known as a Tuxedo domain. An Oracle Tuxedo Message Queue 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*.

An Oracle Tuxedo Message Queue 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 Oracle Tuxedo Message Queue application running different releases of the Oracle Tuxedo Message Queue 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 Message Queue software in the application. For more information about Oracle Tuxedo Message Queue applications, see Oracle Tuxedo Message Queue Product Overview.

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

System Requirements

The system requirements for Oracle Tuxedo Message Queue 12*c* Release 2 (12.1.3) are given in Table 1-1.

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

Component	Requirement
Platform	Any platform identified in "Oracle Tuxedo Message Queue 12c Release 2 (12.1.3) Platform Data Sheets".
Hard disk drive	As stated in the data sheet for the target platform in "Oracle Tuxedo Message Queue 12c Release 2 (12.1.3) Platform Data Sheets".

To install Oracle Tuxedo Message Queue 12*c* Release 2 (12.1.3) on an existing Oracle Tuxedo version, you must select full installation in Oracle Tuxedo 12*c* Release 2 (12.1.3) installation.

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 Message Queue 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".

The amount of temporary storage space needed depends upon the target platform, as stated in the data sheets in "Oracle Tuxedo Message Queue 12c Release 2 (12.1.3) Platform Data Sheets".

When you start OUI, it automatically copies some executable files and link files into the default `/tmp` directory (`C:\Documents and Settings\<user ID>\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

This procedure is the same as the Oracle Tuxedo 12*c* Release 2 (12.1.3) installation.

For more information about configuring IPC resource, see "IPC Resource Configuration on a UNIX System" and "Performing Post-Installation Tasks" in *Installing the Oracle Tuxedo System*.

Note: Before installing Oracle Tuxedo Message Queue software on a UNIX system, Oracle recommends that you adjust the IPC parameters on the target machine.

Oracle Installation Program

The Oracle Tuxedo Message Queue 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 Message Queue software on Windows or UNIX systems.

This procedure is the same as the Oracle Tuxedo 12*c* Release 2 (12.1.3) installation.

For more information about configuring IPC resource, see "IPC Resource Configuration on a UNIX System" and "Performing Post-Installation Tasks" in *Installing the Oracle Tuxedo System*.

Note: Before installing Oracle Tuxedo Message Queue software on a UNIX system, Oracle recommends that you adjust the IPC parameters on the target machine.

Installation Methods

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

- Graphical user interface (GUI) installation, described in “Installing Oracle Tuxedo Message Queue Using GUI-Mode Installation”.
- Silent installation, described in “Installing Oracle Tuxedo Message Queue Using Silent Installation”.

Cancelling Installation

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

You can either install Oracle Tuxedo Message Queue 12*c* Release 2 (12.1.3) in a previous Tuxedo product directory, or install the Oracle Tuxedo Message Queue 12*c* 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 Message Queue 12*c* Release 2 (12.1.3) offers the following install types:

- Install on top of existing Oracle Tuxedo Installation

  Only OTMQ components are copied, the existing Oracle Tuxedo binary is not changed.

- Standalone Installation

  All related Oracle Tuxedo binary and QTMQ components are copied to the target directory.

Oracle Home Directory

When you install Oracle Tuxedo Message Queue, 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 Message Queue 12*c* 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 Message Queue 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 Message Queue 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 Message Queue 12*c* Release 2 (12.1.3), see one of the following sections:

- "Installing Oracle Tuxedo Message Queue Using GUI-Mode Installation"
- "Installing Oracle Tuxedo Message Queue Using Silent Installation"

If you want to uninstall your Oracle Tuxedo Message Queue software, see "Post-Installation".

CHAPTER 3

Installing Oracle Tuxedo Message Queue Using GUI-Mode Installation

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

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 Message Queue on a UNIX system with a non-graphics console, use silent install mode.

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 Message Queue 12c Release 2 (12.1.3) Platform Data Sheets".
2. Log in to the Windows system as the `Administrator` or as a member of the `Administrator` group.

   You need administrative privileges for standalone installation mode on a Windows system; however, if you are going to install on top of existing Oracle Tuxedo Installation, you do not need administrative privileges.
3. Ensure that you have enough free space for the Oracle Tuxedo Message Queue installation.

   For disk space requirements, see "Oracle Tuxedo Message Queue 12c Release 2 (12.1.3) Platform Data Sheets".
4. Install Oracle Tuxedo Message Queue by downloading from the Oracle Web site:

   a. Go to http://www.oracle.com/technology/software/index.html and download the Oracle Tuxedo Message Queue 12*c* Release 2 (12.1.3) installation file specific to your platform.

   b. Go to the directory where you downloaded the Oracle Tuxedo Message Queue 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".
6. Re-login to the system after Oracle Tuxedo Message Queue 12*c* 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 Message Queue 12c Release 2 (12.1.3) Platform Data Sheets"
2. Log in to the UNIX system as the Oracle Tuxedo Message Queue administrator.
3. Ensure that you have enough free space for the Oracle Tuxedo Message Queue installation.

   For disk space requirements, see "Oracle Tuxedo Message Queue 12c Release 2 (12.1.3) Platform Data Sheets".
4. Install Oracle Tuxedo Message Queue by downloading from the Oracle Web site:

a. Go to http://www.oracle.com/technology/software/index.html and download the Oracle Tuxedo Message Queue 12*c* 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 Oracle Tuxedo Message Queue 12*c* 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".

Running GUI-Mode Installation

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

The Oracle Tuxedo Message Queue installer program prompts you to enter specific information about your system and configuration. For instructions on responding to the prompts during installation, see Table 3-1.

Table 3-1 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 3-1 Instructions

In This Window . . .	Perform the Following Action . . .
Choose install type	Select the install type that you want installed on your system. The following choices are available: • Install on top of existing Oracle Tuxedo Installation (the default) - only Oracle Tuxedo Message Queue files. • Standalone installation - all Oracle Tuxedo Message Queue files, and minimal Tuxedo files. For a detailed list of software components for each install type, see "Install Types". 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).

