OPTIMIZE

MERCURY APPLICATION MAPPING™

Preparing the Database Environment

MERCURY™

BUSINESS TECHNOLOGY OPTIMIZATION

Mercury Application Mapping

Preparing the Database Environment

Version 6.5

Document Release Date: November 20, 2006

MERCURY™

Mercury Application Mapping, Version 6.5
Preparing the Database Environment

Mercury provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. Mercury makes no representations or warranties whatsoever as to site content or availability.

Mercury Interactive Corporation
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Toll Free: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (650) 603-5300

If you have any comments or suggestions regarding this document, please send them by e-mail to documentation@mercury.com.

MAM_DB6.1/01

Table of Contents

PART I: INTRODUCING THE DATABASE ENVIRONMENT

PART II: DEPLOYING AND MAINTAINING THE MS SQL SERVER DATABASE

PART III: DEPLOYING AND MAINTAINING THE ORACLE SERVER DATABASE

PART IV: APPENDIX

Welcome to Preparing the Database Environment

This guide describes how to deploy and maintain MS SQL Server and Oracle Server databases for use with Mercury Application Mapping.

Note: This guide is not relevant to Mercury Managed Services customers.

How This Guide Is Organized

The guide contains the following chapters:

Part I **Introducing the Database Environment**

Describes the types of databases used with Mercury Application Mapping.

Part II **Deploying and Maintaining the MS SQL Server Database**

Describes how to install and configure MS SQL Server, how to create and configure databases on MS SQL Server, and how to maintain databases.

Part III **Deploying and Maintaining the Oracle Server Database**

Describes how to install Oracle Server, how to create and configure databases on Oracle Server, how to set up the Mercury Application Mapping database environment for Oracle, how to configure the Oracle Client, and how to maintain databases.

Who Should Read This Guide

This guide is intended for the following users of Mercury Application Mapping:

- Mercury Application Mapping administrators
- Database administrators

Readers of this guide should be knowledgeable and highly skilled in database administration.

Getting More Information

For information on using and updating the Mercury Application Mapping Documentation Library, reference information on additional documentation resources, typographical conventions used in the Documentation Library, and quick reference information on deploying, administering, and using Mercury Application Mapping, refer to *Getting Started with Mercury Application Mapping*.

Part I

Introducing the Database Environment

1

Databases Used in Mercury Application Mapping

To work with Mercury Application Mapping, you must set up the following types of databases:

- **Configuration management database (CMDB).** For storage of configuration information that is gathered from the various Mercury Application Mapping and third-party applications and tools. This information is used when building Mercury Application Mapping views.
- **Configuration management history database (CMDB).** For storage of changes, over time, of the CMDB configuration items (CIs). Users can view CI changes as well as snapshots. For details, see "Environment Comparison" on page 85.

You can set up the CMDB on either a Microsoft SQL Server or an Oracle Server, depending on the type of database server used in your organization.

- To work with an MS SQL Server database, see Part II, "Deploying and Maintaining the MS SQL Server Database."
- To work with an Oracle Server database, see Part III, "Deploying and Maintaining the Oracle Server Database."

Note: If Mercury Application Mapping is installed on a Solaris platform, the CMDB must be set up on an Oracle Server.

Part II

Deploying and Maintaining the MS SQL Server Database

2

Installing and Configuring MS SQL Server

This chapter contains information on the MS SQL Server installation procedure and configuration settings.

Installing MS SQL Server

You install MS SQL Server by running **setupsql.exe** from the **<MS SQL Server root directory>\x86\setup** directory. Although the installation process is not difficult, it is important that you familiarize yourself with all of the installation details so that you select the appropriate options. Note that selecting the default options, in some cases, may negatively affect the MS SQL Server's performance.

The following sections describe the dialog boxes to which you should pay particular attention during the installation process.

This section includes the following topics:

- "Instance Name Dialog Box" on page 8
- "Setup Type Dialog Box" on page 9
- "Select Components Dialog Box" on page 10
- "Services Accounts Dialog Box" on page 10
- "Authentication Mode Dialog Box" on page 12

➤ "Collation Settings Dialog Box" on page 13

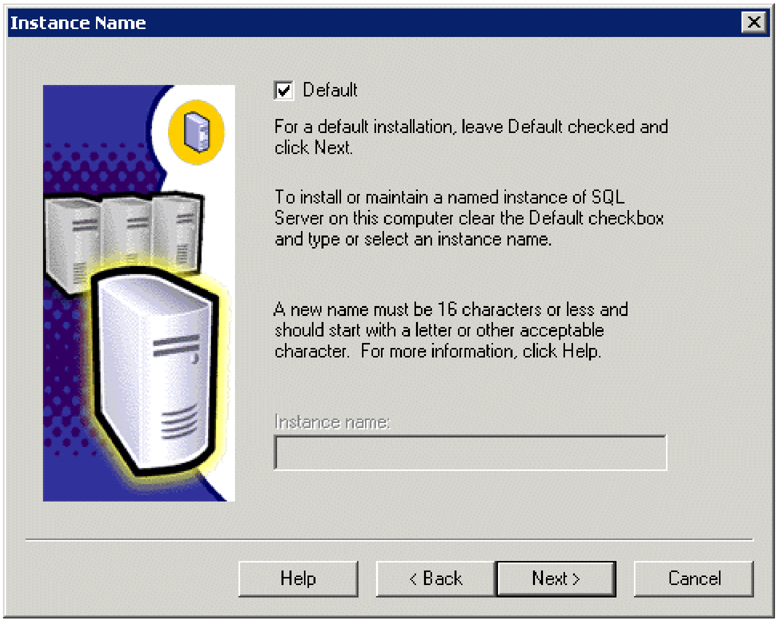
➤ "Network Libraries Dialog Box" on page 16

Instance Name Dialog Box

You can install multiple instances of MS SQL Server 2000 on the same machine. Each instance has its own MS SQL Server and MS SQL Agent services, and is completely independent of the other instances.

You can install only one instance as the default instance. All the other instances must be installed as named instances, the names of which you provide during the installation process. You access a default instance by specifying the server name or IP address, and a named instance by specifying the server name or IP address followed by \<instance_name>, for example, **server1\inst1**.

When working with Mercury Application Mapping, you must install MS SQL Server 2000 as the default instance. You do so by selecting the **Default** check box in the Instance Name dialog box.



Instance Name

☑ Default

For a default installation, leave Default checked and click Next.

To install or maintain a named instance of SQL Server on this computer clear the Default checkbox and type or select an instance name.

