

Oracle® Tuxedo Mainframe Adapter for TCP

Installation Guide

12c Release 2 (12.1.3)

August 2014

ORACLE®

Copyright © 2007, 2014, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Preparing to Install the Oracle Tuxedo Mainframe Adapter for TCP

Oracle Universal Installer (OUI)	1-1
Installing Oracle Tuxedo Mainframe Adapter for TCP 12c Release 2 (12.1.3) with Oracle Home.	1-2
Oracle Universal Installer Inventory	1-3
Customizing and Creating Response Files	1-9
Modes of Installation.	1-23
Installation Media	1-23
Special Instructions for UNIX Users.	1-23
Deinstalling Oracle Products.	1-25
About Oracle Universal Installer Log Files.	1-27
Oracle Internationalization	1-27
Oracle Tuxedo Mainframe Adapter for TCP Components.	1-28
Hardware and Software Prerequisites.	1-28
System Requirements	1-28
Temporary Storage Space Requirements.	1-28
Oracle Installation Program	1-29
Installation Methods	1-29
Oracle Home Directory.	1-30
Choosing an Oracle Home Directory	1-30
Multiple Oracle Homes	1-30
Installation Road Map.	1-31

Installing Oracle Tuxedo Mainframe Adapter for TCP Using GUI-Mode Installation

What Is GUI-Mode Installation?	2-1
--	-----

Starting GUI-Mode Installation on a UNIX System	2-2
Running GUI-Mode Installation.	2-2
Assigning File Ownership on a UNIX System.	2-4

Installing Oracle Tuxedo Mainframe Adapter for TCP Using Silent Installation

What Is Silent Installation?	3-1
Using Silent Installation: Main Steps	3-1
Creating a Response File	3-1
Installing with a Response File	3-3
Response File Examples	3-5

Installing Oracle Tuxedo Mainframe Adapter for TCP (IMS)

Preparing to Install	4-1
Installing TMA TCP for IMS	4-1
Task 1: Allocate z/OS Datasets	4-2
Task 2: Unload the Distribution Files	4-6
Task 3: Link-Edit TMA TCP for IMS	4-6
Task 4: Define TMA TCP for IMS to MVS	4-8
Task 5: Create an TMA TCP for IMS Configuration File.	4-8
Task 6: Create JCL to Run TMA TCP for IMS	4-11
Task 7: Start Oracle TMA TCP for IMS	4-12
Task 8: Test the Installation.	4-13
Distribution Libraries and Executables.	4-13

Installing Oracle Tuxedo Mainframe Adapter for TCP (CICS)

Preparing for Installation	5-1
Task 1: Verify System Requirements.	5-1
Task 2: Check Pre-Installation Considerations	5-2

Installing TMA TCP for CICS	5-2
Task 1: Allocate Temporary Data Sets	5-2
Task 2: Transfer the Installation JCL.	5-3
Task 3: Create the PDS Libraries	5-4
Task 4: Upload the Distribution Files	5-6
Task 5: Define the CICS VSAM Files.	5-11
Task 6: Verify Contents of PROCS	5-12
Task 7: Change Resource Names (Optional).	5-13
Task 8: Define the CICS Table Entries	5-13
Task 9: Build TMA TCP for CICS Executables	5-13
Task 10: Verify the CICS Setup.	5-16
Task 11: Assemble and Link the CICS DCT.	5-16
Task 12: Modify the CICS Startup JCL.	5-17
List of Distribution Files and Members	5-20
BEATCPC.DIST.CONTROL	5-20
BEATCPC.DIST.JCL	5-20
BEATCPC.DIST.SOURCE	5-21
BEATCPC.DIST.INCLUDE	5-21
BEATCPC.DIST.PLKED	5-22
BEATCPC.DIST.OBJECT	5-22

Post Installation Tasks for Oracle Tuxedo Mainframe Adapter for TCP

Uninstalling Oracle Tuxedo Mainframe Adapter for TCP	6-1
Uninstalling Oracle Tuxedo Mainframe Adapter for TCP in GUI-Mode	6-1
Uninstalling Oracle Tuxedo Mainframe Adapter for TCP in Silent Mode	6-2
Distribution Libraries and Executables.	6-3
Solaris	6-3

AIX	6-3
Linux	6-4
Contents of SOURCE(BEACFGSV).....	A-1
Changing Resource Names for SOURCE(BEACFGSV).....	A-2
Contents of CONTROL(CSDU)	A-3
Changing Resource Names for CONTROL(CSDU)	A-4

Oracle Tuxedo Mainframe Adapter for TCP Supported Platforms, Hardware Requirements, and Software Requirements

Oracle Tuxedo Mainframe Adapter for TCP Supported Platforms	B-1
Oracle Tuxedo Mainframe Adapter for TCP Hardware Requirements	B-2
Oracle Tuxedo Mainframe Adapter for TCP Software Requirements.....	B-2
Oracle Tuxedo Mainframe Adapter for TCP Supported Stacks	B-2
Oracle TMA TCP for IMS	B-3
Oracle TMA TCP for CICS	B-3

Preparing to Install the Oracle Tuxedo Mainframe Adapter for TCP

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

- [Oracle Universal Installer \(OUI\)](#)
- [Oracle Tuxedo Mainframe Adapter for TCP Components](#)
- [Hardware and Software Prerequisites](#)
- [Oracle Installation Program](#)
- [Oracle Home Directory](#)
- [Installation Road Map](#)

Oracle Universal Installer (OUI)

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

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

Installing Oracle Tuxedo Mainframe Adapter for TCP 12c Release 2 (12.1.3) with Oracle Home

When you install Oracle Tuxedo Mainframe Adapter for TCP 12c 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 Mainframe Adapter for TCP 12c Release 2 (12.1.3)

To install Oracle Tuxedo Mainframe Adapter for TCP 12c 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.

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

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_oracleInst.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.

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`

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

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

Replace the `<response_file_name>`

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

```
./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 Mainframe Adapter for TCP specific response file parameters, see [“Required Variables in Oracle Tuxedo Mainframe Adapter for TCP Response File” on page 3-2](#).

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” on page 1-3](#).

Specifying a Response File

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

```
On 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:

```
./runInstaller -help
```

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 on UNIX.
- In the inventory logs directory for hosts that already had an inventory.

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

Setting Response File Variables from the Command Line

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

For more information, see [“Response File Format” on page 1-11](#).

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>
```

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

On UNIX:

```
./runInstaller 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>
```

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.

Installation Media

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

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 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:

```
./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", "1.0.1.2.0" }
```

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 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 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 Mainframe Adapter for TCP Components

Oracle Tuxedo Mainframe Adapter for TCP 12c Release 2 (12.1.3) contains the following component:

- Oracle Tuxedo Mainframe Adapter for TCP

The component that contains the Oracle Tuxedo Mainframe Adapter for TCP software files.

Hardware and Software Prerequisites

The Oracle Tuxedo Mainframe Adapter for TCP must be installed into an Oracle Home which has already installed Oracle Tuxedo 12c Release 2 (12.1.3) server component, and Rolling Patch 003 or higher releases.

System Requirements

Oracle Tuxedo Mainframe Adapter for TCP 12c Release 2 (12.1.3) must be installed on a supported OS version for the given platform. Refer to [“Oracle Tuxedo Mainframe Adapter for TCP Supported Platforms”](#) for the supported platforms.

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 Mainframe Adapter for TCP on the target system. During the installation process, your temporary directory must contain sufficient space to accommodate the compressed Java Runtime Environment (JRE) bundled with the installer and an uncompressed copy of the JRE that is expanded into the temporary directory. The installation program moves the JRE from the temporary directory to the *Oracle Home directory* at the end of the installation process. For information about the Oracle Home directory, see [“Oracle Home Directory” on page 1-30](#).

By default, the installation program uses the temporary directories shown in [Table 1-1](#).

Table 1-1 Temporary Directories

Platform	Directory
UNIX	/tmp

To ensure there is adequate temporary space, you may want to allocate an alternate directory for use as a temporary directory for the installation. To do so, perform the appropriate step in [Table 1-2](#) and clean up files as more as possible in the temporary directory before starting the Oracle Installation program.

Table 1-2 Perform This Step

On This Platform...	Perform This Step...
UNIX	<p>Enter the following command at the shell prompt:</p> <pre>export IATEMPDIR=tmpdirname</pre> <p>Replace <code>tmpdirname</code> with the name of a temporary directory of your choice.</p>

Oracle Installation Program

The Oracle Tuxedo Mainframe Adapter for TCP 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 Mainframe Adapter for TCP software on UNIX systems.

Installation Methods

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

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

Canceling Installation

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

If you installed Oracle Tuxedo Mainframe Adapter for TCP 12c Release 2 (12.1.3) in a previous Tuxedo product directory, and you want to return to your original configuration, you must re-install your previous Oracle Tuxedo Mainframe Adapter for TCP version.

Oracle Home Directory

When you install Oracle Tuxedo Mainframe Adapter for TCP, 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 Mainframe Adapter for TCP 12c Release 2 (12.1.3), you are prompted to choose an existing Oracle Home directory, where Oracle Tuxedo Mainframe Adapter for TCP is installed.

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 Mainframe Adapter for TCP 12c Release 2 (12.1.3), see one of the following sections:

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

If you want to uninstall your Oracle Tuxedo Mainframe Adapter for TCP software, see [“Post Installation Tasks for Oracle Tuxedo Mainframe Adapter for TCP”](#) on page 6-1.

Installing Oracle Tuxedo Mainframe Adapter for TCP Using GUI-Mode Installation

The following sections describe how to install Oracle Tuxedo Mainframe Adapter for TCP using graphical user interface (GUI) mode installation on UNIX systems:

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

What Is GUI-Mode Installation?

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

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

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

The Oracle Tuxedo Mainframe Adapter for TCP installer program can be displayed in English or Japanese. You can set certain language-related environment variable on the target platform to specify the language displayed during the installation and the language-specific message catalog installed during the installation. If the language-related environment variable is not set or set to an unknown value, the installer program displays in English.

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 [“Hardware and Software Prerequisites”](#) on page 1-28.
2. Log in to the UNIX system as the Oracle Tuxedo Mainframe Adapter for TCP administrator.
3. Ensure that you have enough free space for the Oracle Tuxedo Mainframe Adapter for TCP installation.
4. Install Oracle Tuxedo Mainframe Adapter for TCP by downloading from the Oracle Web site:
 - a. Go to <http://www.oracle.com/technology/software/index.html> and download the Oracle Tuxedo Mainframe Adapter for TCP 12c Release 2 (12.1.3) installation file specific to your platform.
 - b. Go to the directory where you downloaded the installer, unzip the installer file, and then go to the `Disk1\install` directory, run the `runInstaller` to invoke the installation procedure.

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

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

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

Running GUI-Mode Installation

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

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
Welcome	Click Next to proceed with the installation. You may cancel the installation at any time by clicking Cancel.
Specify Home Details	<p>Name: Enter a name for the Oracle home. This name identifies the program group associated with a particular home and the Oracle services installed on this home. The Oracle home name must be between 1 to 127 characters long, and can include only alphanumeric characters and underscores.</p> <p>Path: 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 Oracle Tuxedo Mainframe Adapter for TCP is installed.</p> <p>If you enter an invalid path for Oracle Home directory, you cannot continue the installation.</p> <p>If the selected ORACLE_HOME is not empty and does not recorded in inventory, the installer will show a warning.</p> <p>If the selected ORACLE_HOME does not include Tuxedo Server component, you cannot continue the installation.</p>
Summary	<p>Click Install to start the installation.</p> <p>If previous version components are already installed, the installation program just upgrades the higher version sub-components in the same ORACLE_HOME by default. If you want to reinstall the same version sub-components, you can specify the value of DEPENDENCY_LIST. For example:</p> <pre data-bbox="501 1222 1040 1274">./runInstaller -force "DEPENDENCY_LIST={ "TMATCP:12.1.3.0.0" }</pre>
Install	When this panel appears, files of the selected installation type are being copied to the target installation directory.
End of Installation	Click Exit to exit the installation program.

Congratulations! Your installation of the Oracle Tuxedo Mainframe Adapter for TCP software is complete!

Assigning File Ownership on a UNIX System

If you installed the Oracle Tuxedo Mainframe Adapter for TCP on a UNIX system, Oracle strongly recommends that you create a separate user account for the Oracle Tuxedo administrator and grant the Oracle TMA files ownership to that account. To change ownership of the Oracle Tuxedo Mainframe Adapter for TCP on a UNIX machine, enter the following command:

```
chown -R adm_account full_path_of_tux_prod_dir
```

```
chgrp -R adm_group full_path_of_tux_prod_dir
```

- `adm_account`: Oracle Tuxedo administrator account
- `adm_group`: Oracle Tuxedo administrator group ID
- `full_path_of_tux_prod_dir`: The full pathname of the product directory where you installed the Oracle Tuxedo Mainframe Adapter for TCP.

Installing Oracle Tuxedo Mainframe Adapter for TCP Using Silent Installation

The following sections describe how to install Oracle Tuxedo Mainframe Adapter for TCP using silent installation on UNIX systems:

- [What Is Silent Installation?](#)
- [Using Silent Installation: Main Steps](#)

What Is Silent Installation?

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

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

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

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 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 Mainframe Adapter for TCP Response File

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

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

RESPONSEFILE_VERSION

The format version number of response file.

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.

DEPENDENCY_LIST

List of dependents that need to be installed along with this product.

Example:

```
DEPENDENCY_LIST={ "TMATCP:12.1.3.0.0" }
```

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 UNIX:

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

Notes:

- To install Oracle Tuxedo Mainframe Adapter for TCP 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 UNIX:

```
./runInstaller -help
```

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.
- `-force` - Use the `-force` flag with the `-responseFile` flag to suppress the dialog when specifying a new folder for `ORACLE_HOME`.
- `-waitforcompletion` - Use the `-waitforcompletion` flag with `-silent` flag to wait for completion instead of spawning the java engine and exiting.

For example, on a UNIX machine, enter the following command:

```
./runInstaller -responseFile /tma/tmatcp_linux_silent.rsp -silent
```

Note:

- Using the `-nowelcome` option with the `-silent` option is unnecessary since the Welcome screen does not appear when you use the `-silent` option.
- If the `ORACLE_HOME` is not empty, use the `-force` option to overwrite it; otherwise, you will get the following error message:

```
SEVERE:OUI-10029:You have specified a non-empty directory to
install this product. It is recommended to specify either an empty
or a non-existent directory. You may, however, choose to ignore
```

this message if the directory contains Operating System generated files or subdirectories like lost+found.

Response File Examples

[Listing 3-1](#) and [Listing 3-2](#) show response file examples that apply to a silent installation of Oracle Tuxedo Mainframe Adapter for TCP 12c Release 2 (12.1.3).

Listing 3-1 Example - A Brief Response File

```
RESPONSEFILE_VERSION=2.2.1.0.0
ORACLE_HOME="/home/test/OraHome_1"
ORACLE_HOME_NAME="OUIHome1"
INSTALL_TYPE="Full"
SAMPLES_SELECT=true
CRM_FOLDER="/home/test/OraHome_1/tuxedo12.1.3.0.0"
```

Listing 3-2 Example - A Complete Response File

```
#####
## Copyright (c) 1999, 2013 Oracle. All rights reserved.      ##
##                                                            ##
## Specify values for the variables listed below to customize ##
## your installation.                                         ##
##                                                            ##
## Each variable is associated with a comment. The comment   ##
## identifies the variable type.                             ##
##                                                            ##
## Please specify the values in the following format:        ##
##                                                            ##
```

```

##          Type          Example          ##
##          String        "Sample Value"      ##
##          Boolean       True or False      ##
##          Number        1000                ##
##          StringList    {"String value 1","String Value 2"} ##
##
## The values that are given as <Value Required> need to be ##
## specified for a silent installation to be successful.    ##
##
##
## This response file is generated by Oracle Software      ##
## Packager.                                               ##
#####

```

```

RESPONSEFILE_VERSION=2.2.1.0.0

```

```

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

```



```

#-----
-----
#Name      : FROM_LOCATION
#Datatype  : String
#Description: Complete path to the products.xml.
#Example: FROM_LOCATION = "../stage/products.xml"
#-----
-----
FROM_LOCATION="/home/test/tma/stage_tma_linux/Disk1/stage/products.xml"

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

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

```

```

#-----
-----
ORACLE_HOME="/home/test/OraHome_1"

#-----
-----
#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="OUIHome1"

#-----
-----
#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 = {"tmabase_tcp", "12.1.3.0.0"}
#-----
-----
DEINSTALL_LIST={"tmabase_tcp", "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 = {"tmabase_tcp", "12.1.3.0.0"}
#-----
-----

```

```
TOPLEVEL_COMPONENT={"tmabase_tcp","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 : tmabase_tcp
```

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

```
SELECTED_LANGUAGES={"en"}
```

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

```
#Name      : COMPONENT_LANGUAGES
```

```
#Datatype  : StringList
```

```
#Description: Languages supported by this component.List of supported
languages : {"en","ja"}
```

```
#Component : tmabase_tcp
```

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

```
COMPONENT_LANGUAGES={"en"}
```

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

```
#Name      : INSTALL_TYPE
```

```
#Datatype  : String
```

```
#Description: Installation type of the component.
```

```
#Component : tmabase_tcp
```

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

```
INSTALL_TYPE="Full"
```

Installing Oracle Tuxedo Mainframe Adapter for TCP (IMS)

Installing Oracle Tuxedo Mainframe Adapter for TCP (IMS) (hereafter referenced as TMA TCP for IMS) consists of the following basic tasks.

- [Preparing to Install](#)
- [Installing TMA TCP for IMS](#)

Note: To upgrade from previous releases of eLink for Mainframe TCP, refer to *Oracle Tuxedo Mainframe Adapter for TCP Release Notes*.

Preparing to Install

Complete the following tasks prior to installing the TMA TCP for IMS software:

- Read the *Oracle Tuxedo Mainframe Adapter for TCP Release Notes*
- Verify the TCP/IP stack software is installed and configured as part of your operating system installation.

Installing TMA TCP for IMS

To install TMA TCP for IMS, complete the following tasks:

- [Task 1: Allocate z/OS Datasets](#)
- [Task 2: Unload the Distribution Files](#)
- [Task 3: Link-Edit TMA TCP for IMS](#)

- [Task 4: Define TMA TCP for IMS to MVS](#)
- [Task 5: Create an TMA TCP for IMS Configuration File](#)
- [Task 6: Create JCL to Run TMA TCP for IMS](#)
- [Task 7: Start Oracle TMA TCP for IMS](#)
- [Task 8: Test the Installation](#)

Task 1: Allocate z/OS Datasets

The directory structures on the TMA TCP for IMS software CD include the following partitioned datasets:

Partitioned Datasets (PDS)	Format	Contents
CONTROL	ASCII	Sample IMS definitions, sample Oracle TMA TCP for IMS configuration files
INCLUDE	ASCII	C header files, COBOL copybooks for use by IMS Client and Server transactions
JCL	ASCII	Job Control Language for linking and executing Oracle TMA TCP for IMS
SOURCE	ASCII	Sample IMS Client and Server transactions for use with Oracle TMA TCP for IMS
OBJECT	BINARY	Object component files of Oracle TMA TCP for IMS
MSGCAT (sequential dataset)	BINARY	File containing runtime messages that TMA TCP for IMS issue

1. FTP the JCL file `INSTALL` from the JCL subdirectory.
2. Execute this job to allocate datasets using high-level qualifiers appropriate to your installation (*hlq1* and *hlq2* are dataset name high-level qualifiers appropriate to your installation).

The following datasets are allocated to receive the Tuxedo Mainframe Adapter for TCP file distribution.

Listing 4-1 Allocating z/OS Datasets

```
DSNAME: hlq1.hlq2.CONTROL
DSORG: PO
DIRBLKS: 10
RECFM: FB
LRECL: 80
BLKSIZE: 23440 (or other valid blocksize)
SPACE: 2 tracks
Usage: IMS sample definitions, sample configuration file

DSNAME: hlq1.hlq2.INCLUDE
DSORG: PO
DIRBLKS: 10
RECFM: FB
LRECL: 80
BLKSIZE: 23440 (or other valid blocksize)
SPACE: 2 tracks
Usage: C header file, COBOL copybooks

DSNAME: hlq1.hlq2.JCL
DSORG: PO
DIRBLKS: 10
RECFM: FB
LRECL: 80
BLKSIZE: 23440 (or other valid blocksize)
SPACE: 2 tracks
Usage: Sample LINK and execute JCL