Table 3-1 Instructions

In This Window . . .	Perform the Following Action . . .
Specify Oracle Home	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. 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. 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. For silent installations, if a non-empty, existing directory is specified, an error is logged in the console and in the `installActions<timestamp>.log` file. Also, the OUI aborts. To override this condition, use the -force flag on the command line. The effect of using the `-force` flag is the same as selecting Yes while installing in interactive mode. You receive a warning message, but the installation continues. If you install on top of existing Oracle Tuxedo installation, Oracle Tuxedo Message Queue product will be installed on Tuxedo Product Installation Directory. If you choose standalone installation, Oracle Tuxedo Message Queue Product Installation Directory is `"$ORACLE_HOME/otmq12.1.3.0.0"` by default (UNIX) and cannot be changed after the `ORACLE_HOME` is determined. Oracle Tuxedo Message Queue installer does not allow to install two same version Oracle Tuxedo Message Queue products into the same `ORACLE_HOME`. If the selected `ORACLE_HOME` already installed the same version Oracle Tuxedo Message Queue, the installer will show a warning.

Table 3-1 Instructions

In This Window . . .	Perform the Following Action . . .
Check prerequisite (install on top of existing Oracle Tuxedo installation only)	If the certain Tuxedo component (published in Tuxedo12.1.3.0.0) is not installed on the target ORACLE_HOME, you cannot continue the installation until you specify a proper ORACLE_HOME. If the certain Tuxedo Server component (published in Tuxedo12.1.3.0.0) is not installed on the target ORACLE_HOME, you cannot continue the installation until you specify a proper ORACLE_HOME. Oracle Tuxedo 12*c* Release 2 (12.1.3) Rolling Patch 004 or above is required on UNIX platforms; if this requirement is not met, you will get an error message. For windows platforms, if the registry table for Tuxedo12.1.3.0.0 is missing, you will get an error message.
Check prerequisite (standalone installation only)	If the certain Tuxedo component (published in Tuxedo12.1.3.0.0) is installed on the target ORACLE_HOME, you cannot continue the installation until you specify a proper ORACLE_HOME. For windows platforms, if Oracle Tuxedo with this version is already installed on this machine, you will get an error message.
Select samples installation	This screen appears to confirm install the samples.
Configure tlisten choice (standalone installation only)	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. This window appears only if you have installed the Server Install.

Table 3-1 Instructions

In This Window . . .	Perform the Following Action . . .
Configure `tlisten` password (standalone installation only)	This window appears only if you select to configure tlisten. 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. **Note:** Machines on the same domain must have the same `tlisten` password defined. 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 `tlisten(1)` or `wlisten(1)` gateway processes, Oracle Tuxedo authenticates using the `tlisten` password. Enter and re-enter a `tlisten` password and then click Next to continue.
Configure `tlisten` port (Windows platforms only)	Set the `tlisten` port value. **Note:** For Windows platforms, the default value is 3050. There is no default value for UNIX platforms
`tlisten` encryption method (Windows platforms only)	Specify the `tlisten` encryption method: LLE or SSL. The default is LLE.
Choose encryption min bit (Windows platforms only)	Specify `tlisten` minimum encryption bit. • SSL: 112, 128, 256 can be selected. The default minimum value is 112. • LLE: 0, 56, 128, 256 can be selected. The default minimum value is 0.
Choose encryption max bit (Windows platforms only)	Specify `tlisten` maximum encryption bit. • SSL: 112, 128, 256 can be selected. The default maximum value is 256. • LLE: 0, 56, 128, 256 can be selected. The default maximum value is 256.
SSL parameter input (standalone installation only)	If the SSL encryption method is chosen, you must enter Security Principal name, location, and password.

Table 3-1 Instructions

In This Window . . .	Perform the Following Action . . .
SSL installation choice (standalone installation only)	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. 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; otherwise, select **No**.
LDAP configuration for SSL support (standalone installation only)	Enter the following LDAP configuration information and then click **Next** to continue. • LDAP Service Name: Fully qualified domain name of the LDAP server (for example, pcwiz.mydomain.com) • LDAP PortID: Port number through which the local machine communicates with the LDAP server (for example, 389) • LDAP BaseObject: Distinguished name of the search base object in the LDAP server (for example, o=beasys.com)
LDAP filter file choice (standalone installation only)	LDAP filter file location is `tux_prod_dir/udataobj/security/bea_ldap_filter.dat` by default. `tux_prod_dir` represents the product directory, where you install Oracle Tuxedo Message Queue software; the default LDAP filter file is generated after installation is complete.
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 Message Queue software is complete!

What Do I Do Next?

To configure your Oracle Tuxedo Message Queue software, verify that your software is installed correctly, or to uninstall Oracle Tuxedo Message Queue software, see "Post-Installation".

CHAPTER 4

Installing Oracle Tuxedo Message Queue Using Silent Installation

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

- What Is Silent Installation?
- 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.

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 Oracle Tuxedo Message Queue Response File

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

Note: For String type variables, you must contain the value using the double quote.

RESPONSEFILE_VERSION

The format version number of the response file. You must enter as follows.

```
RESPONSEFILE_VERSION=2.2.1.0.0
```

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
Standalone installation	This install type installs all Oracle Tuxedo Message Queue files, and installs minimal Tuxedo files.
Install on top of existing Oracle Tuxedo installation	This install type installs Oracle Tuxedo Message Queue server files only.

For more information, see “Install Types”.

INSTALL_SAMPLES

Configures whether to install samples for Oracle Tuxedo Message Queue products.

Set INSTALL_SAMPLES=true to install samples for Oracle Tuxedo Message Queue products.

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.

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 Message Queue 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 4-1 applies to a silent installation of Oracle Tuxedo Message Queue 12*c* Release 2 (12.1.3).

