Using Microsoft Transaction Server with Oracle8

Release 8.1.6

January 2000

Part No. A73029-01

ORACLE®

Using Microsoft Transaction Server with Oracle8, Release 8.1.6

Part No. A73029-01

Primary Author: Mark Kennedy

Contributors: Michael Hussey, Greg Ishii, Eng Khor, Sergio Nakai, and Vivek Raja

Contents

Contact Us! vii

How to Contact Oracle Technical Publications viii
How to Contact Oracle Support Services ix
Resources for Oracle Partners and Developers xiv

Before You Begin xvii

Prerequisites xviii
Intended Audience xviii
How This Guide Is Organized xviii
Documentation and Code Conventions Explained xx

1 Using Microsoft Transaction Server with an Oracle8 Database

Using Microsoft Transaction Server with an Oracle8 Database 1-2
What is Microsoft Transaction Server? 1-2
How are Microsoft Transaction Server and an Oracle8 Database Integrated? 1-3
How Do I Get Started? 1-5

2 Installing and Migrating Oracle Products

Oracle Product Layouts 2-2
What Do You Need to Install? 2-3
Migrating From a Previous Installation of Oracle Service for MTS 2-5

3 Creating a New Oracle Service for MTS

What Do You Need to Configure? 3-2
Verifying Initialization Parameter File Values 3-3
Verifying the Oracle Service for MTS User Name 3-4
Creating a New Oracle Service for MTS 3-6

4 Using the Microsoft Application Demo

Using OCI with the Microsoft Application Demo 4-2
Microsoft Application Demo Overview 4-3
Ensuring the Oracle8 Database Includes the Proper Tables 4-3
Running the Microsoft Application Demo 4-5
Using Oracle's ODBC Driver with the Microsoft Application Demo 4-7

5 Programming with Microsoft Transaction Server and an Oracle8 Database

Component Integration In a Transaction 5-2
Programming with Microsoft Transaction Server and an Oracle8 Database 5-4
Registering Microsoft Transaction Server Components 5-4
Components Running in a Microsoft Transaction Server-Coordinated Transaction 5-6
Components Not Running in a Microsoft Transaction Server-Coordinated Transaction, But Using MS DTC 5-7
Integrating OCI with Microsoft Transaction Server 5-8
OCI Function Overview 5-8
OraMTSSvcGet() 5-12
OraMTSSvcRel() 5-15
OraMTSSvcEnlist() 5-16
OraMTSSvcEnlistEx() 5-19
OraMTSTransTest() 5-20
OraMTSOCIErrGet() 5-21
Integrating ODBC with Microsoft Transaction Server 5-22
Setting the Connection Attribute 5-22
Using Oracle's ODBC Driver 5-22
Using Microsoft's Oracle ODBC Driver 5-24
Integrating Oracle Objects for OLE with Microsoft Transaction Server 5-25
Integrating Oracle Provider for OLE DB with Microsoft Transaction Server 5-25
Integrating Other APIs with Microsoft Transaction Server 5-25

6 Tuning Oracle Service for MTS Performance

Automatically Restarting Oracle Service for MTS 6-2
Improving Performance 6-3
Managing Connections 6-4
Increasing the Transaction Timeout Parameter 6-10
Changing Initialization Parameter File Settings 6-12

7 Troubleshooting

Using Trace Files 7-2
Correcting Windows NT Explorer Crashes 7-5
Identifying Oracle Service for MTS Startup Problems 7-5
Using the Event Viewer 7-7
Accessing the Event Viewer 7-7
Reading the Event Viewer 7-8
Using the Event Viewer 7-9
Starting MS DTC 7-10
Viewing Oracle Service for MTS Configuration Information 7-11
Correcting Net8 Changes that Impact Connection Pooling 7-13
Ensuring Oracle Service for MTS Connectivity and Security 7-14
Changing a Server Package's Windows NT User Account 7-16
Common Questions 7-17

A Manually Creating Oracle Service for MTS Users

Manually Creating the MTSSYS User A-2
Manually Creating a Custom Oracle Service for MTS User A-5
Creating a New Operating System-Authenticated Oracle Service for MTS A-8
Enabling an Existing Oracle Service for MTS to be Operating System-Authenticated A-11

B Deleting or Modifying an Existing Oracle Service for MTS

How Do I Modify or Delete an Oracle Service for MTS? B-2
Preparing to Modify or Delete an Existing Oracle Service for MTS B-3
Deleting an Existing Oracle Service for MTS B-5
Modifying Oracle Service for MTS Connection Information B-7
Modifying Oracle Service for MTS Trace File and MS DTC Information B-10
Manually Deleting or Modifying the Oracle Service for MTS with the Registry B-13
Deleting Roles and Privileges of an Inactive Oracle Service for MTS User B-16

Index

Contact Us!

Using Microsoft Transaction Server with Oracle8, Release 8.1.6

Part No. A73029-01

This document describes how to contact Oracle Corporation if you have issues with the documentation or software. It also provides a list of useful resources for Oracle partners and developers.

Read the section...	If you...
"How to Contact Oracle Technical Publications" on page viii	Have issues with Documentation
"How to Contact Oracle Support Services" on page ix	Have issues with Software
"Resources for Oracle Partners and Developers" on page xiv	Want to join an Oracle partner or application developer program

How to Contact Oracle Technical Publications

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this guide?
- Do you have suggestions for improvement? Please indicate the chapter, section, and page number (if available).

You can send comments regarding documentation in the following ways:

- Electronic mail - ntdoc@us.oracle.com
- FAX - (650) 506-7370 Attn: Oracle Windows NT Server Documentation
- Postal service:
  Oracle Corporation
  NT Server Documentation Manager
  500 Oracle Parkway MS 1OP8
  Redwood Shores, CA 94065
  USA

If you would like a reply, please provide your name, address, and telephone number.

How to Contact Oracle Support Services

Please copy this form and distribute within your organization as necessary.

Oracle Support Services can be reached at the following telephone numbers and Web sites. The hours of business are detailed in your support contract and the *Oracle Customer Support Guide* in your kit.

Oracle Support Services In...	Call...
United States of America	+ (650) 506-1500 for customers with support contracts. + (650) 506-5577 to obtain a support contract.
Europe	+44 1344 860 160 or the local support center in your country.
All other locations	The telephone number for your country listed at the following Web site: `http://www.oracle.com/support/contact_us/sup_hot_phone.html` Oracle Support Services telephone numbers are also listed in the *Oracle Customer Support Guide* in your kit.

Please complete the following checklist before you call. If you have this information ready, your call can be processed much quicker.

❑ Your CPU Support Identification Number (CSI Number) if applicable.

_____

_____

❑ The hardware name on which your application is running.

_____

_____

_____

- ❑ The operating system name and release number on which your application is running.
  - To verify the operating system version on Windows NT, enter the following at the MS-DOS command prompt:

    ```
    C:\> WINMSD
    ```

    The *Windows NT Diagnostics* dialog box displays the operating system and Service Pack version.

_____

_____

_____

- ❑ The release numbers of the Oracle Server and associated products involved in the current problem. For example, Oracle8*i* Enterprise Edition release 8.1.6.0.0 and Oracle Enterprise Manager release 2.1.0.0.0.
  - To verify the release number of the Oracle Server, connect to the database using a tool such as SQL*Plus. The release number is displayed. For example:

    ```
    Connected to:
    Oracle8i Enterprise Edition Release 8.1.6.0.0 - Production
    With the Partitioning and Java options
    PL/SQL Release 8.1.6.0.0 - Production
    ```

_____

_____

_____

_____

- ❑ The third-party software version you are using.
  - To verify an application version, from the application's Help menu, select About...

_____

_____

_____

- ❑ The exact error codes and messages. Please write these down as they occur. They are critical in helping Oracle Support Services to quickly resolve your problem. Note whether there were no errors reported.

_____

_____

_____

_____

_____

- ❑ A description of the issue, including:
  - **What happened?** For example, the command used and its result.

    _____

    _____

    _____

  - **When did it happen?** For example, during peak system load, or after a certain command, or after an operating system upgrade. In addition, what was happening when the problem occurred?

    _____

    _____

    _____

  - **Where did it happen?** For example, on a particular system, or within a certain procedure or table.

    _____

    _____

    _____

- **What is the extent of the problem?** For example, production system unavailable, or moderate impact but increasing with time, or minimal impact and stable.
- Did the problem affect one user, several users, or all users?
- Has anything changed? For example, if this is an operation that used to work and now fails, what is different? Can you undo any recent changes, to verify whether they are relevant to the issue?

_____

_____

_____

- **Can the problem be reproduced?** This is a critical question for support analysts. For example, did the problem recur on the same system, under the same circumstances? Can the problem be reproduced on another system? Additionally:
- Does installing a software component fail on all client machines, or just one?
- Do all clients fail to connect to the server, or just one?
- If you are able to restart the server or database, does restarting the database or rebooting the server or client machine (if applicable) make a difference?

_____

_____

_____

- Keep copies of the Oracle alert log, any trace files, core dumps, and redo log files recorded at or near the time of the incident. Oracle Support Services may need these to further investigate your problem.

  To help analyze problems:

  - Archive or delete old alert logs. When the database is started without an alert log, a new one is created. In some cases, if you force the problem to recur with a new alert log, the timestamps for the recorded events may indicate which events are relevant.
  - Archive or delete old trace files. To check whether the file was modified, right-click and select Properties. The *Properties* dialog box displays the modification date.
  - Check the operating system error logs, especially the System log and Application log. These files are relevant to the Oracle Server. To view these files, from the Start menu, choose Programs > Administrative Tools > Event Viewer, and choose System or Application from the Log main menu.

Resources for Oracle Partners and Developers

This section provides information on partner programs and resources for Oracle database administrators and application developers.

Information Source	Description
Oracle Corporation Home Page `http://www.oracle.com`	This Web site is the starting point for general information on Oracle Corporation.
Alliance Online `http://alliance.oracle.com`	Oracle provides leading-edge technology, education, and technical support that enables you to effectively integrate Oracle into your business. By joining the Oracle Partner Program, you demonstrate to customers that you are committed to delivering innovative Oracle-based solutions and services. The greater your commitment to Oracle, the more we can help you grow your business. It's that simple. The value you derive is associated directly with your level of commitment.
Oracle Education `http://education.oracle.com/`	Customers come to Oracle Education with a variety of needs. You may require a complete curriculum based on your job role to enable you to implement new technology. Or you may seek an understanding of technology related to your key area of responsibility to help you meet technical challenges. You may be looking for self-paced training that can be used as an ongoing resource for reference and hands-on practice. Or, you may be interested in an overview of a new product upgrade. Whatever your training need, Oracle Education has the solution.
Oracle Technology Network `http://technet.oracle.com/`	The Oracle Technology Network is your definitive source for Oracle technical information for developing for the Internet platform. You will be part of an online community with access to free software, Oracle Technology Network-sponsored Internet developer conferences, and discussion groups on up-to-date Oracle technology. Membership is free.
Oracle Store `http://oraclestore.oracle.com/`	This is Oracle's online shopping center. Come to this site to find special deals on Oracle software, documentation, publications, computer-based training products, and much more.

Information Source	Description
Oracle Support Services' Support Web Center `http://www.oracle.com/support/`	Oracle Support Services offers a range of programs so you can select the support services you need and access them in the way you prefer: by telephone, electronically, or face to face. These award-winning programs help you maintain your investment in Oracle technology and expertise. Here are some of the resources available in the Support Web Center:
Oracle*MetaLink* `http://www.oracle.com/support/elec_sup/index.html`	Oracle*MetaLink* is Oracle Support Services' premier Web support service. It is available to Oracle*metals* customers (Gold, Silver, Bronze), 24 hours a day, seven days a week.
Oracle*Lifecycle* `http://www.oracle.com/support/sup_serv/lifecycle/index.html`	Oracle*Lifecycle* is designed to deliver customized, industry-focused, full life-cycle support solutions that enable industry leaders to use Oracle technology to make smart business decisions, achieve operational excellence, and succeed in their markets.
Expert*ONLINE* `http://www.oracle.com/support/sup_serv/online/index.html`	Oracle Support Services has launched a new line of services called Expert*ONLINE*. These services provide online database administration for companies looking to supplement their existing DBA staff or fill a DBA role. Services range from Expert*DETECT*, a monitoring, diagnostic, and recommendation service, to Expert*DBA*, a full online database administration service.
Virtual Support Analyst (VSA) `http://www.oracle.com/support/sup_serv/vsa_start.html`	VSA is Oracle's Internet e-mail service; it is available to U.S. customers with an Oracle*metals* support agreement. With VSA, you can initiate a request for assistance through e-mail, bypassing the queues you may encounter when using telephone support. VSA also enables you to access Oracle's bug database.
Customer Service `http://www.oracle.com/support/cus_serv/index.html`	This site provides resources to make your interactions with Oracle as easy as possible. Among the things you can do are: ■ Learn what is a CPU Support Identification (CSI) number ■ Update your technical contact information ■ Find out whom to contact for invoice and collection issues ■ Request product update shipments ■ Access a glossary of Oracle Support Services terms

Information Source	Description
U.S. Customer Visit Program `http://www.oracle.com/support/cus_serv/cus_visit.html`	This U.S.-based program has been established to help our customers understand and obtain maximum benefit from the support services they have purchased. The visit typically offers a customized orientation presentation, a comprehensive overview and demonstration of Oracle's electronic services, and helpful tips on working more effectively with Oracle Support Services.
Support Web Center Library `http://www.oracle.com/support/library/index.html`	This site contains articles, guides, and other documentation to help you leverage the wealth of knowledge and reference material that Oracle Support Services produces.

Before You Begin

This guide is your primary source of introduction, installation, configuration, usage, and administration information for using Microsoft Transaction Server with Oracle8.

Specific topics discussed are:

- Prerequisites
- Intended Audience
- How This Guide Is Organized
- Documentation and Code Conventions Explained

Prerequisites

This guide assumes that you are familiar with Windows NT and have installed and tested it on your personal computer (PC) and network hardware.

Intended Audience

This guide is necessary for anyone using Oracle8 with Microsoft Transaction Server.

How This Guide Is Organized

This guide is organized as follows:

Chapter 1, "Using Microsoft Transaction Server with an Oracle8 Database"

This chapter describes the integration of Microsoft Transaction Server and an Oracle8 database.

Chapter 2, "Installing and Migrating Oracle Products"

This chapter describes installation and migration requirements for your Microsoft Transaction Server and Oracle8 database environments.

Chapter 3, "Creating a New Oracle Service for MTS"

This chapter describes how to create a new Oracle Service for MTS that uses the user name MTSSYS.

Chapter 4, "Using the Microsoft Application Demo"

This chapter describes how to use the sample Microsoft COM-based application demo that is integrated with Microsoft Transaction Server.

Chapter 5, "Programming with Microsoft Transaction Server and an Oracle8 Database"

This chapter describes how to program with Microsoft Transaction Server and an Oracle8 database.

Chapter 6, "Tuning Oracle Service for MTS Performance"

This chapter describes how to tune Oracle Service for MTS performance.

Chapter 7, "Troubleshooting"

This chapter describes how to troubleshoot Oracle Service for MTS problems.

Appendix A, "Manually Creating Oracle Service for MTS Users"

This appendix describes how to manually create the MTSSYS user, a custom user, or an operating system-authenticated user.

Appendix B, "Deleting or Modifying an Existing Oracle Service for MTS"

This appendix describes how to delete or modify an existing Oracle Service for MTS.

Documentation and Code Conventions Explained

The following conventions are used in this guide:

Convention	Example	Meaning
All uppercase plain	C:\ORACLE\ORA81	Indicates command names, SQL reserved words, and keywords, as in ALTER DATABASE. All uppercase plain is also used for directory names and file names.
Italic	■ Italic is used to indicate a variable: *file name* ■ Italic is used for emphasis: The WHERE clause may be used to *join* rows in different tables.	Indicates a value that you must provide. For example, if a command asks you to type *file name*, you must type the actual name of the file. Italic is also used for emphasis in the text and to indicate the titles of other guides.
Square brackets [ ]	*X*:\[*PATHNAME*]\ORACLE*HOME_NAME*	Encloses optional items. For example, when you create an Optimal Flexible Architecture (OFA)-compliant Oracle home directory, you can place an optional pathname before the \ORACLE pathname. Square brackets also indicate a function key, for example [Enter].
Choose Start >	Choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant	How to start a program. For example, to start Net8 Assistant, you must click the Start button on the taskbar and then choose Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
C:\>	C:\ORACLE\ORADATA>	Represents the Windows NT command prompt of the current hard disk drive. Your prompt reflects the subdirectory in which you are working. This is referred to as the MS-DOS command prompt in this guide.
Backslash (\) before a directory name	\ORADATA	Indicates that the directory is a subdirectory of the root directory.

Convention	Example	Meaning
ORACLE_HOME and *ORACLE_BASE*	Go to the *ORACLE_HOME*\ADMIN directory	In releases prior to 8.1, when you installed Oracle8, all subdirectories were located under a top level *ORACLE_HOME* directory, that by default was: ■ C:\ORANT for Windows NT ■ C:\ORAWIN95 for Windows 95 ■ C:\ORAWIN98 for Windows 98 ■ C:\ORAWIN for Windows 3.1 or whatever you called your Oracle home. In this OFA-compliant release, all subdirectories are no longer under a top level *ORACLE_HOME* directory. There is a new top level directory called *ORACLE_BASE* that by default is C:\ORACLE. If you install Oracle8 release 8.1.6 on a clean computer (that is, there is no other Oracle software on the computer), the default setting for the first Oracle home directory is C:\ORACLE\ORA81. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. See your *Oracle8i Administrator's Guide for Windows NT* for additional information on OFA compliances and for information on installing Oracle products in non-OFA compliant directories.
%*ORACLE_HOME*%	SQL> @%ORACLE_HOME%\ADMIN*DB_NAME*\ADHOC\CATALOG.SQL	In SQL*Plus commands, you may see %ORACLE_HOME%. SQL*Plus is able to locate your Oracle Home directory using the %ORACLE_HOME% variable. This convention can also be used in Server Manager, Export Utility, and Import Utility.

Convention	Example	Meaning
HOME_NAME	Oracle*HOME_NAME*TNSListener	Represents the Oracle home name. The home name can be up to sixteen alphanumeric characters. The only special character allowed in the home name is the underscore.
HOME*ID*	HOME0, HOME1, HOME2	Represents a unique registry subkey for each Oracle home directory in which you install products. A new HOME*ID* is created and incremented each time you install products to a different Oracle home directory on one computer. Each HOME*ID* contains its own configuration parameter settings for installed Oracle products.
Symbols	period . comma , hyphen - semicolon ; colon : equal sign = backslash \ single quote ’ double quote " parentheses ()	Symbols other than brackets and vertical bars must be entered in commands exactly as shown.

1

Using Microsoft Transaction Server with an Oracle8 Database

This chapter describes Microsoft Transaction Server and Oracle8 database integration.

Specific topics discussed are:

- Using Microsoft Transaction Server with an Oracle8 Database
- What is Microsoft Transaction Server?
- How are Microsoft Transaction Server and an Oracle8 Database Integrated?
- How Do I Get Started?

Using Microsoft Transaction Server with an Oracle8 Database

This chapter describes Oracle8 database integration with Microsoft Transaction Server. This integration enables developers to:

- Develop and deploy Component Object Model (COM)-based applications using Microsoft Transaction Server with an Oracle8 database
- Have application transactions for an Oracle8 database coordinated by Microsoft Transaction Server

What is Microsoft Transaction Server?

Microsoft Transaction Server is a proprietary COM-based transaction processing system that runs on an Internet or network server. Microsoft Transaction Server is used to deploy and manage application and database transaction requests on behalf of a client computer. Microsoft Transaction Server provides the following:

- An ActiveX/Distributed Component Object Model (DCOM)-based programming model to develop distributed applications and a runtime environment in which to deploy these applications
- Atomicity Consistency Isolation Durability (ACID) properties to components in transactions
- Access to performance enhancing features such as component caching and database connection pooling

Microsoft Transaction Server is a component of the three-tiered, server-centric architecture model. This enables the presentation, business logic, and data elements of applications to be clearly separated onto different computers connected in a network.

> **See Also:** Microsoft's documentation for additional information on Microsoft Transaction Server

How are Microsoft Transaction Server and an Oracle8 Database Integrated?

Without any special integration, you can deploy a COM component in Microsoft Transaction Server that connects to an Oracle8 database release 8.0.6 or higher. However, the following features require the installation of Oracle Services for Microsoft Transaction Server:

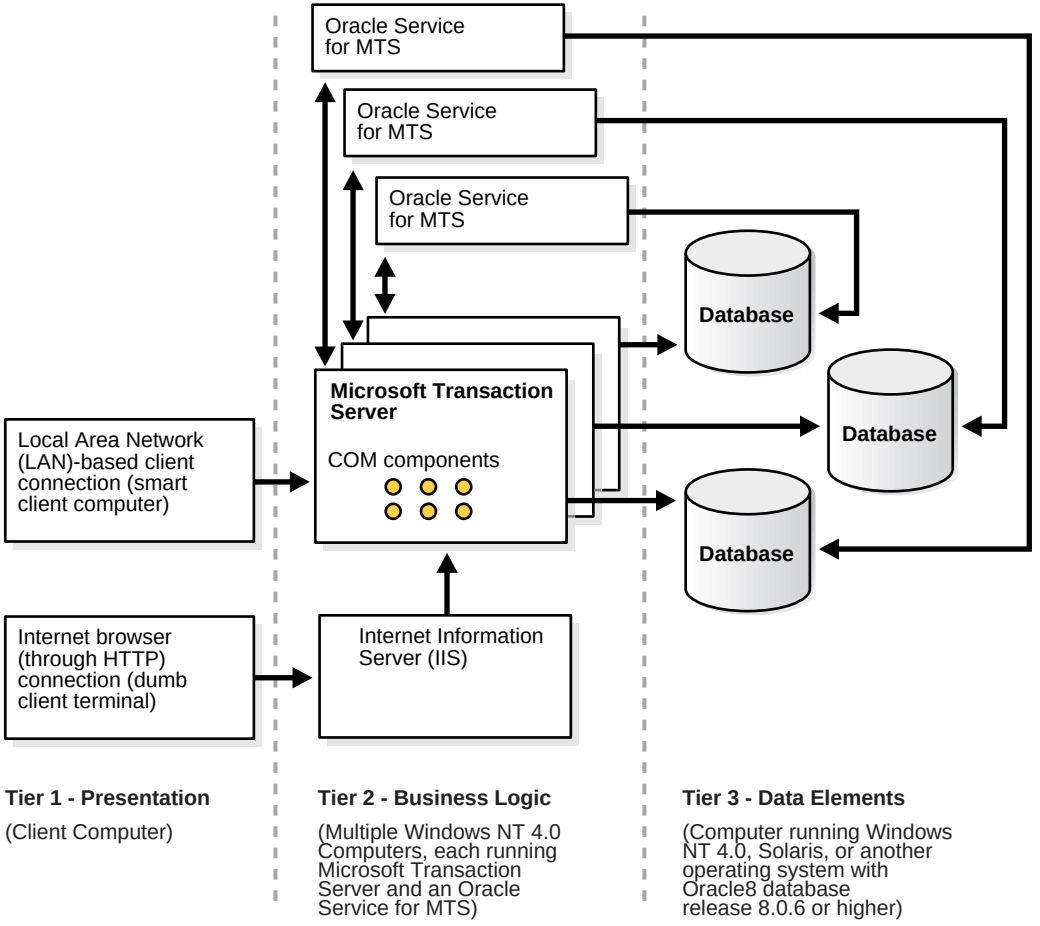
- The COM component to be marked as transactional (using the *Properties* dialog box of the component in the Microsoft Management Console Explorer) and have Microsoft Transaction Server control the transaction
- Client-side connection pooling used in Microsoft Transaction Server

After Oracle Services for Microsoft Transaction Server is installed, you create a special Windows NT service (referred to as an Oracle Service for MTS) for each Oracle8 database to which your COM component connects. Use the Oracle Manager for MTS Services snap-in located in the Microsoft Management Console Explorer to create and manage these services. The Oracle Service for MTS provides the following:

- Maps all of Microsoft's proprietary Object, Linking, and Embedding (OLE) transactions to Oracle transactions
- Handles any recovery scenarios

The COM component can be created using the Oracle Call Interface (OCI), Oracle Objects for OLE (OO4O), Open Database Connectivity (ODBC), or Oracle Provider for OLE DB. Additional application program interfaces (APIs) may be integrated in the future. For a given Oracle8 database, you only need one Oracle Service for MTS on the network.

This illustration shows Microsoft Transaction Server and Oracle8 database integration.



How Do I Get Started?

You are now ready to use Microsoft Transaction Server with an Oracle8 database. To get started quickly, follow these chapters in the *exact* order listed:

To...	See...
Install the Oracle and Microsoft products required for Microsoft Transaction Server and Oracle8 database integration or migrate from a previous version of Oracle Service for MTS	Chapter 2, "Installing and Migrating Oracle Products"
Create a new Oracle Service for MTS	Chapter 3, "Creating a New Oracle Service for MTS"
Run the Microsoft application demo	Chapter 4, "Using the Microsoft Application Demo". The demo consists of an application that: ■ uses COM components marked as transactional ■ enlists in a transaction controlled by Microsoft Transaction Server ■ accesses the Oracle8 database through an Oracle Service for MTS
Create your own COM-based applications	Chapter 5, "Programming with Microsoft Transaction Server and an Oracle8 Database" for instructions on using OCI, OO4O, ODBC, or Oracle Provider for OLE DB with your COM-based applications.