DSNAME: hlq1.hlq2.MSGCAT
DSORG: PS
DIRBLKS: 10
RECFM: FB
LRECL: 100
BLKSIZE: 32000 (or other valid blocksize)
```

SPACE: 2 tracks
 Usage: Message text

 DSNAME: *hlq1.hlq2.OBJECT*
 DSORG: PO
 DIRBLKS: 10
 RECFM: FB
 LRECL: 80
 BLKSIZE: 3200
 SPACE: 1 cylinder
 Usage: Pre-linked object files

 DSNAME: *hlq1.hlq2.SOURCE*
 DSORG: PO
 DIRBLKS: 10
 RECFM: FB
 LRECL: 80
 BLKSIZE: 23440 (or other valid blocksize)
 SPACE: 1 track
 Usage: Sample IMS client and server transactions

The following runtime datasets are also allocated.

Runtime Datasets	Format	Contents
LOAD	Binary	Load library for Oracle TMA TCP for IMS
CONFIG	ASCII	Configuration file for Oracle TMA TCP for IMS
MSGLOG	ASCII	Message log for Oracle TMA TCP for IMS
SVRLOG	ASCII	A log file containing server responses for Oracle TMA TCP for IMS

Listing 4-2 Allocating Runtime Datasets

```
DSNAME: hlq1.hlq2.LOAD
DSORG: PO
DIRBLKS: 10
RECFM: U
LRECL: 0
BLKSIZE: 6133 (or other valid blocksize)
SPACE: 1 cylinder
Usage: Load library

DSNAME: hlq1.hlq2.CONFIG
DSORG: PS
RECFM: FB
LRECL: 80
BLKSIZE: 23440 (or other valid blocksize)
SPACE: 1 track
Usage: Configuration File

DSNAME: hlq1.hlq2.MSGLOG
DSORG: PS
RECFM: VB
LRECL: 1028
BLKSIZE: 6144 (or other valid blocksize)
SPACE: 1 cylinder
Usage: Message Log

DSNAME: hlq1.hlq2.SVRLOG
DSORG: PS
RECFM: VB
LRECL: 32756
BLKSIZE: 32760 (Note: required blocksize)
SPACE: 1 cylinder
Usage: Server Response Log
```

Task 2: Unload the Distribution Files

The TMA TCP for IMS product is distributed on a single CD-ROM (with other members of the Tuxedo Mainframe Adapter for TCP product). The TMA TCP for IMS distribution files are grouped under one of the following directory structures:

- For UNIX, the directory is `/cdrom/unixmf`.

In this directory structure, `/cdrom` is the directory where the drive is mounted.

- For Windows, the directory is `D:\ntmf`.

In this directory structure, `D:` is the CD-ROM drive.

Because the distribution files are located on a CD-ROM, you probably use a UNIX-based system or a PC-based system to access the files on the CD-ROM and file transfer them to z/OS.

Transfer the contents of each subdirectory to the corresponding distribution library on the z/OS system. Use a file transfer program (such as FTP) or any other mechanism that allows you to transfer files to your z/OS system. The file transfer program must be able to:

- Properly transfer both text and binary files
- Properly create members in a target library (PDS)

The IBM File Transfer Program (FTP, distributed as part of TCP/IP for z/OS) satisfies both of these requirements.

Each file in each subdirectory on the distribution CD-ROM corresponds to a member of the associated target library. For example, file `BEATCPI` in the `JCL` subdirectory corresponds to member `BEATCPI` in the `JCL` distribution library.

Note: All files in the `CONTROL`, `INCLUDE`, `JCL`, and `SOURCE` subdirectories are text files; be sure to specify text (or ASCII) mode when you transfer these files from the CD-ROM to their corresponding z/OS distribution library members.

The `MSGCAT` file and all files in the `OBJECT` subdirectory are binary files; specify binary (or image) mode when you transfer these files from the CD-ROM to the corresponding z/OS distribution library members.

Task 3: Link-Edit TMA TCP for IMS

The TMA TCP for IMS program is delivered as a pre-linked object file. Use the `LNKBEA34` JCL file for linking with IBM's TCP/IP.

Tailor the JCL as required to match your installation's standards and naming conventions. Review the following items and make the necessary changes before you submit the job.

1. Supply an appropriate Job Card for the job.
2. Change the names of product datasets as required to match those in your particular installation, including LE/370 datasets (for example, SYS1.SCEELKED), TCP/IP for z/OS datasets (for example, SYS1.SEZACMTX), and IMS (for example, IMSV5R1.RESLIB).
3. Change the names of TMA TCP for IMS datasets as required to match those you allocated in Step 1 (for example, BEATCPI.DIST.OBJECT).
4. Change the name of the link-editor (HEWL) as required to match the name used by your installation.

Note: The link job references components supplied by LE/370, TCP/IP for z/OS, and IMS.

When the job completes successfully, you receive a return code of 0 from the link-edit steps and an executable load module is placed into the specified target load library. The load module should have attributes of AMODE=31, RMODE=24. There should be no unresolved external references.

Sample JCL for Linking TMA TCP for IMS

[Listing 4-3](#) is JCL to link the distributed object files into an executable load module.

Listing 4-3 Sample JCL for Linking Distributed Object Files

```
//JOBNAME JOB ETC.
//LKED EXEC PGM=HEWL,COND=(4,LT),
// REGION=2M,PARM='AMODE=31,MAP,XREF'
//SYSLIB DD DSNAME=SYS1.SCEELKED,DISP=SHR
// DD DSNAME=SYS1.SEZACMTX,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DISP=SHR,DSNAME=BEATCPI.DIST.LOAD
//SYSUT1 DD UNIT=VIO,SPACE=(TRK,(10,10))
//RESLIB DD DISP=SHR,DSN=IMSV5R1.RESLIB
//SYSLIN DD DISP=OLD,DSN=BEA.HLQ.OBJECT(PREBEA34)
// DD *
INCLUDE RESLIB(DFSLI000)
ENTRY CEESTART
NAME BEATCPI(R)
```

```
/*  
//
```

Task 4: Define TMA TCP for IMS to MVS

To integrate TMA TCP for IMS into your IMS environment, you must properly define it to MVS.

OTMA Definition

To use OTMA to integrate TMA TCP for IMS into your z/OS environment, you must properly define it in the program property table (PPT). [Listing 4-4](#) is a sample program entry for the Tuxedo Mainframe Adapter for TCP product. Also, you must link the program to an APF authorized library.

Listing 4-4 Sample Program Property Table