Listing 4-1 UNIX response File

```
##############################################################################
## Copyright (c) 1999, 2014 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="/nfs/ucfhomes/beadev/OTMQ12cR2/installers/Disk1/stage/produ
cts.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_2"

#-------------------------------------------------------------------------
#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="OraHome2"

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

```
#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 = {"otmq.top","12.1.3.0.0"}

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

DEINSTALL_LIST={"otmq.top","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 = {"otmq.top","12.1.3.0.0"}
#-------------------------------------------------------------------------
TOPLEVEL_COMPONENT={"otmq.top","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  : otmq.top
#-------------------------------------------------------------------------

SELECTED_LANGUAGES={"en"}

#-------------------------------------------------------------------------
#Name       : COMPONENT_LANGUAGES
#Datatype   : StringList
#Description: Languages supported by this component.List of supported
languages : {"en","ja"}
#Component  : otmq.top
#-------------------------------------------------------------------------

COMPONENT_LANGUAGES={"en"}

#-------------------------------------------------------------------------
#Name       : INSTALL_TYPE
#Datatype   : String
#Description: Installation type of the component.
#Component  : otmq.top
#-------------------------------------------------------------------------

INSTALL_TYPE="Standalone installation"
```

```
#-------------------------------------------------------------------------
#Name       : TLISTEN_PORT
#Datatype   : String
#Description: This variable is for configure tlisten port
#Component  : tuxedoServerBase
#-------------------------------------------------------------------------

TLISTEN_PORT="3050"

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

CONFIG_TLISTEN=true

#-------------------------------------------------------------------------
#Name       : TLISTEN_PASSWORD
#Datatype   : String
#Description: This variable is for configure tlisten password
#Component  : tuxedoServerBase
#-------------------------------------------------------------------------

TLISTEN_PASSWORD=<Value Unspecified>

#-------------------------------------------------------------------------
#Name       : SSL_PARAMETERS
#Datatype   : StringList
#Description: This variable is for configure SSL Parameters Security
#Principal (Name, Location, Password)
#Component  : tuxedoServerBase
#-------------------------------------------------------------------------
```

```
SSL_PARAMETERS=<Value Unspecified>

#-------------------------------------------------------------------------
#Name       : MIN_CRYPT_BITS_CHOOSE
#Datatype   : Number
#Description: This variable is for configure Min Encryption Bits,
#for LLE:
#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.
#
#for SSL:
#0 means max encryption bit set to 112 ,
#1 means max encryption bit set to 128 ,
#2 means max 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,
#
```

```
#for LLE:
#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.
#
#for SSL:
#0 means max encryption bit set to 112 ,
#1 means max encryption bit set to 128 ,
#2 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  : tuxedoClientCore
#---------------------------------------------------------------------------

LDAP_SUPPORT_SSL=true

#---------------------------------------------------------------------------
#Name       : LDAP_FILTER_FILE
#Datatype   : String
```

```
#Description: This variable is for configure LDAP filter file
#Component   : tuxedoClientCore
#---------------------------------------------------------------------------

LDAP_FILTER_FILE="/scratch/beadev/OraHome_2/otmq12.1.3.0.0/udataobj/securi
ty/bea_ldap_filter.dat"