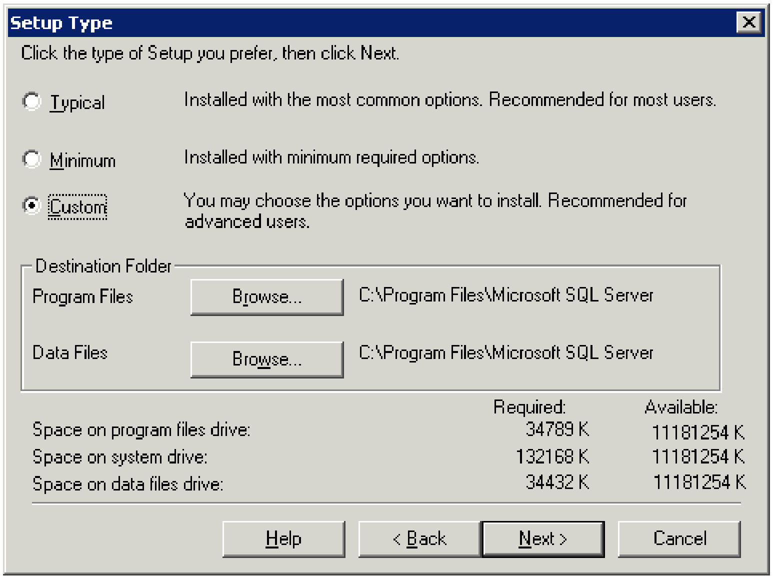
A new name must be 16 characters or less and should start with a letter or other acceptable character. For more information, click Help.

Instance name:

For information on verifying this setting after the installation process, see “Verifying and Modifying Server and Database Settings” on page 55.

Setup Type Dialog Box

In the Setup Type dialog box, you must select the **Custom** option. If you do not select this option, you will not be able to view the Component Selection, Collation Settings, and Network Libraries dialog boxes, in which you must change certain default options that are not appropriate for working with MS SQL Server in conjunction with Mercury Application Mapping.



Setup Type

Click the type of Setup you prefer, then click Next.

○ Typical — Installed with the most common options. Recommended for most users.

○ Minimum — Installed with minimum required options.

● Custom — You may choose the options you want to install. Recommended for advanced users.

Destination Folder

Program Files: C:\Program Files\Microsoft SQL Server

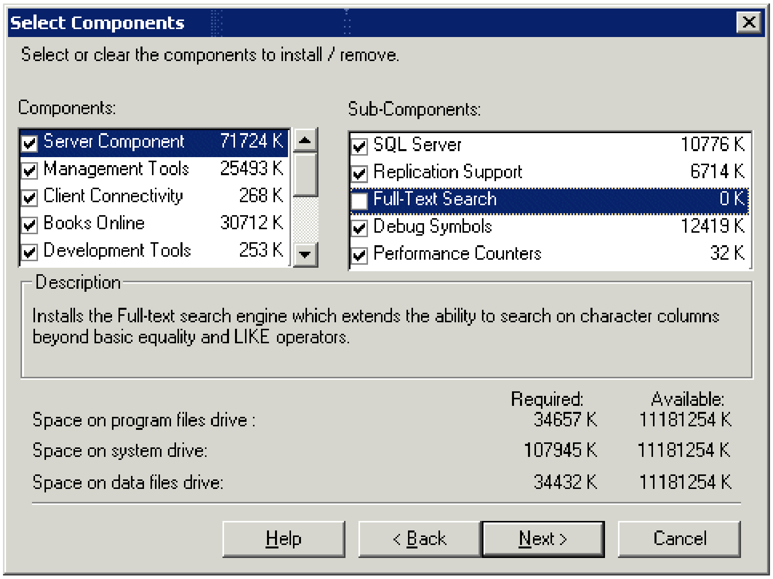
Data Files: C:\Program Files\Microsoft SQL Server

	Required:	Available:
Space on program files drive:	34789 K	11181254 K
Space on system drive:	132168 K	11181254 K
Space on data files drive:	34432 K	11181254 K

Under Destination Folder, you set the Program Files directory in which MS SQL Server executables are stored, and the Data Files directory, in which system databases, and user databases for which a file location is not specified, are stored. Note that these settings cannot be changed later on. Although you can relocate tempdb files (using the ALTER DATABASE tempdb MODIFY FILE command) and user database files (detach, move, attach) at a later stage, you cannot move master, msdb, and model files. Even though these system databases are fairly small, they are essential for the operation of MS SQL Server. Ensure that the Data Files directory is stored on a fault-tolerant disk system, for example, RAID 1.

Select Components Dialog Box

In the Select Components dialog box, ensure that the **Full-Text Search** check box is not selected. Mercury Application Mapping does not require this indexing service.



Select Components

Select or clear the components to install / remove.

Components:

☑	Server Component	71724 K
☑	Management Tools	25493 K
☑	Client Connectivity	268 K
☑	Books Online	30712 K
☑	Development Tools	253 K

Sub-Components:

☑	SQL Server	10776 K
☑	Replication Support	6714 K
□	Full-Text Search	0 K
☑	Debug Symbols	12419 K
☑	Performance Counters	32 K

Description

Installs the Full-text search engine which extends the ability to search on character columns beyond basic equality and LIKE operators.

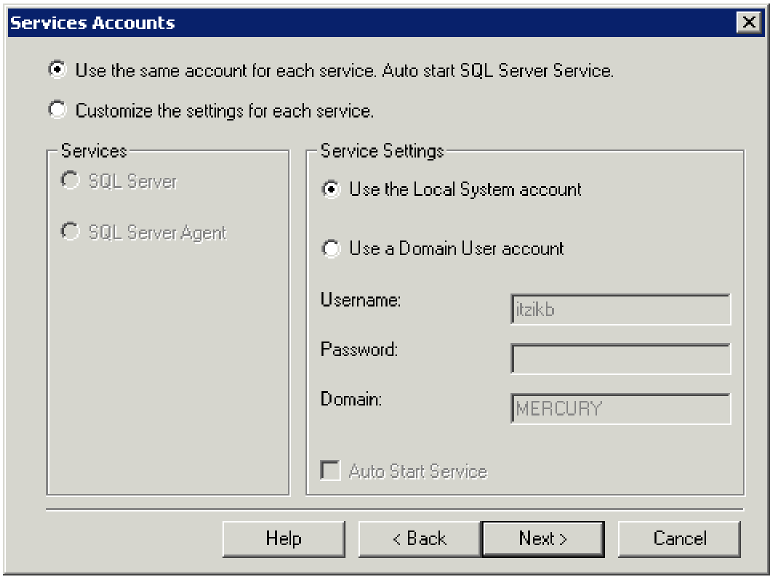
	Required:	Available:
Space on program files drive :	34657 K	11181254 K
Space on system drive:	107945 K	11181254 K
Space on data files drive:	34432 K	11181254 K

Services Accounts Dialog Box

In the Services Accounts dialog box, you select the services and enter the appropriate service settings for your MS SQL Server.

If the MS SQL Server service (MSSQLServer) must perform activities outside the local machine—for example, backing up files to a shared network directory—select **Use a Domain User account** and specify the user name, password, and domain of a user that is a member of the local machine's administrator group, and that has the appropriate permissions for network resources.

If all MS SQL Server activities are limited to the local machine, select **Use the Local System account**, which provides administrative privileges to the MS SQL Server service for the local machine only.



Services Accounts

● Use the same account for each service. Auto start SQL Server Service.

○ Customize the settings for each service.