```
PPT PGMNAME(BEATCP34)      /* PROGRAM NAME = BEATCP34          */  
      CANCEL                /* PROGRAM CAN BE CANCELED          */  
      KEY(7)                /* PROTECT KEY ASSIGNED IS 7       */  
      SWAP                  /* PROGRAM IS SWAPPABLE            */  
      NOPRIV                /* PROGRAM IS NOT PRIVILEGED       */  
      DSI                   /* REQUIRES DATA SET INTEGRITY    */  
      PASS                  /* CANNOT BYPASS PASSWORD PROTECTION */  
      SYST                  /* PROGRAM IS A SYSTEM TASK        */  
      AFF(NONE)            /* NO CPU AFFINITY                 */  
      NOPREF                /* NO PREFERRED STORAGE FRAMES     */
```

Task 5: Create an TMA TCP for IMS Configuration File

Oracle TMA TCP for IMS makes use of a text configuration file which is read and processed during initialization to establish the configuration. Any text editor (such as the ISPF Editor) can be used to create the configuration file. `CONFIGOT`, the OTMA sample in the `CONTROL` distribution library is a sample configuration file which you may use as a starting point for creating configuration files tailored to your particular installation.

Place your completed configuration file in the configuration dataset allocated in the [“Task 1: Allocate z/OS Datasets”](#) section. A configuration file includes the following basic sections:

- `SYSTEM` Statement Definition
- Local Gateways Definition
- Remote Gateways Definition
- Remote Services Definition
- Local Services and IMS Host Systems Definition

Note: You must define local services and IMS host systems if running OTMA.

For detailed information about syntax and parameter definitions for the configuration file, refer to the Oracle Tuxedo Mainframe Adapter for TCP IMS User Guide.

SYSTEM Statement Definition

`SYSTEM` parameters control the overall operational environment of Oracle TMA TCP for IMS. Because TMA TCP for IMS cannot “wait” on the IMS Message Queue, the queue is periodically interrogated (polled) to determine if any new work has arrived (for example, IMS Client requests or Server responses). Set the `SleepTime` parameter to specify the amount of time that TMA TCP for IMS “sleeps” (waits) during idle periods before checking the IMS Message Queue for work.

Local Gateways Definition

You must provide at least one `GATEWAY TYPE=LOCAL` statement to define access to your local (IMS) gateway by remote systems. In this statement, specify the following:

- The IP Address that remote systems use to establish TCP/IP connections with this gateway.
- The IP Address should be that of the z/OS machine on which Oracle TMA TCP for IMS is running.
- The Port Number that remote systems use to establish TCP/IP connections with this gateway.
- You should specify a Port Number that is not a well-known port number and one that is not used by any other TCP/IP application.
- The Account ID and Password (if used) that must be supplied by remote systems to successfully establish an inbound session with this gateway

- The maximum number of concurrent inbound sessions (TCP/IP connections initiated by remote systems) that is accepted via the specified port

Make sure that remote systems are correctly configured with the corresponding IP Address, Port Number, Account ID (if used), and Password (if used). Otherwise, attempts by remote systems to connect to this gateway fail.

Remote Gateways Definition

`GATEWAY TYPE=REMOTE` statements define access to remote systems by the IMS gateway. In particular, each `GATEWAY TYPE=REMOTE` statement specifies:

- The IP Address to use to establish a TCP/IP connection (that is, connect) with the remote gateway
- The Port Number to use to establish a TCP/IP connection (that is, connect) with the remote gateway
- The Account ID and Password (if used) that must be supplied to establish an outbound session with the remote gateway
- The minimum and maximum number of concurrent, outbound sessions that are established with the remote gateway
- If you specify a value other than zero for the minimum number of sessions, TMA TCP for IMS attempts to establish the specified number of outbound sessions with the remote system during initialization.

The IP Address, Port Number, Account ID, and Password must match the configuration of the remote system. Otherwise, attempts to establish an outbound session with the remote system fails.

Remote Services Definition

`SERVICE TYPE=REMOTE` statements define remote services to which TMA TCP for IMS has access. Each statement specifies:

- The corresponding local and remote service names
- The Local Service Name is the name used by an IMS Client transaction to request the service. The Remote Service Name is the name of the service as defined on the remote system. These need not be the same, but note that the Remote Service Name *must* match the configuration of the remote system.
- The remote system (Logical Machine ID) that offers the service

- The remote system offering the service is identified by a Logical Machine ID, which must correspond to the Logical Machine ID specified in a `GATEWAY TYPE=REMOTE` statement.
- A time limit for processing the request
- The Time Limit (in seconds) specifies how long TMA TCP for IMS “waits” for a response to a request sent to a remote system before assuming that the request has “timed out” and that no response is forthcoming. Specify a value that makes a reasonable allowance for network latency and overall system load. Small values should be avoided because they may result in requests timing out under otherwise normal circumstances.

Note: It is permissible to have more than one `SERVICE TYPE=REMOTE` statement for the same (local) service name. When this is the case, each statement must identify a different remote Logical Machine ID. This allows TMA TCP for IMS to “spread” requests for a particular service among multiple remote systems offering the service, thus balancing the load and improving overall throughput.

Local Services and IMS Host Systems Definition

`SERVICE TYPE=LOCAL` statements define local services offered by a local IMS host. There can be multiple `SERVICE TYPE=LOCAL` statements and multiple statements defining the same service, provided each specifies a different local `IMSID`. Each statement requires the following information:

- The corresponding local and remote service names

The Local Service Name is the name used by an IMS Client transaction to request the service. The Remote Service Name is the name of the service as defined on the remote system.

Note: The local and remote service names can be different; however, the Remote Service Name *must* match the configuration of the remote system.

- The `IMSID` (Logical IMS ID) that offers the service

The `IMSID` is a symbolic name that uniquely identifies an IMS system definition. This name must match a logical IMS ID (`IMSID`) specified in a `HOST, TYPE=IMS` statement.

Task 6: Create JCL to Run TMA TCP for IMS

You must create JCL to run the Oracle TMA TCP for IMS product. The following section specifies how to create JCL for running as an OTMA client.

JCL for OTMA Clients

When running the product as an OTMA client, TMA TCP for IMS runs in its own address space under z/OS as a started task or long running job. Member `RUNOTM` in the JCL distribution library is a sample job for executing TMA TCP for IMS.

When tailoring the JCL for your OTMA installation, pay particular attention to the following:

- You need to supply an appropriate Job Card for the job. The job card should specify `TIME=1440`, since TMA TCP for IMS executes as a non-ending job.
- TMA TCP for IMS is written in the C language and requires access to the LE/370 runtime environment. Add DD cards to the `STEPLIB` for the LE/370 runtime datasets, including `SYS1.SCEERUN` and `SYS1.SCEELKED`. Use whatever dataset names are appropriate for your installation.
- For diagnostic purposes, you may wish to add DD cards for `SYSUDUMP` and/or `CEEDUMP`. In the event of an abnormal termination (`ABEND`), LE/370 writes a formatted dump to `CEEDUMP`.
- Add a DD card for `SYSTCPD` pointing to the TPC/IP Data file (for example, `TCPIP.V3R1.DATA` – change the dataset name as necessary to conform to your installation). This dataset is used by TCP/IP in the TMA TCP for IMS address space (when using IBM TCP/IP).
- If you want to preserve messages from previous executions of TMA TCP for IMS, code `DISP=MOD` in the `MSGLOG` DD statement; new messages are appended to the end of the dataset (existing messages are preserved). If you want the Message Log to be overwritten with each new execution of TMA TCP for IMS, code `DISP=OLD` (or `DISP=SHR`) in the `MSGLOG` DD statement; existing messages are lost.

Task 7: Start Oracle TMA TCP for IMS

Before you attempt to start TMA TCP for IMS the first time, make sure that the proper IMS definitions are in place.

If you are using RACF for security, you must set up a `FACILITY` Class Profile in RACF of `IMSXCF.group.client`. You can configure the `group` and `client` names for TMA TCP for IMS.

Start TMA TCP for IMS by submitting the JCL prepared in the “[Task 6: Create JCL to Run TMA TCP for IMS](#)” section.

For additional information on operating this product, refer to the Oracle Tuxedo Mainframe Adapter for TCP IMS User Guide.

Task 8: Test the Installation

When you start TMA TCP for IMS for the first time, you should do so in a controlled environment, using a small, simple configuration to methodically test the system to verify your installation and the configuration. For additional information on using this product, refer to the Oracle Tuxedo Mainframe Adapter for TCP IMS User Guide.

Distribution Libraries and Executables

The Tuxedo Mainframe Adapter for TCP CD-ROM contains the following libraries and executable programs. After installing the TMA TCP for IMS software, verify that these libraries and programs are installed on your system.

Directory	Files
(top)	msgcat
CONTROL	beacrp01 beacrq01 beacrr01 beasvr01 beatcpi configbp configot
INCLUDE	beatcpi client server
JCL	INSTALL LNKBEA34 RUNBMP RUNOTM
OBJECT	PREBEA34
SOURCE	beacrp01 beacrq01 beacrr01 beaotm beasvr01 dfsydru0 dfsyprx0

Installing Oracle Tuxedo Mainframe Adapter for TCP (CICS)

Installing Oracle Tuxedo Mainframe Adapter for TCP (CICS) (hereafter referenced as TMA TCP for CICS) consists of the following basic tasks and topics.

- [Preparing for Installation](#)
- [Installing TMA TCP for CICS](#)

Note: To upgrade from previous releases of eLink for Mainframe TCP, refer to *Oracle Tuxedo Mainframe Adapter for TCP Release Notes*.

Preparing for Installation

Before you install TMA TCP for CICS, complete the following tasks.

- [Task 1: Verify System Requirements](#)
- [Task 2: Check Pre-Installation Considerations](#)

Task 1: Verify System Requirements

Prior to beginning the installation, verify hardware and software requirements. For system requirements, refer to the *Oracle Tuxedo Mainframe Adapter for TCP Release Notes*. Ensure that your installation workstation is connected to z/OS through TCP/IP, and is capable of running FTP.

Task 2: Check Pre-Installation Considerations

Verify that all block sizes in the supplied JCL are appropriate for the device types being used. By default, 3380 is assumed.

Installing TMA TCP for CICS

Complete the following tasks to install Oracle TMA TCP for CICS

- [Task 1: Allocate Temporary Data Sets](#)
- [Task 2: Transfer the Installation JCL](#)
- [Task 3: Create the PDS Libraries](#)
- [Task 4: Upload the Distribution Files](#)
- [Task 5: Define the CICS VSAM Files](#)
- [Task 6: Verify Contents of PROCS](#)
- [Task 7: Change Resource Names \(Optional\)](#)
- [Task 8: Define the CICS Table Entries](#)
- [Task 9: Build TMA TCP for CICS Executables](#)
- [Task 10: Verify the CICS Setup](#)
- [Task 11: Assemble and Link the CICS DCT](#)
- [Task 12: Modify the CICS Startup JCL](#)

Task 1: Allocate Temporary Data Sets

Allocate a temporary data set on z/OS direct access storage device (DASD) for the installation job as shown in the following example.

Note: Specify either the Volume Serial parameter or the Generic Unit parameter, but not both.

Listing 5-1 Allocating the Install Data Set

```
Data Set Name:YOURHLQ.TEMP.INSTALL
Volume serial:nnnnn          (Blank for authorized default volume)
```



```

Generic unit:                (Generic group name or unit address)
Space units: TRACK           (BLKS, TRKS, CYLS, KB, MB or BYTES)
Primary quantity:1          (In above units)
Secondary quantity:1        (In above units)
Directory blocks:0          (Zero for sequential data set)
Record format:FB
Record length:80
Block size:6160
Expiration date:             (YY/MM/DD, YYYY/MM/DD YY.DDD,
                              YYYY.DDD in Julian form DDDD for
                              retention period in days or blank)

Enter "/" to select option
Allocate Multiple Volumes

```

Task 2: Transfer the Installation JCL

The TMA TCP for CICS product is distributed on a single CD-ROM (with other members of the Oracle TMA TCP product). The TMA TCP for CICS distribution files are grouped under one of the following directory structures:

- To FTP from UNIX to the mainframe, the distribution files are in:
 - /cdrom/unixmf/cics directory,
 - /cdrom is the directory where the drive is mounted.
- To FTP from Windows to the mainframe, the distribution files are in:
 - D:\ntmf\cics
 - D: is the CD-ROM drive.

FTP the file called `INSTALL` to the temporary data set created in the [“Task 1: Allocate Temporary Data Sets”](#) section. Listing 3-2 is the process for downloading the `INSTALL` file using FTP.

Listing 5-2 Uploading Distribution Files

```

% ftp zos-machine
Connected to zos-machine.
Name (zos-machine:unix-id): tso-id
331 Send password please.

```

```

Password: tso-password
230 tso-id is logged on.
Remote system type is z/OS.
ftp> quot site rec=fb
ftp> quot site lr=80
ftp> put INSTALL 'data set name allocated'
200 Port request OK.
125 Storing data set 'data set name allocated'
2749 bytes sent in 0.00 seconds (555.12 Kbytes/s)
ftp> quit
221 Quit command received. Goodbye.

```

Task 3: Create the PDS Libraries

Tailor the JCL as required to match your installation's standards and naming conventions. Review the following items and make the necessary changes before you submit the job.

Note: This is an instream PROC; the variables are at the end of the PROC.

1. Modify the JOBCARD to your standards.
2. Modify the variable "YOURHLQ." If you intend to rename them, modify the variables JCL, OBJECT, and SOURCE.
3. Submit the revised INSTALL JCL. This JCL creates six partitioned data sets based on the "high-level-qualifier" you have specified as the value of the JCL parameter (&HLQ) in the JCL.

The INSTALL JCL includes the name BEATCPC as part of the data set name. For example.

Listing 5-3 Sample INSTALL JCL

```

//JDOE1  JOB (INSTALL), 'J DOE', NOTIFY=&SYSUID,
//        CLASS=C,MSGCLASS=X,MSGLEVEL=(1,1),TIME=1440
//*
//INSTALL  PROC  HLQ=,                ** YOUR HIGH LEVEL QUALIFIER
//        DASD=,                    ** DASD DEVICE TYPE
//        VOLUME=,                  ** SERIAL NUMBER OF TARGER VOLUME
//        BLKFB=6160,              ** BLKSIZE FOR FIXED BLOCK FILE
//*=====
//*      THESE JCL WILL CREATE THE PDS FILES REQUIRED FOR
//*      THE INSTALLATION OF BEA TMA TCP FOR CICS
//*=====
//        PDCNTL= 'CONTROL',        ** 'YOURHLQ'.BEATCPC.CONTROL
//        PDSINCL= 'INCLUDE',      ** 'YOURHLQ'.BEATCPC.INCLUDE

```