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

LDAP_CONFIG={"1","1","1"}
```

What Do I Do Next?

To configure your Oracle Tuxedo Message Queue software, verify that your software is installed correctly, or to uninstall Oracle Tuxedo Message Queue software, see “Post-Installation”.

CHAPTER 4

Post-Installation

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

- Running Samples to Verify Your Installation
- Uninstalling Oracle Tuxedo Message Queue
- Reinstalling Oracle Tuxedo Message Queue

Running Samples to Verify Your Installation

One of the ways to verify that your Oracle Tuxedo Message Queue software is installed correctly is to run one or more of the sample applications included with the installation. The `simpqapp` 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 Message Queue software is installed.

The `simpqapp` sample consists of a queue server `TuxMsgQ` offering the queue service, and three clients: `enqclt`, `deqclt`, and `filter`.

The `enqclt` client attaches the queue `QSPACE.QUEUE1` and enqueues one message into the queue `QSPACE.QUEUE2`.

The `deqclt` client attaches the queue `QSPACE.QUEUE2` and dequeues the message from the `enqclt` client.

The `filter` client demo illustrates how to use the simple filter and complex filter to dequeue messages from the queue.

QSPACE is the queue space name and is advertised as a service by queue server TuxMsgQ. QUEUE1 and QUEUE2 are the queues in queue space QSPACE.

Running simpqapp

On a UNIX system, it can be found in the directory $TUXDIR/samples/OTMQ/simpqapp.

Note: Samples are not installed unless they are selected to be installed during Oracle Tuxedo Message Queue installation. The procedure presented in the following section is also provided in the README.txt file in the simpqapp directory.

To configure and run the sample on a UNIX system, do the following steps:

1. Log in to the target machine
2. Change to the directory where the Oracle Tuxedo Message Queue is installed.

   ```
   cd $TUXDIR
   ./tux.env
   ```

3. Modify the environment variables IPCKEY or QIPCKEY in samples/OTMQ/simpqapp/setenv.sh if there is a conflict with other application.
4. Set the environment

   ```
   cd $TUXDIR/samples/OTMQ/simpqapp
   . ./setenv.sh
   ```

5. Compile and run simpqapp

   ```
   sh  runme.sh
   ```

This command is a script that builds the client, create the QUEUE and transaction log, boot the application, run the client, and then remove everything. Once this script has been run successfully, it may be instructive to run each command individually outside of the script. If the script does not report an error, it means success.

Uninstalling Oracle Tuxedo Message Queue

Uninstalling Oracle Tuxedo Message Queue12*c* Release 2 (12.1.3) does not remove the Oracle Home directory associated with the installation but does remove all Oracle Tuxedo Message Queue 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 Message Queue

When you start the Oracle Tuxedo Message Queue Installation program on system that already has a copy of Oracle Tuxedo Message Queue installed, the installation program detects the existing Oracle Tuxedo Message Queue home directory and displays a warning that the selected `ORACLE_HOME` already installed Oracle Tuxedo Message Queue. If you want to continue the installation, the components which not exist in the selected `ORACLE_HOME` are installed. Existing Oracle Tuxedo Message Queue component is not reinstalled by default.

APPENDIX A

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

Table A-1 lists Oracle Tuxedo Message Queue 12*c* Release 2 (12.1.3) supported Oracle Tuxedo 12*c* Release 2 (12.1.3) platforms. For more information, see "Oracle Tuxedo 12c Release 2 (12.1.3) Platform Data Sheets" in *Installing the Oracle Tuxedo System*.

Table A-1 Oracle OTMQ 12*c* Release 2 (12.1.3) Supported Tuxedo Platforms

Platform	GA Port/ Post-GA Port & Certification	Release Date	OS EOL Date
HP-UX 11i v3 (64-bit) on Itanium **Note:** For all OTMQ install sets and components, 1 GB is the minimum disk space requirement for installation on HP-UX 11i v3 (64-bit) on Itanium systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	GA	2014.12	TBD
IBM AIX 6.1 (32-bit) on IBM PowerPC **Note:** For all OTMQ install sets and components, 449,800 KB is the minimum disk space requirement for installation on IBM AIX 6.1 (32-bit) on IBM PowerPC systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	GA	2014.12	TBD
IBM AIX 6.1 (64-bit) on IBM PowerPC **Note:** For all OTMQ install sets and components, 449,800 KB is the minimum disk space requirement for installation on IBM AIX 6.1 (64-bit) on IBM PowerPC systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	Post-GA	2015.07	TBD

Table A-1 Oracle OTMQ 12*c* Release 2 (12.1.3) Supported Tuxedo Platforms

Platform	GA Port/ Post-GA Port & Certification	Release Date	OS EOL Date
IBM AIX 7.1 (64-bit) on IBM PowerPC **Note:** For all OTMQ install sets and components, 449,800 KB is the minimum disk space requirement for installation on IBM AIX 7.1 (64-bit) on IBM PowerPC systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	Post-GA	2015.08	TBD
Oracle Linux 5.6 (32-bit) on x86-64 **Note:** For all OTMQ install sets and components, 265 MB is the minimum disk space requirement for installation on Oracle Linux 5.6 (32-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	GA	2014.12	TBD
Oracle Linux 5.6 (64-bit) on x86-64 **Note:** For all OTMQ install sets and components, 193,464 KB is the minimum disk space requirement for installation on Oracle Linux 5.6 (64-bit) on x86-64 systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	GA	2014.12	TBD

Table A-1 Oracle OTMQ 12*c* Release 2 (12.1.3) Supported Tuxedo Platforms

Platform	GA Port/ Post-GA Port & Certification	Release Date	OS EOL Date
Oracle Linux 6.0 (64-bit) on x86-64 **Note:** For all OTMQ install sets and components, 193,464 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 set. The temporary storage space has the same requirement also.	Post-GA	2015.03	TBD
Oracle Linux 5.6 (64-bit) on Exalogic 2.0 **Note:** For all OTMQ install sets and components, 193,464 KB is the minimum disk space requirement for installation on Oracle Linux 5.6 (64-bit) on Exalogic 2.0 systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	Post-GA	2015.03	TBD
Oracle Linux 5.8 (64-bit) on Exalogic 2.0 **Note:** For all OTMQ install sets and components, 193,464 KB is the minimum disk space requirement for installation on Oracle Linux 5.8 (64-bit) on Exalogic 2.0 systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	Post-GA	2015.03	TBD

Table A-1 Oracle OTMQ 12*c* Release 2 (12.1.3) Supported Tuxedo Platforms

Platform	GA Port/ Post-GA Port & Certification	Release Date	OS EOL Date