2

Installing and Migrating Oracle Products

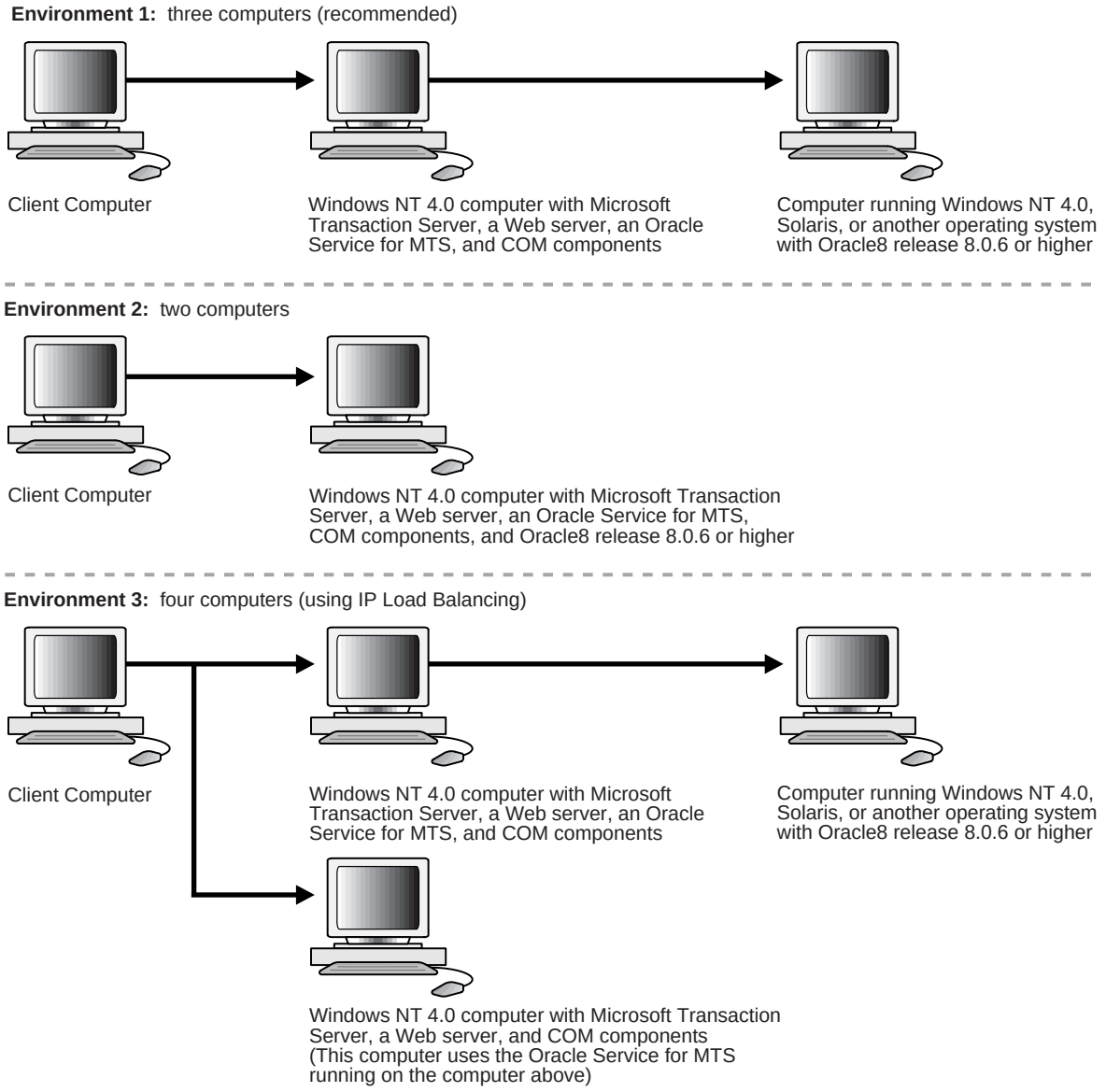
This chapter describes installation and migration requirements for your Microsoft Transaction Server and Oracle8 database environment.

Specific topics discussed are:

- Oracle Product Layouts
- What Do You Need to Install?
- Migrating From a Previous Installation of Oracle Service for MTS

Oracle Product Layouts

This figure shows three possible installation layouts for your Microsoft Transaction Server and Oracle8 database environment:



What Do You Need to Install?

The table on the following page lists the Oracle and non-Oracle products you must install. After reviewing your installation responsibilities, see your Oracle installation documentation for instructions on installing the required Oracle products. On Windows NT, for example, these instructions are included in *Oracle8i Installation Guide for Windows NT* and *Oracle8i Client Installation Guide for Windows NT*. Key guidelines to follow are:

- You need at least 128 MB of RAM if the Oracle8 database, Microsoft Transaction Server, and Oracle Services for Microsoft Transaction Server are installed on the same computer.
- Install the Oracle Services for Microsoft Transaction Server on a stable computer that is always running, reliable, fast, and easily recoverable. If the service is not running, a connection fails if the component is marked transactional.
- One installation of the Oracle Services for Microsoft Transaction Server can create multiple services.
- There *must be only one* Oracle Service for MTS for each database instance. This service does not need to be on the same computer as the Oracle8 database.
- Microsoft Transaction Server must be installed for the Oracle Service for MTS to run. However, the Oracle Service for MTS can use a Microsoft Distributed Transaction Coordinator (MS DTC) on a different computer. Configure this when creating or modifying the Oracle Service for MTS. See "Component Integration In a Transaction" on page 5-2 for additional information on the role of MS DTC.

Note: Microsoft Transaction Server integration with an Oracle8 database is *not* currently supported on Windows 95 or Windows 98.

Note: To use the Oracle Manager for MTS Services snap-in online help in the Microsoft Management Console, you must download and install version 1.2 of Microsoft HTML Help. If you use an earlier version, you can have problems displaying the online help.

	What Products Must Be Installed?	
On The...	**Oracle Products**	**Non-Oracle Products**
Client computer	■ None required	■ Fully-functioning networking protocol software ■ Web browser
Windows NT computer where Microsoft Transaction Server is installed	■ Oracle Services for Microsoft Transaction Server[1] ■ Net8 Client[2] ■ Oracle Objects for OLE[3] ■ ODBC[3] ■ Oracle Provider for OLE DB[3] ■ Net8 Assistant[2] ■ SQL*Plus	■ Microsoft Internet Information Server (IIS)[4] ■ Microsoft Transaction Server version 2.0[5] ■ Windows NT 4.0 Service Pack 4 or greater or a special distributed transaction coordinator stand-alone setup program[6] ■ Fully-functioning networking protocol software
Computer where the Oracle8 database is installed	■ Oracle8 Server (the Oracle8 database)[7] ■ Net8 Server ■ SQL*Plus	■ Fully-functioning networking protocol software

[1] See the *Oracle8i Installation Guide for Windows NT* or *Oracle8i Client Installation Guide for Windows NT* for installation instructions.

[2] Automatically installed with the Oracle Services for Microsoft Transaction Server.

[3] ODBC, Oracle Provider for OLE DB, and Oracle Objects for OLE are only required if you are building or using components with which they are required. If you are building or using components only with OCI, then ODBC, Oracle Provider for OLE DB, and Oracle Objects for OLE are not required.

[4] An applications server like IIS is not required, but is beneficial if using Microsoft Transaction Server. Install the applications server on the same computer as Microsoft Transaction Server. The applications server you choose must support extensions.

[5] Microsoft Transaction Server is currently available as part of the Microsoft Windows NT 4.0 Option Pack. Microsoft Management Console is automatically installed with Microsoft Transaction Server.

[6] After installing Microsoft Transaction Server 2.0, Oracle Corporation recommends that you install Windows NT 4.0 Service Pack 4 or higher. If you cannot install this, you can download a special DTC stand-alone setup program that enables the Oracle8 database and Microsoft Transaction Server to interact properly. Use a Web browser to access this Web site:

`ftp.microsoft.com/bussys/distapps/mts/public-fixes/usa/dtc/oracle/readme.htm`

See the README file at this location for instructions on download and executing the update file. Note that installing Service Pack 4 or higher also corrects several EXPLORER.EXE problems. If you have already installed Service Pack 4 or higher, do *not* install this special MS DTC update file.

[7] If your Oracle8 database is a release prior to 8.0.6, you cannot take advantage of the integration described in this guide. See section "Using Microsoft's Oracle ODBC Driver" on page 5-24 for a description of what functionality is available to you.

Note: Oracle Service for MTS is supported on Windows 2000.

Migrating From a Previous Installation of Oracle Service for MTS

The Oracle Service for MTS migration procedures to follow depend upon whether you are migrating or deleting your release 8.1.3, 8.1.4, or 8.1.5 Oracle8 database. This table references the sections of this guide to follow for both situations:

If You...	Then...
Migrated your release 8.1.3, 8.1.4, or 8.1.5 Oracle8 database to release 8.1.6	You have two choices: **Choice 1: Delete and Re-create:** ■ Delete the Oracle Service for MTS by carefully following the procedures in "Deleting an Existing Oracle Service for MTS" on page B-5. ■ Create a new Oracle Service for MTS by following the procedures in Chapter 3, "Creating a New Oracle Service for MTS". **Choice 2: Rename:** ■ If you are using release 8.1.3 or 8.1.4, rename the registry parameters for the existing Oracle Service for MTS. All parameter names that began with OMTS in releases 8.1.3 or 8.1.4 now begin with ORAMTS. See section "Manually Deleting or Modifying the Oracle Service for MTS with the Registry" on page B-13 for information on the registry location of these parameters. You do not need to rename these parameters if your release is 8.1.5.
Deleted your release 8.1.3, 8.1.4, or 8.1.5 Oracle8 database and instead created a new installation of release 8.1.6	*No* Oracle Service for MTS migration is required. You can: ■ Delete the Oracle Service for MTS. Note that it is not a requirement that you follow the procedures in Appendix B, "Deleting or Modifying an Existing Oracle Service for MTS". ■ Create a new Oracle Service for MTS by following the procedures in Chapter 3, "Creating a New Oracle Service for MTS".

3

Creating a New Oracle Service for MTS

This chapter describes how to create a new Oracle Service for MTS that connects to the Oracle8 database with the user name MTSSYS.

Specific topics discussed are:

- What Do You Need to Configure?
- Verifying Initialization Parameter File Values
- Verifying the Oracle Service for MTS User Name
- Creating a New Oracle Service for MTS

What Do You Need to Configure?

You must configure your Microsoft Transaction Server and Oracle8 database environments after installing the Oracle Services for Microsoft Transaction Server. Review this table to identify what you need to configure:

On The...	Is Any Configuration Required?
Client computer	No.
Computer where the Oracle8 database is installed	It depends upon your database environment. There are two areas that you may want to verify: **1.** Verify several Oracle8 database initialization parameter file settings. See section: "Verifying Initialization Parameter File Values" on page 3-3 **2.** Verify that the special Oracle Service for MTS user name MTSSYS exists in your Oracle8 database. See section: "Verifying the Oracle Service for MTS User Name" on page 3-4
Windows NT computer where Microsoft Transaction Server is installed	Yes. Create a new Oracle Service for MTS for each Oracle8 database to which COM components can connect. See section: "Creating a New Oracle Service for MTS" on page 3-6

Additional Information: Non-Oracle products may also require configuration. See the documentation that came with those products for any configuration responsibilities.

Verifying Initialization Parameter File Values

You can verify that several initialization parameters are set to sufficient values. The values to which to set these parameters are determined by the workload environment of your database.

To verify initialization parameter file values:

1. Go to the computer on which the Oracle8 database is installed.
2. Start SQL*Plus:

   ```
   C:\> SQLPLUS
   ```

3. Connect with the INTERNAL user name:

   ```
   ENTER USER-NAME: INTERNAL
   ```

4. Check the value for the SESSIONS parameter:

   ```
   SQL> SHOW PARAMETER SESSIONS
   ```

5. Check the value for the PROCESSES parameter:

   ```
   SQL> SHOW PARAMETER PROCESSES
   ```

   The current settings for both parameters should be sufficient for running the Microsoft application demo described in Chapter 4, "Using the Microsoft Application Demo". For creating and deploying your own COM-based applications, the values to which to set these parameters depend upon the anticipated workload for your database environment. For example, if you anticipate 100 concurrent connections to your Oracle8 database, consider setting both values to 200 to account for any system overload. Ensure that you do not set these parameters too high, as they are resource-intensive. See *Oracle8i Reference* and *Oracle8i Administrator's Guide* for information on these parameters.

6. If you want to set these parameters to different values, see section "Changing Initialization Parameter File Settings" on page 6-12 for instructions.

Verifying the Oracle Service for MTS User Name

You must assign a user name to the new Oracle Service for MTS. A special Oracle Service for MTS user name is typically included in your Oracle8 database for Windows NT:

User Name	Password	Description
MTSSYS	MTSSYS	User name with: ■ CONNECT, RESOURCE, and SELECT_CATALOG_ ROLE database roles ■ FORCE ANY TRANSACTION, CREATE PUBLIC SYNONYM, and DROP PUBLIC SYNONYM database privileges ■ SELECT, INSERT, UPDATE, and DELETE rights on the Oracle Service for MTS table in the Oracle8 database

This section describes how to connect as user name MTSSYS with the new Oracle Service for MTS. Review the table below to determine if your Oracle8 database includes this user name:

If You Create Your Oracle8 Database Through These Methods...	MTSSYS is...
■ Through the Typical or Minimal installation type of the Oracle8*i* Enterprise Edition or Oracle8*i* database type ■ Through the Typical option of the Oracle Database Configuration Assistant where you select to copy existing database files from the CD ■ Manually using the database creation script available on your Oracle 8.1 CD-ROM (On Windows NT, this script is named BUILDALL.SQL and is described in the *Oracle8i Administrator's Guide for Windows NT*) ■ Manually using your own SQL script, and then run the SCOTT.SQL and ORAMTS.SQL scripts in that order against the database[1]	Automatically included on Windows NT

If You Create Your Oracle8 Database Through These Methods...	MTSSYS is...
■ Through the Custom option of the Oracle Database Configuration Assistant ■ Through the Typical option of the Oracle Database Configuration Assistant where you select to create new database files ■ Manually using your own SQL script, and do *not* run ORAMTS.SQL against the database ■ Using the CREATE DATABASE syntax in SQL*Plus line mode ■ In *any* available method on Solaris or any other operating system other than Windows NT ■ In *any* available method with Oracle8 database release 8.0.6. **Note:** Release 8.0.6 users must obtain and run the ORAMTS.SQL script included with release 8.1.*x*.	Not created. See "Manually Creating the MTSSYS User" on page A-2 or "Manually Creating a Custom Oracle Service for MTS User" on page A-5

[1] If you run ORAMTS.SQL before SCOTT.SQL or do not run SCOTT.SQL at all, numerous error messages appear when running ORAMTS.SQL. Ignore these messages. The product functions properly, but the sample application demo described in section "Using OCI with the Microsoft Application Demo" on page 4-2 will not work.

To ensure that your Oracle8 database includes MTSSYS, try to log on with this user name. In addition, Oracle Corporation strongly recommends that you change the MTSSYS password.

To verify the MTSSYS user name and change the password:

1. Go to the computer on which your Oracle8 database is installed.
2. Start SQL*Plus:

   ```
   C:\> SQLPLUS
   ```

3. Connect with the MTSSYS user name and MTSSYS password:

   ```
   ENTER USER-NAME: MTSSYS/MTSSYS
   ```

   If the connection is successful, this verifies that MTSSYS is included in your Oracle8 database. If the connection is not successful, you must create an Oracle Service for MTS user name. See section "Manually Creating the MTSSYS User" on page A-2 or section "Manually Creating a Custom Oracle Service for MTS User" on page A-5.

4. Change the password for the MTSSYS user:

   ```
   SQL> ALTER USER MTSSYS IDENTIFIED BY NEW_PASSWORD;
   ```

   This changes your MTSSYS user name password.

Note: You can also have your Oracle Service for MTS user name authenticated by the operating system. In this case, your Windows NT domain user name is used instead of MTSSYS. See "Creating a New Operating System-Authenticated Oracle Service for MTS" on page A-8 for instructions.

Creating a New Oracle Service for MTS

Follow the procedures below to create a new Oracle Service for MTS to which to assign the MTSSYS user name.

To create a new Oracle Service for MTS:

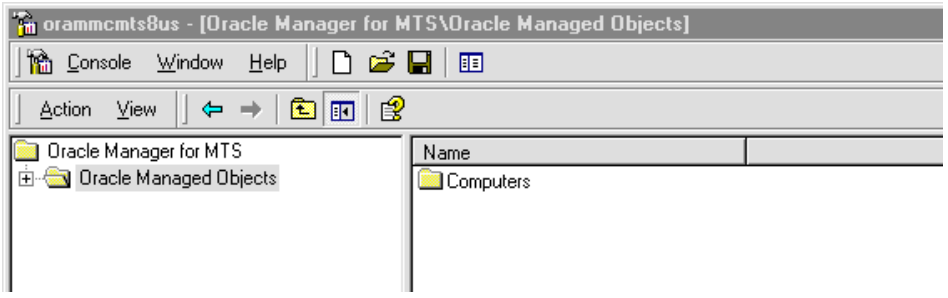
1. Go to the computer where you installed Oracle Services for Microsoft Transaction Server.

   Note: You can configure the Oracle Service for MTS on this computer or use this computer to configure the Oracle Service for MTS on a remote computer.

2. Use Net8 Assistant to create the net service name *MTSDEMO* that connects to your Oracle8 database. See the *Net8 Administrator's Guide* for instructions. The *MTSDEMO* connection enables you to quickly use the Microsoft application demo described in section "Using OCI with the Microsoft Application Demo" on page 4-2.

3. Choose Start > Programs > Oracle - *HOME_NAME* > Application Development > Oracle Manager for Microsoft Transaction Server.

   The Microsoft Management Console Explorer appears:

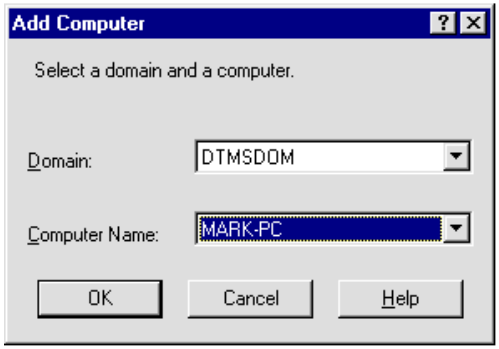


4. Right-click Computers.

A menu appears with several options.

5. Choose New > Computer.

   The *Add Computer* dialog box appears:

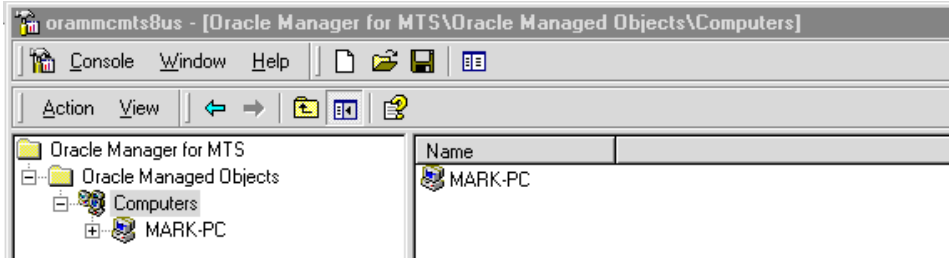


6. Enter the domain and name of the computer on which to configure the Oracle Service for MTS and click OK.

   Note: You must have Administrator privileges on this computer.

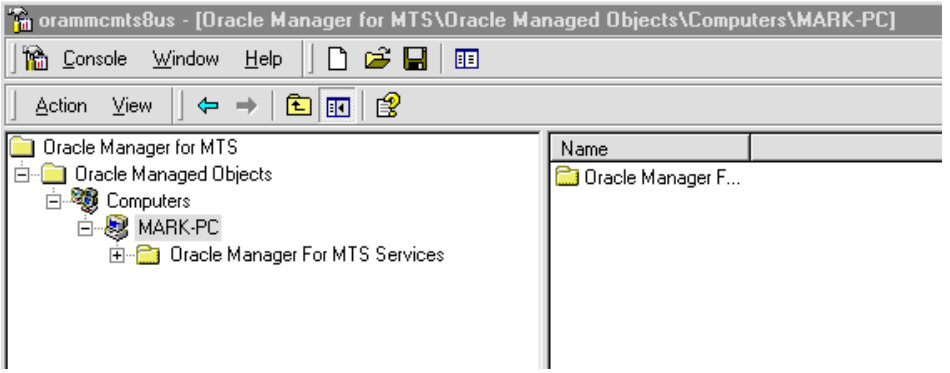
   Note: If creating an Oracle Service for MTS on a remote computer, Microsoft Transaction Server and Oracle Services for Microsoft Transaction Server must already be installed on the remote computer.

   A new icon is created under Computers in the Microsoft Management Console Explorer:



7. Double-click the new icon.

The Oracle Manager for MTS Services snap-in appears:

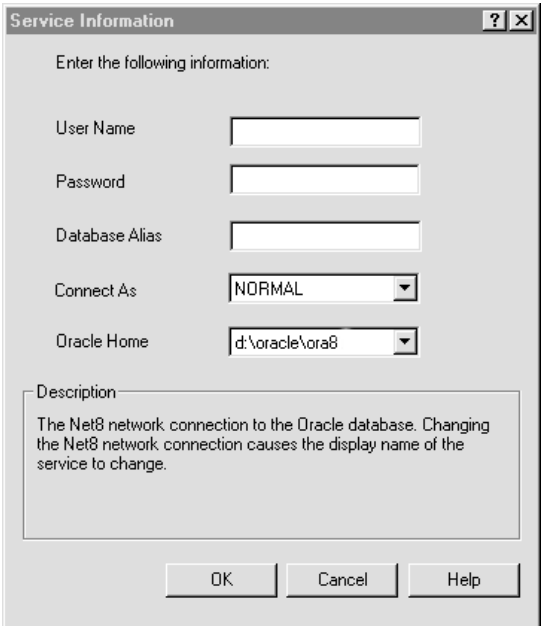


8. Right-click the Oracle Manager for MTS Services snap-in.

   A menu appears with several options.

9. Choose New > Service.

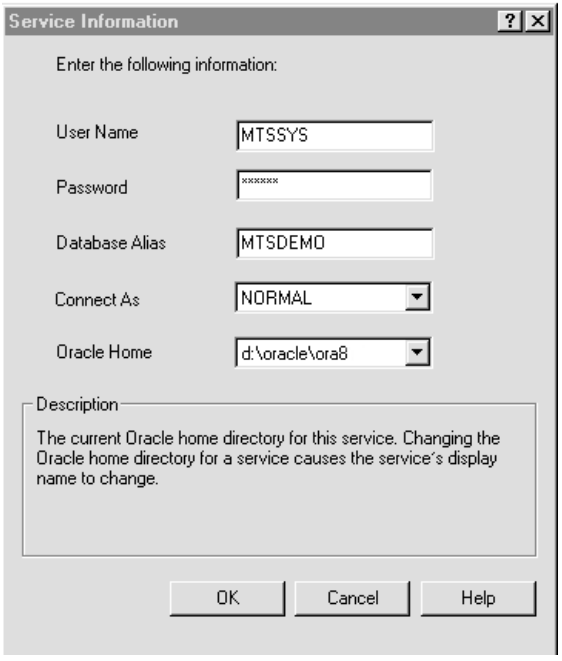
   The *Service Information* dialog box appears.



10. Enter the following information:

Where...	Is...
User Name	MTSSYS
Password	MTSSYS
Database Alias	The net service name that connects to the Oracle8 database created in step 2. For this example, the network connection name *MTSDEMO* is entered.
Connect As	The privileges with which to connect to the Oracle8 database (NORMAL, SYSOPER, or SYSDBA).
Oracle Home	The unique home name for the Oracle8 database installation.

When complete, the *Service Information* dialog box appears as follows:

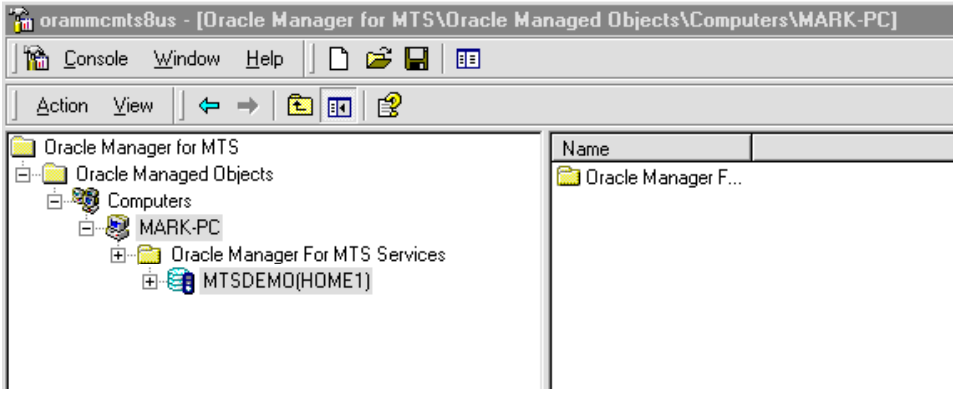


11. Click OK.

The following message indicates that the Oracle Service for MTS was created and started:

12. Click OK.

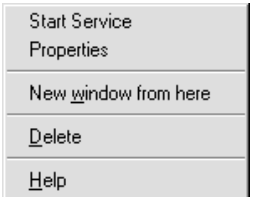
A new Oracle Service for MTS icon named *MTSDEMO* appears beneath the Oracle Manager for MTS Services snap-in. This icon is named after the net service name entered in the Database Alias field of the *Service Information* dialog box in step 10.



13. If the Oracle Service for MTS does not start, follow these instructions:

a. Right-click the new Oracle Service for MTS icon (for example, *MTSDEMO*) in the Microsoft Management Console Explorer.

A menu appears with several options:



b. Choose Start Service.

The Oracle Service for MTS starts.

Note: If the Oracle Service for MTS still does not start, see "Identifying Oracle Service for MTS Startup Problems" on page 7-5 for troubleshooting information.