Services

○ SQL Server

○ SQL Server Agent

Service Settings

● Use the Local System account

○ Use a Domain User account

Username: itzikb

Password:

Domain: MERCURY

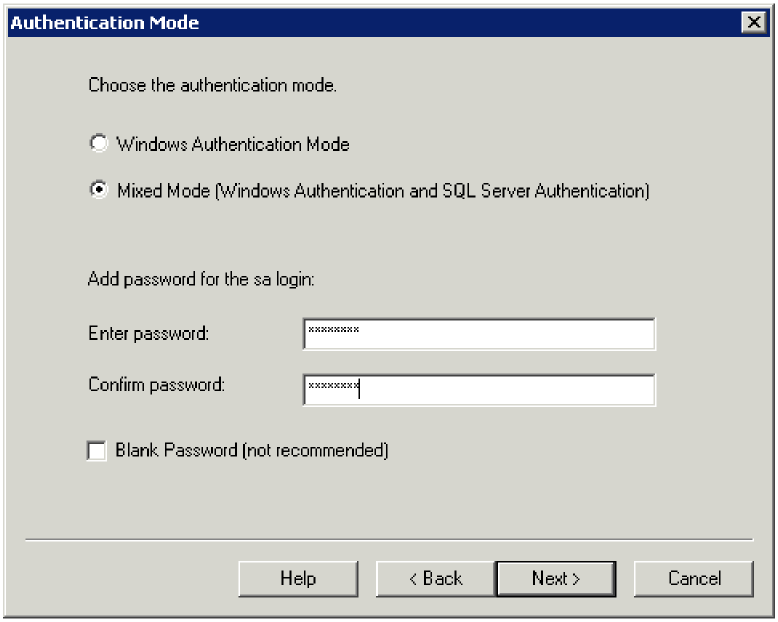
□ Auto Start Service

Similarly, if you want the MS SQL Server Agent service (SQLServerAgent) to perform activities that require permissions outside the local machine (for example, replication with other servers, ActiveX script job steps, or CmdExec job steps), select **Use a Domain User account** and specify the user name, password, and domain of a user that is a member of the local machine's administrator group.

If all MS SQL Server Agent activities are limited to the local machine, select **Use the Local System account**.

Authentication Mode Dialog Box

In the Authentication Mode dialog box, you select the type of authentication you want MS SQL Server to use. Mercury Application Mapping works with MS SQL Server authentication, which is disabled by default. To enable MS SQL Server authentication, select **Mixed Mode** and provide a complex password for the **sa** login.



Authentication Mode

Choose the authentication mode.

○ Windows Authentication Mode

● Mixed Mode (Windows Authentication and SQL Server Authentication)

Add password for the sa login:

Enter password: xxxxxxxx

Confirm password: xxxxxxxx

□ Blank Password (not recommended)

Note: To enhance the security of your MS SQL Server, it is important to enter a password.

The sa user has system administrator privileges and can therefore perform all actions within the MS SQL Server. Similarly, the sa user can perform all operating system/network actions in the context of the MSSQLServer service account, by using the **xp_cmdshell** extended procedure.

For information on modifying or verifying the authentication mode after installation, see "Verifying and Modifying Server and Database Settings" on page 55.

Collation Settings Dialog Box

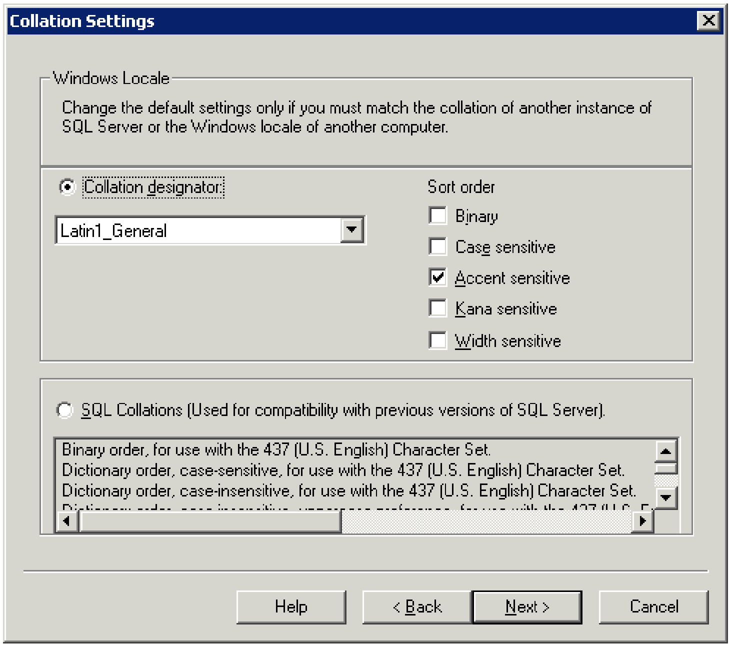
In the Collation Settings dialog box, you set the language, dictionary/binary order, and sensitivity of character data types.

Select one of the following options:

- **Windows Locale.** Select this option only if you do not need to create backward compatibility with previous versions of MS SQL Server (for example, for replication purposes). If you select this option, set the following for Mercury Application Mapping certification:
  - **Collation Designator.** The language for regular character data types (char, varchar, text). Select the default option, **Latin1_General**, to support English.
  - **Sort Order.** Binary order and sensitivity (case, accent, kana, width) of character data types. Select the **Accent sensitive** option only, and ensure that the other options are cleared.

Note: Mercury Application Mapping does not support case sensitivity with MS SQL Server 2000.

- **SQL Collations.** Select this option if the current version of MS SQL Server must be compatible with previous versions, for example, if you are replicating data between servers.



Collation Settings

Windows Locale

Change the default settings only if you must match the collation of another instance of SQL Server or the Windows locale of another computer.

● Collation designator: Latin1_General

Sort order

□ Binary

□ Case sensitive

☑ Accent sensitive

□ Kana sensitive

□ Width sensitive

○ SQL Collations (Used for compatibility with previous versions of SQL Server).

Binary order, for use with the 437 (U.S. English) Character Set.
Dictionary order, case-sensitive, for use with the 437 (U.S. English) Character Set.
Dictionary order, case-insensitive, for use with the 437 (U.S. English) Character Set.

The above settings affect only the system databases and serve as the default settings for user databases. Databases can have different collation settings from the server's default settings, and a table column can have different collation settings from the database's default settings. Because of the flexibility in collation management in MS SQL Server 2000, you can restore or attach a database that has different collation settings.