Red Hat Enterprise Linux 5.6 (64-bit) on x86-64 **Note:** For all OTMQ install sets and components, 193,464 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 set. The temporary storage space has the same requirement also.	Post-GA	2015.03	TBD
Red Hat Enterprise Linux 6.0 (64-bit) on x86-64 **Note:** For all OTMQ install sets and components, 193,464 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 set. The temporary storage space has the same requirement also.	Post-GA	2015.03	TBD
Novell SUSE Linux Enterprise Server 10 (32-bit) x86-64 **Note:** For all OTMQ install sets and components, 193,464 KB is the minimum disk space requirement for installation on Novell SUSE Linux Enterprise Server 10 (32-bit) x86-64 systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	Post-GA	2015.03	TBD

Table A-1 Oracle OTMQ 12*c* Release 2 (12.1.3) Supported Tuxedo Platforms

Platform	GA Port/ Post-GA Port & Certification	Release Date	OS EOL Date
Oracle Solaris 10 (64-bit) on SPARC **Note:** For all OTMQ install sets and components, 447,686 KB is the minimum disk space requirement for installation on Oracle Solaris 10 (64-bit) on SPARC systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	GA	2014.12	TBD
Oracle Solaris 11 (64-bit) on SPARC **Note:** For all OTMQ install sets and components, 447,686 KB is the minimum disk space requirement for installation on Oracle Solaris 11 (64-bit) on SPARC systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also.	Post-GA	2015.03	TBD
Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2013 **Note:** For all OTMQ install sets and components, 232,560 KB is the minimum disk space requirement for installation on Microsoft Windows Server 2008 R2 (64-bit) on x86-64 with MS Visual Studio 2013 systems. This requirement assumes installation of the default components for the selected install set. The temporary storage space has the same requirement also. **Note:** Only supports OTMQ standalone installation mode.	Post-GA	2017.01	TBD

Note: For each platform's hardware and software requirements, refer to this platform's data sheets on "Oracle Tuxedo 12c Release 2 (12.1.3) Platform Data Sheets" in *Installing the Oracle Tuxedo System*.

APPENDIX A

Migrating from OMQ to OTMQ 12c

This chapter includes the following topics:

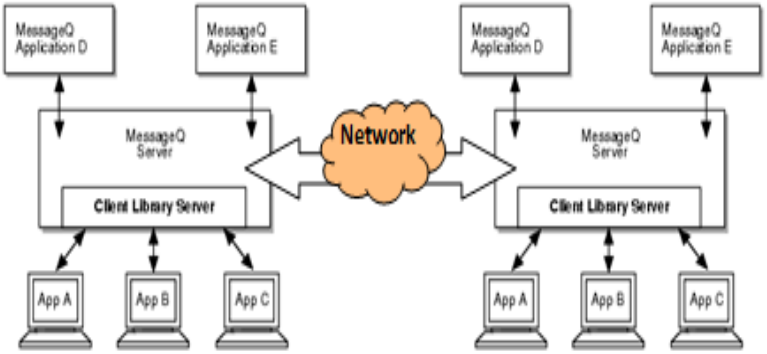
Overview

Oracle Tuxedo Message Queue (OTMQ) component improves and enhances Oracle Tuxedo queuing services. Besides the existing queuing features of the Oracle Tuxedo /Q component, OTMQ also provides richer queuing features (such as Reliable Message Delivery, Synchronous/Asynchronous Messaging, Publish/Subscribe, Message Filtering, Dynamic Queue Alias (Naming), Journal, etc.). Because OTMQ is implemented based on Oracle Tuxedo infrastructure, it can provide support for transactions, security, scalability and HA.

OMQ Architecture

Oracle MessageQ (OMQ) message queuing bus forms the data highway used to transfer messages between applications by creating a logical interconnection of message queues in a networked environment as shown in Figure B-1.

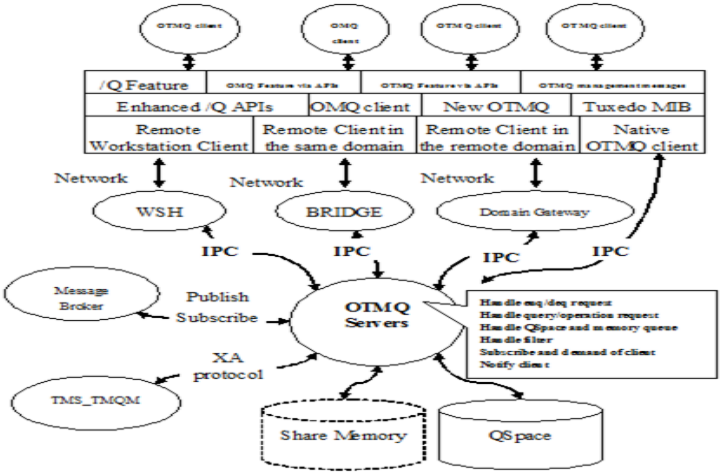
Figure B-1 OMQ Architecture



OTMQ Architecture

Figure B-2 shows the system structure of an application using OTMQ.

Figure B-2 OTMQ Architecture



Feature Compatibility

Table 2 provides a summary of key OMQ and OTMQ compatibility features.

Table B-2 Feature Compatibility

Key Features		OMQ	OTMQ	OTMQ
Communication Model	P2P	yes	yes	
	Pub/Sub	yes	yes	
	Remote client	yes (thru CLS)t	yes (thru WSH)	
Basic Messaging	synchronous (WF)	yes	yes	
	asynchronous (AK)	yes	yes	
Recoverable Messaging (DIPs and UMAs)		yes	yes	OTMQ does not support `UMA=DISCL`
Persistent/non-persistent message		yes	yes	
Message-Based Services		yes	yes	OTMQ supports parts of MBS, also provides MIB operation to check status or manage queing resources
Naming		yes	yes	
Message Filter		yes	yes	
Journal		yes (Journal files)	yes (Journal files)	
WS SAF		yes	yes	
PAMS API		yes	yes	OMQ clients using PAMS APIs can be recompile and relink with OTMQ tools and libraries, and run with configured OTMQ servers.

Deployment

OTMQ applications can be deployed centrally or decentrally. By taking advantage of flexible Oracle Tuxedo deployment models, an OTMQ application configures and deploy resources using `UBBCONFIG` and `DOMCONFIG` files.

Migration Tasks

Application Migration

- Application configuration files re-generation
- Application resources (Q space and queues) reconstruction
- Application code changes: API migration

Configuration File Migration

Configuration file migration involves the following OTMQ components:

- `UBBCONFIG`/`DMCONFIG` (configuration files)
- `tmloadcf` (administration command)

OMQ application resources, OMQ groups and queues should be migrated to OTMQ QSpace and queues accordingly.

OTMQ provides a tool ("`dmqclconv`"), to migrate the `omq.init` file into OTMQ configuration files (`UBBCONFIG`/`DMCONFIG`).

For more information, see Oracle Tuxedo Message Queue Command Reference.

OTMQ Commands

OTMQ uses the "`tmqadmin`" command to create/manage queuing resources (Qspace and queues). OTMQ applications should also use the following commands to build clients and servers:

- `buildqclient`

  Used to construct an OTMQ client module

- `buildqserver`

  Used to construct an OTMQ load module which can run as an Oracle Tuxedo application server.

For more information, see the Oracle Tuxedo Message Queue Command Reference.

OTMQ System Services

OTMQ provides following system services:

- `TuxMsgQ`: Message Queue Manager.
- `TuxMQFWD`: Message Queue Off-line Trade Driver.
- `TMQ_NA`: Message Queue Naming Server.
- `TMS_TMQM`: TMS server for OTMQ resource manager.
- `TMQ_EVT`: TMQ Event reporting process.
- `TMQFORWARDPLUS`: Message Forwarding server.
- `TuxMsgQLD`: TMQ Link Driver server.

These services can be configured and deployed in the `UBBCONFIG` file based on application requirements. For more information, see the Oracle Tuxedo Message Queue UBB Server Reference.

Limitation(s)

- `OMQ UMA_SAF`

  For OMQ application compatibility, OTMQ supports `UMA_SAF` in a revised edition. For an OMQ client, if `UMA_SAF` for a recoverable message is applied, the message is stored in the local `DQF` . If OMQ is unable to write to the local journal disk file, UMA is used. OTMQ follows this methodology; however, OTMQ stores the message in the local `SAF` to ensure that the message is recoverable. If the action fails, then UMA is used.

Examples

- OMQ Configuration File-Setup.init
- Converter Results

OMQ Configuration File-Setup.init

Listing 1 provides an example OMQ configuration file-Setup.init

Listing B-1 OMQ Configure File-Setup.init