4

Using the Microsoft Application Demo

This chapter describes how to use the sample Microsoft Component Object Model (COM)-based application demo that is integrated with Microsoft Transaction Server.

Specific topics discussed are:

- Using OCI with the Microsoft Application Demo
- Using Oracle's ODBC Driver with the Microsoft Application Demo

Using OCI with the Microsoft Application Demo

You can use the Oracle Call Interface (OCI) with the sample banking application demo that Microsoft provides with Microsoft Transaction Server. In most cases, OCI is automatically integrated with the Microsoft application demo. Review the table below to determine if OCI and the Microsoft application demo are integrated in your environment, and what you can do if they have not been integrated.

If...	Then...
Microsoft Transaction Server is *already* installed when you install Oracle Services for Microsoft Transaction Server	Oracle Universal Installer automatically backs up and substitutes several Visual C++ files in the banking demo with files that integrate the OCI.DLL and ORAMTS.DLL files. This enables you to use OCI with the banking demo.
Microsoft Transaction Server is *not* installed when you install Oracle Services for Microsoft Transaction Server	Perform the following procedures: **1.** Install Microsoft Transaction Server. **2.** Back up the *ROOTDRIVE*:\PROGRAM FILES\MTS\SAMPLES\PACKAGES\VCACCT.DLL file on your Windows NT computer to a different location. **3.** Copy the VCACCT.DLL file in *ORACLE_BASE**ORACLE_HOME*\ORAMTS\SAMPLES\ACCOUNT.VC\RELEASE to the Windows NT directory listed in step 2.

Note: The directory path examples in this guide follow Optimal Flexible Architecture (OFA) guidelines (for example, *ORACLE_BASE**ORACLE_HOME*\ORAMTS\TRACE). If you specified non-OFA compliant directories during installation or installed a release 8.0.6 database, your directory paths will differ. See the following documentation for additional information:

- For Oracle release 8.1.6, see your *Oracle8i Administrator's Guide for Windows NT*
- For Oracle release 8.0.6, see your *Getting Started* guide

Microsoft Application Demo Overview

The Microsoft application demo is installed under *ORACLE_BASE**ORACLE_HOME*\ORAMTS\ SAMPLES\ACCOUNT.VC and is an OCI implementation of the Visual C++ Sample Bank package that ships with Microsoft Transaction Server. The demo component uses the user account SCOTT and password TIGER to connect to a database whose net service name is MTSDEMO. You can change this information in the ORAMISC.H file. The demo also uses two tables:

- ACCOUNT
- RECEIPT

These tables are part of the schema of user SCOTT in the default Oracle8 database created during installation. See the following section "Ensuring the Oracle8 Database Includes the Proper Tables" to ensure that your Oracle8 database includes this schema.

Ensuring the Oracle8 Database Includes the Proper Tables

If the default Oracle8 database is not being used or your Oracle8 database does not include the user SCOTT, the tables required to run the sample banking Microsoft application demo can be created in the relevant user's schema.

To ensure the Oracle8 database includes the proper tables:

1. Review the following table to determine if your Oracle8 database includes the proper tables:

If You Create Your Oracle8 Database Through These Methods...	Then...
■ Through the Typical or Minimal installation type of the Oracle8*i* Enterprise Edition or Oracle8*i* database type ■ Through the Typical option of the Oracle Database Configuration Assistant where you select to copy existing database files from the CD ■ Manually using the database creation script available on your Oracle 8.1 CD-ROM (On Windows NT, this script is named BUILDALL.SQL and is described in your *Oracle8i Administrator's Guide for Windows NT* guide.) ■ Manually using your own SQL script, and then explicitly run the SCOTT.SQL and ORAMTS.SQL scripts in that order against the database[1]	Your Oracle8 database includes the proper tables. Go to section "Running the Microsoft Application Demo" on page 4-5 to run the sample demo.

If You Create Your Oracle8 Database Through These Methods...	Then...
■ Through the Custom option of the Oracle Database Configuration Assistant ■ Through the Typical option of the Oracle Database Configuration Assistant where you select to create new database files ■ Manually using your own SQL script, and do *not* run ORAMTS.SQL against the database ■ Using the CREATE DATABASE syntax in SQL*Plus line mode ■ In *any* available method on Solaris or any other operating system.	Your Oracle8 database does *not* include the proper tables. Perform steps 1 through 6 in this section before proceeding to section "Running the Microsoft Application Demo" on page 4-5 to run the sample demo.

[1] If you run ORAMTS.SQL before running SCOTT.SQL or do not run SCOTT.SQL at all, you receive numerous error messages when running ORAMTS.SQL. These messages can be ignored. The product functions properly, but the sample application demo described in this chapter will not work.

If your Oracle8 database does not include the proper tables, you must create them manually.

To manually create the proper tables:

1. Start SQL*Plus:

   ```
   C:\> SQLPLUS
   ```

2. Connect with the SYSTEM user name:

   ```
   ENTER USER-NAME: SYSTEM/PASSWORD@NET_SERVICE_NAME
   ```

   where *PASSWORD* is MANAGER after Oracle8 database installation, unless you changed it, and *NET_SERVICE_NAME* is the net service name that connects to the Oracle8 database.

3. Create the user MTSDEMOUSR:

   ```
   SQL> CREATE USER MTSDEMOUSR IDENTIFIED BY MTSDEMOUSR;
   ```

   This creates the user MTSDEMOUSR, which is used to run the Microsoft sample application.

4. Assign the following roles to user MTSDEMOUSR:

   ```
   SQL> GRANT CONNECT, RESOURCE TO MTSDEMOUSR;
   ```

5. Connect with user MTSDEMOUSR:

   ```
   SQL> CONNECT MTSDEMOUSR/MTSDEMOUSR@NET_SERVICE_NAME
   ```

6. Execute the following SQL script:

   ```
   SQL> @\ORACLE_BASE\ORACLE_HOME\ORAMTS\SAMPLES\ACCOUNT.VC\OMTSSAMP.SQL
   ```

   This creates the ACCOUNT and RECEIPT tables in the schema of user MTSDEMOUSR.

Note: The Oracle Service for MTS (with user MTSSYS) and the sample application components (with user MTSDEMOUSR) use separate user accounts to connect to the Oracle8 database.

Running the Microsoft Application Demo

To run the Microsoft application demo:

Note: The sample project was created using Visual C++ 5.0. To build the project using Visual C++ 6.0, the Microsoft .IDL file must be processed with the new version of the MIDL compiler included with Visual C++ 6.0:

1. Go to the directory *ORACLE_BASE**ORACLE_HOME*\ORAMTS\SAMPLES\ACCOUNT.VC.
2. Enter the following at the MS-DOS command prompt:

   ```
   MIDL ACCOUNT.IDL
   ```

When opening a Visual C++ 5.0 project, Visual C++ 6.0 automatically updates it to the new version.

---

1. Ensure that you have installed the Sample Bank Client application shipped with Microsoft Transaction Server. The sample component DLLs are typically installed under *ROOTDRIVE*:\PROGRAM FILES\MTS\SAMPLES\PACKAGES. You can also verify installation through the Control Panel:
   a. Choose Start > Settings > Control Panel.
   b. Double-click Add/Remove Programs.

      The *Add/Remove Programs Properties* dialog box appears.

   c. Select Windows NT 4.0 Option Pack and click Add/Remove.

      The setup takes several seconds to initialize. When complete, the *Microsoft Windows NT 4.0 Option Pack Setup* dialog box appears.

   d. Click Next.

e. Click Add/Remove.

f. Scroll through the list until you find Transaction Server.

g. Double-click Transaction Server.

h. Ensure that the Transaction Server Development subcomponent check box is selected; this indicates that the demo is installed. If this subcomponent is not selected, then select it and follow steps i through m. Otherwise, go to step 2.

i. Click OK.

j. Click Next on the *Microsoft Windows NT 4.0 Option Pack Setup* dialog box.

   Installation takes several seconds.

k. Exit the *Add/Remove Programs Properties* dialog box.

l. Exit the Control Panel.

m. Go to step 2.

2. Open the project ACCOUNT.DSP located under *ORACLE_BASE**ORACLE_HOME*\ORAMTS\SAMPLES\ACCOUNT.VC using Microsoft Developer Studio.
3. Build VCACCT.DLL.
4. Back up the file VCACCT.DLL installed under *ROOTDRIVE*:\PROGRAM FILES\MTS\SAMPLES\PACKAGES.
5. Overwrite VCACCT.DLL with the one you built in step 3.
6. Choose Start > Programs > Windows NT 4.0 Option Pack > Microsoft Transaction Server > Bank Client.

   This starts the Sample Visual Basic Bank application and runs VBBANK.EXE to test the component.
7. Click the Visual C++ radio button of the Language field.
8. Click the Account or the MoveMoney radio boxes under the Component field.
9. Enter the account number(s) and the amount for the operation.
10. Click Submit.
11. Start SQL*Plus

```
C:\> SQLPLUS
```

12. Connect with the user name SCOTT and password TIGER (unless you changed it):

```
ENTER USER-NAME: SCOTT/TIGER
```

13. Verify the success of the operation:

```
SQL> SELECT * FROM ACCOUNT;

SQL> SELECT * FROM RECEIPT;
```

Using Oracle's ODBC Driver with the Microsoft Application Demo

You can use Oracle's ODBC release 8.1 driver with the Microsoft sample application. See section "Using Oracle's ODBC Driver" on page 5-22 for instructions on integrating Oracle's ODBC with the sample application.

5

Programming with Microsoft Transaction Server and an Oracle8 Database

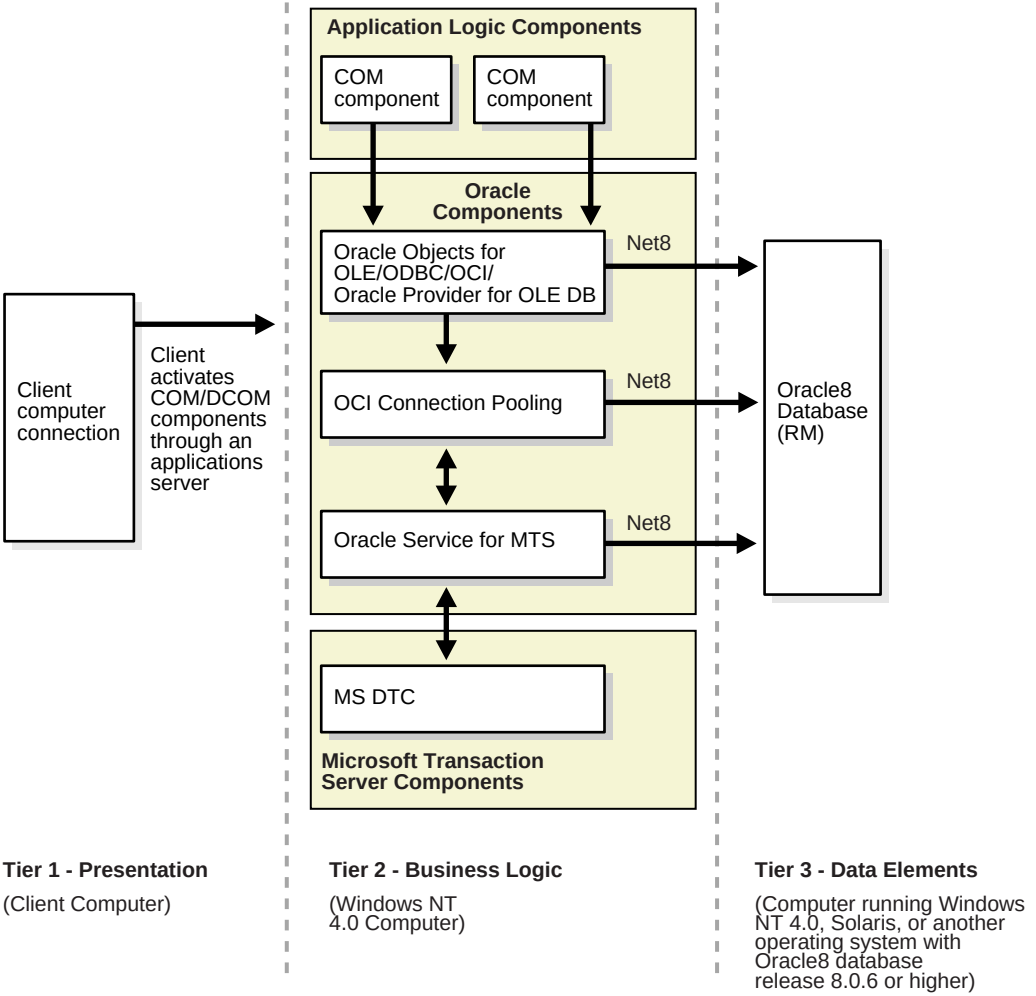
This chapter describes how to program with Microsoft Transaction Server and an Oracle8 database.

Specific topics discussed are:

- Component Integration In a Transaction
- Programming with Microsoft Transaction Server and an Oracle8 Database
- Integrating OCI with Microsoft Transaction Server
- Integrating ODBC with Microsoft Transaction Server
- Integrating Oracle Objects for OLE with Microsoft Transaction Server
- Integrating Oracle Provider for OLE DB with Microsoft Transaction Server
- Integrating Other APIs with Microsoft Transaction Server

Component Integration In a Transaction

When a client computer initiates a transaction request, Microsoft Transaction Server enlists the Oracle8 database to act as a resource manager (RM) in the transaction process. The focal point of the transaction process is a component of Microsoft Transaction Server called Microsoft Distributed Transaction Coordinator (MS DTC). The figure and table below provide an overview of how these and other components perform a transaction:



Component	Major Responsibilities
Client Computer connection	■ Activates the application components using a Web browser or some other remote connection.
Application logic components	■ Embed the business logic. If the component is transactional, Microsoft Transaction Server initiates a transaction. ■ Acquire pooled connections to an Oracle8 database through the Oracle resource dispenser and OCI, Oracle's ODBC driver, Oracle Provider for OLE DB, or Oracle Objects for OLE (OO4O). ■ Decide the outcome of the operation by notifying Microsoft Transaction Server of its decision to commit or abort the changes to all RMs.
OO4O, ODBC, OCI, and Oracle Provider for OLE DB	■ Obtain a service context to the Oracle8 database through the OCI connection pooling component. ■ Provide connection pooling resources, if necessary (ODBC).
OCI Connection pooling	Performs the following for transaction components: ■ Contacts the Oracle8 database for the location of its Oracle Service for MTS. ■ Connects and sends a message to the Oracle Service for MTS requesting to enlist the RM (Oracle8 database) in the transaction. ■ Starts an Oracle global transaction corresponding to the Microsoft Transaction Server transaction of which the component is a part. and also: ■ Acts as a resource dispenser to perform client-side connection pooling.
Oracle Service for MTS	■ Enlists the Oracle8 database in the transaction. ■ Provides the COM communication interface between Microsoft Transaction Server (and its MS DTC component) and the Oracle8 database. MS DTC commits and aborts transactions through the Oracle Service for MTS. ■ Provides transaction support in the MS DTC. ■ Performs recovery of transactions in case of failure (for example, Microsoft Transaction Server goes down, the Oracle8 database goes down, the client application goes down, etc.). ■ Runs on a Windows NT computer, but can communicate through a Net8 network connection with Oracle8 databases running on different operating systems (such as Windows NT and Solaris).
MS DTC (part of Microsoft Transaction Server)	■ Commits and aborts transactions using the two-phase commit protocol. ■ Keeps track of transactions that require recovery. Multiple MS DTCs can be involved in a single transaction. When the RM (Oracle8 database) is enlisted in a transaction, a connection is opened between the client MS DTC and the RM MS DTC. When the client MS DTC commits or aborts a transaction, it sends the request through all involved RM MS DTCs. The transaction request is then passed to the Oracle Service for MTS, which sends it to the Oracle8 database.
Oracle8 database	■ Acts as an RM for Microsoft Transaction Server. This is the database on which the client transaction request is performed.

Programming with Microsoft Transaction Server and an Oracle8 Database

See the *Microsoft Transaction Server Programmer's Guide* in the Microsoft Transaction Server's Help file for an explanation of how to develop application components for Microsoft Transaction Server.

Regardless of the application program interface (API) you use, OCI connection pooling is used in nearly all cases to coordinate a transaction. Review the following sections for information on how a transaction is registered and OCI connection pooling is used to coordinate your transaction:

- Registering Microsoft Transaction Server Components
- Components Running in a Microsoft Transaction Server-Coordinated Transaction
- Components Not Running in a Microsoft Transaction Server-Coordinated Transaction, But Using MS DTC

Registering Microsoft Transaction Server Components

Application components that run in the Microsoft Transaction Server environment are created as dynamic link libraries (DLLs) that are registered with Microsoft Transaction Server using the Microsoft Transaction Server Explorer graphical user interface (GUI) tool. When you register the application component, you mark it as one of the following:

Type	The Component...
Requires a transaction	Must execute in a transaction. If the transaction does not currently exist, Microsoft Transaction Server automatically creates a new transaction for the component.
Requires a new transaction	Must execute within their own transaction. Microsoft Transaction Server automatically creates a new transaction for the component.
Supports transactions	Can execute within the client's transaction. When a new component is created, its context inherits the transaction from the context of the client. If the client does *not* have a transaction, the new context is also created without one.
Does not support transactions	Does not run within a transaction. The new component is created without a transaction, regardless of whether the client has a transaction.

How you register an application component determines if it runs in a Microsoft Transaction Server-coordinated transaction.

If Your Application Component...	Then...
Runs in a Microsoft Transaction Server-coordinated transaction	OCI connection pooling is *always* used and Microsoft Transaction Server and its MS DTC component coordinate the creation, startup, management, and commitment phases of the transaction. Microsoft Transaction Server ensures that all changes made by the component are committed if the transaction succeeds, or are aborted if the transaction fails. See section "Components Running in a Microsoft Transaction Server-Coordinated Transaction" on page 5-6.
Does *not* run in a Microsoft Transaction Server-coordinated transaction	The component runs in a Microsoft Transaction Server environment, but its MS DTC component may or may not coordinate communication between the Oracle Service for MTS and the Oracle8 database. If your transaction is not MS DTC-coordinated, your client application must create, start, manage, and commit the transaction. See section "Components Not Running in a Microsoft Transaction Server-Coordinated Transaction, But Using MS DTC" on page 5-7. OCI connection pooling may be used, depending upon how you program your client application and the OCI API.

Components Running in a Microsoft Transaction Server-Coordinated Transaction

This section describes how OCI connection pooling, Microsoft Transaction Server, and MS DTC operate with application components in a Microsoft Transaction Server-coordinated transaction environment.

1. The client API being used (ODBC, OCI, OO4O, or Oracle Provider for OLE DB) calls OCI function OraMTSSvcGet() to obtain a service context from the OCI connection pooling component.
2. The OCI connection pooling component:
   a. Contacts the Oracle8 database for the hostname and pipe name of its Oracle Service for MTS.
   b. Opens a pipe to the Oracle Service for MTS.
   c. Requests to enlist in the transaction, which is to be coordinated by the MS DTC component of Microsoft Transaction Server.

   These actions return OCI service and environment handles to client applications.
3. The Oracle Service for MTS enlists the Oracle8 database in the MS DTC-coordinated transaction.
4. The client application:
   a. Performs the database operations.
   b. Calls OCI function OraMTSSvcRel() to release the OCI pooling connection obtained at the beginning of the transaction.
   c. Calls SetComplete (to commit database operations) or SetAbort (to abort database operations) on the Microsoft Transaction Server context object associated with the component.
5. MS DTC performs the two-phase commit protocol to prepare and commit (or abort) the transaction, which notifies the OCI connection pooling component and ends the transaction.
6. The Oracle Service for MTS is notified and performs the necessary steps to complete phase one (the prepare phase).
7. The Oracle Service for MTS is notified and performs the necessary steps to complete the commit (or abort) phase.

Components Not Running in a Microsoft Transaction Server-Coordinated Transaction, But Using MS DTC

This section describes how OCI connection pooling, Microsoft Transaction Server, and MS DTC operate with application components in this type of environment.

1. The client application starts an MS DTC-coordinated transaction and connects to the Oracle8 database. OCI function OraMTSSvcEnlist() enlists either a:
   - pooled OCI connection obtained previously through OraMTSSvcGet(...,…,ORAMTS_CFLG_NOIMPLICIT) and not yet released with OraMTSSvcRel()
   - nonpooled OCI connection obtained through some other method
2. The client application then calls OraMTSSvcEnlist() or OraMTSSvcEnlistEx(), passing in the MS DTC-coordinated transaction.
3. The OCI connection pooling component:
   a. Contacts the Oracle8 database for the hostname and pipe name of its Oracle Service for MTS.
   b. Opens a pipe to the Oracle Service for MTS.
   c. Requests to enlist in the transaction, which is to be coordinated by the MS DTC component of Microsoft Transaction Server.
4. The Oracle Service for MTS enlists the Oracle8 database in the MS DTC-coordinated transaction.
5. The client application:
   a. Performs database operations.
   b. Calls OCI function OraMTSSvcEnlist(NULL) regardless of whether pooled or nonpooled OCI connections were used.
   c. Calls OCI function OraMTSSvcRel() to release them, *if* pooled OCI connections originally obtained from OraMTSSvcGet(...,…,ORAMTS_CFLG_NOIMPLICIT) were used.
   d. Calls the commit or Abort method on the transaction object returned by MS DTC (for example, pTransaction->Commit() or pTransaction->Abort()).
6. MS DTC performs the two-phase commit protocol to commit the transaction.
7. The Oracle Service for MTS is notified and performs the necessary steps to complete phase one (the prepare phase).
8. The Oracle Service for MTS is notified and performs the necessary steps to complete the commit (or abort) phase.

Integrating OCI with Microsoft Transaction Server

The following OCI functions enable you to integrate your OCI client application with Microsoft Transaction Server and an Oracle8 database. Review the following sections for information on this integration:

- OCI Function Overview
- OraMTSSvcGet()
- OraMTSSvcRel()
- OraMTSSvcEnlist()
- OraMTSSvcEnlistEx()
- OraMTSTransTest()
- OraMTSOCIErrGet()

OCI Function Overview

You must use OCI version 8.1. OCI releases earlier than 8.1 are not supported.

WARNING: As with any C++ Microsoft Transaction Server component, obtain the object context and call SetAbort(), SetComplete(), EnableCommit(), or DisableCommit(), depending on the state of your component's work. Do *not* make any OCI transaction calls such as OCITransCommit() or OCITransAbort(); this corrupts your data!

The only change to make in your code is in obtaining and releasing your OCI service context handle. An OCI service context handle and environment handle are acquired when you obtain a pooled OCI connection to the database with the OCI function OraMTSSvcGet(). Include the ORAMTS.H header and link with the ORAMTS.LIB library. When you are finished, call OCI function OraMTSSvcRel() to release your service context handle and environment handle. Using OraMTSSvcGet() enables you to receive connection pooling and implicit transaction support (if you registered your application component to run in a Microsoft Transaction Server transaction). See sections "OraMTSSvcGet()" on page 5-12 and "OraMTSSvcRel()" on page 5-15 for more information.

If you use OraMTSSvcGet() and OraMTSSvcRel(), and you did *not* register your component to run in a Microsoft Transaction Server transaction, you can still enlist your database in a Microsoft Transaction Server-coordinated transaction. Use OCI function OraMTSSvcEnlist() (for pooled or nonpooled connections) or, for better

performance, OraMTSSvcEnlistEx() (for nonpooled connections obtained through standard OCI methods). In these cases, the MS DTC component of Microsoft Transaction Server coordinates the transaction. See sections "OraMTSSvcEnlist()" on page 5-16 and "OraMTSSvcEnlistEx()" on page 5-19 for more information.