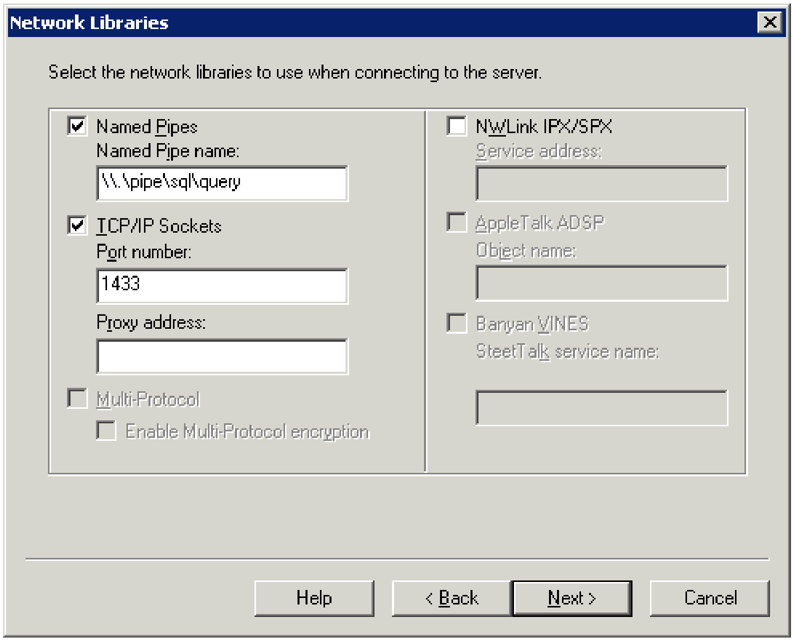
Note: Changing any of the above settings requires scripting all system objects and routines (logins, user defined system messages, master stored procedures, and so forth), reinstalling MS SQL Server (or running the RebuildM.exe utility) with the new settings, recreating all system objects from the saved scripts, and attaching the user databases. It is therefore recommended that you select the appropriate options during the installation process.

For information on verifying the MS SQL Server 2000 collation settings, see “Verifying and Modifying Server and Database Settings” on page 55.

Network Libraries Dialog Box

In the Network Libraries dialog box, you set the session level protocol by which MS SQL Server listens to client connections.

By default, MS SQL Server listens to client connections via both Named Pipes and TCP/IP, which are both supported by Mercury Application Mapping. For Mercury Application Mapping certification, however, you should select the **TCP/IP Sockets** option only.



Select the network libraries to use when connecting to the server.

☑ Named Pipes
Named Pipe name: \\.\pipe\sql\query

☑ TCP/IP Sockets
Port number: 1433
Proxy address:

□ Multi-Protocol
□ Enable Multi-Protocol encryption

□ NWLink IPX/SPX
Service address:

□ AppleTalk ADSP
Object name:

□ Banyan VINES
SteetTalk service name:

You can configure the client machine to connect to MS SQL Server using both TCP/IP and Named Pipes. However, for Mercury Application Mapping certification, you should configure the client machine to connect to MS SQL Server using TCP/IP only.

For information on modifying or verifying the above settings after installation, see "Verifying and Modifying Server and Database Settings" on page 55.

Note: Install the Service Pack according to the release system requirements. For details, see "System Requirements for Working with MS SQL" on page 22.

Configuring MS SQL Server

This section describes the service and server options you can configure once you have installed MS SQL Server.

This section includes the following topics:

- "Service Configuration Options" on page 17
- "Server Configuration Options" on page 17

Service Configuration Options

If you installed Full-Text Search, ensure that it is disabled (locate the service in the Services applet using Microsoft Search) so that no resources are unnecessarily wasted.

Unless you are using distributed transactions, ensure that the Distributed Transactions Coordinator service is also disabled or set to manual mode.

Similarly, ensure that all unnecessary services are not set to automatic startup mode.

Server Configuration Options

Most server configuration options are dynamically configured by MS SQL Server (usually set to 0). For Mercury Application Mapping certification, you should not change the default options unless you are instructed to do so by Mercury Customer Support.

There are specific situations in which you may want to change the default settings. You can change these settings in the sp_configure stored procedure, or in the various dialog boxes in MS SQL Server Enterprise Manager (mainly the Server Properties dialog box).

The following table describes the configuration options available in MS SQL 2000, their default settings, and the settings required for Mercury Application Mapping certification:

Configuration Option	**Default**	**Certification in MS SQL Server 2000**
affinity mask	0	Default
allow updates	0	Default
awe enabled (Enterprise Edition)	0	Default, unless the server needs to access 4 to 64 GB of memory
c2 audit mode	0	Default
cost threshold for parallelism	5	Default
cursor threshold	-1	Default
default full-text language	1033	Default
default language	0	Default
fill factor	0	Default
index create memory	0	Default
lightweight pooling	0	Default
locks	0	Default
max async IO	32	N/A
max degree of parallelism	0	Default
max server memory	2,147,483,647	Default
max text repl size	65,536	Default
max worker threads	255	Default
media retention	0	Default

Configuration Option	**Default**	**Certification in MS SQL Server 2000**
min memory per query	1024	Default
min server memory	0	Default
Using Nested Triggers	1	Default
network packet size	4096	Default
open objects	0	Default
priority boost	0	Default
query governor cost limit	0	Default
query wait	-1	Default
recovery interval	0	Default
remote access	1	Default
remote login timeout	20	Default
remote proc trans	0	Default
remote query timeout	600	Default
scan for startup procs	0	Default
set working set size	0	Default
show advanced options	0	Default
spin counter	0, 10,000	N/A
time slice	100	N/A
two digit year cutoff	2049	Default
user connections	0	Default
user options	0	Default

You can view all of the above configuration options by running

```
EXEC sp_configure 'show advanced options', 1
reconfigure with override
```

To view the current values for each of the options, run EXEC sp_configure.

For large installations, you may need to set the awe enabled option. For more information, see Chapter 5, "MS SQL Server Sizing Guidelines."

Note: It is strongly recommended that no significant processes other than a single MS SQL Server installation be installed on the server that hosts the Mercury Application Mapping databases. When MS SQL Server is the sole significant process on the machine, you should not change the default memory settings. You should allow the MS SQL Server to manage memory dynamically (except when you configure awe enabled support).

To reconfigure an option, run **EXEC sp_configure** '<**option**>'**,** <**value**>. Note that some options take effect after running reconfigure with override, while others require restarting the MSSQLServer service.

3

Overview of MS SQL Server Deployment

You can set up CMDB and History CMDB databases on an MS SQL Server. This chapter describes deploying MS SQL Servers for use with Mercury Application Mapping.

About MS SQL Server Deployment

To deploy MS SQL Server for use with Mercury Application Mapping, see “Installing and Configuring MS SQL Server” on page 7.

Information is provided in this section for both recommended and supported MS SQL Server environments. Mercury Application Mapping recommendation indicates that Mercury quality assurance personnel have rigorously tested the recommended environment/option. A supported environment or option means that Mercury quality assurance personnel have successfully performed basic tests on the environment/option.

System Requirements for Working with MS SQL

The following table describes the system requirements for working with MS SQL Server in conjunction with Mercury Application Mapping:

Component	Supported		Recommended	
	Version/Edition	**Service Pack**	**Version/Edition**	**Service Pack**
Operating System	Windows 2000 Server / Advanced Server	Service Pack 4	Windows 2003 Server – standard / enterprise	Service Pack 1
MS SQL Server	MS SQL Server 2000 Enterprise	Service Pack 4	MS SQL Server 2000 Enterprise	Service Pack 4
Microsoft Data Access Components (MDAC) on Mercury Application Mapping servers	2.5; 2.52; 2.61; 2.62; 2.7 SP1 Refresh		2.7 SP1 Refresh.	