```
%VERSION 4.0

%PROFILE

ACCEPT_KILL_COMMAND     YES      ! accept kill commands from DMQ Monitor
!ACCEPT_KILL_COMMAND      NO      ! accept kill commands from DMQ Monitor
ENABLE_XGROUP           YES      ! allow cross-group communications
!ENABLE_XGROUP           NO      ! allow cross-group communications
!XGROUP_VERIFY           YES       ! verify cross group links
XGROUP_VERIFY           NO       ! do not verify cross group links
FIRST_TEMP_QUEUE        200      ! minimum is 101,  maximum is 850
GROUP_MAX_USER_QUEUE   999
XGROUP_TABLE_SIZE       20       ! size of the group link table
GROUP_MAX_MESSAGE_SIZE  4194304  ! largest message size permitted in this
group
ATTACH_TMO              600      ! timeout in seconds for locates
!RCV_MSG_QUOTA_METHOD       MIN
ENABLE_MRS              YES      ! start recoverable messaging servers
!ENABLE_MRS               NO      ! start recoverable messaging servers
ENABLE_JRN             YES      ! enable PCJ journaling when MRS is enabled
!ENABLE_JRN             NO      ! enable PCJ journaling when MRS is enabled
ENABLE_SBS            YES     ! start selective broadcast and AVAIL/UNAVAIL

DEFAULT_NAMESPACE_PATH   /home/beadev/BMQ5064rp/qa/mq/ddtf ! the default
namespace path for naming agent
```

```
%EOS

%MRS
MRS_JOURNAL_PATH           /home/beadev/BMQ5064rp/qa/mq/ddtf  ! recovery
journal path
MRS_POST_CONFIRMATION_PATH /home/beadev/BMQ5064rp/qa/mq/ddtf
MRS_DEAD_LETTER_PATH      /home/beadev/BMQ5064rp/qa/mq/ddtf  ! dead-letter
journal path
MAX_CACHE_BYTES            4194304      ! maximum number of unwritten bytes
%EOS

%CLS
47100  TCPIP 1
!47100 TCPIP 1
%EOS

%XGROUP

!Group  Group  Node/  Init Thresh- Buffer Recon-   Window   Trans- End-
!Name   Number Host        old     Pool   nect   Delay Size port   point
!G1       1    reboo    Y    2000000 .      30     10   25000 TCPIP  47001
G1      1    localhost   Y    2000000 .      30     10   25000 TCPIP  47001
G3      3    titan    Y    2000000 .      30     10   25000 TCPIP  47001
!G1     1    gemini   Y    2000000 .      30     10   25000 TCPIP  47001
!* For XGROUP DGA test
!G1       1    reboo    Y    2000000 .      10     30   25000 TCPIP  47001
!G2       2    reboo    Y    2000000 .      20     40   35000 TCPIP  47002
!*
```

```
G2      2     localhost    Y    2000000 .      30     10   25000 TCPIP  47002
!G2     2     localhost  Y    2000000  .      30     10   25000 TCPIP  47002
!G2      2     titan    Y    2000000 .      30     10   25000 TCPIP  47002
!G2     2     gemini   Y    2000000 .      30     10   25000 TCPIP  47002
%EOS

!%route
!3 2
!%EOS
%QCT

!Queue    Queue  Byte    Msg    Quota  UCB  Queue Owner Conf  Perm    Name
!Name     Number Quota   Quota  Enable Send Type  Queue Style Active Scope
Security

TEMPLATE       0 262144    .    .       .    P          0 II    N      L     N
NON_QUOTA      1       .    .   None    .    P          0 II    N      L     N
QUOTA          2       .    5   Msg     .    P          0 II    N      L     N
MRQ            3       .  1000  Msg     .    M          0 II    Y      L     N
NR_MRQ        4       .    .   None    .    M          0 II    Y      L     N
R_MRQ          5       .    .   None    .    M          0 II    Y      L     N
G_IN_QCT       6       .    .   None    .    P          0 II    Y      G     N
New_MRQ       7       .    .   None    .    M          0 II    Y      L     N
!QUOTA2       8       5    4   None    .    P          0 II    N      L     N
!QUOTA2       8       5    4   Msg     .    P          0 II    N      L     N
QUOTA2        8       5    4   Byte    .    P          0 II    N      L     N
!QUOTA2       8       5    4   All     .    P          0 II    N      L     N
byte_quota_q  9   64000    .   Byte    .    P          0 II    N      L     N
```

```
msg_quota_q  10      .   5   Msg    .    P          0 II    N      L     N
all_quota_q  11  64000   5   All    .    P          0 II    N      L     N
no_quota_q   12  64000   5   None   .    P          0 II    N      L     N
ii_queue     13  64000   5   None   .    P          0 II    N      L     N
ei_queue     14  64000   5   None   .    P          0 EI    N      L     N
eo_queue     15  64000   5   None   .    P          0 EO    N      L     N
QUEUE1      16      .   .   None   .    P         0 II    N      L     N
QUEUE2       17      .   .   None   .    P          0 II    Y      L     N
QUEUE3      18      .   .   None   .    S         16 II    N      L     N
BIND_PQ1     20      .   .   None   .    P          0 II    N      L     N
BIND_PQ2     21      .   .   None   .    P          0 II    N      L     N
BIND_SQ      22      .   .   None   .    S         20 II    N      L     N
spare1      90      .   .   None   .    P          0 II    N      L     N
*
EXAMPLE_Q_1          193   64000  100   .    .    .    .    .    N    L    N
EXAMPLE_Q_2          194   64000  100   .    .    .    .    .    N    L    N
IVP_unowned_sq       195  250000  100   .    .    S    0    .    N    L    N
%EOS