```

//          PDSJCL='JCL',          ** 'YOURHLQ'.BEATCPC.JCL
//          PDSOBS='OBJECT',      ** 'YOURHLQ'.BEATCPC.OBJECT
//          PDSLKED='PLKED',      ** 'YOURHLQ'.BEATCPC.PLKED
//          PDSSRCE='SOURCE',     ** 'YOURHLQ'.BEATCPC.SOURCE
//          PDSLKED='PLKED',      ** 'YOURHLQ'.BEATCPC.PLKED
//          PDSSRCE='SOURCE'     ** 'YOURHLQ'.BEATCPC.SOURCE
//*-----
//* WE RECOMMEND THAT YOU KEEP THE NODE 'BEATCPC' IN THE data set NAMES
//*-----
//EXEC PGM=IEFBRI4
//PDCNTL DD DSNNAME=&HLQ.BEATCPC.&PDCNTL., DISP=(NEW,CATLG,DELETE),
//          DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=&BLKFB),
//          UNIT=&DASD,VOL=SER=&VOLUME,SPACE=(CYL,(1,1,5))
//*
//PDSJCL DD DSNNAME=&HLQ.BEATCPC.&PDSJCL., DISP=(NEW,CATLG,DELETE),
//          DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=&BLKFB),
//          UNIT=&DASD,VOL=SER=&VOLUME,SPACE=(CYL,(1,1,5))
//*
//PDSINCL DD DSNNAME=&HLQ.BEATCPC.&PDSINCL., DISP=(NEW,CATLG,DELETE),
//          DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=&BLKFB),
//          UNIT=&DASD,VOL=SER=&VOLUME,SPACE=(CYL,(1,1,5))
//*
//PDSOBS DD DSNNAME=&HLQ.BEATCPC.&PDSOBS., DISP=(NEW,CATLG,DELETE),
//          DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=3200),
//          UNIT=&DASD,VOL=SER=&VOLUME,SPACE=(CYL,(3,1,5))
//*
//PDSLKED DD DSNNAME=&HLQ.BEATCPC.&PDSLKED., DISP=(NEW,CATLG,DELETE),
//          DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=&BLKFB),
//          UNIT=&DASD,VOL=SER=&VOLUME,SPACE=(CYL,(3,1,5))
//*
//PDSSRCE DD DSNNAME=&HLQ.BEATCPC.&PDSSRCE., DISP=(NEW,CATLG,DELETE),
//          DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=&BLKFB),
//          UNIT=&DASD,VOL=SER=&VOLUME,SPACE=(CYL,(1,1,5))
//*
//          PEND
//*-----
//* CUSTOMIZE THE PARAMETER ACCORDING TO YOUR INSTALLATION
//*-----
//          EXEC PROC=INSTALL,
//          DASD=3390,** YOUR DEVICE TYPE
//          VOLUME=MVS201,** YOUR VOLUME SERIAL NUMBER
//          HLQ='PRODUCTS.BEA.TMA',** YOUR HIGH LEVEL QUALIFIER
//          PDCNTL='CONTROL',** BEATCPC CONTROL
//          PDSINCL='INCLUDE',** BEATCPC INCLUDE
//          PDSJCL='JCL',** BEATCPC JCL
//          PDSOBS='OBJECT',** BEATCPC OBJECT
//          PDSLKED='PLKED',** BEATCPC PLKED
//          PDSSRCE='SOURCE'** BEATCPC SOURCE

```

Task 4: Upload the Distribution Files

Upload all the files on the distribution medium to the partitioned data sets allocated by the job `INSTALL` in the “[Task 3: Create the PDS Libraries](#)” section.

The TMA TCP for CICS distribution files are grouped under one of the following directory structures:

- To FTP the distribution files from UNIX to the mainframe, the files are in:

`/cdrom/unixmf/cics`

`/cdrom` is the directory where the drive is mounted.

- To FTP files from Windows to the mainframe, the files are in:

`D:\ntmf\cics`

`D:` is the CD-ROM drive.

The previous directory structures include the following partitioned data sets.

Table 5-1 Partitioned Data Sets

From	To
<code>./control</code>	Your PDS for CONTROL (YOURHLQ.BEATCPC.CONTROL) (ASCII mode)
<code>./jcl</code>	Your PDS for JCL (YOURHLQ.BEATCPC.JCL) (ASCII mode)
<code>./source</code>	Your PDS for SOURCE (YOURHLQ.BEATCPC.SOURCE) (ASCII mode)
<code>./include</code>	Your PDS for INCLUDE (YOURHLQ.BEATCPC.INCLUDE) (ASCII mode)
<code>./object</code>	Your PDS for OBJECT (YOURHLQ.BEATCPC.OBJECT) (Binary mode)
<code>./plked</code>	Your PDS for PLKED (YOURHLQ.BEATCPC.PLKED) (Binary mode)

Listing 5-4 Uploading Distribution Files Using FTP

```

% ftp zos-machine
Connected to zos-machine.
Name (beavs:tso-id):
331 Send password please.
Password: tso-password
230 tso-id is logged on.
Remote system type is z/OS.
ftp> prompt off
Interactive mode off
ftp> quot site rec=fb
200 SITE command was accepted
ftp> quot site lr=80
200 SITE command was accepted

```

Upload the CONTROL files.

Note: You must set ASCII mode on.

Listing 5-5 Upload control Files

```

ftp> lcd /cdrom/unixmf/cics/control
Local directory now /cdrom/unixmf/cics/control
ftp> cd 'YOURHLQ.BEATCPC.CONTROL'
250 "'YOURHLQ.BEATCPC.CONTROL'" partitioned data set is working directory
ftp>ascii
200 Representation type is Ascii
ftp> mput *
local: CSDU remote: CSDU
200 Port request OK.
125 Storing data set "YOURHLQ".BEATCPC.CONTROL (CSDU)
250 Transfer completed successfully.
5740 bytes sent in 0.01 seconds (747.10 Kbytes/s)
.
.

```

Upload the JCL files.

Note: You must set ASCII mode on.

Listing 5-6 Uploading JCL Files

```
ftp> lcd /cdrom/unixmf/cics/jcl
Local directory now /cdrom/unixmf/cics/jcl
ftp> cd 'YOURHLQ.BEATCPC.JCL'
250 "'YOURHLQ.BEATCPC.JCL'" partitioned data set is working directory.
ftp>ascii
200 Representation type is Ascii
ftp> mput *
local: BLDVSAM remote: BLDVSAM
200 Port request OK.
125 Storing data set "YOURHLQ".BEATCPC.JCL(BLDVSAM)
250 Transfer completed successfully.
2214 bytes sent in 0.00 seconds (563.49 Kbytes/s)
.
.
.
```

Upload the SOURCE files.

Note: You must set ASCII mode on.

Listing 5-7 Uploading SOURCE Files

```
ftp> lcd /cdrom/unixmf/cics/source
Local directory now /cdrom/unixmf/cics/source
ftp> cd 'YOURHLQ.BEATCPC.SOURCE'
```

```

250 "'YOURHLQ.BEATCPC.SOURCE'" partitioned data set is working directory.
ftp>ascii
200 Representation type is Ascii
ftp> mput *
local: BEACCLN1 remote: BEACCLN1
200 Port request OK.
125 Storing data set YOURHLQ.BEATCPC.SOURCE(BEACCLN1)
250 Transfer completed successfully.
10578 bytes sent in 0.01 seconds (1013.85 Kbytes/s)
.
.
.

```

Upload the INCLUDE files.

Note: You must set ASCII mode on.

Listing 5-8 Uploading INCLUDE Files

```

ftp> ascii
ftp> lcd /cdrom/unixmf/cics/include
ftp> cd "'YOURHLQ".BEATCPC.INCLUDE'
250 "'YOURHLQ.BEATCPC.INCLUDE'" partitioned data set is working directory
ftp>ascii
200 Representation type is Ascii
ftp> mput *
200 Port request OK.
125 Storing data set "YOURHLQ".BEATCPC.INCLUDE'(CLIENT)
250 Transfer completed successfully.
local: CLIENT remote: CLIENT
3736 bytes sent in 0.0081 seconds (4.5e+02 Kbytes/s)
.
.
.

```

Upload the OBJECT files.

Note: You must set binary mode on.

Listing 5-9 Uploading OBJECT Files

```
ftp> binary
200 Representation type is IMAGE.
ftp> lcd /cdrom/unixmf/cics/object
Local directory now /cdrom/unixmf/cics/object
ftp> cd 'YOURHLQ.BEATCPC.OBJECT'
250 "'YOURHLQ.BEATCPC.OBJECT'" partitioned data set is working directory.
ftp>binary
200 Representation type is Image
ftp> mput *
local: BEACFGSV remote: BEACFGSV
200 Port request OK.
125 Storing data set "YOURHLQ".BEATCPC.OBJECT(BEACFGSV)
250 Transfer completed successfully.
6240 bytes sent in 0.00 seconds (1466.96 Kbytes/s)
.
.
.
```

Upload the PLKED files.

Note: You must set binary mode on.

Listing 5-10 Uploading PLKED Files

```
ftp> binary
200 Representation type is Image
ftp> lcd /cdrom/unixmf/plked
```



```

ftp> cd ' "YOURHLQ".BEATCPC.PLKED'
250 "'YOURHLQ.BEATCPC.PLKED'" partitioned data set is working directory
ftp>binary
200 Representation type is Image
ftp> mput *
200 Port request OK.
125 Storing data set "YOURHLQ.BEATCPC.PLKED(BEACIC00)
250 Transfer completed successfully.
local: BEACIC00 remote: BEACIC00
64560 bytes sent in 0.7 seconds (90 Kbytes/s)
.
.
.

```

Verify that the distribution libraries were uploaded. For the listing of files that should be on your system, refer to the [“List of Distribution Files and Members”](#) section.

Task 5: Define the CICS VSAM Files

Define the CICS VSAM files (Connections, Requesters, Service Names, User Connection, Inbound Service).

1. Modify DELETE/DEFINE control cards VCON, VREQ, VSVC, VUSR, and VISN in the PDS "YOURHLQ".BEATCPC.CONTROL. [Listing 5-11](#) is an example of a modified DELETE/DEFINE.

Listing 5-11 Sample DELETE/DEFINE Control Card