Note: To check which version of MDAC is installed on your machine, download and run the component checker tool from: **http://www.microsoft.com/downloads/details.aspx?FamilyID=8f0a8df6-4a21-4b43-bf53-14332ef092c9&displaylang=en**

4

Maintaining MS SQL Server Databases

This chapter describes the various maintenance tasks that are recommended for Mercury Application Mapping databases created on MS SQL Servers, such as backing up databases, checking database integrity and handling fragmentation, and monitoring databases.

Backing Up Databases

MS SQL Server supports three main types of database backup: full, differential, and log. It also supports file/file group backup. To develop a backup policy that provides the required recovery needs, it is important to thoroughly understand each backup type and the recovery model database configuration option explained in the previous section.

You can automate backup operations using MS SQL Agent jobs. The MS SQL Agent (represented by the SQLServerAgent service) is installed automatically when you install MS SQL Server. Ensure that the MS SQL Agent is configured to autostart in the operating system’s Services applet when the server is started.

The following points are applicable to all backup types:

- The backup includes all changes made until the backup is complete.
- The backup can be performed online, but it is recommended to back up the database during periods of low activity, since the backup procedure can negatively impact on your system's performance.
- The following operations should not be performed during a backup procedure:
  - adding or removing files
  - shrinking the database
- The backup target can be a disk device (local or on a shared network that the MS SQL Server service account needs permission to access) or tape (only local).

This section includes the following topics:

Full Backup

When you perform a full database backup, all information about the backed up database is contained within the backup, including data, meta data, and file information. A full backup is the basis for differential and log backups. With small databases, it is recommended to perform a full backup every day (for example, system databases that store mainly meta data). For large databases (such as the CMDB and History CMDB databases), it is generally recommended to have longer intervals between full backups (for example, once a week).

The storage requirements for a full backup are about the same as the storage requirements for the occupied data portion of the files. For example, if the total size of the data files is 20 GB, but only 15 GB are used (there are 5 GB of free space), the full backup size of the database should be approximately 15 GB.

Differential Backup

You use a differential backup to back up the extents (blocks of 8 contiguous 8K pages) that were changed since the last full backup. When restoring a database, you need only restore the last differential backup performed after the full backup. Note that the differential backup performance of MS SQL Server 2000 is superior to that of MS SQL Server 7.0. The difference in performance is due to the fact that MS SQL Server 7.0 scans the entire database, reading each extent to check whether it was changed after the last full backup, whereas MS SQL Server 2000 has bitmaps in each file mapping the changed extents, and the differential backup directs the disk arms to read only the relevant extents.

After performing operations that affect large portions of data, such as index rebuilds or defragmentations, it is recommended that you perform a full backup. Otherwise, differential backups can become very large. For more information on index rebuilds and defragmentation, see “Database Integrity and Fragmentation” on page 27.

Differential backup is usually scheduled at intervals between full backups. For example, if you perform a full backup once a week, you may want to perform a differential backup every day, or even several times a day.

The storage requirements for a differential backup are the total size of the extents (64 KB blocks) that were changed since the last full backup.

Log Backup

A log backup—unlike full and differential backups which are mainly based on backing up an image of extents—backs up transactions from the transaction log and replays them upon restoration. In order to perform a log backup, the database must be set to the full or bulk-logged recovery model. If you want to perform a point-in-time or log mark restoration, or back up changes recorded in the log when the data crashes, you must set the database to the full recovery model. Otherwise, all changes made since the last performed backup are lost.

A log backup is incremental in nature, and backs up only the transactions performed since the previous log backup. When restoring a database, you must restore all log backups after the last differential (or full) backup you restored.

A log backup also marks the portion of the log that was backed up as available for reuse. In a database that is set to the full or bulk-logged recovery model, log portions that were not backed up cannot be reused. When the log is full, and MS SQL Server cannot cycle to its beginning to reuse log space, it must expand. The frequency of your log backups, therefore, is a factor in determining the required size of the transaction log. Frequent log backups allow you to keep a smaller transaction log. It is recommended that you back up your log as frequently as possible, for example, every 30 minutes.

File/File Group Backup

Instead of backing up the entire database, you can back up a file or file group. However, when you restore a single file or file group, you must apply all log backups up to and including the point of failure, in order to synchronize (same point-in-time) the file/file group with the rest of the database. This type of backup is generally useful with very large databases, for which you cannot frequently perform a full backup.

Maintenance Plan

In the MS SQL Server Enterprise Manager, under the Management tree view, there is a graphic tool called Database Maintenance Plans. This tool allows you to define and automate common maintenance tasks (full and log backups, integrity checks, index rebuilds, and statistics collection).

Transaction Log Issues

In terms of maintenance, the log is sensitive. When it is full, the log first tries to cycle and reuse inactive backed up log space, but if such space does not exist, the log tries to expand the file. If there is no room for the file to expand, MS SQL Server rejects data modification requests. To avoid log explosion, ensure that the log is large enough and that it is frequently backed up (ideally, by schedule). In addition, note that the active portion of the log starts with the oldest open transaction and continues until the current pointer in the log. The active portion cannot be reused or truncated. If a transaction remains open for a long time, it will inevitably lead to log explosion at some point, even though the log is backed up. To identify whether such a problem exists, run DBCC OPENTRAN to obtain the transaction that has been open for the longest period of time. To terminate the process running the transaction and roll back the transaction's activity, use the following command: `KILL <process id>`

Note: In MS SQL Server 2000, the DBCCSHRINKFILE command should always be successful.

Database Integrity and Fragmentation

It is important to periodically check the physical integrity of your database objects, and to handle index fragmentation issues that are the main cause of performance degradation.

This section includes the following topics:

- "Database Integrity" on page 28
- "Understanding File System Fragmentation" on page 28
- "Handling CMDB Index Fragmentation" on page 29
- "Distribution Statistics" on page 32

Database Integrity

It is recommended that you run DBCC CHECKDB periodically to check the allocation and structural integrity of the objects in the database. You can automate and schedule the DBCC CHECKDB command using MS SQL Agent jobs. Use the following command syntax:

DBCC CHECKDB ('database name')

Note: You can use the WITH NO_INFOMSGS option to reduce processing and tempdb usage. You can also run a quick physical-only test (page structure and record headers) using the PHYSICAL_ONLY option.

Because the MS SQL Server 2000 database holds only schema locks (which prevent schema changes) and not data changes, the DBCC CHECKDB command can be run online. It is recommended, however, to run the DBCC CHECKDB command during periods of low activity, since it can negatively impact on your system's performance (DBCC CHECKDB is CPU- and disk-intensive, and uses tempdb for sorting).

Understanding File System Fragmentation

File system fragmentation is relevant to all disk files, not just database files. It refers to the scattering of parts of the same disk file over different areas of the disk, as new parts of the file are added and existing parts are deleted. File system fragmentation slows disk access and degrades the overall performance of disk operations, although usually not severely.