*
%GNT

! Name          Address Scope
AVAIL_TARGET         1.1    L
INTEROP_TARGET       1.1     L
REMOTE_NON_QUOTA     1.1    L
REMOTE_QUOTA          1.2     L
```

```
REMOTE_MRQ            1.3     L
LOCAL_NR_RCVR          0.4      L
LOCAL_R_RCVR           0.5      L
G1_NR_RCVR            1.4     L
G1_R_RCVR             1.5     L
G2_NR_RCVR            2.4     L
G2_R_RCVR             2.5     L
MY_GRP_Q_1            0.0     L
MY_GRP_Q_2            0.0     L
MY_BUS_Q_1            0.0     G
MY_BUS_Q_2            0.0     G
MY_Q_1                0.0     L
MY_Q_2                0.0     L
IVP_test_bindq1                               0.0         L
IVP_test_bindq2                               0.0         L
IVP_private_MOT1                              4999        L
IVP_universal_MOT1                            5001        L
%EOS

%NAM
!NA_GROUP 0
NA_GROUP 1
!NA_GROUP 2
%EOS

%END
```

Converter Results

After running the OTMQ `dmqclconv` command as follows:

`dmqclconv -g 1 -f dmq.ini`, the converted results are shown in Listing 2.

Listing B-2 Converted Results

```
UBB:
*RESOURCES
MASTER              SITE1
IPCKEY              @IPCKEY@
MAXACCESSERS        25
MAXSERVERS          30
MODEL               SHM
MAXSERVICES         102

*MACHINES
"localhost"     LMID="SITE1"
        MAXACCESSERS=100
        MAXWSCLIENTS=50
        TUXCONFIG="@TUXCONFIG@"
        TUXDIR="@TUXDIR@"
        APPDIR="@APPDIR@"
        TLOGDEVICE="@TLOGDEVICE@"
        TLOGNAME="TLOG"

*GROUPS
"OTMQ1"     LMID="SITE1"        GRPNO=1
            OPENINFO="TUXEDO/TMQM:@QMCONFIG@:1"
```

```
                TMSNAME="TMS_TMQM"
                TMSCOUNT=2

"OTMQ2"         LMID="SITE1"        GRPNO=2
                OPENINFO="TUXEDO/TMQM:@QMCONFIG@:2"
                TMSNAME="TMS_TMQM"
                TMSCOUNT=2

"GRPDM"         LMID="SITE1"        GRPNO=3
"GRPGW"         LMID="SITE1"        GRPNO=4
"GRPEVT"        LMID="SITE1"        GRPNO=5
"WSLGROUP"      LMID="SITE1"        GRPNO=6

*SERVERS
"TuxMsgQ"       SRVGRP="OTMQ1"      SRVID=1
                CLOPT="-s 1:TuxMsgQ -- -i 10"
"TuxMQFWD"      SRVGRP="OTMQ1"      SRVID=2
                CLOPT="-- -i 2"
"TMQ_NA"        SRVGRP="OTMQ1"      SRVID=3
                CLOPT="--  -g 1"
"TuxMsgQ"       SRVGRP="OTMQ2"      SRVID=4
                CLOPT="-s 2:TuxMsgQ -- -i 10"
"TuxMQFWD"      SRVGRP="OTMQ2"      SRVID=5
                CLOPT="-- -i 2"
"TMQ_NA"        SRVGRP="OTMQ2"      SRVID=6
                CLOPT="--  -g 1"
"DMADM"         SRVGRP="GRPDM"      SRVID=7
```

```
                CLOPT="-A"
"GWADM"         SRVGRP="GRPGW"      SRVID=8
                CLOPT="-A"
"GWTDOMAIN"     SRVGRP="GRPGW"      SRVID=9
                CLOPT="-A"
"TMQEVT"        SRVGRP="GRPEVT"     SRVID=10
                CLOPT="-A"
"WSL"           SRVGRP="WSLGROUP"       SRVID=12
                CLOPT="-A -- -n //localhost:47100 -M 1"

*SERVICES
"TuxMQATH1"         BLOCKTIME=60
"1_LOCAL_NA"        BLOCKTIME=60
"TuxMQATH2"         BLOCKTIME=60
"2_LOCAL_NA"        BLOCKTIME=60

*SERVICES

DUBB:
*DM_RESOURCES
*DM_LOCAL
"DOM1"      GWGRP="GRPGW"
            ACCESSPOINTID="DOM1"
            CONNECTION_POLICY=ON_STARTUP
            RETRY_INTERVAL=30
##gnumber is 1,gname is G1,node is localhost,port is 47001
```

```
"DOM2"      GWGRP="GRPGW"
            ACCESSPOINTID="DOM2"
            CONNECTION_POLICY=ON_STARTUP
            RETRY_INTERVAL=30
##gnumber is 2,gname is G2,node is localhost,port is 47002

*DM_REMOTE
"DOM3"      ACCESSPOINTID="DOM3"
##gnumber is 3,gname is G3,node is titan,port is 47001

*DM_TDOMAIN
"DOM1"      NWADDR="//localhost:47001"
##gnumber is 1,gname is G1

"DOM2"      NWADDR="//localhost:47002"
##gnumber is 2,gname is G2

"DOM3"      NWADDR="//titan:47001"
##gnumber is 3,gname is G3

*DM_EXPORT
"1"             LACCESSPOINT="DOM1"
"GLOBAL_NA"     LACCESSPOINT="DOM1"
"2"             LACCESSPOINT="DOM2"

*DM_IMPORT
"3"             RACCESSPOINT="DOM3"
```

```
                TYPE="QSPACE"