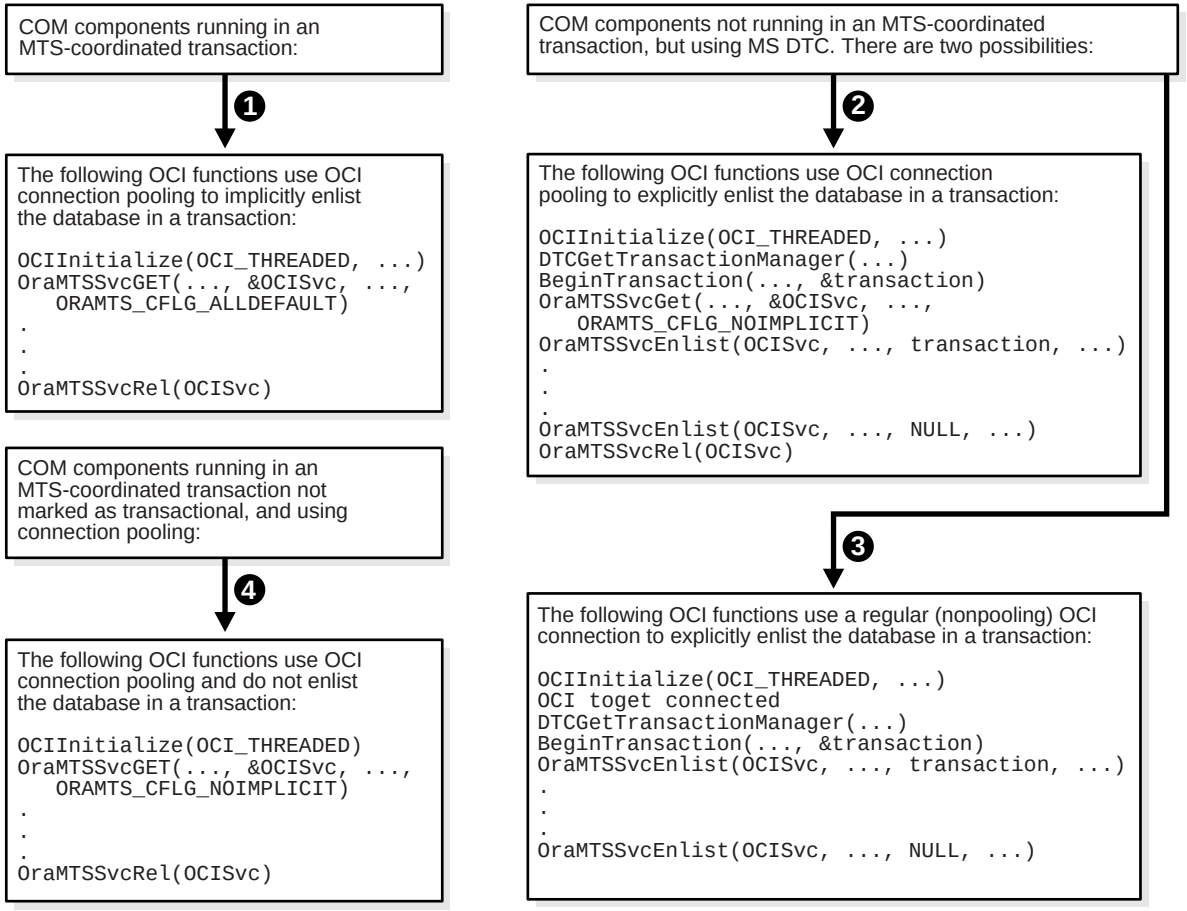
Ensure that on a per process basis, you call OCIInitialize at least once before executing any other OCI calls. This initializes the OCI process environment. In addition, you must pass it the OCI_THREADED flag. If you are using Microsoft's Internet Information Server (IIS) and your components are being called as in-process libraries, then OCIInitialize is already called for you.

```
#include <oci.h>
#include  <oramts.h>
#include  <xolehlp.h>
// other MTS relevent includes ...

// prototype for the error handler.
BOOL Chekerr(sword swOCIStat, OCIError *OCIErrh);

// MTS component method
HRESULT OCITestMethod()
{
 IObjectContext *pObjectContext = NULL;
 OCIEnv    *myenvh = NULL;
 OCISvcCtx *mysvch = NULL;
 OCIError  *myerrh = NULL;
 OCIStnt   *mystmh = NULL;
 DWORD      dwStat;
 HRESULT    hRes = S_OK;
 sword      swOCIStat;
 BOOL       bCommit = FALSE;
 char      *lpzStmt = "UPDATE EMP SET SAL = SAL + 1000";

 // Initialize the OCI environment first -- request OCI_THREADED
 OCIInitialize(OCI_THREADED, (dvoid*)NULL,NULL,NULL,NULL);
 // attempt to get a connection to the database via the resource dispenser
 OraMTSSvcGet(
"scott","tiger","finprod_db",&mysvch, &myenvh, ORAMTS_CFLG_ALLDEFAULT);
 // validate return status
```

```
 if(dwStat != ORAMTS_ERR_NOERROR)
 {
   printf("error: failed to obtain a connection to the database - %ld",
dwStat);
   goto cleanup;
 }
 // successful logon and enlistment in the MTS transaction. allocate statement
 // handles and other handles using the OCI environment handle myenvh ....
 swOCIStat = OCIHandleAlloc(myenvh, (void *)&myerrh,OCI_HTYPE_ERROR, 0 , NULL);
 if (Checkerr(swOCIStat, myerrh)) goto cleanup;
 swOCIStat = OCIHandleAlloc(myenvh, (dvoid *)&mystmh,OCI_HTYPE_STMT, 0,NULL);
 if (Checkerr(swOCIStat, myerrh)) goto cleanup;
 // prepare a DML statement
 OCIStmtPrepare(mystmh, myerrh, lpzStmt, lstrlen(lpzStmt), OCI_NTV_SYNTAX,
OCI_DEFAULT)
 Checkerr(swOCIStat, myerrh);
 // execute the statement -- ensure that AUTOCOMMIT is not requested.
 OCIStmtExecute(mysvch, mystmh, myerrh, 1, 0, NULL, NULL, OCI_DEFAULT);
 if (Checkerr(swOCIStat, myerrh)) goto cleanup;
 // all's well so far choose to go for a commit
 bCommit = TRUE;
cleanup:
 if (mystmh) OCIHandleFree((void*)mystmh, OCI_HTYPE_STMT);
 if (myerrh  OCIHandleFree((void*)myerrh, OCI_HTYPE_ERROR);
 if (mysvch) OraMTSSvcRel(mysvch);
 if (bCommit)
     pObjectContext->SetComplete();
 else
     pObjectContext->Abort();
 return(bCommit ? S_OK : E_FAIL);
}
```

See the ACCOUNT.VC files in the *ORACLE_BASE**ORACLE_HOME*\ORAMTS\SAMPLES\ACCOUNT.VC\RELEASE directory for code samples.

This illustration provides a high-level overview of how to use the OCI functions OraMTSSvcGet(), OraMTSSvcRel(), and OraMTSSvcEnlist().



Non-COM applications (also known as standalone applications) can also use methods 2, 3, and 4 above. However, non-COM applications cannot use the Microsoft Transaction Server Explorer GUI. Method 3 can also use OraMTSSvcEnlistEx() instead of OraMTSSvcEnlist() for better performance.

OraMTSSvcGet()

Purpose

OraMTSSvcGet() obtains a pooled connection (also known as an OCI service context) from the OCI connection pool. The pooled connection includes an OCI service context handle and OCI environment handle.

Syntax

```
DWORD  OraMTSSvcGet(
                     text       *lpUname,
                     text       *lpPsswd,
                     text       *lpDbnam,
                     OCISvcCtx **pOCISvc,
                     OCIEnv    **pOCIEnv,
                     ub4         dwConFlgs
                   );
```

Parameters

Data Type	Parameter	Description
text	*lpUname(IN)	User name for connecting to the Oracle8 database
text	*lpPsswd(IN)	Password for the above user name
text	*lpDbnam(IN)	Net service name for connecting to the database (created with Net8 Assistant)
OCISvcCtx	**pOCISvc(OUT)	Pointer to the OCI service context handle
OCIEnv	**pOCIEnv(OUT)	Pointer to the OCI environment handle
ub4	dwConFlgs(IN)	Connection flags, for which there are the following possible values: ■ ORAMTS_CFLG_ALLDEFAULT Obtains a pooled connection and enlists the connection in any MTS transaction, if one exists. If the component is nontransactional, no enlistment request is issued.

Data Type	Parameter	Description
		■ ORAMTS_CFLG_NOIMPLICIT Obtains a pooled connection, but does not enlist the resource in any Microsoft Transaction Server transaction even if the component is transactional. This flag must be used if the component wants to manually enlist the connection resource later using OraMTSSvcEnlist(). Prior to releasing a connection obtained in this fashion, the client must de-enlist the resource if enlisted. ■ ORAMTS_CFLG_UNIQUESRVR Requests a single OCI session per OCI Server. In this release, multiplexing is not supported. Therefore, this option is always used. ■ ORAMTS_CFLG_SYSDBALOGN Use this flag if connecting as SYSDBA. ■ ORAMTS_CFLG_SYSOPRLOGN Use this flag if connecting as SYSOPER. ■ ORAMTS_CFLG_PRELIMAUTH Use this flag if connecting as INTERNAL.

Returns

Returns ORAMTSERR_NOERROR upon successful acquisition of an OCI pooling connection (OCI service context).

Comments

OraMTSSvcGet() returns a pooled OCI connection to the caller, enabling a database transaction using OCI to begin. Use OraMTSSvcGet if you want to implicitly enlist the OCI connection in a transaction coordinated by Microsoft Transaction Server. In this type of transaction, Microsoft Transaction Server controls the creation, startup, management, and commitment phases of the transaction through its MS DTC component.

OraMTSSvcGet() can also be used to simply provide connection pooling without enlisting the Oracle8 database in an MTS transaction. This is done by setting OraMTSSvcGet() as follows:

```
OraMTSSvcGet(...,ORAMTS_CFLG_NOIMPLICIT)
```

In *all* cases where OraMTSSvcGet() is used, you must always use OraMTSSvcRel() to release the connection when you are done.

> **Note:** Connection pooling is used regardless of whether you enlist or do not enlist your COM component in a transaction.

Use the flags ORAMTS_CFLG_SYSDBALOGN, ORAMTS_CFLG_SYSOPRLOGN, and ORAMTS_CFLG_PRELIMAUTH when connecting as SYSDBA, SYSOPER, and INTERNAL, respectively. For instance, to obtain an enlisted connection using the INTERNAL account, call OraMTSSvcGet() as follows:

```
OraMTSSvcGet("INTERNAL", "oracle", "oracle", &OCISvc, &OCIEnv, ORAMTS_CFLG_
ALLDEFAULT | ORAMTS_CFLG_PRELIMAUTH);
```

To obtain a nonenlisted connection using the SCOTT/TIGER account, call OraMTSSvcGet() as follows:

```
OraMTSSvcGet("scott", "tiger", "oracle", &OCISvc, &OCIEnv, ORAMTS_CFLG_
ALLDEFAULT | ORAMTS_CFLG_NOIMPLICIT);
```

OraMTSSvcGet does not support placing the user name (lpUname), password (lpPsswd), and net service name syntax (lpDbname) together in the user name argument (for example, "SCOTT/TIGER@PROD_FIN"). Instead, the caller must fill in lpUname, lpPsswd, and lpDbname separately (as shown in the two examples above). Calling OraMTSSvcGet() with the user name and password as NULL strings causes external authentication (operating system authentication) to be used for the connection.

OraMTSSvcRel()

Purpose

OraMTSSvcRel() releases a pooled OCI connection (OCI service context) back to the connection pool. OraMTSSvcRel() must be used to release connections that were acquired with OraMTSSvcGet().

Syntax

```
DWORD OraMTSSvcRel(OCISvcCtx *OCISvc);
```

Parameters

Data Type	Parameter	Description
OCISvcCtx	*OCISvc(IN)	OCI service context for pooled connection

Returns

Returns ORAMTSERR_NOERROR upon successful release of a pooled OCI connection.

Comments

An OCI pooled connection obtained through a previous call to OraMTSSvcGet() is released back to the connection pool. Once released back to the connection pool, the OCI service context, its environment handle, and all child handles are invalid.

A nontransactional client component must explicitly issue OCITransCommit() or OCITransAbort() prior to releasing a connection obtained through OraMTSSvcGet(…,…,ORAMTS_CFLG_ALLDEFAULT) back to the pool. Otherwise, all changes made in that session are rolled back. A transaction component uses the SetComplete or SetAbort methods on its Microsoft Transaction Server object context.

Components that have called OraMTSSvcGet(…,…,ORAMTS_CFLG_ NOIMPLICIT) to obtain a connection resource must first de-enlist the resource if enlisted. If the connection was enlisted explicitly, pTransaction->Commit() or pTransaction->Abort() must be called. Otherwise, OCITransCommit() or OCITransAbort() must be called before releasing the connection back to the pool.

OraMTSSvcEnlist()

Purpose

OraMTSSvcEnlist() enlists or de-enlists an OCI connection in a transaction coordinated by MS DTC.

Syntax

```
DWORD OraMTSSvcEnlist(
                      OCISvcCtx  *OCISvc,
                      OCIError   *OCIErr,
                      void       *lpTrans,
                      unsigned    dwFlags
                     );
```

Parameters

Data Type	Parameter	Description
OCISvcCtx	*OCISvc(IN)	OCI service context for pooled connections obtained by calling OraMTSSvcGet() or a service context obtained through calls to OCI 8.1. The latter have an internal context object created by the resource dispenser when they enlist through OraMTSSvcEnlist().
OCIError	*OCIErr(IN / OUT)	OCI environment handle for errors (ignored if the OCI service context represents a pooled connection that was obtained using OraMTSSvcGet()).
void	*lpTrans(IN)	Pointer to the MS DTC-controlled transaction in which to enlist. If NULL, the OCI connection is de-enlisted from the MS DTC-controlled transaction.
unsigned	dwFlags(IN)	Enlistment flags, for which there are the following possible values:

Data Type	Parameter	Description
		■ ORAMTS_ENFLG_DEFAULT If enlisting, then start a new Oracle global transaction. If de-enlisting, then detach from any global Oracle transaction and delete the context object if the OCI service context represents a nonpooled connection. ■ ORAMTS_ENFLG_RESUMTX Used to re-enlist a temporarily de-enlisted service context by resuming the global Oracle transaction. ■ ORAMTS_ENFLG_DETCHTX Used to temporarily de-enlist from a transaction by just detaching from the global Oracle transaction. Wrapping context information for nonpooled OCI service contexts is retained by the resource dispenser.

Returns

Returns ORAMTSERR_NOERROR upon successful acquisition of an OCI connection.

Comments

OraMTSSvcEnlist() enlists two types of OCI connections:

- pooled OCI connections obtained previously through OraMTSSvcGet() with the ORAMTS_CFLG_NOIMPLICIT flag and not yet released with OraMTSSvcRel()
- OCI connections obtained through some other method (standard OCI programming)

With both connection types, your application must manually begin the MS-DTC-coordinated transaction.

For pooled OCI connections, the underlying object must be explicitly enlistable. When the transaction is complete, you must de-enlist OraMTSSvcEnlist(), passing NULL as the transaction pointer as follows:

```
OraMTSSvcEnlist(OCISvc, OCIenv, NULL, ORAMTS_ENFLG_DEFAULT)
```

If OraMTSSvcGet() is also involved in obtaining the connection, you must use OraMTSSvcRel() to release the connection when done.

Callers must:

1. Allocate a connection.
2. Enlist the connection.
3. Perform work.
4. De-enlist the connection.
5. Release the connection.
6. Attempt to commit or abort.

For nonpooled OCI connections, the enlistment creates a context wrapper object within the resource dispenser. This has a transaction handle, error handle, and other information pertaining to the enlistment. The transaction handle must be undisturbed until the service context is finally disposed. Once a nonpooled OCI connection has been enlisted, it can be detached and attached to the underlying Oracle transaction through the same call using the dwFlags parameter. To detach from the Oracle transaction, set lpTrans to NULL and dwFlags to ORAMTS_ENFLG_DETCHTX. To resume the current transaction, lpTrans is not set to NULL and dwFlags is set to ORAMTS_ENFLG_RESUMTX.

OraMTSSvcEnlistEx()

Purpose

OraMTSSvcEnlistEx() enlists an OCI connection or service context in an MS DTC transaction.

Syntax

```
DWORD OraMTSSvcEnlistEx(
                         OCISvcCtx  *OCISvc,
                         OCIError   *OCIErr,
                         void       *lpTrans,
                         unsigned   dwFlags,
                         char       *lpDBName
                                            );
```

Parameters

Data Type	Parameter	Description
OCISvcCtx	*OCISvc	OCI service context for a pooled connection
OCIError	*OCIErr	OCI error handle for errors. This is ignored if the service context represents a pooled connection
void	*lpTrans	Pointer to an MS DTC-controlled transaction
unsigned	dwFlags	Enlistment flags
char	*lpDBName	Net service name for connecting to the database (created with Net8 Assistant)

Returns

Returns ORAMTSERR_NOERROR on success.

Comments

This call is identical to OraMTSSvcEnlist() except for the addition of an lpDBName parameter. The lpDBName parameter is only used when enlisting nonpooled connections. The parameter is used to cache information to improve enlistment performance (regarding the Oracle Service for MTS for the Oracle8 database). This parameter is ignored for pooled connections and also for de-enlistment requests.

OraMTSTransTest()

Purpose

OraMTSTransTest() tests if you are running inside a Microsoft Transaction Server-initiated transaction.

Syntax

```
BOOL OraMTSTransTest();
```

Parameters

None.

Returns

Returns TRUE if running inside a Microsoft Transaction Server transaction. Otherwise, FALSE is returned.

Comments

This can be used by Microsoft Transaction Server transactional components to check if a component is executing within the context of a Microsoft Transaction Server transaction. Note that this call can *only* test Microsoft Transaction Server-initiated transactions. Transactions started by directly calling the MS DTC are not detected.

OraMTSOCIErrGet()

Purpose

OraMTSOCIErrGet() retrieves the OCI error code and message text (if any) from the last OraMTSSvc operation (typically OraMTSSvcGet(), OraMTSSvcEnlist(), or OraMTSSvcEnlistEx()).

Syntax

```
BOOL OraMTSOCIErrGet(DWORD *dwErr, LPTSTR lpcEMsg, DWORD *lpdLen);
```

Parameters

Data Type	Parameter	Description
DWORD	*dwErr	(OUT) error code
LPCHAR	lpcEMsg	(OUT) buffer for the error message if any
DWORD	*lpdLen	(IN / OUT) size of lpcEmsg in; msg bytes out

Returns

Returns TRUE if an OCI error was encountered. Otherwise, returns FALSE. If TRUE and lpcEMsg and lpdLen are valid, and there is a stashed error message, up to lpdLen bytes are copied into lpcEMsg. lpdLen is set to the actual number of message bytes.

Comments

This function retrieves the OCI error code and OCI error message text, if any, from the last OraMTSSvc operation on this thread. For example:

```
DWORD dwStat = OraMTSSvcGet("scott", "invalid_password","fin_prod",
db",&mysvch, &myenvh, ORAMTS_CFLG_ALLDEFAULT);
        if (dwStat != ORAMTS_ERR_NOERROR)
        {
                DWORD   dwOCIErr;
                char    errBuf[MAX_PATH];
                DWORD   errBufLen = sizeof(effBuf);

                if (OraMTSOCIErrGet(&dwOCIErr, &errBuf, &errBufLen))
                        printf("OCIError %d: %s"\n);
        }
```

Integrating ODBC with Microsoft Transaction Server

This section describes how to use ODBC with Microsoft Transaction Server and an Oracle8 database. Specific topics discussed are:

- Setting the Connection Attribute
- Using Oracle's ODBC Driver (recommended)
- Using Microsoft's Oracle ODBC Driver

OCI connection pooling operates as described in section "Programming with Microsoft Transaction Server and an Oracle8 Database" on page 5-4, with no changes to OCI code required for ODBC to operate.

Setting the Connection Attribute

To use Microsoft Transaction Server with either Oracle's ODBC Driver 8.1 or Microsoft's Oracle ODBC driver, you must set the connection attribute. Use the function SQLSetConnectAttr to call the parameter SQL_ATTR_ENLIST_IN_DTC in your ODBC code. This enables you to receive connection pooling and implicit transaction support. See "Setting Up MTS to Access Oracle" in the Microsoft Transaction Server online Help for instructions.

Using Oracle's ODBC Driver

The ODBC Driver Manager distributed with ODBC 3.0 is a Resource Dispenser that supports connection pooling (see the Microsoft Transaction Server SDK for information). Oracle's ODBC driver release 8.1 integrates with the ODBC 3.0 Driver Manager by supporting the SQLSetConnectAttr(...,..., SQL_ATTR_ENLIST_IN_DTC) call to enlist / de-enlist the ODBC connection in / from MS DTC-coordinated transactions.

You must use release 8.1 of Oracle's ODBC driver. Previous versions do *not* work with Microsoft Transaction Server. Use Oracle's ODBC Driver 8.1 with:

- Your own applications
- The sample banking application that Microsoft provides with Microsoft Transaction Server (see Chapter 4, "Using the Microsoft Application Demo")

To configure Oracle's ODBC Driver 8.1:

1. Choose Start > Settings > Control Panel.

   The *Control Panel* window appears.

2. Double-click ODBC.

   The *ODBC Data Source Administrator* dialog box appears.

3. Choose the File DSN tab.

4. If you want Oracle's ODBC Driver to work with Microsoft's sample banking application demo, follow substeps a through d. Otherwise, go to step 5.

   a. Back up Microsoft's MTSSAMPLES.DSN file. This file is located in *ROOTDRIVE*:\PROGRAM FILES\COMMON FILES\ODBC\DATA SOURCES.

   b. Select MTSSAMPLES.DSN and click Remove.

   c. Click Yes when prompted.

      This deletes the configuration file that enables the Microsoft Transaction Server sample application demo to use Microsoft's ODBC driver.

   d. Go to step 5.

5. Click Add to create a new File data source name (DSN).

   The *Create New Data Source* wizard appears.

6. Select Oracle ODBC Driver 8.1.

7. Click Advanced.

8. Add the following information in the keywords and values field:

   ```
   SERVER=DATABASE_ALIAS
   USERNAME=SCOTT
   PASSWORD=TIGER
   ```

Where...	Is...
SERVER	the database alias used by the Oracle Service for MTS to access the Oracle8 database
USERNAME	SCOTT (Oracle8 database user name for this application)
PASSWORD	TIGER (Oracle8 database password for user name SCOTT)

9. Click OK.
10. Click Next to continue with the *Create New Data Source* wizard.
11. Enter the name of the file DSN to which you want to save this connection information:

If Using Oracle's ODBC For...	Then Enter...
Microsoft's sample application	MTSSAMPLES.DSN (Microsoft's ODBC name). This name must exactly match the name you removed in substep 4b.
Your *own* applications	Any appropriate name.

12. Complete the remaining *Create New Data Source* wizard pages.
13. Click OK to exit the *ODBC Data Source Administrator* dialog box.
14. Exit the *Control Panel* window.

Using Microsoft's Oracle ODBC Driver

If the Oracle8 database version is 8.0.5 or earlier, you cannot use the integration described in this guide. However, there is a solution if you use Microsoft's Oracle ODBC driver. No other APIs are supported.

You can use Microsoft's Oracle ODBC Driver included in Windows NT Option Pack 4 to enable applications to interact with Microsoft Transaction Server and an Oracle8 database. If you use this driver, the rest of the information in this guide does not apply and you do not receive the performance benefits, other API support of Oracle integration, or Oracle 8.1 client support. See "Setting Up MTS to Access Oracle" in the Microsoft Transaction Server online Help for instructions on enabling Microsoft's Oracle ODBC Driver. After following those instructions, perform these additional steps:

To configure Microsoft's Oracle ODBC Driver:

1. Install Oracle Required Support Files (RSF) release 7.3.4 and SQL*Net 2.3 on the computer where Microsoft's Oracle ODBC Driver is operating.
2. Run the *ORACLE_BASE**ORACLE_HOME*\ORAMTS\SAMPLES\ACCOUNT.VC\OMTSSAMP.SQL script.
3. Use SQL*Net Easy Config to set up a database alias connection. This is the alias that the MTSSAMPLES.DSN file uses.

4. If you installed the release 7.3.4 RSFs in a home that has Net8 installed, be sure to set the following registry parameter at HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE:

   ```
   ORAOCI = ORA73.DLL
   ```

Integrating Oracle Objects for OLE with Microsoft Transaction Server

There are no special requirements for using Oracle Objects for OLE. You must use version 8.1 of Oracle Objects for OLE. See the Oracle Objects for OLE online Help file for additional information on using Oracle Objects for OLE with Microsoft Transaction Server.

Connection pooling operates as described in section "Programming with Microsoft Transaction Server and an Oracle8 Database" on page 5-4, with no changes required to the Oracle Objects for OLE code.

Integrating Oracle Provider for OLE DB with Microsoft Transaction Server

See the *Oracle Provider for OLE DB User's Guide* for information on using Oracle Provider for OLE DB with Microsoft Transaction Server.

Integrating Other APIs with Microsoft Transaction Server

Currently, other APIs are not supported, unless they use Oracle's ODBC Driver 8.1, such as ADO.

6

Tuning Oracle Service for MTS Performance

This chapter provides Oracle Service for MTS performance tuning information:

Specific topics discussed are:

- Automatically Restarting Oracle Service for MTS
- Improving Performance
- Managing Connections
- Increasing the Transaction Timeout Parameter
- Changing Initialization Parameter File Settings

Automatically Restarting Oracle Service for MTS

With release 8.1.5, the Oracle Service for MTS automatically shut down in the following situations:

- the Oracle database was down
- the Oracle Service for MTS was unable to connect to the Oracle database for whatever reason

With release 8.1.5, you needed to manually restart the Oracle Service for MTS. For these situations in releases 8.0.6 and 8.1.6, the Oracle Service for MTS automatically restarts and attempts to re-establish a connection to the Oracle database. Upon re-establishing the connection, all existing, active, enlisted transactions are aborted. Transactions prepared before the Oracle Service for MTS shut down are either committed or aborted, depending upon the outcome of the transaction (determined by the Microsoft Distributed Transaction Coordinator (MS DTC).

For example, if component A was enlisted in a transaction when the Oracle Service for MTS shut down, then:

- the Oracle Service for MTS is restarted
- component A's transaction is aborted and subsequent database operations by component A result in errors

However, if component A called the SetComplete() method on the Microsoft Transaction Server transaction context object, and the MS DTC successfully prepared the transaction before the Oracle Service for MTS shut down, the service on restart:

- queries the MS DTC for the final outcome of component A's transaction as determined by the MS DTC
- commits or aborts the Oracle transaction accordingly

Users can check an entry in the Event Viewer to determine if the Oracle Service for MTS has been restarted. See "Using the Event Viewer" on page 7-7 for instructions on using the Event User.

Attention: If Oracle Service for MTS does not restart, transactions remain "in-doubt" and must be manually committed or aborted. See *Oracle8i Distributed Database Systems* for information on manually committing or aborting these transactions.

Improving Performance

You can improve performance when you:

- Optimize your hardware configuration

  Optimize your network interconnects between the computer running the Component Object Model (COM) components and the computer running the Oracle Service for MTS (if they are different) and the interconnects between these computers and the computer running the Oracle8 database.

- Optimize your programming methods

  Placing all your code for a given transaction into one COM component means you do not need to mark that component as transactional. This eliminates the overhead of going through Microsoft Transaction Server and the Oracle Service for MTS. You can then use the Oracle commit or rollback functions to control that transaction in your component. If you are using OCI, you can still use ORAMTSSvcGet(), but you can also use the ORAMTS_CFLG_NOIMPLICIT flag. If you are updating across two or more Oracle8 databases, use database links and connect to one database from your COM component.