To defragment a file system, you rewrite parts of a file to contiguous sectors on a hard disk. This increases the speed of data access and retrieval. In order to avoid fragmentation of your database files, create the files with as large an initial size as possible (so that they can accommodate changes in the future), and manually expand them with large increments as they become full.

If you cannot anticipate the future size of a database file, use a large value as the file growth increment in order to avoid small fragmented parts. Do not use too large a value, however, or you will experience client request timeouts when the file autogrows (for more details, see “Creating Databases” on page 37). In addition, avoid using the autoshrink database option, because it increases the chances of fragmentation as the database files continually shrink and grow.

Note: It is recommended that you periodically run a defragmentation utility.

Handling CMDB Index Fragmentation

Since Microsoft SQL Server 2000 maintains indexes to reflect updates to their underlying tables, these indexes can become fragmented. Depending on workload characteristics, this fragmentation can adversely affect workload performance.

Fragmentation exists when indexes have pages in which the logical ordering, based on the key value, does not match the physical ordering inside the data file. All leaf pages of an index contain pointers to the next and previous pages in the index. This forms a doubly linked list of all index/data pages. Ideally, the physical order of the pages in the data file should match the logical order. Overall disk throughput is increased significantly when the physical and logical ordering of the data match. This leads to much better performance for certain types of queries. When the physical ordering does not match the logical ordering, disk throughput can become less efficient because the disk head must move back and forth to gather the index pages, instead of scanning forward in one direction only. Fragmentation affects I/O performance, but has no effect on performance of queries whose data pages reside in the SQL Server data cache.

When indexes are first built, little or no fragmentation should exist. Over time, as data is inserted, updated, and deleted, fragmentation levels on the underlying indexes may begin to rise.

The main criteria used to determine if indexes are fragmented are:

- **Logical scan fragmentation.** Shows the ratio of pages that are out of order. This percentage should be between 0% and 10% with anything higher indicating external fragmentation.
- **Scan Density.** Shows the ratio between the Best Count (the number of pages if they were all full) and the Actual Count of extents. The Actual Count is higher as pages are not full due to fragmentation. This percentage should be as close as possible to 100%.

Mercury Application Mapping provides two utilities that can be used to detect and rebuild fragmented indexes in the CMDB schema. The utilities use the Logical scan fragmentation and Scan Density criteria to detect, and if instructed to rebuild, fragmented indexes. The operation of listing the fragmented tables has a very small impact on system performance and can be executed on line. The operation of rebuilding the indexes may hinder performance as tables are partially locked during the process, and CPU and I/O are heavily utilized. It is recommended to rebuild the indexes in a maintenance window. The utilities should be run by a systems or database administrator.

The utilities are located in the **<Mercury Application Mapping server>\J2F\CMDB\dbscripts\ms\utils** directory on the Mercury Application Mapping server. For details, see “Mercury Application Mapping Database Script” on page 149.

This section includes the following topics:

- “Utility to Rebuild All Indexes in Database” on page 31
- “Utility to Rebuild Indexes Based on the Fragmentation Level of Each Index” on page 31

Utility to Rebuild All Indexes in Database

The **rebuild_indexes.bat** utility runs through all tables in the database and rebuilds them.

To run the rebuild_indexes.bat utility:

Execute **rebuild_indexes.bat** with the following parameters:

- SQL Server name
- Database name
- SA password

Output from the procedure is located in the **rebuild_indexes.log** file in the same directory.

Utility to Rebuild Indexes Based on the Fragmentation Level of Each Index

The **rebuild_fragmented_indexes.bat** utility has two working modes:

- **List fragmented tables.** In this mode, a list of the fragmented tables (that is, tables with over 30% fragmentation) is returned, together with the commands needed to rebuild the tables at a later time.
- **Rebuild fragmented tables.** In this mode, all fragmented tables (that is, tables with over 30% fragmentation) are rebuilt.

To run the rebuild_fragmented_indexes.bat utility:

Execute **rebuild_fragmented_indexes.bat** with the following parameters:

- SQL Server name
- Database name
- SA password
- Working mode – 0 to provide a rebuild script for later use; 1 to rebuild indexes automatically.

Output from the procedure (a list of fragmented tables and the rebuild commands) is located in the **rebuild_indexes.log** file in the same directory.

Distribution Statistics

MS SQL Server 2000 allows statistical information regarding the distribution of values in a column to be created. This statistical information can be used by the query processor to determine the optimal strategy for evaluating a query. When an index is being created, SQL Server automatically stores statistical information regarding the distribution of values in the indexed column(s). The query optimizer in SQL Server uses these statistics to estimate the cost of using the index for a query. As the data in a column changes, index and column statistics can become out-of-date and cause the query optimizer to make less-than-optimal decisions on how to process a query.

It is recommended to update index statistics to provide the query optimizer with up-to-date information about the distribution of data values in the tables. This allows the query optimizer to make better judgments about the best way to access data, as it has more information about the data stored in the database.

By default, the auto update statistics database option is enabled, but if this option has been disabled, it is strongly recommended that you create an automatic task to update statistics for the CMDB and History CMDB databases, on a daily basis, as the data is frequently changed. The job should execute the sp_updatestats API against the specific database.

In addition to running a daily task to update statistics for the CMDB databases if the auto update statistics database option is not enabled, it is recommended that you manually refresh statistics for CMDB if major changes to the CMDB schema objects have occurred, usually caused by bulk insert transactions.

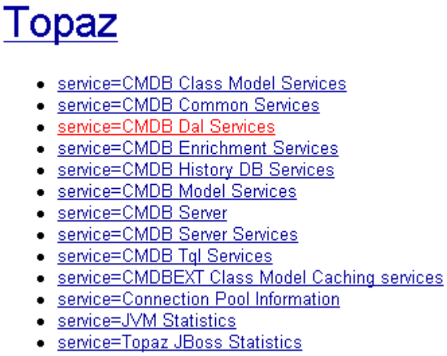
The following scenario warrants a manual refresh of CMDB statistics:

- **Automated Discovery tasks.** Discovery Manager is the process responsible for automatically detecting configuration items (CIs) and inserting them into CMDB.

Running a Manual Statistics Refresh

When major data changes have been carried out on CMDB, you refresh the statistics by one of the following methods:

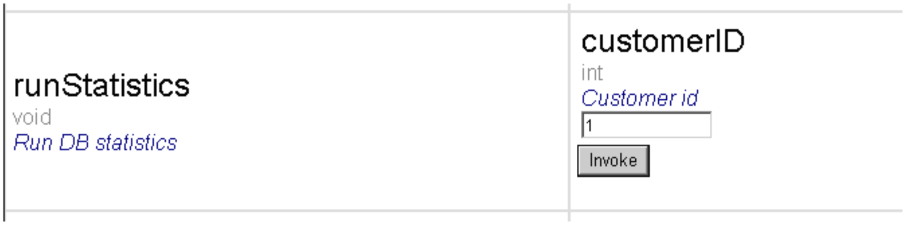
1. In a Web browser, open: **http://<Mercury Application Mapping server machine name>:8080/jmx-console**.
2. In the **Topaz** section, select **CMDB Dal Services**.