CRQUE:
crdl @QMCONFIG@ 0 40000
echo

qspacecreate -n 8192B -Q 100 -T 200 -N @QALIAS@
1
@QIPCKEY1@
16384
200
100
100
100
errque
y
16
y
y

qopen 1
qcreate -t PQ -a N -c II -f
0
fifo
none
0
30
```

```
100%
0%

qcreate -t PQ -a N -c II -f
1
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
2
fifo
none
0
30
100%
0%

qcreate -t MRQ -a Y -c II -f
3
fifo
none
0
30
```

```
100%
0%

qcreate -t MRQ -a Y -c II -f
4
fifo
none
0
30
100%
0%

qcreate -t MRQ -a Y -c II -f
5
fifo
none
0
30
100%
0%

qcreate -t PQ -a Y -c II -f
6
fifo
none
0
30
```

```
100%
0%

qcreate -t MRQ -a Y -c II -f
7
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
8
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
9
fifo
none
0
30
```

```
100%
0%

qcreate -t PQ -a N -c II -f
10
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
11
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
12
fifo
none
0
30
```

```
100%
0%

qcreate -t PQ -a N -c II -f
13
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c EI -f
14
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c EO -f
15
fifo
none
0
30
```

```
100%
0%

qcreate -t PQ -a N -c II -f
16
fifo
none
0
30
100%
0%

qcreate -t PQ -a Y -c II -f
17
fifo
none
0
30
100%
0%

qcreate -t SQ -a N -c II -f -o 16
18
fifo
none
0
30
```

```
100%
0%

qcreate -t PQ -a N -c II -f
20
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
21
fifo
none
0
30
100%
0%

qcreate -t SQ -a N -c II -f -o 20
22
fifo
none
0
30
```

```
100%
0%

qcreate -t PQ -a N -c II -f
90
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c EO -f
193
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c EO -f
194
fifo
none
0
30
```

```
100%
0%

qcreate -t SQ -a N -c EO -f -o 0
195
fifo
none
0
30
100%
0%

qspacecreate -n 8192B -Q 100 -T 200 -N @QALIAS@
2
@QIPCKEY2@
16384
200
100
100
100
errque
y
16
y
y

qopen 2
```

```
qcreate -t PQ -a N -c II -f
0
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
1
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
2
fifo
none
0
30
100%
0%
```

```
qcreate -t MRQ -a Y -c II -f
3
fifo
none
0
30
100%
0%

qcreate -t MRQ -a Y -c II -f
4
fifo
none
0
30
100%
0%

qcreate -t MRQ -a Y -c II -f
5
fifo
none
0
30
100%
0%
```

```
qcreate -t PQ -a Y -c II -f
6
fifo
none
0
30
100%
0%

qcreate -t MRQ -a Y -c II -f
7
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
8
fifo
none
0
30
100%
0%
```

```
qcreate -t PQ -a N -c II -f
9
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
10
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
11
fifo
none
0
30
100%
0%
```

```
qcreate -t PQ -a N -c II -f
12
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
13
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c EI -f
14
fifo
none
0
30
100%
0%
```

```
qcreate -t PQ -a N -c EO -f
15
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
16
fifo
none
0
30
100%
0%

qcreate -t PQ -a Y -c II -f
17
fifo
none
0
30
100%
0%
```

```
qcreate -t SQ -a N -c II -f -o 16
18
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
20
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
21
fifo
none
0
30
100%
0%
```

```
qcreate -t SQ -a N -c II -f -o 20
22
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c II -f
90
fifo
none
0
30
100%
0%

qcreate -t PQ -a N -c EO -f
193
fifo
none
0
30
100%
0%
```

```
qcreate -t PQ -a N -c EO -f
194
fifo
none
0
30
100%
0%

qcreate -t SQ -a N -c EO -f -o 0
195
fifo
none
0
30
100%
0%

q
```

Listing 3 shows a QALIAS example

Listing B-3 QALIAS

```
TEMPLATE            0.0     L
NON_QUOTA           0.1     L
QUOTA               0.2     L
```

```
MRQ                 0.3     L
NR_MRQ              0.4     L
R_MRQ               0.5     L
G_IN_QCT            0.6     G
New_MRQ             0.7     L
QUOTA2              0.8     L
byte_quota_q        0.9     L
msg_quota_q         0.10    L
all_quota_q         0.11    L
no_quota_q          0.12    L
ii_queue            0.13    L
ei_queue            0.14    L
eo_queue            0.15    L
QUEUE1              0.16    L
QUEUE2              0.17    L
QUEUE3              0.18    L
BIND_PQ1            0.20    L
BIND_PQ2            0.21    L
BIND_SQ             0.22    L
spare1              0.90    L
EXAMPLE_Q_1         0.193   L
EXAMPLE_Q_2         0.194   L
IVP_unowned_sq      0.195   L
AVAIL_TARGET        1.1     L
INTEROP_TARGET      1.1     L
REMOTE_NON_QUOTA    1.1     L
REMOTE_QUOTA        1.2     L
```

```
REMOTE_MRQ           1.3     L
LOCAL_NR_RCVR        0.4     L
LOCAL_R_RCVR         0.5     L
G1_NR_RCVR           1.4     L
G1_R_RCVR            1.5     L
G2_NR_RCVR           2.4     L
G2_R_RCVR            2.5     L
MY_GRP_Q_1           0.0     L
MY_GRP_Q_2           0.0     L
MY_BUS_Q_1           0.0     G
MY_BUS_Q_2           0.0     G
MY_Q_1               0.0     L
MY_Q_2               0.0     L
IVP_test_bindq1      0.0     L
IVP_test_bindq2      0.0     L
IVP_private_MOT1     4999    L
IVP_universal_MOT1   5001    L
```

Listing 4 shows a CRTLOG example.

Listing B-4 CRTLOG

```
echo
crdl -b 200 -z @TLOGDEVICE@
crlog -m SITE1
q
```