  See "Integrating OCI with Microsoft Transaction Server" on page 5-8 for more information on using ORAMTSSvcGet().

Managing Connections

When a COM component ends a session with the Oracle8 database, the connection by default does not immediately terminate. Instead, the connection remains idle in a connection pool, where it is available for reuse by another COM component attempting a new connection to the Oracle8 database.

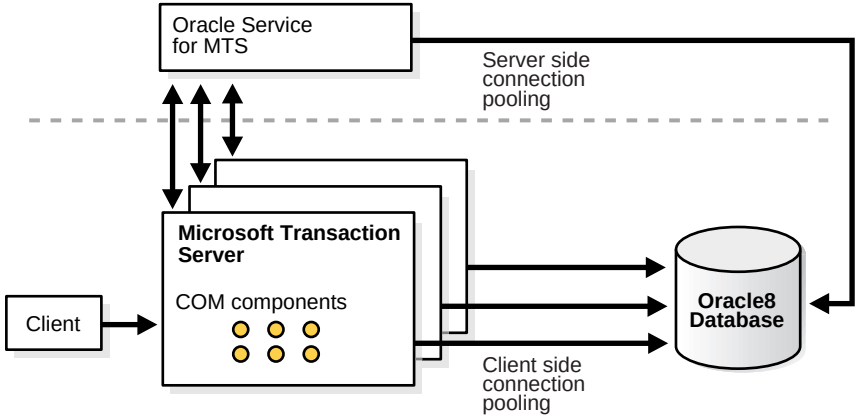
The idle period during which a connection can be reused reduces the resource costs associated with opening a new connection. The amount of time that the connection remains idle and available in the connection pool is determined by several registry parameter settings that you can modify on the computers on which the client Microsoft Transaction Server components and the Oracle Service for MTS are installed. The Oracle Service for MTS uses a similar connection pool. The connection pool is located in two places:

Connection Pool Location	This Type of Pooling is Used for Connections Between...
Client side	Microsoft Transaction Server client components and the Oracle8 database
Server side	The Oracle Service for MTS and the Oracle8 database

The following illustration identifies the connection pool locations and the registry parameters associated with each pool:

Server side connection pooling registry parameters:

ORAMTS_SVC_CONN_POOL_TIMEOUT
ORAMTS_SVC_NET_CACHE_TIMEOUT
ORAMTS_SVC_NET_CACHE_MAXFREE
Server side parameters are set in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\OracleMTSService*N*.



Client side connection pooling registry parameters:

ORAMTS_CONN_POOL_TIMEOUT
ORAMTS_NET_CACHE_TIMEOUT
ORAMTS_NET_CACHE_MAXFREE
ORAMTS_OSCREDS_MATCH_LEVEL

Client side parameters are set in HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*ID*.

This table describes the client side registry parameters that you can modify to manage connection pooling:

Client Side Parameter	Description	Default Value Entry
ORAMTS_CONN_POOL_TIMEOUT	This parameter enables you to set how long a connection remains idle and available for reuse in the client side connection pool before timing out. After timing out, the connection is released.	60 seconds
ORAMTS_NET_CACHE_TIMEOUT	This parameter enables you to set how long the server portion of the connection remains idle and available for reuse in the client side connection pool before timing out. Each database connection consists of two portions: ■ The portion of the connection associated with ORAMTS_CONN_POOL_TIMEOUT is responsible for session issues such as user name, password, and Net8 connection information. ■ The portion of the connection associated with ORAMTS_NET_CACHE_TIMEOUT is responsible for server issues (the physical network connection). The server connection is the more expensive operation. It is advisable to keep this value set higher then the session timeout value associated with ORAMTS_CONN_POOL_TIMEOUT. After ORAMTS_CONN_POOL_TIMEOUT times out, the server portion of the connection associated with ORAMTS_NET_CACHE_TIMEOUT remains available for a slightly longer period of time. This portion remains available because when you create a connection to an Oracle8 database, the server portion of the connection requires more resources to initially establish than the session portion of the connection. A server connection can then be reused by creating a new session with it.	120 seconds **Note:** This value is in addition to the value you set for ORAMTS_CONN_POOL_TIMEOUT. For example, if you set ORAMTS_CONN_POOL_TIMEOUT to 180, and set ORAMTS_NET_CACHE_TIMEOUT to 60, the time period before a connection is *completely* terminated is 240 seconds.
ORAMTS_NET_CACHE_MAXFREE	This parameter enables you to set the maximum number of free server connections to maintain in the client side connection pool at any given time.	5

Client Side Parameter	Description	Default Value Entry
ORAMTS_ OSCREDS_ MATCH_LEVEL	This parameter enables you to set the degree of Windows NT security checking to perform on a connection when OS_ROLES is set to TRUE in the INIT.ORA file. When a user connects to the Oracle8 database (for example, with the CONNECT / command), there are certain database roles and privileges associated with their Windows NT user name. When the user disconnects, the connection becomes idle and available in the pool. When another user issues the CONNECT / command, the Windows NT identity of both users must match or the second user can receive the same database roles and privileges as the first user. This can be a security concern if the second user has only the CONNECT and RESOURCE database roles, but accidently receives the DBA database role associated with the first user. For this situation, setting this parameter to OS_ AUTH_LOGIN ensures that Windows NT security checking is performed. Furthermore, if the Oracle8 database has OS_ROLES set to TRUE, the roles of the operating system user are associated with a connection regardless of whether "CONNECT /" or "CONNECT *USER NAME*/*PASSWORD*" is performed. To enable Windows NT security checking in this case, set this parameter to ALWAYS. Windows NT security checking is an expensive operation. There is always a cost associated with Windows NT verifying the operating system credentials prior to reusing a connection. For performance reasons, it is advisable to set this parameter to NEVER. However, if you set OS_ROLES to TRUE or use operating system-authenticated connections, ensure that you set this parameter accordingly.	There are three possible values: ■ ALWAYS Windows NT security checking is always performed. This setting is the most secure, because it does not permit a second user to accidently receive the database roles and privileges of the first user. ■ OS_AUTH_LOGIN Windows NT security checking is only done if the user name and password are NULL. This is the default value. ■ NEVER *No* Windows NT security checking is performed. This setting is the least expensive of the three. Use this setting if you are *not* setting OS_ROLES to TRUE or *not* using operating system-authenticated connections.

This table describes the server side registry parameters that you can modify to manage connection pooling:

Server Side Parameter	Description	Default Value Entry
ORAMTS_SVC_CONN_POOL_TIMEOUT	This parameter enables you to set how long a connection remains idle and available for reuse in the server side connection pool before timing out. After timing out, the connection is released.	300 seconds
ORAMTS_SVC_NET_CACHE_TIMEOUT	This parameter enables you to set how long the server portion of the connection remains idle and available for reuse in the server side connection pool before timing out. Each database connection consists of two portions: ■ The portion of the connection associated with ORAMTS_SVC_CONN_POOL_TIMEOUT is responsible for session issues such as user name, password, and Net8 connection information. ■ The portion of the connection associated with ORAMTS_SVC_NET_CACHE_TIMEOUT is responsible for server issues (the physical network connection). The server connection is the more expensive operation. It is advisable to keep this value set higher then the session timeout value associated with ORAMTS_SVC_CONN_POOL_TIMEOUT. After ORAMTS_SVC_CONN_POOL_TIMEOUT times out, the server portion of the connection associated with ORAMTS_SVC_NET_CACHE_TIMEOUT remains available for a slightly longer period of time. This portion remains available because when you create a connection to an Oracle8 database, the server portion of the connection requires more resources to initially establish than the session portion of the connection. A server connection can then be reused by creating a new session with it.	0 **Note:** This value is in addition to the value you set for ORAMTS_SVC_CONN_POOL_TIMEOUT. For example, if you set this value to 60, and set ORAMTS_SVC_NET_CACHE_TIMEOUT to 60, the time duration before a connection is *completely* terminated is 120 seconds. The default values for ORAMTS_SVC_NET_CACHE_TIMEOUT and ORAMTS_SVC_NET_CACHE_MAXFREE are set to zero because the Oracle Service for MTS uses a single user to connect to the database. Therefore, the parameter ORAMTS_SVC_CONN_POOL_TIMEOUT can be used to configure the pool.
ORAMTS_SVC_NET_CACHE_MAXFREE	This parameter enables you to set the maximum number of free server connections to maintain in the server side connection pool at any given time.	0

To edit the connection pool registry settings:

1. Log on to the appropriate computer:

For...	Log on to the computer where...
Server side parameters	Oracle Service for MTS is installed
Client side parameters	Client Microsoft Transaction Server components are installed

2. Enter the following at the MS-DOS command prompt:

```
C:\> REGEDT32
```

3. Go to the appropriate registry location:

For...	Go to...
Server side parameters	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\OracleMTSService*N*
Client side parameters	HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*ID* where *ID* is the unique registry subkey of your Oracle home

4. Double-click the parameter to edit on the right side of the *Registry Editor* window.
5. Modify the value in the dialog box that appears, and click OK.
6. Choose Exit from the Registry main menu.

Increasing the Transaction Timeout Parameter

If your transaction requests are timing out before completing, it may be because the transaction timeout parameter is set too low. Increase the transaction timeout parameter to ensure that your transactions have sufficient time to complete.

To increase the transaction timeout parameter:

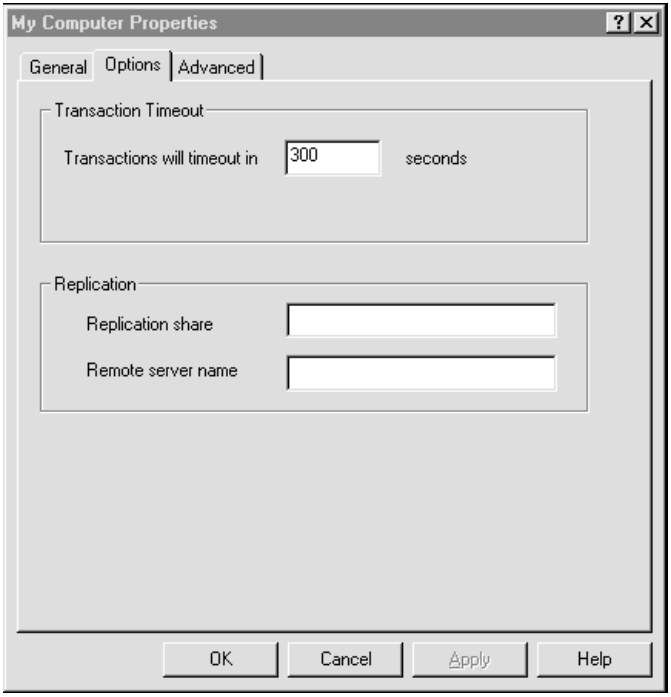
1. Go to the computer on which Microsoft Transaction Server is installed and the Oracle Service for MTS is configured.
2. Choose Start > Programs > Windows NT 4.0 Option Pack > Microsoft Transaction Server > Transaction Server Explorer.

   The Microsoft Management Console appears.
3. Double-click Console Root in the Microsoft Management Console Explorer window.
4. Double-click Microsoft Transaction Server.
5. Double-click Computers.
6. Right-click My Computer.

   A menu appears with several options.
7. Choose Properties.

   The *My Computer Properties* dialog box appears.
8. Choose the Options tab.

9. Enter a value in the Transaction Timeout field and click OK:



The transaction timeout value is increased. For most environments, 60 seconds may be sufficient. However, if your transaction is competing with numerous concurrent transactions, this value may be too low.

Changing Initialization Parameter File Settings

You may need to set several initialization parameters to the values described below in order to use your Oracle8 database with Microsoft Transaction Server. The values to which to set these parameters are based upon your database workload environment. See section "Verifying Initialization Parameter File Values" on page 3-3 for information on determining proper parameter settings.

To set initialization parameters:

1. Go to the computer on which the Oracle8 database is installed.
2. Go to the initialization parameter file for your Oracle8 database release:

If Using An...	Go to...
8.1.*x* databases	*ORACLE_BASE*\ADMIN*DB_NAME*\PFILE\INIT.ORA
8.0.6 databases	*ORACLE_HOME*\DATABASE\INIT*SID*.ORA

3. Set the following initialization parameters to at least these values:
   - SESSIONS = 200 (or larger if anticipating heavier loads)
   - PROCESSES = 200 (or larger if anticipating heavier loads)
4. Start SQL*Plus:

   ```
   C:\> SQLPLUS
   ```

5. Connect with the INTERNAL account:

   ```
   ENTER USER-NAME: INTERNAL
   ```

6. Shut down the Oracle8 database:

   ```
   SQL> SHUTDOWN
   ```

7. Restart the Oracle8 database:

   ```
   SQL> STARTUP
   ```

8. Exit SQL*Plus:

   ```
   SQL> EXIT
   ```

7

Troubleshooting

This chapter provides Oracle Service for MTS troubleshooting information.

Specific topics discussed are:

- Using Trace Files
- Correcting Windows NT Explorer Crashes
- Identifying Oracle Service for MTS Startup Problems
- Using the Event Viewer
- Starting MS DTC
- Viewing Oracle Service for MTS Configuration Information
- Correcting Net8 Changes that Impact Connection Pooling
- Ensuring Oracle Service for MTS Connectivity and Security
- Changing a Server Package's Windows NT User Account
- Common Questions

Using Trace Files

Trace files record information on Oracle Service for MTS performance. The information provided includes:

- Any errors
- Enlistment requests and outcomes
- Prepare, commit, and abort requests and outcomes

Use the Advanced tab of the Oracle Service for MTS *Properties* dialog box to specify trace levels and the output file name. You access this dialog box through the Oracle Manager for MTS Services snap-in of the Microsoft Management Console Explorer. See section "Modifying Oracle Service for MTS Trace File and MS DTC Information" on page B-10 for instructions.

In addition, there are two registry parameters to handle tracing within ORAMTS.DLL. ORAMTS.DLL implements the API for integrating the Oracle8 database with Microsoft Transaction Server. ORAMTS.DLL is a resource dispenser providing pooled OCI connections. ORAMTS.DLL also allows clients with non-pooled OCI connections to enlist in MS DTC-initiated transactions. The ORAMTS.DLL communicates with the Oracle Service for MTS to enlist the Oracle8 database in MS DTC-initiated transactions.

Parameter	Description	Data Type	Value Range
ORAMTS_CP_TRACE_LEVEL	Traces the Resource Dispenser layer within ORAMTS.DLL. A trace file is generated in the working directory of the process using ORAMTS.DLL. The trace file name uses the following format: ORAMTSCP*PID*.TRC where *PID* is the identification number of the process. New trace information is always added to the bottom of the file.	REG_EXP_SZ	0 to 4
ORAMTS_RPC_TRACE_LEVEL	Traces the remote procedure call (RPC) layer within ORAMTS.DLL. The RPC is used between the Oracle Service for MTS and the client to enlist in MS DTC transactions. A trace file is generated in the working directory of the process using ORAMTS.DLL. The trace file name uses the following format: ORAMTSRCP*PID*.TRC where *PID* is the identification number of the process. New trace information is always added to the bottom of the file.	REG_EXP_SZ	0 to 4

This table describes the range of trace values:

Level	Description
0	No tracing[1]
1	Traces errors only
2	Traces important events in addition to errors
4	Traces function entry / exit, important events, and errors

[1] If neither registry parameter is set in the registry or as an environment variable, then tracing is disabled (the same as setting the level to 0). Note also that Level 3 is not currently supported. If you set either parameter to 3, level 2 tracing is instead enabled.

Set these parameters either as environmental variables for in- or out-of-process Microsoft Transaction Server components at the MS-DOS command prompt or in the registry:

To Set...	Then...

As an environment variable for in-process Microsoft Transaction Server components	In-process Microsoft Transaction Server components run as DLLs in the address of the clients invoking them.
	1. Set ORAMTS_CP_TRACE_LEVEL to a value at the MS-DOS command prompt prior to launching your application:
	`C:\> SET ORAMTS_CP_TRACE_LEVEL=4`
	`C:\> MYINPROCAPP.EXE`

As an environment variable for out-of-process Microsoft Transaction Server components	Out-of-process Microsoft Transaction Server components are typically hosted by an external process, typically *ROOTDRIVE*:\WINNT\SYSTEM32\MTX.EXE.
	The environment for this process is the system environment. Therefore, set ORAMTS_CP_TRACE_LEVEL as a system-level or user-level environment variable. However, this requires the user to reboot the server. To avoid this, perform the following steps:
	1. Stop all running MTX.EXE processes:
	`C:\> MTXSTOP`
	2. Set the environment variable:
	`C:\> SET ORAMTS_CP_TRACE_LEVEL=2`
	3. Manually start the surrogate process. MTX.EXE takes an argument, which is the package name of interest.
	`C:\> START MTX.EXE /p:"OraMTSSample"`
	`C:\> MYOUTPROCAPP.EXE`

To Set...	Then...
As a registry variable	**1.** Enter the following at the MS-DOS command prompt: `C:\> REGEDT32` **2.** Go to HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*ID*, where *ID* is the identifier for your Oracle home directory. **3.** Choose the Add Value option in the Edit menu. The *Add Value* dialog box appears. **4.** Enter the appropriate parameter in the Value Name field. **5.** Select REG_EXPAND_SZ in the Data Type field and click OK. The *String Editor* dialog box appears. **6.** Enter an appropriate trace setting in the *String Editor* dialog box and click OK. **7.** Exit the registry.

Warning: Tracing at level 4 can lead to performance degradation and large trace files. In general, turn on tracing only for debugging purposes. ORAMTS.DLL is also used by the Oracle Service for MTS. Setting the parameters in the registry also controls tracing within the Oracle Service for MTS.

Oracle Service for MTS tracing is controlled through the Oracle Manager for MTS Services snap-in.

Correcting Windows NT Explorer Crashes

If you experience Windows NT Explorer crashes or other unexpected Windows NT problems when using Microsoft Transaction Server with an Oracle8 database, install the Windows NT 4.0 Service Pack 4 or greater (available from Microsoft).

Identifying Oracle Service for MTS Startup Problems

When the Oracle Service for MTS is correctly configured and running properly, the trace file displays the following message:

```
========================================
OracleMTSService0
9/11/1998 19:58:09
----------------------------------------
...
530515: [273] Oracle MTS Service - Accepting new enlistment requests.
...
```

If this message is not present and the Oracle Service for MTS is running (that is, the process has not exited), then the Oracle Service for MTS cannot complete the startup process. These are the most common causes for this problem:

Cause	Solution
The database has not been started or the account information of the service is incorrect.	Try connecting with SQL*Plus using the user name and net service name that the Oracle Service for MTS is configured to use. If SQL*Plus fails to connect, the Oracle Service for MTS also fails. To correct this problem, start up the database and/or correct the login information of the Oracle Service for MTS. See Appendix B, "Deleting or Modifying an Existing Oracle Service for MTS" for information.
MS DTC is not running.	Using the Microsoft Transaction Server Explorer, verify that MS DTC is running. If MS DTC is not running, see section "Starting MS DTC" on page 7-10 for startup instructions.

Cause	Solution
Recovery is being performed or cannot complete.	The Oracle Service for MTS does not start new transactions before all active transactions from an older run of the service are aborted. The following message is written to the trace file when recovery starts: `529765: [273] Recovery - Starting. In-doubt: number, Active: number.` The number following "Active:" indicates the number of active transactions to abort. If the trace file shows that active transactions cannot be aborted during recovery, do the following: **1.** Stop the Oracle Service for MTS. **2.** Shut down the Oracle8 database. **3.** Start the Oracle8 database. This procedure terminates all active transactions. **4.** Start the Oracle Service for MTS. The service should now start without any problems.

Windows NT 4.0 Service Pack 4 or greater is not installed.	Install Service Pack 4 or greater.
	When Service Pack 4 or greater is not installed, the following message is written to the trace file:
	`790172218: [135] DTCMgr::Initialize - Error: QueryInterface(IID_IResourceManager2) failed with error -2147467262.  Retrying.`
	`790172218: [135] DTCMgr::Initialize - Error: please make sure MS DTC is running.`
	`790172718: [299] OracleMTSService0 is exiting.`

Using the Event Viewer

Oracle Service for MTS problems and other significant occurrences are recorded as events. These events are recorded in an application event log. View and manage these recorded events in the Event Viewer. The types of events displayed include:

- MS DTC startup or shutdown status
- Recovery failure
- Oracle8 database startup or shutdown status
- Fatal errors that prevent Oracle Service for MTS startup

Accessing the Event Viewer

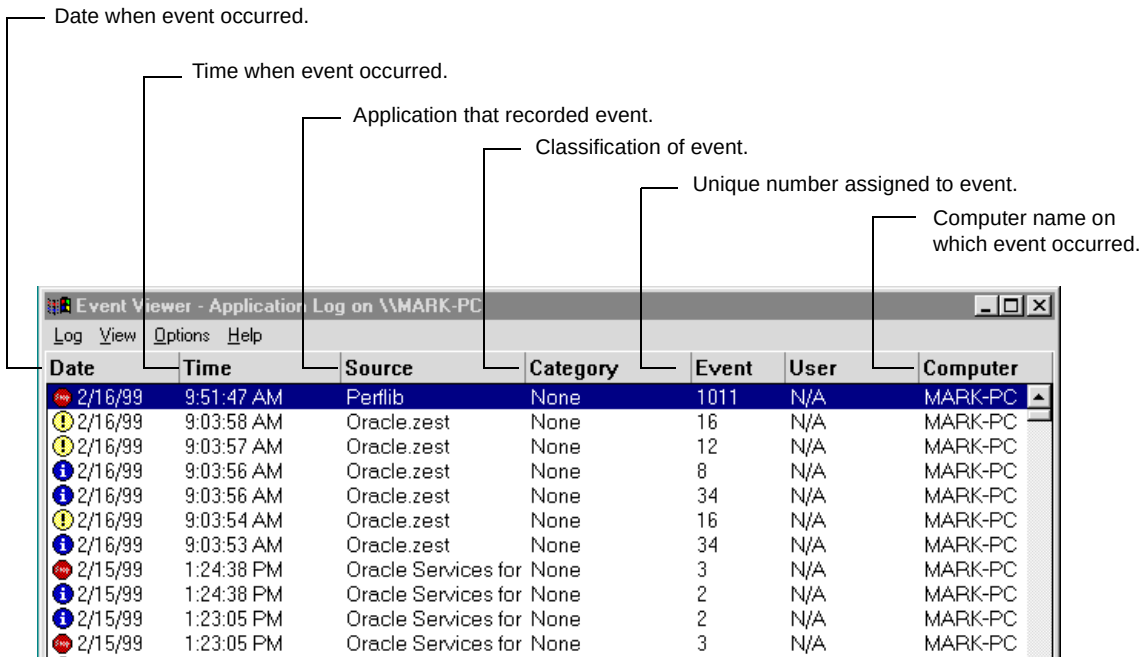
To access the Event Viewer:

1. Choose Start > Programs > Administrative Tools > Event Viewer.

   The *Event Viewer* window appears.

2. Choose Application from the Log menu.

   The *Application* view displays the following information:



Reading the Event Viewer

The icons beside each event determine the type of event:

Icon	Event Type	Description
red (stop sign)	Error	Indicates an error. Always check these icons.
blue (informational)	Information	Indicates a non-critical system event. You can ignore these icons unless you want to track a specific event.
yellow (exclamation point)	Warning	Indicates a special event, such as the termination of an instance or the shutdown of services. Investigate these icons, but they are usually non-critical.

The Oracle Service for MTS events display with a source of Oracle Service for MTS. Oracle Service for MTS consists of the following event IDs:

Event ID	Description
1	The Oracle Service for MTS started successfully.
2	The Oracle Service for MTS exited.
3	The Oracle Service for MTS failed to start. See the trace file for details.
4	The Oracle Service for MTS failed to start due to insufficient memory.
5	The Oracle Service for MTS failed to start. The service configuration parameters are missing or invalid.
6	The Oracle Service for MTS failed to start. Communication with the Service Control Manager failed.
7	The Oracle Service for MTS failed to start because CreateEvent failed.
8	The Oracle Service for MTS failed to start. The recovery mechanism failed. See the trace file for details.
9	The Oracle8 database has shut down, and the Oracle Service for MTS is stopping.
10	The Oracle Service for MTS has been notified that MS DTC is shutting down and exiting.

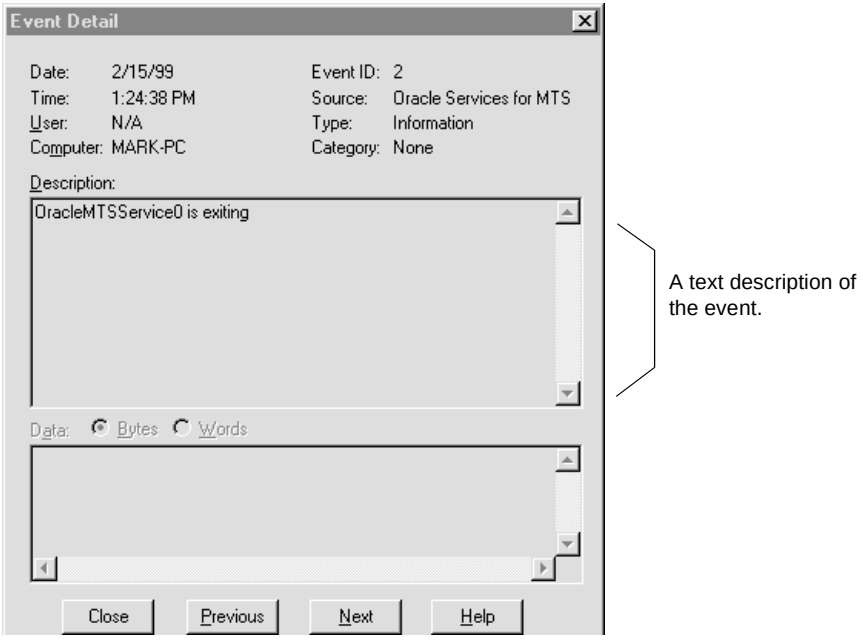
Event ID	Description
11	The trace file cannot be created and / or opened. The Oracle Service for MTS runs, but no trace information can be generated.
12	A parameter is missing.
13	A parameter is invalid.