Topaz

- service=CMDB Class Model Services
- service=CMDB Common Services
- service=CMDB Dal Services
- service=CMDB Enrichment Services
- service=CMDB History DB Services
- service=CMDB Model Services
- service=CMDB Server
- service=CMDB Server Services
- service=CMDB Tql Services
- service=CMDBEXT Class Model Caching services
- service=Connection Pool Information
- service=JVM Statistics
- service=Topaz JBoss Statistics

3. Under **runStatistics**, enter the customer ID in the **customerID** box. The default customer ID for an individual Mercury Application Mapping system (that is, one not managed by Mercury Managed Services) is 1:



runStatistics
void
Run DB statistics

customerID
int
Customer id
1

4. Click **Invoke**. The CMDB statistics are regenerated.

Database Maintenance – References

For extensive information on MS SQL Server performance tuning, refer to the following documents:

- An approach to database optimization:
  http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/sql/reskit/sql2000/part10/c3361.asp
- Troubleshooting application performance with MS SQL Server:
  http://support.microsoft.com/default.aspx?scid=KB;EN-US;q224587
- Performance counters:
  http://www.databasejournal.com/features/mssql/article.php/1477311#disk
- How to perform an MS SQL Server performance audit:
  http://www.sql-server-performance.com/sql_server_performance_audit.asp
- MS SQL Server performance tuning tips:
  http://www.sql-server-performance.com/best_sql_server_performance_tips.asp
- Operations guide:
  http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/sql/maintain/operate/opsguide/default.asp
- General MS SQL Server performance tuning page with other links:
  http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/rdbmspft.mspx
- MS SQL Server MSDN performance tuning section:
  http://msdn.microsoft.com/SQL/sqlperf/ default.aspx

5

MS SQL Server Sizing Guidelines

This chapter contains guidelines for the server and database configuration settings that should be used when working with MS SQL Server and Mercury Application Mapping. Note that the recommended settings generally differ according to the size of your Mercury Application Mapping deployment.

Data File Property Settings

You can configure the following MS SQL Server data file properties:

- **SIZE.** The MS SQL Server allocates the size specified in this parameter when the database is created, and "zeroes" the space allocated (this may take a while). Since MS SQL Server tries to acquire as large a consecutive amount of space as possible, to avoid file system fragmentation caused by future file growth, it is recommended that you specify as large a value as possible.
- **FILEGROWTH.** The automatic growth increment of the file can be specified as either a percentage of the existing file size, or as a fixed size. Using a small growth increment is not recommended because it increases file system fragmentation. On the other hand, if you have a very large increment, modifications sent by clients might incur connection timeouts while waiting for the automatic expansion to finish. This problem can occur a while after the database was created and come as a surprise when the files are set with the default ten percent growth increment. Note that when the database is small, the automatic growth is completed in a short time, but when the database becomes large (for example, several gigabytes), the default ten percent growth increment may take a while to allocate and initialize.

The tempdb Database Settings

The frequent expansion of the tempdb system database can affect the database's performance, especially in large MS SQL Server installations. The size of the tempdb, therefore, should be large enough to avoid the need for early expansion. Its growth increment should be large enough to avoid fragmentation, yet not too large to expand in a reasonable amount of time. Create the tempdb with a minimum, initial size of 500 MB and with a growth increment of 50 MB. The tempdb database should be striped across several disks, ideally on a RAID 0+1 controller. It is recommended to move the tempdb database to its own set of disks.

6

Creating and Configuring MS SQL Server Databases

This chapter describes the creation and configuration of Mercury Application Mapping databases on an MS SQL Server.

Creating Databases

CMDB and History CMDB databases are created before installing the Mercury Application Mapping server. For details about installing Mercury Application Mapping server, see "Introduction to Installation" in *Mercury Application Mapping Installation Guide*.

This section includes the following topics:

- "Database Permissions" on page 37
- "System Databases" on page 39

Database Permissions

To create a database, you must have CREATE DATABASE permissions. To connect to an existing database, the login account to which you are connecting must be mapped to dbo in the database.

Note: Members of the sysadmin server role automatically have CREATE DATABASE permissions, and are also mapped to dbo in all databases. A database owner is automatically mapped to dbo in the database.

To grant CREATE DATABASE permissions to a user, the user's login must first be mapped to a database user in the master database.

Use one of the following three methods to map a user's login to dbo in the required database:

- Make the user a member of the sysadmin server role (remember that this gives the login the strongest possible permissions for all server activities).
- Make the user a database owner, using EXEC sp_changedbowner 'login'.
- Alias the user login as the dbo in the database, using EXEC sp_addalias 'login', 'dbo'.

To check whether a user is a database owner, run the following:

```
EXEC sp_helpdb <database name>
```

To check whether a user has CREATE DATABASE permissions, log in to the Query Analyzer with the login account of the user whose permissions you want to check, and run the following:

```
USE master
IF PERMISSIONS() & 1 = 1
  PRINT 'User has CREATE DATABASE permissions'
ELSE
  PRINT 'User does not have CREATE DATABASE permissions'
```

To check whether a user is mapped to dbo in the database, log in to the Query Analyzer with the login account of the user whose mapping you want to check. Change the database context to the required database, and run the following:

```
SELECT USER_NAME()
```

System Databases

The following system databases are especially important for the smooth performance of MS SQL Server:

- **tempdb.** Numerous MS SQL Server activities—such as creating local and global temporary tables, creating work tables behind the scenes to spool intermediate query execution results, and sorting—implicitly or explicitly use the tempdb system database. If your system is not configured properly, the tempdb database can become a performance bottleneck, so it is very important to determine the tempdb database's original size correctly. For more information on setting database sizes, see Chapter 5, "MS SQL Server Sizing Guidelines." To move tempdb's files, use the ALTER DATABASE tempdb MODIFY FILE command, and restart MS SQL Server.
- **master, msdb, model.** These databases, although crucial for the operation of MS SQL Server, are smaller than tempdb because they store only meta data. It is strongly recommended to use a fault tolerant disk—ideally, RAID 1—for these databases.

Note: For Mercury Application Mapping certification, place system databases on fault tolerant disks.

To check the database's properties, run the following:

EXEC sp_helpdb <database name>

Configuring Databases

Once you have created the necessary databases, you can add new files to the databases, change some of the existing database file properties, and set the database configuration options appropriately.

This section includes the following topics:

- Database File Configuration
- Database Configuration Options

Database File Configuration

You can change certain database file properties, as well as add or drop files, using the Properties dialog box in the Enterprise Manager or the ALTER DATABASE command. For details, refer to MS SQL Server 2000 Books Online.

This section includes the following topics:

- “Adding Files” on page 40
- “Dropping Files” on page 40
- “Changing File Properties” on page 40

Adding Files

Data files can be added to an existing file group in a database, or to a new file group. There are no special restrictions or requirements.