```

DELETE YOURHLQ.BEATCPC.BEAVSVC           -
      CLUSTER PURGE                       -
DEFINE CLUSTER (                          -
      NAME (YOURHLQ.BEATCPC.BEAVSVC)      -
      SHAREOPTIONS (2 3)                  -
      KEYS (16 0)                          -
      CISZ (1024)                          -
      TRACKS (5 1)                         -

```

```

VOLUMES (CICS01)           -
DATA (                       -
  RECORDSIZE (56, 56)       -
NAME (YOURHLQ.BEATCPC.BEAVSVC.DATA) -
INDEX (                     -
NAME (YOURHLQ.BEATCPC.BEAVSVC.INDEX)

```

Change all instances of `YOURHLQ.BEATCPC.BEAVSVC` to a physical data set name that meets your standards. Change the `TRACKS` and `VOLUMES` parameters to appropriate values for your installation. The provided value for the `TRACKS` parameter works for most installations, but this value can be increased if necessary.

2. For the `BLDVSAM JCL`, complete the following steps.
 - a. Modify the `JOB` to your standards.
 - b. Modify `YOURHLQ.BEATCPC.XXXX`, where `XXXX` is `BEAVCON`, `BEAVREQ`, `BEAVSVC`, `BEAVUSR`, and `BEAVISN`.
 - c. Submit `BLDVSAM`.
3. Verify the results. There should be three data sets in `YOURHLQ.BEATCPC` corresponding to each of the five `VSAM` files. The three data sets are as follows:
 - One data set has no extension
 - One data set has `DATA` as the extension
 - One data set has `INDEX` as the extension

Note: A condition code of 8 is acceptable on `DELETE` steps. If you receive condition codes other than zero for `DEFINE` steps, you should evaluate this for potential problems.

Task 6: Verify Contents of PROCS

Before you go on to the next step, we recommend that you check the following items in `PROCS` `CMPBC`, `CMPLMAP`, `CMPLKED`, and `LNKIBM` in the `PDS` `YOURHLQ.BEATCPC.JCL`.

You may need to modify them to meet your standards. You will need to review the following:

- All references to `CICS` and `TCP/IP` data set names.

- The data set names associated with the STEPLIB, SYSMSGS, SYSXMSCS, SYSPATH, and SYSLIB DD names.
- Unit = references.

Task 7: Change Resource Names (Optional)

The default names for Oracle TMA TCP for CICS Mapsets, Programs, Transids, and VSAM data sets are listed in Appendix A, “Changing Resource Names.” If you want to use a different naming convention, use the procedure for changing names included after each table.

Note: You must have the C 370 compiler to customize resource names.

Task 8: Define the CICS Table Entries

1. Verify the CICS table entry definitions (PCTs, PPTs, FCTs) in CSDU (from the PDS "YOURHLQ".BEATCPC.CONTROL).
2. Modify the group and list name in CSDU (from the PDS "YOURHLQ".BEATCPC.CONTROL).
3. In the JCL in CSDUPD (from the PDS "YOURHLQ".BEATCPC.JCL), complete the following steps.
 - a. Modify the JOBCARD to your standards.
 - b. Modify the STEPLIB and DFHCSD data set names.
 - c. Modify the SYSIN DSN=YOURHLQ.BEATPC.CONTROL.
 - d. Submit CSDUPD.
 - e. Verify the results.

Note: A “COND CODE 0004” may result due to the DELETE commands for non-existent CSD objects.

Task 9: Build TMA TCP for CICS Executables

To build the executables for Oracle TMA TCP for CICS, you need to complete the following tasks.

1. If there is not a PDS allocated where the executables for your CICS region reside, allocate such a PDS before running any JCL that builds executables.

2. Link the CICS Admin maps (MainMenu, Connection, Requester, Outbound Service Name, Inbound Service Name, User Connection Account, Handler Configuration). In the JCL in MAKLMAP (from the PDS "YOURHLQ".BEATCPC.JCL"), complete the following steps.
 - a. Modify the JOBCARD to your standards.
 - b. Modify the PROCS JCLLIB card with the correct HLQ.
 - c. Modify the variables YOURHLQ, PDSOBS and LNKLIB for your PDS.
Note: LNKLIB is where your CICS region executable binary files reside.
 - d. Modify the variable PROG, if you renamed mapsets.
 - e. Submit MAKLMAP.
 - f. Verify the results.
3. Complete one of the following procedures depending on your compiler:
 - [Pre-Link/Link TMA TCP for CICS](#)
 - [Linking without the C 370 Compiler](#)

Pre-Link/Link TMA TCP for CICS

The following sections explain the procedures for linking programs and mapsets prior to running the Oracle TMA TCP for CICS product.

Note: The following sections apply for a C 370 compiler.

Linking the CICS Admin Programs

Pre-Link/Link the CICS Admin Programs (MainMenu, Connection, Requester, Outbound Service Name, Inbound Service Name, User Connection Account, Handler Configuration). In the JCL in MAKL1 (from the PDS "YOURHLQ".BEATCPC.JCL"), complete the following steps.

1. Modify the JOBCARD to your standards.
2. Modify the PROCS JCLLIB card with the correct HLQ.
3. Modify the variables YOURHLQ, PDSOBS, TCPLIB and LNKLIB for your PDS.
Note: LNKLIB is where your CICS region executable binary files reside.
4. Modify the LKED.SYSIN program names, if you renamed programs.
5. Submit MAKL1.

6. Verify the results.

Note: A "COND CODE = 004" is normal.

Linking the CICS Utility Programs

Pre-Link/Link the CICS utility Programs (Pre-Requester, Application Handler, Shutdown, and Configuration Converter). In the JCL in MAKL0 (from the PDS "YOURHLQ".BEATCPC.JCL"), complete the following steps.

1. Modify the JOBCARD to your standards.
2. Modify the PROCS JCLLIB card with the correct HLQ.
3. Modify the variables YOURHLQ, PDSOBS, TCPLIB and LNKLIB for your PDS.

Note: LNKLIB is where your CICS region executable binary files reside.

4. Modify the LKED.SYSIN program names, if you renamed programs.
5. Submit MAKL0.
6. Verify the results.

Note: A "COND CODE = 004" is normal.

Linking the CICS TCP Programs

Pre-Link/Link the CICS TCP Programs (Requester and Handler). In the JCL in MAKL3 (from the PDS "YOURHLQ".BEATCPC.JCL"), complete the following steps.

1. Modify the JOBCARD to your standards.
2. Modify the variables YOURHLQ, PDSOBS, TCPLIB and LNKLIB for your PDS.

Note: LNKLIB is where your CICS region executable binary file resides.

3. Modify the INCLUDE OBJECT program name, if you renamed programs.
4. Modify the LKED.SYSIN program names, if you renamed programs.
5. Submit MAKL3.
6. Verify the results.

Note: A "COND CODE = 004" is normal.

Linking without the C 370 Compiler

If you do NOT have the C 370 compiler, use the following procedure to Linkedit the PreLinked objects. Link the Prelinked objects of the following programs.

- CICS TCP Programs (Requester and Handler).
- CICS Utility Programs (Pre-Requester, Application Handler, Shutdown, and Configuration Converter).
- CICS Admin Programs (MainMenu, Connection, Requester, Service Name).

In the JCL in MAKLKED (from the PDS "YOURHLQ.BEATPCP.JCL"), complete the following steps.

1. Modify the JOBCARD to your standards.
2. Modify the variables YOURHLQ, TCPLIB and LNKLIB for your PDS.
Note: LNKLIB is where your CICS region executable binary file resides.
3. Submit MAKLKED.
4. Verify the results.

Task 10: Verify the CICS Setup

Log on to your CICS region to install and verify the components you set up for Oracle TMA TCP for CICS.

- CEDA INSTALL GROUP (____)
This is the group name you used in CSDU
- CEMT INQUIRE PROG (____)
Verify the mapsets and programs
- CEMT INQUIRE TRAN (____)
Verify the transids
- CEMT INQUIRE FILE (____)
Verify the VSAM files

Task 11: Assemble and Link the CICS DCT

Note: Recent releases of CICS allow this task to be with CSD instead of this method.

1. Add the following entries to the CICS DCT.

Entry
BEALOG DFHDCT TYPE=SDSCI,
DSCNAME=BEALOG,
BLKSIZE=136,
RECSIZE=132,
RECFORM=VARUNB,
TYPEFLE=OUTPUT,
BUFNO=1
BEAL DFHDCT TYPE=EXTRA,
DESTID=BEAL,
DSCNAME=BEALOG

2. Assemble and link the DCT.

Task 12: Modify the CICS Startup JCL

1. Add the Oracle Tuxedo Mainframe Adapter load library (LNKLIB) to the DFHRPL concatenation.
2. Add the following DD statement to CICS file concatenation.


```
//BEALOG DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
```
3. Add DD statement to make the connection between the logical names of the VSAM files and the DSN.

Note: If DSN statements were added to the CSDU, this step is not necessary.

The following is an example of a CICS startup JCL.

Listing 5-12 Sample CICS Startup JCL with TMA TCP additions