Using the Event Viewer

To use the Event Viewer:

1. View the icons.
2. Double-click an icon to analyze (especially red icons).

   The *Event Detail* dialog box appears with more information about the selected event:



See Also: Your Microsoft Windows NT documentation for more information on using the Windows NT Event Viewer

Starting MS DTC

If the Oracle Service for MTS is running, but you do not receive a message in your trace file indicating that it is accepting new enlistment requests, ensure that the Microsoft Transaction Server's MS DTC component is started. MS DTC must be running to enable communication with the Oracle Service for MTS.

To start MS DTC:

1. Go to the computer on which Microsoft Transaction Server is installed and the Oracle Service for MTS is configured.
2. Choose Start > Programs > Windows NT 4.0 Option Pack > Microsoft Transaction Server > Transaction Server Explorer.

   The Microsoft Management Console appears.
3. Double-click Console Root in the Microsoft Management Console Explorer window.
4. Double-click Microsoft Transaction Server.
5. Double-click Computers.
6. Right-click My Computer.

   A menu appears with several options.
7. Choose Start MS DTC.

   MS DTC starts.

Viewing Oracle Service for MTS Configuration Information

The Oracle Manager for MTS Services snap-in includes an Identification tab in the Oracle Service for MTS *Properties* dialog box that enables you to view information that can be useful when debugging problems with Oracle Support Services. Note that none of the information in the Identification tab can be changed.

To view configuration information:

1. Go to the computer from which to modify the Oracle Service for MTS. The Oracle Service for MTS can be running on this computer or on a remote computer that you can access from this computer.
2. Choose Start > Programs > Oracle - *HOME_NAME* > Application Development > Oracle Manager for Microsoft Transaction Server.

   The Microsoft Management Console appears.
3. Find the Oracle Service for MTS to modify in the Explorer window.
4. Right-click the Oracle Service for MTS icon to modify.

   A menu appears with several options:

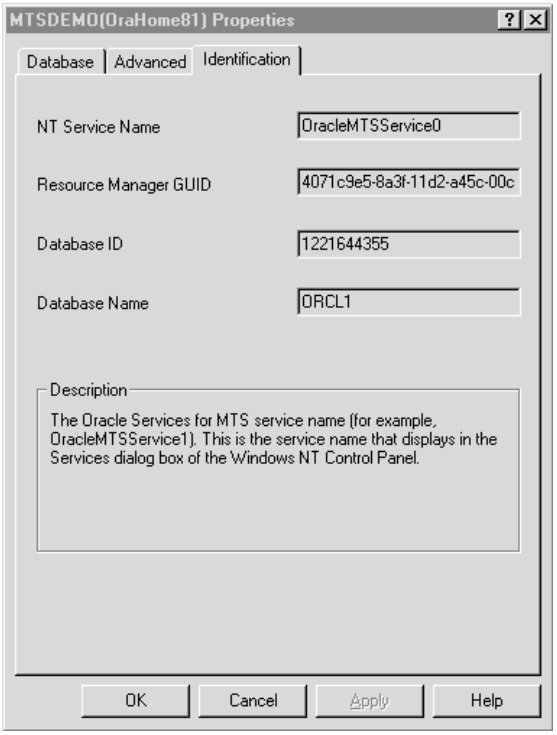
5. Choose Stop Service.

   A message indicates that the Oracle Service for MTS has stopped.
6. Right-click the same Oracle Service for MTS icon again.

   A menu appears with several options.
7. Choose Properties.

   The *Properties* dialog box appears.
8. Click the Identification tab.

9. View the following information:



Field	Description
NT Service Name	Displays the Oracle Service for MTS service name (for example, OracleMTSService0). This is the service name that displays in the *Services* dialog box of the Windows NT Control Panel.
Resource Manager GUID	Displays the resource manager global unique identifier (GUID) for identifying an Oracle8 database and its associated Oracle Service for MTS to MS DTC.
Database ID	Displays a value unique to each Oracle8 database. The Oracle Service for MTS retrieves this value from the database and sets it the first time it ever connects to that database.
Database Name	Displays the Oracle8 database name (as identified by the DB_NAME INIT.ORA file parameter value). The Oracle Service for MTS retrieves this value from the database and sets it the first time it connects to that database.

Correcting Net8 Changes that Impact Connection Pooling

The connection pool provided by the ORAMTS layer (that is, ORAMTS.DLL) uses a connection's Net8 net service name to identify pooled connections for an application. If changes are made to the Net8 net service name (for example, altering the host or the database system identifier [SID] for the Net8 net service name in TNSNAMES.ORA), and there are currently pooled connections, the application using the connection pool must be stopped and restarted.

This ensures that all currently pooled connections corresponding to the old Net8 net service name are destroyed and any new pooled connections use the changes made to the Net8 net service name. This includes both the Oracle Service for MTS and any application hosting Microsoft Transaction Server components.

Note: It is not recommended that you change the Net8 net service name that the Oracle Service for MTS uses to connect to a different database. If you must do this, follow the procedures in section "Preparing to Modify or Delete an Existing Oracle Service for MTS" on page B-3 before changing the Net8 net service name.

To empty connection pools:

1. Follow the instructions below:

If Your Application is an...	Then...
Out-of-process Microsoft Transaction Server component	Run the following application: `C:\> MTXSTOP` This empties the connection pools.
In-process Microsoft Transaction Server component	Kill the application, which also empties the connection pool.

Note: See the table in section "Using Trace Files" on page 7-2 for definitions of out-of-process and in-process Microsoft Transaction Server components.

Ensuring Oracle Service for MTS Connectivity and Security

Microsoft Transaction Server components are typically hosted within the Internet Information Server (IIS) or by a surrogate process (MTX.EXE). These processes typically log on to the operating system with their own user credentials. When a Microsoft Transaction Server component acquires an enlisted connection to an Oracle8 database, the component uses name pipe-based interprocess communication (IPC) to contact the Oracle Service for MTS for that database. To ensure that this pipe can be opened under all circumstances, perform the following tasks on the Windows NT computer on which the Oracle Service for MTS is located:

To ensure that the pipe can be opened:

1. Enter the following command at the MS-DOS command prompt:

   ```
   C:\> NET SHARE IPC$
   ```

2. Choose Start > Programs > Administrative Tools > User Manager.
3. Double-click the built-in Windows NT GUEST account (if it is not already enabled).

   The *User Properties* dialog box appears.
4. Ensure that the Account Disabled option is not selected.
5. Set the password to the empty string ("") in the Password and Confirm Password fields.
6. Click OK.
7. Go to the computer on which the Oracle database is located.
8. Start SQL*Plus:

   ```
   C:\> SQLPLUS
   ```

9. Log on with the SYSTEM account:

   ```
   SQL> CONNECT SYSTEM/PASSWORD
   ```

   where *PASSWORD* is MANAGER by default, unless you changed it after installation.
10. Enter the following query to obtain the name of the pipe used by the service:

    ```
    SQL> SELECT PIPENAME FROM MTS_PROXY_INFO;
    ```

    Your output is of the form PSIPE*n*, where *n* is some number.
11. Start the registry:

    ```
    C:\> REGEDT32
    ```

12. Go to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters.

    The parameter NullSessionPipes appears on the right side of the window. This parameter uses the data type REG_MULTI_SZ (for a list of strings).

13. Double-click NullSessionPipes.

    The *Multi-String Editor* dialog box appears.

14. Add the pipe name identified in step 10 to the list of pipes.

15. Click OK.

16. Exit the registry.

Changing a Server Package's Windows NT User Account

Packages that are activated as "server packages" and log on with the Windows NT System user account can have trouble enlisting in Microsoft Transaction Server transactions even when the Oracle Service for MTS and the Oracle database are up and running. This can be corrected by setting the package's identity in the Microsoft Transaction Server Explorer to a user account other than the Windows NT System account.

To set a package's identity to a different user account:

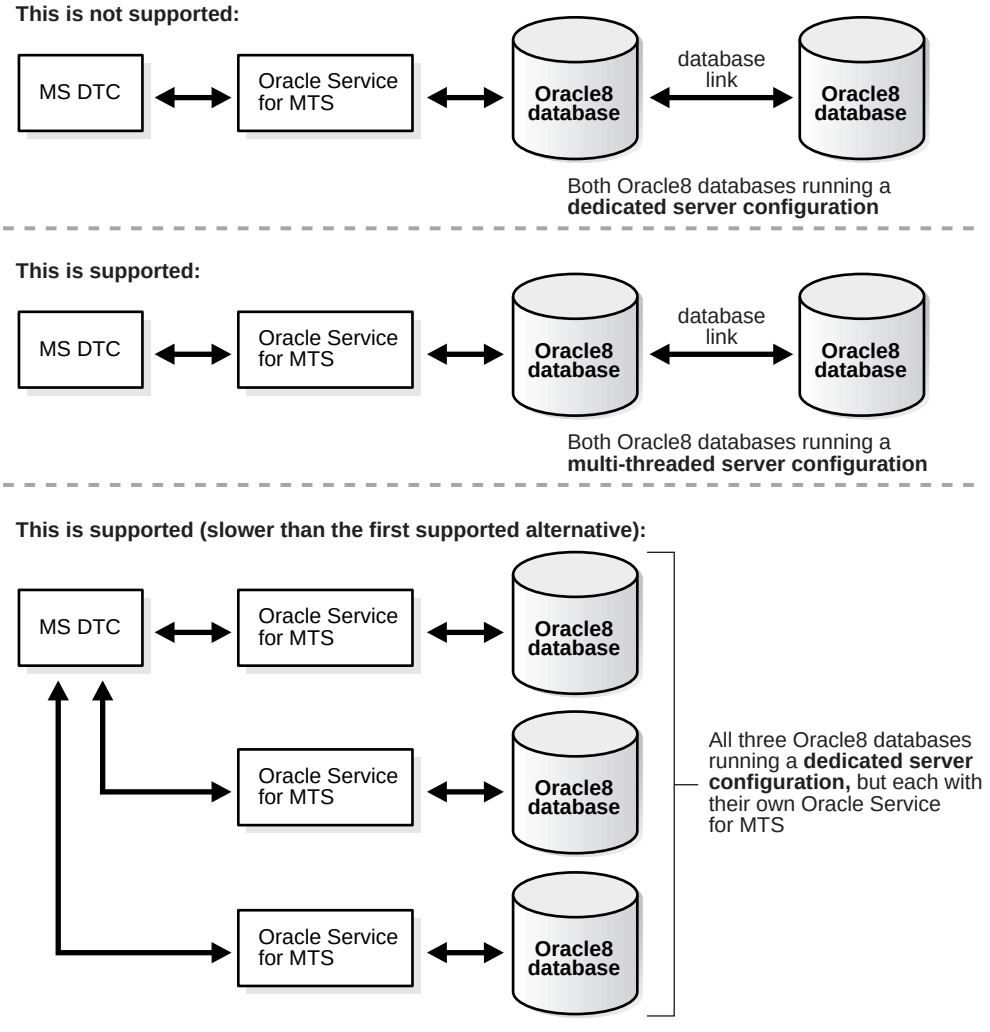
1. Choose Programs > Windows NT 4.0 Option Pack > Microsoft Transaction Server > Transaction Server Explorer to start Microsoft Transaction Server Explorer.
2. Double-click Computers.
3. Double-click MyComputer.
4. Double-click Packages Installed.
5. Select the appropriate package.
6. Right-click properties.
7. Click the Identity tab.
8. Change the logon account of the package.
9. Click OK.
10. Exit Microsoft Transaction Server Explorer.

Common Questions

This section presents answers to common questions.

Question: How do I design an application when I have multiple Oracle8 databases?

Oracle Service for MTS does not support the use of database links between Oracle8 databases running in *dedicated server* configurations. As alternatives, the Oracle Service for MTS supports using database links in a multi-threaded server environment or providing each Oracle8 database with its own Oracle Service for MTS in a dedicated server environment. This figure illustrates these alternatives:



Question: What are the differences between Oracle's Net8 connection pooling, OCI connection pooling, and Microsoft Transaction Server connection pooling?

Net8 connection pooling is a server-side feature that is implemented only if your Oracle8 database is configured for multi-threaded server support. Net8 connection pooling enables you to minimize the number of physical network connections to a multi-threaded server. This is achieved by sharing a dispatcher's set of connections among multiple client processes.

OCI connection pooling is a client-side feature that allows for the caching of Net8 connections to Oracle8 databases. It also allows for the multiplexing of multiple sessions to an Oracle8 database over a single Net8 connection. This second feature is not supported in ORAMTS.DLL for this release because of potential deadlock issues.

Microsoft Transaction Server connection pooling maintains a pool of Net8 connections and only uses one OCI session per Net8 connection.

Question: Can you use operating system authentication with your Oracle Service for MTS user name?

Yes, but only on Windows NT. Operating system authentication is not supported when connecting to databases on other platforms. See the following sections in this guide for instructions on creating an operating system-authenticated Oracle Service for MTS user:

To Enable Operating System Authentication For...	See...
A new Oracle Service for MTS	"Creating a New Operating System-Authenticated Oracle Service for MTS" on page A-8
An existing Oracle Service for MTS	"Enabling an Existing Oracle Service for MTS to be Operating System-Authenticated" on page A-11

A

Manually Creating Oracle Service for MTS Users

This appendix describes how to manually create the MTSSYS user, a custom user, or an operating system-authenticated user on Windows NT.

Specific topics discussed are:

- Manually Creating the MTSSYS User
- Manually Creating a Custom Oracle Service for MTS User
- Creating a New Operating System-Authenticated Oracle Service for MTS
- Enabling an Existing Oracle Service for MTS to be Operating System-Authenticated

Manually Creating the MTSSYS User

This section describes how to manually create the MTSSYS user (if it is not included in your Oracle8 database) and execute the scripts that enable you to run the sample application demo described in section "Using OCI with the Microsoft Application Demo" on page 4-2. See section "Verifying the Oracle Service for MTS User Name" on page 3-4 to determine if your Oracle8 database includes this user.

To manually create an MTSSYS user:

1. Log on to the computer where your Oracle8 database is installed.
2. Start SQL*Plus:

   ```
   C:\> SQLPLUS
   ```

3. Connect with the INTERNAL user name:

   ```
   ENTER USER-NAME: INTERNAL
   ```

4. If you want to use the Microsoft application demo described in section "Using OCI with the Microsoft Application Demo" on page 4-2, go to step 4a. Otherwise, go to step 5.

   a. Verify that you can connect with the SCOTT user name.

      ```
      SQL> CONNECT SCOTT/PASSWORD
      ```

      where *PASSWORD* is TIGER after installation, unless you changed it.

   b. Follow the steps below based on whether or not you connected to the Oracle8 database:

If You...	Then...
Connected as user SCOTT	Go to step 5.
Did not connect as user SCOTT	Go to step 4c.

   c. Run the following script:

      ```
      SQL> @ORACLE_BASE\ORACLE_HOME\RDBMS\ADMIN\SCOTT.SQL;
      ```

      This creates the user SCOTT, which is used by the sample application demo.

   d. Go to step 5.

5. Run the ORAMTS.SQL script:

   ```
   SQL> @ORACLE_BASE\ORACLE_HOME\RDBMS\ADMIN\ORAMTS.SQL;
   ```

> **Note:** Oracle database release 8.0.6 does not include the ORAMTS.SQL script. You must obtain this script from Oracle database release 8.1.*x*.

This creates the user MTSSYS and two tables (ACCOUNT and RECEIPT) under user SCOTT that are used by the Microsoft application demo.

> **Note:** If your Oracle8 database did not include user SCOTT and you did not run SCOTT.SQL, error messages appear when running ORAMTS.SQL. Ignore these messages. The product functions properly, but the sample application demo will not work.

6. Connect with the MTSSYS user name and MTSSYS password:

   ```
   SQL> CONNECT MTSSYS/MTSSYS
   ```

7. Change the password for the MTSSYS user:

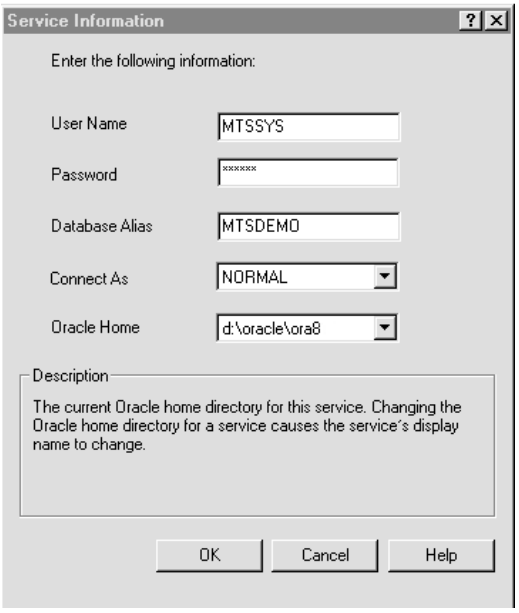
   ```
   SQL> ALTER USER MTSSYS IDENTIFIED BY NEW_PASSWORD
   ```

   This changes your MTSSYS user password.

8. Exit SQL*Plus:

   ```
   SQL> EXIT
   ```

9. Follow the procedures in section "Creating a New Oracle Service for MTS" on page 3-6 to create an Oracle Service for MTS user. When you get to step 10 on page 3-9, enter the MTSSYS user name and password in the *Service Information* dialog box:



10. Complete the remaining procedures (steps 11 through 13) in section "Creating a New Oracle Service for MTS" on page 3-9.

    When complete, this creates an MTSSYS user for the Oracle Service for MTS.

Manually Creating a Custom Oracle Service for MTS User

This section describes how to manually create a custom Oracle Service for MTS user (if you do not want to use the user MTSSYS) and execute the scripts that enable you to run the Microsoft application demo described in section "Using OCI with the Microsoft Application Demo" on page 4-2.

To manually create a custom user:

1. Go to the computer where your Oracle8 database is installed.
2. Start SQL*Plus:

   ```
   C:\> SQLPLUS
   ```

3. Connect with the INTERNAL user name:

   ```
   ENTER USER-NAME: INTERNAL
   ```

4. If you want to use the Microsoft application demo described in section "Using OCI with the Microsoft Application Demo" on page 4-2, go to step 4a. Otherwise, go to step 5.

   a. Verify that you can connect with the SCOTT user name.

      ```
      SQL> CONNECT SCOTT/PASSWORD
      ```

      where *PASSWORD* is TIGER after installation, unless you changed it.

   b. Follow the steps below based on whether or not you connected to the Oracle8 database:

If You...	Then...
Connected as user SCOTT	Go to step 5.
Did not connect as user SCOTT	Go to step 4c.

   c. Run the following script:

      ```
      SQL> @ORACLE_BASE\ORACLE_HOME\RDBMS\ADMIN\SCOTT.SQL;
      ```

      This creates the user SCOTT, which is used by the sample application demo.

   d. Go to step 5.

5. Verify that you can connect with the MTSSYS user name:

   ```
   SQL> CONNECT MTSSYS/PASSWORD
   ```

   where *PASSWORD* is MTSSYS after installation, unless you changed it. The MTSSYS user name is required to run the script that enables you to create a custom Oracle Service for MTS user.

6. Follow the steps below based on whether or not you connected to the Oracle8 database:

If You...	Then...
Connected as user MTSSYS	Go to step 7.
Did not connect as user MTSSYS	Go to step 6a.

   a. Run the ORAMTS.SQL script:

      ```
      SQL> @ORACLE_BASE\ORACLE_HOME\RDBMS\ADMIN\ORAMTS.SQL;
      ```

      This creates the user MTSSYS and two tables (ACCOUNT and RECEIPT) under user SCOTT that are used by the Microsoft application demo.

      Note: If your Oracle8 database did not include user SCOTT and you did not run SCOTT.SQL, error messages appear when running ORAMTS.SQL. Ignore these messages. The product functions properly, but the Microsoft application demo will not work.

   b. Go to step 7.

7. Go to the computer where your Oracle Services for Microsoft Transaction Server is installed.
8. Go to the *ORACLE_BASE**ORACLE_HOME*\ORAMTS\ADMIN directory.
9. Use an ASCII editor to open the file MTSUSER.SQL.
10. Replace all occurrences of MTS_USER with a custom user name (in this example, the user name entered is FRANK).
11. Replace all occurrences of MTS_PASSWORD with a custom password.
12. Save your changes and close the file.

13. Start SQL*Plus:

    ```
    C:\> SQLPLUS
    ```

14. Connect with the INTERNAL user name:

    ```
    ENTER USER-NAME: INTERNAL
    ```

15. Run the MTSUSER.SQL script:

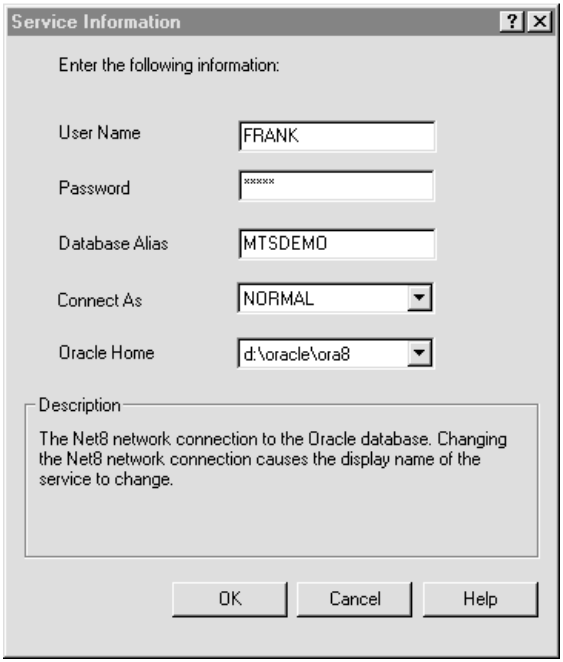
    ```
    SQL> @ORACLE_BASE\ORACLE_HOME\ORAMTS\ADMIN\MTSUSER.SQL;
    ```

    This creates a custom Oracle Service for MTS user.

16. Exit SQL*Plus:

    ```
    SQL> EXIT
    ```

17. Follow the procedures in section "Creating a New Oracle Service for MTS" on page 3-6. When you get to step 10 on page 3-9, enter the custom user name and password in the *Service Information* dialog box:



18. Complete the remaining procedures (steps 11 through 13) in section "Creating a New Oracle Service for MTS" on page 3-9.

    When complete, this creates a custom user for the Oracle Service for MTS.

Creating a New Operating System-Authenticated Oracle Service for MTS

Your Oracle Service for MTS user name can be authenticated by the Windows NT operating system. This provides you with the advantage of not having to enter a user name and password when creating your Oracle Service for MTS.

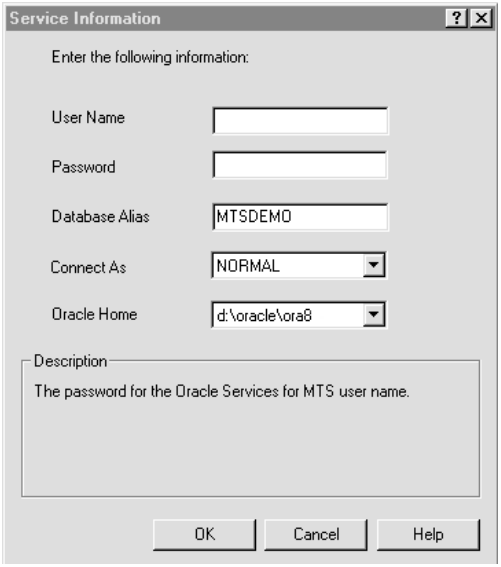
Note: These instructions assume you are familiar with enabling Oracle8 database users to be authenticated by Windows NT. See the following documentation for additional information:

- For Oracle release 8.1.6, see your *Oracle8i Administrator's Guide for Windows NT*
- For Oracle release 8.0.6, see your *Getting Started* guide

To create an operating system-authenticated Oracle Service for MTS user:

1. Authenticate a Windows NT user name by following the instructions in the above-mentioned guides. Ensure that you grant the following roles and privileges to the user:
   - CONNECT, RESOURCE, and SELECT_CATALOG_ROLE roles
   - FORCE ANY TRANSACTION, CREATE PUBLIC SYNONYM, and DROP PUBLIC SYNONYM privileges
2. Log onto Windows NT with the authenticated user name.

3. Follow the procedures in section "Creating a New Oracle Service for MTS" on page 3-6. When you get to step 10 on page 3-9, leave the User Name and Password fields blank in the *Service Information* dialog box:

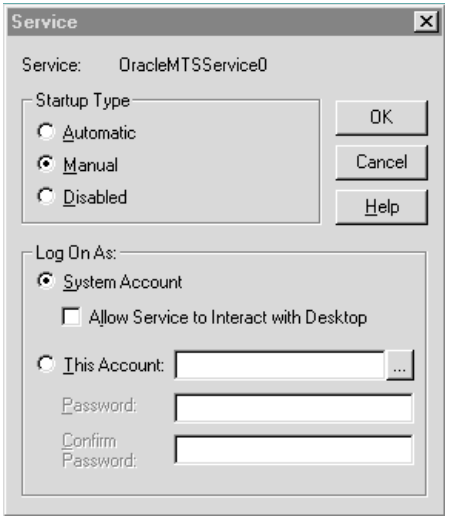


4. Complete the remaining procedures (steps 11 through 13) in section "Creating a New Oracle Service for MTS" on page 3-9.
5. Choose Start > Settings > Control Panel.

   The *Control Panel* window appears.
6. Double-click Services.
7. The *Services* dialog box appears.
8. Select OracleMTSService*n* and click Startup.

   where *n* is the Oracle Service for MTS number you want to modify.