```
//CICSREG JOB
//CICSCNTL EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN DD DISP=SHR,DSN=CICSTS31.SYSIN(DFHRCYES)
//DTCNTL EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN DD DISP=SHR,DSN=CICSTS31.SYSIN(DFHRCNO)
//*
//CICS EXEC PGM=DFHSIP, REGION=32M, TIME=1439, COND=(1,NE,CICSCNTL),
// =START=COLD, SYSIN, SEC=NO, GRPLIST=ETPLIST
//STEPLIB DD DISP=SHR,DSN=CICSTS31.SVTSC.SDFHAUTH
// DD DISP=SHR,DSN=CICSTS31.SDFHAUTH
// DD DISP=SHR,DSN=MASTCAT.MQM112.SCSQAUTH
// DD DISP=SHR,DSN=COB2140.COB2CICS.MODLIB
// DD DISP=SHR,DSN=C370.SEDCLINK
// DD DISP=SHR,DSN=ADLE370.V1R3M0.SCEECICS
// DD DISP=SHR,DSN=ADLE370.V1R3M0.SCEERUN
//DFHRPL DD DISP=SHR,DSN=CICSTS31.SVTSC.SDFHLOAD
// DD DISP=SHR,DSN=CICSTS31.SDFHLOAD
// DD DISP=SHR,DSN=BEA.BEA100.BEALOAD
// DD DISP=SHR,DSN=TCPIP.V3R1.SEZALINK
// DD DISP=SHR,DSN=TCPIP.V3R1.SEZATCP
// DD DISP=SHR,DSN=MQM112.SCSQCICS
// DD DISP=SHR,DSN=MQM112.SCSQLOAD
// DD DISP=SHR,DSN=COB2140.COB2CICS
// DD DISP=SHR,DSN=MASTCAT.COB2140.COB2LIB
// DD DISP=SHR,DSN=MASTCAT.PLI230.PLILINK
// DD DISP=SHR,DSN=C370.SEDCLINK
// DD DISP=SHR,DSN=ERW121.SERWLOAD
// DD DISP=SHR,DSN=ADLE370.V1R3M0.SCEECICS
// DD DISP=SHR,DSN=ADLE370.V1R3M0.SCEERUN
//SYSIN DD DISP=SHR,DSN=CICSTS31.CICS.SYSIN(TCICS)
//DFHCSD DD DISP=SHR,DSN=CICSTS31.DFHCSD
//DFHCMACD DD DISP=SHR,DSN=CICSTS31.DFHCMACD
//DFHJ01A DD DISP=SHR,DSN=CICSTS31.CICS.DFHJ01A
//DFHJ01B DD DISP=SHR,DSN=CICSTS31.CICS.DFHJ01B
//DFHJ01X DD DISP=SHR,DSN=CICSTS31.CICS.DFHJ01X
//DFHAUXT DD DISP=SHR,DSN=CICSTS31.CICS.DFHAUXT,
//DFHBUXT DD DISP=SHR,DSN=CICSTS31.CICS.DFHBUXT,
//DFHDMPA DD DISP=SHR,DSN=CICSTS31.CICS.DFHDMPA
//DFHDMPB DD DISP=SHR,DSN=CICSTS31.CICS.DFHDMPB
//DFHRSD DD DISP=SHR,DSN=CICSTS31.CICS.DFHRSD
//DFHGCD DD DISP=SHR,DSN=CICSTS31.CICS.DFHGCD
//DFHLCD DD DISP=SHR,DSN=CICSTS31.CICS.DFHLCD
//DFHXMSG DD DISP=SHR,DSN=CICSTS31.CNTL.CICS.DFHXRMSG
//DFHXCTL DD DISP=SHR,DSN=CICSTS31.CNTL.CICS.DFHXRCTL
```



```

//DFHTEMP DD DISP=SHR,DSN=CICSTS31.CNTL.CICS.DFHTEMP
//DFHINTRA DD DISP=SHR,DSN=CICSTS31.CNTL.CICS.DFHINTRA
//DFHJACD DD DISP=SHR,DSN=CICSTS31.CICS.DFHJACD
//DFHJPDS DD DISP=SHR,DSN=CICSTS31.DFHJPDS
//DFHJOUT DD SYSOUT=(A,INTRDR)
//DFHCXRF DD SYSOUT=X
//MQMMSG DD SYSOUT=*
//LOGUSR DD SYSOUT=*, DCB=(DSORG=PS,RECFM=V, BLKSIZE=136)
//MSGUSR DD SYSOUT=*, DCB=(DSORG=PS,RECFM=V, BLKSIZE=136)
//TCPDATA DD SYSOUT=*, DCB=(DSORG=PS,RECFM=V, BLKSIZE=136)
//BEALOG DD SYSOUT=*, DCB=(DSORG=PS,RECFM=V, BLKSIZE=136) ** BEA Connect
//COUT DD SYSOUT=*, DCB=(DSORG=PS,RECFM=V, BLKSIZE=137)
//PLIMSG DD SYSOUT=*, DCB=(DSORG=PS,RECFM=V, BLKSIZE=137)
//COUT DD SYSOUT=*, DCB=(DSORG=PS,RECFM=V, BLKSIZE=137)
//SYSABEND DD SYSOUT=*
//TRACEOUT DD SYSOUT=*
//PRINTER DD SYSOUT=*,DCB=BLKSIZE=121
//*
//PRTDMPA EXEC PGM=DFHDUP, PARM=SINGLE, REGION=0M, COND=(1,NE,DTCNTL)
//STEPLIB DD DSN=CICSTS31.SDFHLOAD,DISP=SHR
//SYSIN DD DUMMY
//DFHDMPDS DD DISP=SHR,DSN=CICSTS31.CICS.DFHDMPS
//DFHTINDX DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DFHPRINT DD SYSOUT=*
//*
//PRTDMPB EXEC PGM=DFHDUP, PARM=SINGLE, REGION=0M, COND=(1,NE,DTCNTL)
//STEPLIB DD DSN=CICSTS31.SDFHLOAD,DISP=SHR
//SYSIN DD DUMMY
//DFHDMPDS DD DISP=SHR,DSN=CICSTS31.CICS.DFHDMPS
//DFHTINDX DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DFHPRINT DD SYSOUT=*
//*
/* PRTAUXT EXEC PGM=DFHTUP, REGION=0M, COND=(1,NE,DTCNTL)
/* STEPLIB DD DSN=CICSTS31.SDFHLOAD,DISP=SHR
/* DFHAUXT DD DISP=OLD,DSN=CICSTS31.CICS.DFHAUXT
/* DFHAXPRT DD SYSOUT=*
/* DFHAXPRM DD DUMMY
/*
//PRTBUXT EXEC PGM=DFHTUP, REGION=0M, COND=(1,NE,DTCNTL)
//STEPLIB DD DSN=CICSTS31.SDFHLOAD,DISP=SHR
//DFHAUXT DD DISP=SHR,DSN=CICSTS31.CICS.DFHBUXT
//DFHAXPRT DD SYSOUT=*
//DFHAXPRM DD DUMMY

```

List of Distribution Files and Members

The following tables list and describe the members that make up each distribution partitioned data set.

BEATCPC.DIST.CONTROL

Verify that the following members are installed by Oracle TMA TCP for CICS.

Member	Description
CSDU	Del/Def Parameters for (PCTs, PPTs, FCTs)
VCON	Del/Def Parameters for (VSAM Connections)
VISN	Del/Def Parameters for (VSAM Inbound Services)
VREQ	Del/Def Parameters for (VSAM Requesters)
VSVC	Del/Def Parameters for (VSAM Service Names)
VUSR	Del/Def Parameters for (VSAM User Connections)

BEATCPC.DIST.JCL

Verify that the following members are installed by Oracle TMA TCP for CICS.

Member	Description
BLDVSAM	Jcl to Del/Def VSAM files (BEAVCON, BEAVREQ, BEAVSVC, BEAVUSR, BEAVISN)
CMPBC	Proc to Compile/link C program (Configuration)
CMPLKED	Proc to Link the PreLinked PLKED
CMPLMAP	Proc to Link CICS Mapsets(TCP Admin)
CSDUPD	Jcl to Del/Def CICS (PCTs, PPTs, FCTs)
LNKIBM	Proc to PreLink/Link Pgm objects

Member	Description
MAKBC	Jcl to Compile/link C program (Configuration)
MAKL0	Jcl to PreLink/Link Pgm objects(PreRequester, Shutdown)
MAKL1	Jcl to PreLink/Link Pgm objects(TCP Admin)
MAKL3	Jcl to PreLink/Link Pgm objects(Requester, Handler)
MAKLKED	Jcl to Link the PreLinked PLKED
MAKLMAP	Jcl to Link CICS Mapsets(TCP Admin)

BEATPC.DIST.SOURCE

Verify that the following members are installed by Oracle TMA TCP for CICS.

Member	Description
BEACCLN1	Program Sample Test Client
BEACSVR1	Program Sample Test Server
BEACFGSV	Program Name Configuration (Files, Maps, Programs, Trns)

BEATPC.DIST.INCLUDE

Verify that the following members are installed by Oracle TMA TCP for CICS.

Member	Description
BEACFGSV	Include for the name configuration (Files, Maps, Programs, Trns)
CLIENT	Copybook for COBOL CICS Client Request/Response Header
CLIENTH	Include for C CICS Client Request/Response Header

Member	Description
TWACOPY	Copybook for COBOL CICS Server TWA Message size
TWAINCL	Include for C CICS Server TWA Message size

BEATCPC.DIST.PLKED

Verify that the following members are installed by Oracle TMA TCP for CICS.

Member	Description
BEAAPPLC	Prelinked object of Application Handler
BEACIC00	Prelinked object of Handler
BEAPCON	Prelinked object of Admin Connections
BEAPISN	Prelinked object of Admin(Inbound Service Names)
BEAPMNU	Prelinked object of Admin(Main Menu)
BEAPREQ	Prelinked object of Admin(Requesters)
BEAPRERQ	Prelinked object of Pre-Requester
BEAP SVC	Prelinked object of Admin(Outbound Service Names)
BEAPUSR	Prelinked object of Admin(User Connection Account)
BEAREQST	Prelinked object of Requester
BEASHUT	Prelinked object of Shutdown
CNVTISNC	Prelinked object of Configuration Converter

BEATCPC.DIST.OBJECT

Verify that the following members are installed by Oracle TMA TCP for CICS:

- BEAAPPLC
- BEACFGSV

- BEACIC00
- BEAERRNO
- BEAEVT2
- BEALOGC
- BEAMCON
- BEAMMNU
- BEAMREQ
- BEAMSUB
- BEAMSVS
- BEAMUSR
- BEAMISN
- BEAPCON
- BEAPMNU
- BEAPREQ
- BEAPRERQ
- BEAPSVS
- BEAPUSR
- BEAPISN
- BEAREQST
- BEARQSUB
- BEASHUT
- EBCASC
- BEASUBS
- BEAGHBN
- BEASKIBM

- BEASKVAR
- CNVTISNC

Post Installation Tasks for Oracle Tuxedo Mainframe Adapter for TCP

This section includes the following topics:

- [Uninstalling Oracle Tuxedo Mainframe Adapter for TCP](#)
- [Distribution Libraries and Executables](#)

Uninstalling Oracle Tuxedo Mainframe Adapter for TCP

Uninstalling Oracle Tuxedo Mainframe Adapter for TCP in GUI-Mode

The following steps show uninstallation process in GUI mode.

1. Invoke uninstall TMA introduction panel

Go to <ORACLE_HOME>/oui/bin, run `runInstaller` on Unix/Linux platform. You can completely remove Oracle Tuxedo Mainframe Adapter for TCP 12c Release 2 (12.1.3) just by clicking "**Deinstall Products**".

2. Inventory panel

Select products and click **Remove** to remove those products from your system.

At the beginning, you may not see any product from Inventory panel. That is because your central inventory pointer (`/etc/oraInst.loc` or) is not set up. You will not get the installed product list unless you navigate to a couple of pages where you select the central inventory location and come back to the first page and then select **Installed products**.

Note: Do not choose "Oracle Universal Installer 12.1.0.1.0" if you just want to uninstall Oracle Tuxedo Mainframe Adapter for TCP; otherwise, all the Tuxedo components are uninstalled.

3. Confirm panel

To remove the products, click **Yes** in the Confirmation dialog. Click **No** to cancel the removal.

4. Uninstall complete panel

After uninstall finishes, this panel appears and reports whether it is successful.

5. Inventory panel

Check the product list from Inventory, finding if the product is already uninstalled successfully.

Uninstalling Oracle Tuxedo Mainframe Adapter for TCP in Silent Mode

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 Oracle Universal Installer 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.

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

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

If you want to hide the inventory dialog box during a deinstallation, you can specify the products to be removed in the `DEINSTALL_LIST` parameter of the response file; specify Oracle homes to be removed with the `REMOVE_HOMES` variable; specify the name of the current Oracle home name with `ORACLE_HOME_NAME`. For example, on a UNIX machine, enter:

```
./runInstaller -deinstall -silent  
"DEINSTALL_LIST={"tmabase_tcp", "12.1.3.0.0"}" ORACLE_HOME_NAME="OUIHome2"  
-responseFile <response>
```

Note: When uninstalling in silent mode, you need both `DEINSTALL_LIST` and response file.

Distribution Libraries and Executables

The Oracle Tuxedo Mainframe Adapter for TCP software contains the following libraries and executable programs. After installing the Tuxedo software, verify that these libraries and programs are installed on your system.

Solaris

Verify that the following files are installed by TMA TCP Gateway.

Directory	Files
/bin	GWIDOMAIN
/lib	ConvMVS.so.60 ConvMVSC.so.60 libctxcp_10.so WrapTPS.so.60 WrapTPSD.so.60
/locale/C	LIBGWI_CAT LIBGWI.text
/udataobj/codepage	00819x00037 00819x00273 00819x00278 00819x00284 00819x00285 00819x00297 00819x00500 00819x00860 00819x01047 00912x00870 none tuxedo

AIX

Verify that the following files are installed by TMA TCP Gateway.

Directory	Files
/bin	GWIDOMAIN
/lib	ConvMVS.so.60 ConvMVSC.so.60 libctxcp_10.so WrapTPS.so.60 WrapTPSD.so.60
/locale/C	LIBGWI_CAT LIBGWI.text
/udataobj/codepage	00819x00037 00819x00273 00819x00278 00819x00284 00819x00285 00819x00297 00819x00500 00819x00860 00819x01047 00912x00870 none tuxedo

Linux

Verify that the following files are installed by TMA TCP Gateway.

Directory	Files
/bin	GWIDOMAIN
/lib	ConvMVS.so.60 ConvMVSC.so.60 libctxcp_10.so WrapTPS.so.60 WrapTPSD.so.60

Directory	Files
/locale/C	LIBGWI_CAT LIBGWI.text
/udataobj/codepage	00819x00037 00819x00273 00819x00278 00819x00284 00819x00285 00819x00297 00819x00500 00819x00860 00819x01047 00912x00870 none tuxedo

Changing Resource Names

The default names for Oracle TMA TCP for CICS Mapsets, Programs, Transids, and VSAMs are listed in the following sections. If you want to use a different naming convention, use the procedure for changing names included after each table.

Note: You must have the C 370 compiler to customize resource names.

Contents of SOURCE(BEACFGSV)

The names in the following table are located in PDS "YOURHLQ".BEATCPC.SOURCE(BEACFGSV)

Name	Type	Description
BEAM	Transid	Main Menu
BEAPMNU	Program	Main Menu
BEAMMNU	Mapset	Main Menu
BEAC	Transid	Connection Administration
BEAPCON	Program	Connection Administration
BEAMCON	Mapset	Connection Administration
BEAR	Transid	Requester Administration
BEAPREQ	Program	Requester Administration

Name	Type	Description
BEAMREQ	Mapset	Requester Administration
BEAS	Transid	Service Name Administration
BEAP SVC	Program	Service Name Administration
BEAM SVC	Mapset	Service Name Administration
BEAU	Transid	User Connection Administration
BEAPUSR	Program	User Connection Administration
BEAMUSR	Mapset	User Connection Administration
BEAI	Transid	Inbound Service Administration
BEAPISN	Program	Inbound Service Administration
BEAMISN	Mapset	Inbound Service Administration
BEAVCON	VSAM	Connection Table
BEAVREQ	VSAM	Requester Table
BEAVSVC	VSAM	Service Name Table
BEAVUSR	VSAM	User Connection Table
BEAVISN	VSAM	Inbound Service Table
BEAPRERQ	Program	Pre-requester
BREQ	Transid	To start Requester
BEAL	TDQid	Log Message TD Queue
BEAA	Transid	Application Handler Transid
BEAHBDWN	TSQid	Handler Shutdown TS Queue

Changing Resource Names for SOURCE(BEACFGSV)

To change any of the default names shown in the previous table, complete the following steps.

1. Modify Program (BEACFGSV) located in PDS "YOURHLQ".BEATCPC.SOURCE

2. For the JCL (MAKBC) located in PDS "YOURHLQ".BEATCPC.JCL, complete the following steps.
 - a. Modify the JOBCARD to your standards.
 - b. Modify "YOURHLQ.BEATCPC.SOURCE" and "YOURHLQ.BEATCPC.OBJECT".
 - c. Submit MAKBC.
3. Verify the results.

Contents of CONTROL(CSDU)

The names in the following table are located in PDS "YOURHLQ".BEATCPC.CONTROL(CSDU)

Name	Type	Description
BEAPCON	Program	Connection administration
BEAPMNU	Program	Main menu
BEAPREQ	Program	Requester administration
BEAPSVL	Program	Outbound service administration
BEAPUSR	Program	User connection administration
BEAPISN	Program	Inbound service administration
BEASHUT	Program	Shutdown
BEAPRERQ	Program	Pre-Requester
BEAREQST	Program	Requester
BEACIC00	Program	Handler
BEAAPPLC	Program	Application Handler
BEAC	Transaction	Connection administration
BEAM	Transaction	Main menu
BEAR	Transaction	Requester administration
BEAS	Transaction	Outbound service administration

Name	Type	Description
BEAU	Transaction	User connection administration
BEAI	Transaction	Inbound service administration
BDWN	Transaction	Shutdown
BREQ	Transaction	Requester
BEAH	Transaction	Handler
BEAA	Transaction	Application Handler
BEAMCON	Mapset	Connection administration
BEAMMNU	Mapset	Main menu
BEAMREQ	Mapset	Requester administration
BEAMSVC	Mapset	Outbound service administration
BEAMUSR	Mapset	User connection administration
BEAMISN	Mapset	Inbound service administration
BEAVCON	File	Connection table
BEAVREQ	File	Requester table
BEAVSVC	File	Outbound service table
BEAVUSR	File	User connection table
BEAVISN	File	Inbound service table

Changing Resource Names for CONTROL(CSDU)

After changing resource names in BEACFGSV, you need to modify the file (CSDU) located in PDS "YOURHLQ".BEATCPC.CONTROL.

Changes to the CSDU member must correspond to changes in the BEACFGSV member.

You can also change the names of the BDWN and BEAH transactions. If the name of the BEAH transaction changes, the CICSHAND parameter in the GWICONFIG file of the Oracle TMA TCP Gateway must also reflect the change.

Oracle Tuxedo Mainframe Adapter for TCP Supported Platforms, Hardware Requirements, and Software Requirements

The following sections provide you with information about supported platforms, hardware requirements, and software requirements.

- [Oracle Tuxedo Mainframe Adapter for TCP Supported Platforms](#)
- [Oracle Tuxedo Mainframe Adapter for TCP Hardware Requirements](#)
- [Oracle Tuxedo Mainframe Adapter for TCP Software Requirements](#)

Oracle Tuxedo Mainframe Adapter for TCP Supported Platforms

Oracle Tuxedo Mainframe Adapter for TCP 12c Release 2 (12.1.3) is supported for the following platforms:

Table 1 TMA for TCP Supported Platforms

Platform
HP-UX 11.31 (64-bit) on Itanium
IBM AIX 6.1 (64-bit) on IBM PowerPC
Oracle Linux 5.6 (64-bit) on x86

Table 1 TMA for TCP Supported Platforms

Platform
Oracle Linux 5.9 (64-bit) on Exalogic 2.0
Oracle Solaris 10 (64-bit) SPARC

For more information, see [Oracle Tuxedo Mainframe Adapter for TCP Certified Platform Tables](#).

Oracle Tuxedo Mainframe Adapter for TCP Hardware Requirements

This section describes the minimum hardware requirements.

- An S/390 processor capable of supporting the required software products listed in the section [“Oracle Tuxedo Mainframe Adapter for TCP Software Requirements”](#).
- TCP/IP network connectivity

Oracle Tuxedo Mainframe Adapter for TCP Software Requirements

This section describes the minimum software requirements for the Oracle TMA TCP 12cR2.

Note: Support of TMA on higher releases of z/OS and its subsystems depends on IBM's warranty for the specific upper release's backward compatibility to the release that TMA formally supports. If IBM warrants such backward compatibility to the formally supported release, then the new release can be considered supported.

For more information, see [Oracle Tuxedo Mainframe Adapter for TCP Certified Platform Tables](#).

Oracle Tuxedo Mainframe Adapter for TCP Supported Stacks

Oracle Tuxedo Mainframe Adapter for TCP is designed to work with IBM TCP for MVS stack product.

Oracle TMA TCP for IMS

The following software runs with Oracle TMA TCP for IMS:

- IBM IMS/TM 9, 10, 11, 12, 13
- IBM z/OS 1.11, 1.12, 1.13, 2.1

Oracle TMA TCP for CICS

The following software runs with Oracle TMA TCP for CICS:

- IBM CICS TS 3.2, 4.1, 4.2
- TCP/IP for MVS Version 3.2, 4.1, 4.2 with CICS Socket Interface
- C/370 Runtime Library Version 2.1 or higher
- IBM z/OS 1.11, 1.12, 1.13, 2.1