The *Service* dialog box appears:



9. Select *This Account*.
10. Specify the operating system authenticated user name with which you logged onto the computer in step 2.
11. Enter the password in the Password field.
12. Enter the password a second time in the Confirm Password field.
13. Click OK to exit the *Service* dialog box.
14. Click Close to exit the *Services* dialog box.
15. Exit the *Control Panel* window.
16. Choose Start > Programs > Oracle - *HOME_NAME* > Application Development > Oracle Manager for Microsoft Transaction Server.

    The Microsoft Management Console appears.
17. Find the Oracle Service for MTS to modify in the Explorer window.
18. Right-click the new Oracle Service for MTS icon.

    A menu appears with several options.
19. Choose Stop Service.

The service stops.

20. Right-click the same Oracle Service for MTS icon again.

    A menu appears with several options.

21. Choose Start Service.

    The service starts.

Enabling an Existing Oracle Service for MTS to be Operating System-Authenticated

You can modify an existing Oracle Service for MTS to be associated with an operating system-authenticated user on Windows NT.

Note: These instructions assume you are familiar with enabling Oracle8 database users to be authenticated by Windows NT. See the following documentation for additional information:

- For Oracle release 8.1.6, see your *Oracle8i Administrator's Guide for Windows NT*
- For Oracle release 8.0.6, see your *Getting Started* guide

To modify an existing user to be operating system-authenticated:

1. Authenticate a Windows NT user name by following the instructions in the above-mentioned guides. Ensure that you grant the following roles and privileges to the user:
   - CONNECT, RESOURCE, and SELECT_CATALOG_ROLE roles
   - FORCE ANY TRANSACTION, CREATE PUBLIC SYNONYM, and DROP PUBLIC SYNONYM privileges
2. Log onto Windows NT with the authenticated user name.

3. Choose Start > Settings > Control Panel.

   The *Control Panel* window appears.

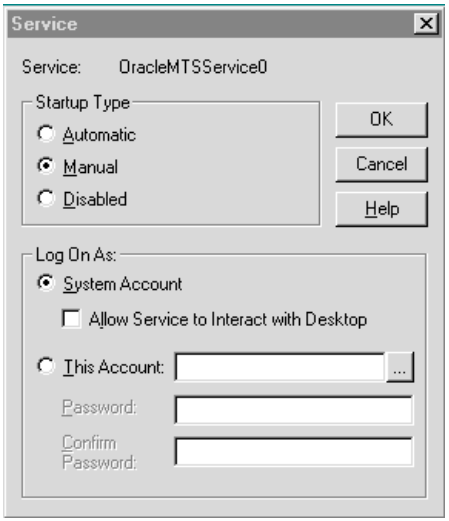
4. Double-click Services.

5. The *Services* dialog box appears.

6. Select OracleMTSService*n* and click Startup.

   where *n* is the Oracle Service for MTS number you want to modify.

   The *Service* dialog box appears:



7. Select This Account.

8. Specify the operating system authenticated user name with which you logged onto the computer in step 2.

9. Enter the password in the Password field.

10. Enter the password a second time in the Confirm Password field.

11. Click OK to exit the *Service* dialog box.

12. Click Close to exit the *Services* dialog box.

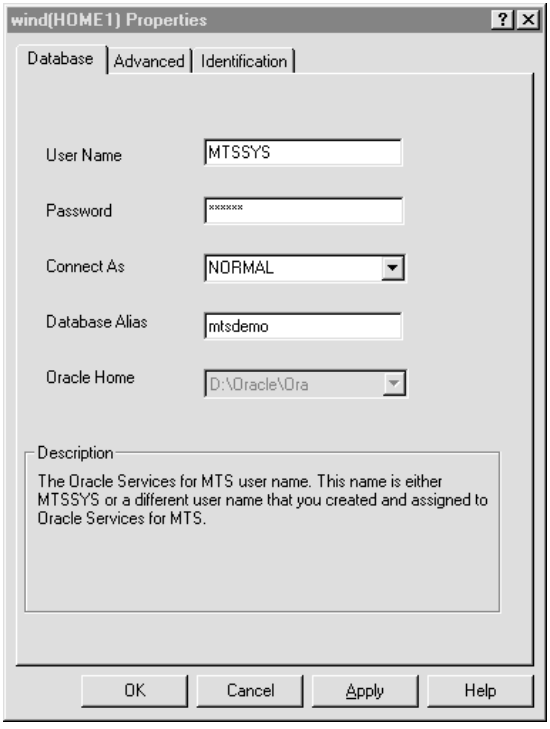
13. Exit the *Control Panel* window.

14. Go to the computer from which to modify the Oracle Service for MTS. The Oracle Service for MTS can be running on this computer or on a remote computer that you can access from this computer.
15. Choose Start > Programs > Oracle - *HOME_NAME* > Application Development > Oracle Manager for Microsoft Transaction Server.

    The Microsoft Management Console appears.
16. Find the Oracle Service for MTS to modify in the Explorer window.
17. Right-click the Oracle Service for MTS icon to modify.

    A menu appears with several options.
18. Choose Properties.

    The *Properties* dialog box contains a Database tab that displays the following information:



19. Remove all information from the User Name and Password fields.

20. Click OK.
21. Right-click the same Oracle Service for MTS icon again.

    A menu appears with several options.
22. Choose Stop Service.

    The service stops.
23. Right-click the same Oracle Service for MTS icon again.

    A menu appears with several options.
24. Choose Start Service.

    The service starts.

B

Deleting or Modifying an Existing Oracle Service for MTS

This appendix describes how to delete or modify an existing Oracle Service for MTS.

Specific topics discussed are:

- How Do I Modify or Delete an Oracle Service for MTS?
- Preparing to Modify or Delete an Existing Oracle Service for MTS
- Deleting an Existing Oracle Service for MTS
- Modifying Oracle Service for MTS Connection Information
- Modifying Oracle Service for MTS Trace File and MS DTC Information
- Manually Deleting or Modifying the Oracle Service for MTS with the Registry
- Deleting Roles and Privileges of an Inactive Oracle Service for MTS User

How Do I Modify or Delete an Oracle Service for MTS?

You cannot modify an existing Oracle Service for MTS to connect to a different Oracle8 database. The Oracle8 database for which the Oracle Service for MTS was originally created must always remain the same. Instead, delete the existing Oracle Service for MTS and create a new one for the second Oracle8 database. If necessary, however, you can modify the database alias, Oracle home, user name, and password of an existing Oracle Service for MTS as long as the Oracle8 database remains the same. The table below describes the procedures to follow for deleting or modifying an existing Oracle Service for MTS:

On The...	What Procedures Must You Perform if You Want to...	
	Delete the Oracle Service for MTS?	**Modify the Oracle Service for MTS?**
Client computer	None.	None.
Windows NT computer where Microsoft Transaction Server is installed	To delete an existing Oracle Service for MTS, see sections: ■ "Preparing to Modify or Delete an Existing Oracle Service for MTS" on page B-3 ■ "Deleting an Existing Oracle Service for MTS" on page B-5 ■ "Deleting Roles and Privileges of an Inactive Oracle Service for MTS User" on page B-16 If Oracle Service for MTS deletion is unsuccessful, see section: ■ "Manually Deleting or Modifying the Oracle Service for MTS with the Registry" on page B-13	To modify an existing Oracle Service for MTS, see sections: ■ "Preparing to Modify or Delete an Existing Oracle Service for MTS" on page B-3 Then see the appropriate section: ■ "Modifying Oracle Service for MTS Connection Information" on page B-7 ■ "Modifying Oracle Service for MTS Trace File and MS DTC Information" on page B-10 If Oracle Service for MTS modification is unsuccessful, see section: ■ "Manually Deleting or Modifying the Oracle Service for MTS with the Registry" on page B-13
Computer where the Oracle8 database is installed	None.	None.

Preparing to Modify or Delete an Existing Oracle Service for MTS

Modifying a currently operating Oracle Service for MTS to connect to a different MS DTC; modifying the net service name, Oracle home name, user name, or password; or deleting an Oracle Service for MTS *may* require the DBA to manually commit or abort transactions that did not successfully complete and / or recover.

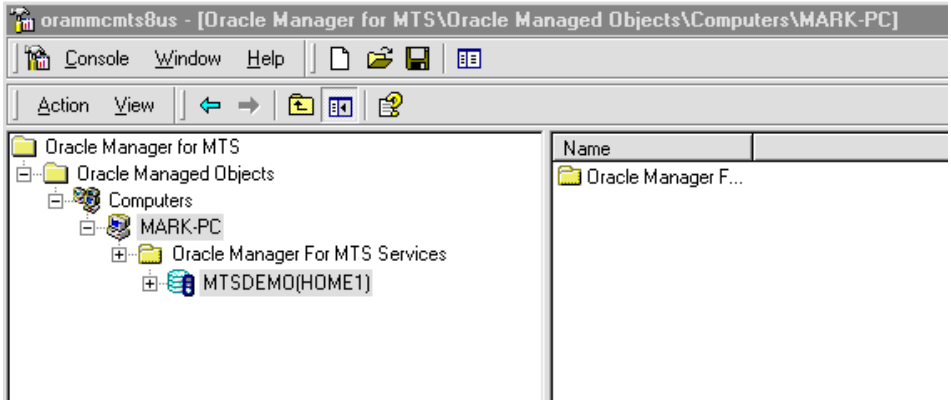
For this reason, you must prepare the Oracle Service for MTS before it can be deleted or modified to access a different MS DTC or use different connection information. Follow these procedures to ensure that:

- all transactions are completed before deleting or modifying the Oracle Service for MTS
- no manual transaction commits or aborts are required

To prepare to modify or delete an existing Oracle Service for MTS:

1. Go to the computer from which to modify the Oracle Service for MTS. The Oracle Service for MTS can be running on this computer or on a remote computer that you can access from this computer.
2. Choose Start > Programs > Oracle - *HOME_NAME* > Application Development > Oracle Manager for Microsoft Transaction Server.

   The Microsoft Management Console appears.
3. Find the Oracle Service for MTS to modify in the Explorer window.
4. Right-click the Oracle Service for MTS icon to modify (*MTSDEMO* in this example):



A menu appears with several options.

5. Choose Stop Service.

   A message indicates that the Oracle Service for MTS has stopped.

6. Click OK.

7. Go to the computer on which the Oracle8 database is running.

8. Start SQL*Plus:

   ```
   C:\> SQLPLUS
   ```

9. Connect with the INTERNAL user name:

   ```
   ENTER USER-NAME: INTERNAL
   ```

10. Shut down the Oracle8 database:

    ```
    SQL> SHUTDOWN
    ```

11. Restart the Oracle8 database:

    ```
    SQL> STARTUP
    ```

12. Exit SQL*Plus:

    ```
    SQL> EXIT
    ```

13. Return to the computer on which to start the Oracle Service for MTS.

14. Choose Start > Programs > Oracle - *HOME_NAME* > Application Development > Oracle Manager for Microsoft Transaction Server.

    The Microsoft Management Console appears.

15. Find the Oracle Service for MTS to start in the Explorer window.

16. Right-click the Oracle Service for MTS icon.

    A menu appears with several options.

17. Choose Start Service.

    A message indicates that the Oracle Service for MTS has started.

18. Click OK.

19. Do not allow any new transactions to use the Oracle Service for MTS.

20. Monitor the Oracle Service for MTS trace file for a message indicating that recovery has completed successfully. This file is located in *ORACLE_BASE**ORACLE_HOME*\ORAMTS\TRACE.

21. Right-click the Oracle Service for MTS icon.

22. Choose Stop Service.

    A message indicates that the Oracle Service for MTS has stopped.

23. Click OK.

24. Go to the computer on which the Oracle8 database is running.

25. Start SQL*Plus:

    ```
    C:\> SQLPLUS
    ```

26. Connect with the INTERNAL user name:

    ```
    ENTER USER-NAME: INTERNAL
    ```

27. Delete this information from the following table:

    ```
    SQL> DELETE FROM MTS_PROXY_INFO;
    ```

    ```
    SQL> COMMIT;
    ```

Deleting an Existing Oracle Service for MTS

This section describes how to delete an existing Oracle Service for MTS. You *must* use the Microsoft Management Console Explorer to delete the Oracle Service for MTS. Deleting the Oracle Service for MTS in any other way (such as with your keyboard's Delete button) can cause data inconsistencies in the Oracle8 database. These inconsistencies require the DBA to manually commit or abort transactions that did not successfully complete and / or recover.

To delete an existing Oracle Service for MTS:

1. Follow the procedures described in section "Preparing to Modify or Delete an Existing Oracle Service for MTS" on page B-3 to ensure that all transactions are resolved *before* deleting an existing Oracle Service for MTS.

2. Go to the computer from which to delete the Oracle Service for MTS. The Oracle Service for MTS can be running on this computer or on a remote computer that you can access from this computer.

3. Choose Start > Programs > Oracle - *HOME_NAME* > Application Development > Oracle Manager for Microsoft Transaction Server.

   The Microsoft Management Console appears.

4. Find the Oracle Service for MTS to delete in the Explorer window.

5. Right-click the Oracle Service for MTS icon.

A menu appears with several options.

6. Choose Stop Service.

   A message indicates that the Oracle Service for MTS has stopped.

7. Click OK.

8. Right-click the same Oracle Service for MTS again.

   A menu appears with several options.

9. Choose Delete.

   If successful, a message indicates that the Oracle Service for MTS has been deleted.

   If unsuccessful, a message indicates that the Oracle Service for MTS has not been deleted. See section "Manually Deleting or Modifying the Oracle Service for MTS with the Registry" on page B-13. After completing those procedures, retry steps 8 and 9 above.

Modifying Oracle Service for MTS Connection Information

The Oracle Service for MTS includes a Database tab in the *Properties* dialog box that enables you to modify the net service name, Oracle home, user name, and password of an existing Oracle Service for MTS.

To modify Oracle Service for MTS connection information:

1. Follow the procedures described in section "Preparing to Modify or Delete an Existing Oracle Service for MTS" on page B-3 to ensure that all transactions are resolved *before* modifying an existing Oracle Service for MTS.
2. Go to the computer from which to modify the Oracle Service for MTS. The Oracle Service for MTS can be running on this computer or on a remote computer that you can access from this computer.
3. Choose Start > Programs > Oracle - *HOME_NAME* > Application Development > Oracle Manager for Microsoft Transaction Server.

   The Microsoft Management Console appears.
4. Find the Oracle Service for MTS to modify in the Explorer window.
5. Right-click the Oracle Service for MTS icon to modify.

   A menu appears with several options.

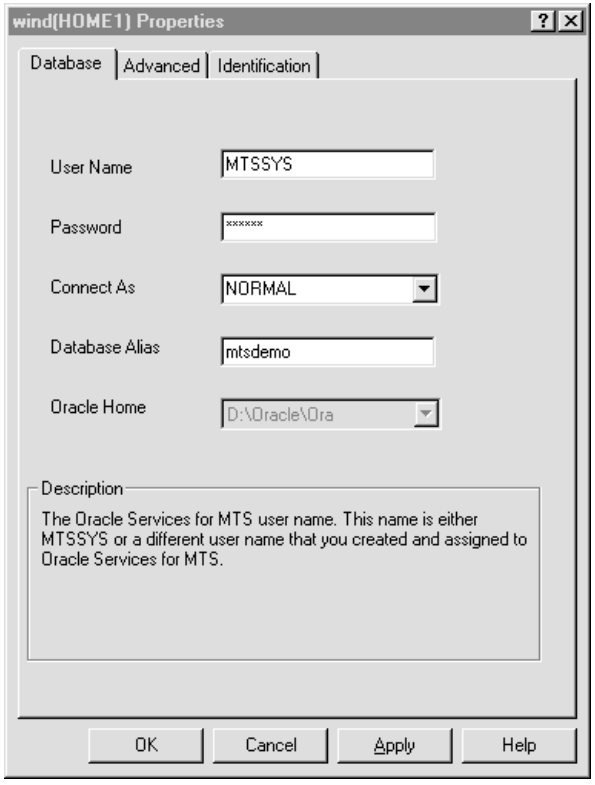
6. Choose Stop Service.

   A message indicates that the Oracle Service for MTS has stopped.
7. Right-click the same Oracle Service for MTS icon again.

   A menu appears with several options.
8. Choose Properties.

   The *Properties* dialog box appears.
9. Click the Database tab.

The Database tab of the *Properties* dialog box displays information similar to the following:



10. Modify the following information, as appropriate:

> **Attention:** The user name, password, database alias, and Oracle home that you specify must still connect to the *same* Oracle8 database for which the Oracle Service for MTS was originally created.

Field	Displays the...
User Name	Oracle Service for MTS user name. This user name is either MTSSYS, a custom name you manually created, or no name (if using operating system authentication).
Password	Password for the Oracle Service for MTS user name (or none if using operating system authentication).
Connect As	Privilege with which the service connects to the Oracle8 database (NORMAL, SYSOPER, or SYSDBA).
Database Alias	Net service name for connecting to the Oracle8 database. Changing the net service name causes the display name of the service to change. For example, if the Oracle home directory was C:\ORACLE\ORA81 (home name ORA81) and the Net8 network connection was changed from FINAPS_TEST.WORLD to FINAPS_PROD.WORLD, then the display name changes from FINAPS_TEST.WORLD(ORA81) to FINAPS_PROD.WORLD(ORA81).
Oracle Home	Oracle home directory for this service. Changing the Oracle home directory for a service causes the service's display name to change. For example, if the net service name for the service is FINAPS.WORLD and the home directory was changed from C:\ORANT81_TEST (home name ORANT81TST) to C:\ORANT81_PROD (home name ORANT81PRD), then the display name changes from FINAPS.WORLD(ORANT81TST) to FINAPS.WORLD(ORANT81PRD).

11. Click OK.

If successful, the Oracle Service for MTS can now connect to the Oracle8 database with the information specified in step 10.

If not successful, the Oracle Service for MTS is unable to connect to the Oracle8 database with the information specified in step 10. See section "Manually Deleting or Modifying the Oracle Service for MTS with the Registry" on page B-13. After completing those procedures, retry steps 7 through 11 above.

Modifying Oracle Service for MTS Trace File and MS DTC Information

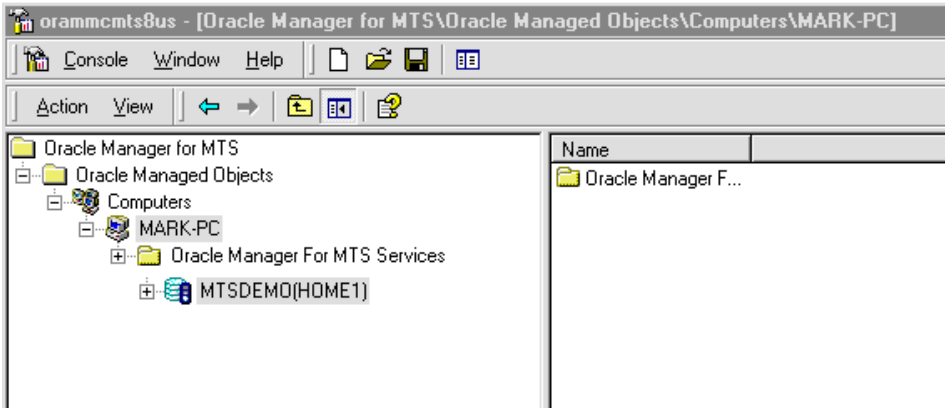
The Oracle Service for MTS includes an Advanced tab in the *Properties* dialog box that enables you to modify the following information:

- trace file name and level of trace detail
- MS DTC location

To modify trace file and MS DTC information:

1. Follow the procedures described in section "Preparing to Modify or Delete an Existing Oracle Service for MTS" on page B-3 to ensure that all transactions are resolved before modifying an existing Oracle Service for MTS.
2. Go to the computer from which to modify the Oracle Service for MTS. The Oracle Service for MTS can be running on this computer or on a remote computer that you can access from this computer.
3. Choose Start > Programs > Oracle - *HOME_NAME* > Application Development > Oracle Manager for Microsoft Transaction Server.

   The Microsoft Management Console appears.
4. Find the Oracle Service for MTS to modify in the Explorer window.
5. Right-click the Oracle Service for MTS icon to modify (*MTSDEMO* in this example):



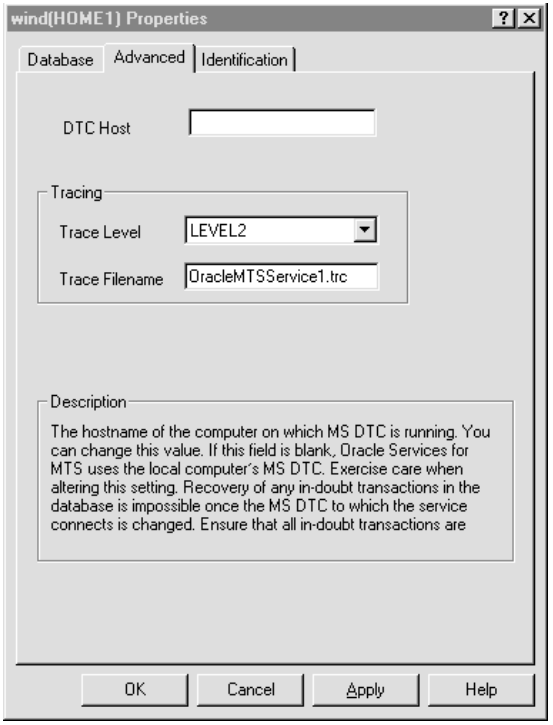
A menu appears with several options.

6. Choose Properties.

   The *Properties* dialog box appears.

7. Click the Advanced tab.

   The Advanced tab of the *Properties* dialog box displays the following information:



8. Make appropriate changes.

Field	Displays the...
DTC Host	Hostname of the computer on which MS DTC is running. You can change this value. If this field is blank, the Oracle Service for MTS uses the local computer's MS DTC. When changing the hostname, ensure that the service can: ■ connect to the new host (has net logon privileges) ■ access the MS DTC on the new host The default service account is "LocalSystem", which may not have privileges to log on to the remote host. In such cases, change the logon account of the service (using the Control Panel) to an account that can access the new host. Exercise care when altering this setting. Recovery of any in-doubt transactions in the database is impossible once the MS DTC to which the service connects is changed. Ensure that all in-doubt transactions are resolved before changing the MS DTC to which the service connects.
Trace Level	Level of detail for trace recording. The range of values are NOTRACE (for no tracing), LEVEL1, LEVEL2, LEVEL3, and LEVEL4 (for highest tracing). The information recorded includes: ■ Any errors ■ Enlistment requests and outcomes ■ Prepare, commit, and abort requests and outcomes LEVEL1 only records errors in the trace file. If no errors occur, no information is written to the trace file and Oracle Service for MTS performance is not impacted. LEVEL2 (the default setting) records both errors and minimal transaction information. This means that Oracle Service for MTS performance can be impacted, particularly in a high traffic environment. Test Oracle Service for MTS performance in your Oracle8 database environment to determine the proper trace level. Resetting the trace level to LEVEL1 is typically sufficient for most Oracle8 database environments. **Note:** The higher you set the trace level, the more Oracle Service for MTS performance is impacted. See section "Using Trace Files" on page 7-2 for additional information on trace settings.
Trace Filename	Trace file name. If a directory path is not provided, the file is created in *ORACLE_BASE**ORACLE_HOME*\ORAMTS\TRACE.

9. Click OK.

Manually Deleting or Modifying the Oracle Service for MTS with the Registry

Before the Oracle Service for MTS can be deleted or modified to connect to a different Oracle8 database, it must be cleanly disassociated from the Oracle8 database to which it has been connecting. Sometimes this disassociation fails. Follow the instructions in this section *only* if the following procedures were unsuccessful:

- Deletion procedures in section "Deleting an Existing Oracle Service for MTS" on page B-5
- Modification procedures in section "Modifying Oracle Service for MTS Connection Information" on page B-7

The Microsoft Management Console Explorer can fail to delete or modify the Oracle Service for MTS if any of the following occur:

Issue	Solution
The Explorer cannot connect to the Oracle8 database using the information in the registry.	Ensure that the Oracle8 database and its listener are started. Use SQL*Plus or a different tool to verify that the Oracle8 database accepts new connections.
The information in the Oracle8 database does not match the information in the registry.	The Explorer is connecting to a different Oracle8 database than the one to which the Oracle Service for MTS connects. If the Explorer and the Oracle Service for MTS run on the same computer, they may be using TNSNAMES.ORA files from different Oracle homes. If they run on different computers (for example, the Explorer is configuring a service on a remote computer), the entry in the TNSNAMES.ORA file that they use points to different databases. Whether it is a local or remote problem, resolve it by ensuring that the entry in the TNSNAMES.ORA file for both the Explorer and the Oracle Service for MTS points to the same database instance.
The Explorer cannot delete the service information stored in the Oracle8 database.	The Oracle8 database is unstable or is not working properly. Check if any database trace files are being created that indicate a database process crash.

To manually delete or modify Oracle Service for MTS with the registry:

1. Start the registry from the MS-DOS command prompt:

   ```
   C:\> REGEDT32
   ```

   The *Registry Editor* window appears.

2. Navigate to HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\OracleMTSService*n*.

   where *n* is the number of the Oracle Service for MTS.

   The right-hand side of the window shows various parameters and values associated with OracleMTSService*n*, including the following:

Parameter	This Parameter Contains the...
ORAMTS_SUNAME	Oracle Service for MTS user name.
ORAMTS_SUPWD	Password for the above account (encrypted in the registry). Do *not* change the password in the registry. You can only change the password from within the Database tab of the Oracle Service for MTS *Properties* dialog box in the Oracle Manager for MTS Services snap-in. See section "Modifying Oracle Service for MTS Connection Information" on page B-7 for instructions.
ORAMTS_ORADB	Database alias for connecting with the Oracle Service for MTS to the Oracle8 database.

3. Start SQL*Plus:

   ```
   C:\> SQLPLUS
   ```

4. Connect to the Oracle8 database with the same user name and Net8 net service name with which the Oracle Service for MTS connects:

   ```
   ENTER USER-NAME: USERNAME/PASSWORD@NET_SERVICE_NAME
   ```

   where *NET_SERVICE_NAME* is the net service name for connecting to the database. The password is stored in the registry in encrypted form. Use plain text passwords when connecting with SQL*Plus.

5. Verify that the Oracle8 database is the same one to which the Oracle Service for MTS connects by checking the following database information:

   ```
   SQL> SELECT NAME, DBID FROM V$DATABASE;
   ```

   which displays information similar to the following:

   ```
   NAME      DBID
   ---------------------
   ORCL      12345678
   ```

6. Check that these values match the registry values ORAMTS_DBNAME (for ORCL) and ORAMTS_DBID (for 12345678).

7. Check the service information:

   ```
   SQL> SELECT RMGUID FROM MTS_PROXY_INFO;
   ```

   which displays information similar to the following:

   ```
   RMGUID

   -------------------------

   2320b23e93e09fff02a231974
   ```

8. Check that this information matches the registry value ORAMTS_RMGUID.

9. Proceed only if all values match.

   If not all values match, the Oracle8 database is not the same one to which the Oracle Service for MTS connects. If you continue, Oracle Service for MTS installation on the database breaks. This can leave the database in an inconsistent state that requires DBA intervention to correct. The reason SQL*Plus connected to a different database than the Oracle Service for MTS is mismatching TNSNAMES.ORA files.

10. Delete the service information stored in the database.

    ```
    SQL> DELETE FROM MTS_PROXY_INFO;

    SQL> COMMIT;
    ```

11. Exit from SQL*Plus.

    ```
    SQL> EXIT
    ```

12. Delete the OracleMTSService*n* service.

13. Reboot your computer.

14. Return to your previous section:

If You Were...	Return to...
Deleting Oracle Service for MTS	Steps 8 and 9 of section "Deleting an Existing Oracle Service for MTS" on page B-6.
Modifying Oracle Service for MTS to access a different Oracle8 database	Steps 7 through 11 of section "Modifying Oracle Service for MTS Connection Information" on page B-7.

Deleting Roles and Privileges of an Inactive Oracle Service for MTS User

Ensure that you delete the roles and privileges assigned to an Oracle Service for MTS user that you no longer want to use.

To delete roles and privileges of an Oracle Service for MTS user:

1. Go to *ORACLE_BASE**ORACLE_HOME*\ORAMTS\ADMIN.
2. Open the file REVOKEUSER.SQL with an ASCII editor.
3. Replace MTS_USER with the user name from which to revoke roles and privileges.

> **Note:** This script uses the user name MTSSYS and the password MTSSYS. If you have changed the password or are using an Oracle Service for MTS user name other than MTSSYS, you need to substitute the correct user name and password.

4. Start SQL*Plus:

```
C:\> SQLPLUS
```

5. Connect with the INTERNAL user name:

```
ENTER USER-NAME: INTERNAL
```

6. Run the modified script:

```
SQL> @ORACLE_BASE\ORACLE_HOME\ORAMTS\ADMIN\REVOKEUSER.SQL;
```

7. Exit SQL*Plus:

```
SQL> EXIT
```

Index

Symbols

%ORACLE_HOME%, explained, xxi

A

ACCOUNT table
  using with Microsoft application demo, 4-3, A-2, A-5
ACCOUNT.IDI, 4-5
Administrator privileges
  for creating an Oracle Service for MTS, 3-7

B

backslash (\), defined, xx
BUILDALL.SQL script
  creating MTSSYS user name, 3-4, 4-3

C

C:\>, defined, xx
C:\ORANT, defined, xxi
C:\ORAWIN, defined, xxi
C:\ORAWIN95, defined, xxi
Choose Start >, defined, xx
client computer
  configuration requirements, 3-2
  installation requirements, 2-4
code conventions, used in this guide, xx
COM components
  deploying, 1-2
  integration in a transaction, 5-2
  marking as transactional, 1-3
  ODBC, 2-4
  Oracle Objects for OLE, 2-4
  programming with OCI and Microsoft Transaction Server, 5-8
  programming with ODBC and Microsoft Transaction Server, 5-22
  programming with Oracle Objects for OLE and Microsoft Transaction Server, 5-25
  programming with Oracle Provider for OLE DB and Microsoft Transaction Server, 5-25
  registering in a Microsoft Transaction Server environment, 5-4
  running in a Microsoft Transaction Server coordinated transactions, 5-6
  running in an MS DTC-coordinated transaction, 5-7
  using OCI, 1-3
  using ODBC, 1-3
  using Oracle Objects for OLE, 1-3
  using Oracle Provider for OLE DB, 1-3
  using with Microsoft's Oracle ODBC Driver and Microsoft Transaction Server, 5-24
  using with Oracle's ODBC Driver and Microsoft Transaction Server, 5-22
computer on which Microsoft Transaction Server is installed
  configuration requirements, 3-2
  installation requirements, 2-4
computer on which Oracle8 database is installed
  configuration requirements, 3-2
  installation requirements, 2-4
configuration requirements
  on client computer, 3-2
  on computer on which Microsoft Transaction Server is installed, 3-2
  on computer on which Oracle8 database is installed, 3-2

verifying INIT.ORA file parameter values, 3-3
connection
managing connection pooling, 6-4
connection attribute
setting with ODBC, 5-22
connection pooling
client side registry parameters, 6-4
differences between OCI, Microsoft Transaction Server, and Net8 connection pooling, 7-18
editing registry parameters, 6-9
emptying connection pools, 7-13
integration with Microsoft Transaction Server, 5-3
managing connections, 6-4
obtaining environment handles, 5-12
obtaining service handles, 5-12
releasing connections, 5-15
responsibilities, 5-3
server side registry parameters, 6-4
using OraMTSSvcGet() function, 5-12
connectivity
for Oracle Service for MTS, 7-14
controlling tracing, 7-4
custom user name
creating for Oracle Service for MTS, A-5
running the MTSUSER.SQL script, A-2, A-6, A-7

D

database alias *See* net service name
database links
with Microsoft Transaction Server, 7-17
DCOM, 1-2
demo
directory installation location, 4-3
Microsoft application demo, 1-5, 4-2, 4-7
running the Microsoft application demo, 4-5
running the OMTSSAMP.SQL script, 4-5
using the MTSDEMOUSR user name, 4-4
using Visual C++, 4-5
verifying Microsoft application demo installation, 4-5
verifying Oracle8 database includes proper tables, 4-3
directory names, convention used, xx
documentation conventions, used in this guide, xx

E

environment handles, 5-12
error messages
appearing in the Event Viewer, 7-7
Event Viewer
accessing, 7-7
blue (informational) icon, 7-8
IDs, 7-8
Oracle Service for MTS events displayed, 7-7
reading, 7-8
red (stop sign) icon, 7-8
using, 7-7
yellow (exclamation point) icon, 7-8

F

file names, convention used, xx

G

getting started
with Microsoft Transaction Server and an Oracle8 database, 1-5

H

hardware configuration
optimizing to improve performance, 6-3
HOME_NAME, defined, xxii
HOME*ID*
defined, xxii

I

in-doubt transaction
manually committing after an Oracle Service for MTS restart, 6-2
INIT.ORA file
changing parameter settings, 6-12
OS_ROLES parameter, 6-7
PROCESSES parameter, 6-12
SESSIONS parameter, 6-12
verifying parameter values, 3-3
in-process components
definition, 7-3
installation
of Microsoft Transaction Server and Oracle8 database on same computer, 2-2

of Microsoft Transaction Server and Oracle8 database on separate computers, 2-2, 5-4
using IP load balancing, 2-2
installation layouts, 2-2
installation requirements
for client computer, 2-4
for computer on which Microsoft Transaction Server is installed, 2-4
for computer on which Oracle8 database is installed, 2-4
for computer running Oracle Service for MTS, 2-3
Microsoft Internet Information Server, 2-4
Microsoft Transaction Server, 2-3, 2-4
Net8 Assistant, 2-4
Net8 Client, 2-4
Net8 Server, 2-4
ODBC, 2-4
one Oracle Service for MTS for each Oracle8 database, 2-3
Oracle Provider for OLE DB, 2-4
Oracle Services for Microsoft Transaction Server, 2-4
Oracle8 database, 2-4
required RAM, 2-3
Service Pack 4.0 or greater, 2-4
special downloadable MS DTC file, 2-4
SQL*Plus, 2-4
IP load balancing, 2-2
italic letters, defined, xx

L

local computer
creating an Oracle Service for MTS, 3-6

M

Microsoft application demo
installation location, 4-3
overview of database contents, 4-3
running, 4-5
running the OMTSSAMP.SQL script, 4-5
setting sufficient INIT.ORA parameter values, 3-3
using Microsoft Developer Studio, 4-6
using OCI, 4-2
using Oracle's ODBC driver, 4-7
using the MTSDEMOUSR user name, 4-4
using Visual C++, 4-5
verifying installation, 4-5
verifying Oracle8 database includes proper tables, 4-3
Microsoft Developer Studio
using with Microsoft application demo, 4-6
Microsoft HTML Help, 2-3, 2-4
Microsoft Internet Information Server
installation requirements, 2-4
Microsoft Management Console
installation methods, 2-4
Microsoft Transaction Server, 1-2
benefits, 1-2
client computer responsibilities, 5-3
COM components running in a transaction, 5-6
components running in an MS DTC-coordinated transaction, 5-7
connection pooling responsibilities, 5-3
database links, 7-17
definition, 1-2
getting started with an Oracle8 database, 1-5
increasing timeout parameter, 6-10
installation requirements, 2-3, 2-4
integration with Oracle8 database, 1-3
MS DTC responsibilities, 5-3
OCI responsibilities, 5-3
ODBC responsibilities, 5-3
Oracle Objects for OLE responsibilities, 5-3
Oracle Provider for OLE DB responsibilities, 5-3
Oracle Service for MTS responsibilities, 5-3
Oracle8 database responsibilities, 5-3
programming with OCI, 5-8
programming with ODBC, 5-22
programming with Oracle Objects for OLE, 5-25
programming with Oracle Provider for OLE DB, 5-25
programming with the Oracle8 database, 5-4
registering COM components, 5-4
starting MS DTC, 7-10
three-tiered architecture model, 1-2
using OCI with the Microsoft application demo, 4-2
using with Microsoft's Oracle ODBC Driver, 5-24

using with Oracle's ODBC Driver, 5-22
Microsoft Transaction Server demos
using Oracle's ODBC driver with the Microsoft application demo, 4-7
MS DTC, 5-16, 5-19
COM components running in an MS DTC-coordinated transaction, 5-7
modifying the location, B-10
responsibilities, 5-2, 5-3
special downloadable file for installation, 2-4
starting, 7-10
using with Oracle Service for MTS, 2-3
MTSDEMO, 4-3
Net8 network connection, 3-9
MTSDEMOUSR user name
using the Microsoft application demo, 4-4
MTSSamples.dsn
using with Oracle's ODBC driver, 5-23
MTSSYS user name
changing the password, 3-5, A-3
connecting with, 3-5
for use with Oracle Service for MTS, 3-4
manually creating for Oracle Service for MTS, A-2
manually creating with the ORAMTS.SQL script, A-2, A-6
password, 3-4
required roles and privileges, 3-4
verifying inclusion in Oracle8 database, 3-4
MTSUSER.SQL script
editing to create a custom user, A-7
running, A-7
MTX.EXE
running, 7-3
MTXSTOP.EXE
running, 7-13
multiple Oracle homes
defined, xxi

N

net service name
changes that impact connection pooling, 7-13
creating with Net8 Assistant, 3-6
defined, 3-9
Net8
MTSDEMO net service name, 3-9
Net8 Assistant
creating a network connection, 3-6
installation requirements, 2-4
Net8 Client
installation requirements, 2-4
Net8 Server
installation requirements, 2-4
network connection *See* net service name
network interconnects
optimizing to improve performance, 6-3
non-COM components
using, 5-11

O

OCI
connection pooling differences with OCI, Net8, and Microsoft Transaction Server, 7-18
enlisting an MS DTC-coordinated transaction, 5-19
enlisting or de-enlisting an MS DTC-coordinated transaction, 5-16
obtaining pooled OCI connections, 5-13
obtaining pooled or standard OCI connections, 5-17
OraMTSOCIErrGet() function, 5-21
OraMTSSvcEnlist() function, 5-16
OraMTSSvcEnlist() function parameters, 5-16
OraMTSSvcEnlistEx() function, 5-19
OraMTSSvcEnlistEx() function parameters, 5-19
OraMTSSvcGet() function, 5-12
OraMTSSvcGet() function parameters, 5-12
OraMTSSvcRel() function, 5-15
OraMTSSvcRel() function parameters, 5-15, 5-20
OraMTSTransTest() function, 5-20
overview of connection pooling, 5-8
programming with Microsoft Transaction Server, 5-8
releasing pooled OCI connections, 5-15
responsibilities, 5-3
sample file location, 5-10
using with the Microsoft application demo, 4-2
warnings against using OCITransCommit() and OCITransAbort(), 5-8
OCI_THREADED flag
passing, 5-9
OCIInitialize

calling, 5-9
ODBC
configuring Microsoft's Oracle ODBC Driver with Microsoft Transaction Server, 5-24
configuring Oracle's ODBC Driver with Microsoft Transaction Server, 5-23
installation requirements, 2-4
programming with Microsoft Transaction Server, 5-22
responsibilities, 5-3
setting the connection attribute, 5-22
using Microsoft's Oracle ODBC Driver with Microsoft Transaction Server, 5-24
using MTSSamples.dsn with Oracle's ODBC driver, 5-23
using Oracle's ODBC Driver with Microsoft Transaction Server, 5-22
using Oracle's ODBC driver with the Microsoft application demo, 4-7
using the SQL_ATTR_ENLIST_IN_DTC parameter, 5-22
using the SQLSetConnectAttr function, 5-22
OLE
integration with Microsoft Transaction Server, 5-3
OMTSSAMP.SQL script, 4-5, 5-24
OO4O. *See* Oracle Objects for OLE
operating system authentication
authenticating a new Oracle Service for MTS user name, A-8
authenticating an existing Oracle Service for MTS user name, A-11
available platforms, 7-18
of Oracle Service for MTS user, 3-6
with OCI function OraMTSSvcGet(), 5-14
Optimal Flexible Architecture
directory naming conventions, 4-2, A-8, A-11
Oracle Manager for MTS Services snap-in, 7-4
definition, 1-3
displaying in the Microsoft Management Console, 3-8
using Microsoft HTML Help, 2-3
Oracle Objects for OLE
programming with Microsoft Transaction Server, 5-25
responsibilities, 5-3
with Microsoft Transaction Server, 5-3
Oracle Provider for OLE DB
installation requirements, 2-4
integration with Microsoft Transaction Server, 5-3
programming with Microsoft Transaction Server, 5-25
Oracle Service for MTS
authenticating a new Oracle Service for MTS user name, A-8
authenticating an existing Oracle Service for MTS user name, A-11
automatically starting after a shutdown, 6-2
creating a custom user name, A-5
creating a new service, 3-6
creating on a local computer, 3-6
creating on a remote computer, 3-6
creating on Windows 2000 computers, 2-4
creating with Administrator privileges, 3-7
definition, 1-3
deleting, B-5
deleting roles and privileges of inactive users, B-16
ensuring connectivity and security, 7-14
installation requirements, 2-3
manually committing transactions after a restart, 6-2
manually creating the MTSSYS user name, A-2
modifying connection information to an Oracle8 database, B-7
modifying Oracle Service for MTS in the registry, B-13
modifying trace file and MS DTC information, B-10
preparing to delete or modify, B-3
reasons for failing to be deleted or modified, B-13
requirements for deleting, B-2
requirements for modifying, B-2
responsibilities, 5-3
special MTSSYS user name, 3-4
starting the service, 3-11
troubleshooting startup problems, 7-5
using with MS DTC, 2-3

verifying the MTSSYS user name, 3-4
Oracle Services for Microsoft Transaction Server
installable product, 1-3, 3-2
installation methods, 2-4
installation requirements, 2-4
ORACLE_BASE
explained, xxi, 4-2, A-8, A-11
ORACLE_HOME
explained, xxi, 4-2, A-8, A-11
Oracle8 database
authenticating an existing Oracle Service for MTS user name, A-11
changing INIT.ORA file parameter settings, 6-12
creating a custom Oracle Service for MTS user name, A-2, A-5
installation requirements, 2-4
integration with Microsoft Transaction server, 1-3
no Microsoft Transaction Server integration with releases prior to 8.1, 2-4
responsibilities, 5-3
verifying inclusion of MTSSYS user name, 3-4
with Microsoft Transaction Server, 1-2
ORAMTS.DLL
definition, 7-2
ORAMTS.SQL script
creating MTSSYS user, 3-4
running, 4-3
running to create the MTSSYS user, A-2, A-6
ORAMTS_CFLG_ALLDEFAULT flag
description, 5-12
ORAMTS_CFLG_NOIMPLICIT flag
description, 5-13
ORAMTS_CFLG_PRELIMAUTH
using, 5-14
ORAMTS_CFLG_PRELIMAUTH flag
description, 5-13
ORAMTS_CFLG_SYSDBALOGN
using, 5-14
ORAMTS_CFLG_SYSDBALOGN flag
description, 5-13
ORAMTS_CFLG_SYSOPRLOGN
using, 5-14
ORAMTS_CFLG_SYSOPRLOGN flag
description, 5-13
ORAMTS_CFLG_UNIQUESRVR flag
description, 5-13
ORAMTS_CONN_POOL_TIMEOUT registry parameter, 6-5, 6-6
ORAMTS_CP_TRACE_LEVEL registry parameter, 7-2
ORAMTS_ENFLG_DEFAULT flag
description, 5-17
ORAMTS_ENFLG_DETCHTX flag
description, 5-17
ORAMTS_ENFLG_RESUMTX flag
description, 5-17
ORAMTS_NET_CACHE_MAXFREE registry parameter, 6-5, 6-6
ORAMTS_NET_CACHE_TIMEOUT registry parameter, 6-5, 6-6
ORAMTS_ORADB registry parameter, B-14
ORAMTS_OSCREDS_MATCH_LEVEL registry parameter, 6-5, 6-7
ORAMTS_RPC_TRACE_LEVEL registry parameter, 7-2
ORAMTS_SUNAME registry parameter, B-14
ORAMTS_SUPWD registry parameter, B-14
ORAMTS_SVC_CONN_POOL_TIMEOUT registry parameter, 6-5
ORAMTS_SVC_CONN_POOL_TIMEOUT registry parameter, 6-5, 6-8
ORAMTS_SVC_NET_CACHE_MAXFREE registry parameter, 6-5, 6-8
ORAMTS_SVC_NET_CACHE_TIMEOUT registry parameter, 6-5, 6-8
ORAMTSERR_NOERROR
returning upon acquiring a connection, 5-17, 5-19
returning upon obtaining a connection, 5-13
returning upon releasing a connection, 5-15
OraMTSOCIErrGet() function, 5-21
OCI function, 5-21
OraMTSSvcEnlist() function
enlisting pooled or standard OCI connections, 5-17
OCI function, 5-16
ORAMTS_ENFLG_DEFAULT flag, 5-17
ORAMTS_ENFLG_DETCHTX flag, 5-17
ORAMTS_ENFLG_RESUMTX flag, 5-17
overview, 5-8, 5-11

parameters, 5-16
returning ORAMTSERR_NOERROR upon acquiring a connection, 5-17
syntax, 5-16
OraMTSSvcEnlistEx() function
differences with OraMTSSvcEnlist() function, 5-19
OCI function, 5-19
overview, 5-8, 5-11
parameters, 5-19
returning ORAMTSERR_NOERROR upon acquiring a connection, 5-19
syntax, 5-19
OraMTSSvcGet() function
OCI function, 5-12
ORAMTS_CFLG_ALLDEFAULT flag, 5-12
ORAMTS_CFLG_NOIMPLICIT flag, 5-13
ORAMTS_CFLG_PRELIMAUTH flag, 5-13
ORAMTS_CFLG_SYSDBALOGN flag, 5-13
ORAMTS_CFLG_SYSOPRLOGN flag, 5-13
ORAMTS_CFLG_UNIQUESRVR flag, 5-13
overview, 5-8, 5-11
parameters, 5-12
responsibilities, 5-12
returning a pooled connection, 5-13
returning ORAMTSERR_NOERROR upon obtaining a connection, 5-13
syntax, 5-12
using with operating system authentication, 5-14
OraMTSSvcRel() function
OCI function, 5-15
overview, 5-8, 5-11
parameters, 5-15, 5-20
releasing a pooled connection, 5-15
returning ORAMTSERR_NOERROR upon releasing a connection, 5-15
syntax, 5-15
OraMTSTransTest() function, 5-20
OCI function, 5-20
ORAOCI
setting in the registry, 5-25
OS_ROLES parameter, 6-7
out-of-process components
definition, 7-3

P

packages
setting the identity of, 7-16
password
changing for MTSSYS user name, 3-5, A-3
for MTSSYS user name, 3-4
performance
improving, 6-3
setting proper trace levels, B-12
pooled connection
releasing, 5-15
privileges
deleting privileges of an inactive Oracle Service for MTS user, B-16
PROCESSES parameter
changing the value, 6-12
verifying INIT.ORA file value, 3-3
programming
with OCI and Microsoft Transaction Server, 5-8
with ODBC and Microsoft Transaction Server, 5-22
with Oracle Objects for OLE and Microsoft Transaction Server, 5-25
with Oracle Provider for OLE DB and Microsoft Transaction Server, 5-25
with Oracle's ODBC Driver and Microsoft Transaction Server, 5-22
programming methods
optimizing to improve performance, 6-3

R

RECEIPT table
using with Microsoft application demo, 4-3
REGEDT32, 6-9, B-14
registry
modifying the connection pooling parameters, 6-9
modifying the Oracle Service for MTS database connection, B-13
REGEDT32, B-14
trace file settings, 7-2
registry parameter
editing connection pooling parameters, 6-9
ORAMTS_CONN_POOL_TIMEOUT, 6-5, 6-6
ORAMTS_CP_TRACE_LEVEL, 7-2
ORAMTS_NET_CACHE_MAXFREE, 6-5, 6-6

ORAMTS_NET_CACHE_TIMEOUT, 6-5, 6-6
ORAMTS_ORADB, B-14
ORAMTS_OSCREDS_MATCH_LEVEL, 6-5, 6-7
ORAMTS_RPC_TRACE_LEVEL, 7-2
ORAMTS_SUNAME, B-14
ORAMTS_SUPWD, B-14
ORAMTS_SVC_CONN_POOL_TIMEOUT, 6-8
ORAMTS_SVC_NET_CACHE_MAXFREE, 6-5, 6-8
ORAMTS_SVC_NET_CACHE_TIMEOUT, 6-5, 6-8
remote computer
creating an Oracle Service for MTS, 3-6
requirements
for modifying or deleting an Oracle Service for MTS, B-2
REVOKEUSER.SQL script
running, B-16
RM
responsibilities, 5-2, 5-3
roles
deleting roles of an inactive Oracle Service for MTS user, B-16

S

samples
Microsoft application demo, 4-2, 4-7
SCOTT user name
using the ACCOUNT and RECEIPT tables, 4-3, A-2, A-5
using with Microsoft application demo, 4-3, 4-7, A-2, A-5
SCOTT.SQL script
running, 4-3, A-2, A-5
running with ORAMTS.SQL, 3-4
security
for Oracle Service for MTS, 7-14
server package
setting the identity of, 7-16
service
creating a new Oracle Service for MTS, 3-6
deleting, B-5
modifying connection information to an Oracle8 database, B-7
preparing to delete or modify, B-3
starting Oracle Service for MTS, 3-11
service handles, 5-12
Service Pack 4.0 or greater
correcting Windows NT Explorer crashes, 7-5
installation requirements, 2-4
SESSIONS parameter
changing the value, 6-12
verifying INIT.ORA file value, 3-3
SQL*Plus
installation requirements, 2-4
square brackets, defined, xx
symbols, list of, xxii
syntax, 5-20

T

three-tiered architecture, 1-2, 1-4, 5-2
timeout parameter
increasing for Microsoft Transaction Server, 6-10
trace files
file name conventions, 7-2
level 4 warning, 7-4
modifying trace levels and file names, B-10
registry settings, 7-2
trace levels, 7-3
using, 7-2
trace levels
available range, B-12
best levels for performance, B-12
controlled by Oracle Manager for MTS Services snap-in, 7-4
file location, B-12
trace parameters
setting as environmental variables, 7-3
setting in the registry, 7-4
transaction
increasing the transaction timeout parameter, 6-10
overview of COM component integration, 5-2
troubleshooting
correcting Net8 changes that impact connection pooling, 7-13
correcting Windows NT Explorer crashes, 7-5

increasing the transaction timeout parameter, 6-10
Oracle Service for MTS startup problems, 7-5
reason that a deletion or modification of Oracle Service for MTS fails, B-13
starting MS DTC, 7-10
using the Event Viewer, 7-7
using trace files, 7-2
tuning
changing the SESSIONS and PROCESSES INIT.ORA parameters, 6-12
improving performance, 6-3
increasing the transaction timeout parameter, 6-10
managing connection pooling, 6-4
setting proper trace levels, B-12

U

user name
custom, A-2, A-6, A-7
MTSSYS, 3-4
operating system authenticated, A-11

V

variables, convention used, xx
VCACCT.DLL, 4-2
building, 4-6
Visual C++
using with the Microsoft application demo, 4-5

W

Windows 2000
creating Oracle Service for MTS on, 2-4
Windows NT Explorer
correcting crashes, 7-5