Dropping Files

To drop a file, you must first empty it using the DBCC SHRINKFILE command’s EMPTYFILE option, which transmits the file data to all the other files in the file group. Once you empty the file, you can use the ALTER DATABASE <database name> DROP FILE command to drop it.

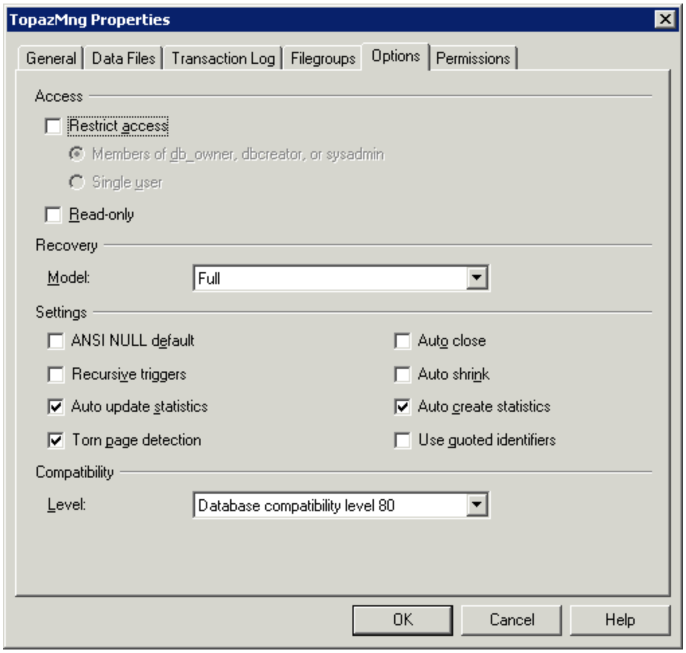
Changing File Properties

You can change the size-related properties for all databases, as well as the filename property for the tempdb database (this takes effect after you restart MS SQL Server). The SIZE, MAXSIZE, and FILEGROWTH properties can be changed using the ALTER DATABASE tempdb MODIFY FILE command. Note that the SIZE property can only be enlarged. To shrink the file, use the DBCC SHRINKFILE command. For details and recommendations concerning file properties, see “Creating Databases” on page 37.

Database Configuration Options

Each database contains a set of configurable options that determine its behavior. You can view or change the database options using:

- The Options tab in the Enterprise Manager's Properties dialog box



TopazMng Properties

Access

□ Restrict access

● Members of db_owner, dbcreator, or sysadmin

○ Single user

□ Read-only

Recovery

Model: Full

Settings

□ ANSI NULL default

□ Recursive triggers

☑ Auto update statistics

☑ Torn page detection

□ Auto close

□ Auto shrink

☑ Auto create statistics

□ Use quoted identifiers

Compatibility

Level: Database compatibility level 80

Note: Not all of the database configuration options are available in this dialog box.

- The sp_dboptions stored procedure
- The ALTER DATABASE <database name> SET command

The following table lists the default configuration options, as well as the configuration settings required for Mercury Application Mapping certification:

Configuration Option	**Description**	**Default**	**Certification in MS SQL Server 2000**
Restrict access	Only single users or members of the db_owner, dbcreator, or sysadmin groups can access the database.	Not set (MULTI_USER)	MULTI_USER
Read only	Database is read only	Not set (READ_WRITE)	READ_WRITE
Recovery	The database recovery model determines the recovery capabilities by controlling the amount of bulk operation logging (such as Select into, Bulk, Insert, Create index, LOB manipu-lation). The higher the recovery model, the higher the recovery capabilities. However, the amount of logging also increases, which may affect performance.	Full	Full (unless you are certain that the lower recovery capabilities are sufficient for your system)
Truncate log on checkpoint	Automatically marks inactive portions of log for reuse on checkpoint	Not set	N/A
Select into/bulk copy	Allows the use of minimally logged Select into/bulk copy operations	Not set	N/A
ANSI NULL default (see note below)	Specifies whether the database columns are defined as NULL or NOT NULL, by default	Not set	Not set
Recursive triggers	Specifies whether recursive triggers are supported	Not set	Not set

Configuration Option	**Description**	**Default**	**Certification in MS SQL Server 2000**
Auto update statistics	Specifies whether out-of-date statistics required by a query for optimization are built automatically during optimization	Set	Set
Auto create statistics	Specifies whether missing statistics required by a query for optimization are built automatically during optimization	Set	Set
Torn page detection	Specifies whether incomplete pages can be detected	Set	Set
Auto close	Specifies whether the database shuts down after its resources are freed and all users exit	Not set	Not set **Note:** If set, it may take a long time for the database to allocate resources every time a user connects, after the database is closed.
Auto shrink	Specifies whether the database is automatically shrunk every hour, leaving 25% of free space	Not set	Not set **Note:** If set, constant growth/shrinkage may cause file system fragmentation.

Configuration Option	Description	Default	Certification in MS SQL Server 2000
Use quoted identifiers	Specifies whether the MS SQL Server enforces ANSI rules regarding quotation marks. Select this option to specify that double quotation marks be used only for identifiers, such as column and table names. Note that character strings must be enclosed in single quotation marks.	Not set	Not set
Compatibility level	The version of MS SQL Server that the database appears to be (for the application)	80	80

Note: Not all ANSI options can be set using the Enterprise Manager. The ANSI database configuration options include: ANSI_NULLS, QUOTED_IDENTIFIER, ANSI_NULL_DEFAULT, ANSI_PADDING, ANSI_WARNINGS, ARITHABORT, NUMERIC_ROUNDABORT, CONCAT_NULL_YIELDS_NULL. Note that the options you set may not take effect, since these options can also be set at a higher level. For example, if the session option QUOTED_IDENTIFIER was turned on, the equivalent database configuration option is irrelevant. Some tools or database interfaces turn certain session options on or off, so that the database configuration options never take effect.

The following table summarizes the characteristics of each recovery model:

	Model/Support		
	Simple	**Bulk Logged**	**Full**
Allows log backup	No	Yes	Yes
Allows point-in-time/log mark restoration	No	No	Yes
Allows backup log when data crashes (saves changes until the crash point)	No	No	Yes
Amount of bulk operation logging (can affect the performance of bulk operations)	Minimal	Minimal	Full

To check the database properties, run the following:

```
EXEC sp_helpdb <database name>
```

7

Manually Creating the MS SQL Server Databases

This chapter describes the manual creation of the MS SQL Server databases required by Mercury Application Mapping.

Overview of the MS SQL Databases

Mercury Application Mapping uses the following databases for its persistency:

- **CMDB.** The configuration management database.
- **CMDB History.** The configuration management history database.

During the Mercury Application Mapping setup, new databases can be set up automatically by the setup procedure, or already existing databases can be used. Existing databases can either have been created manually in advance, or can have been created by a previous installation of the same release of Mercury Application Mapping.

For details about installing Mercury Application Mapping server, see “Introduction to Installation” in *Mercury Application Mapping Installation Guide*.

Note: The Windows Authentication connection method to MS SQL Server is currently not supported. MS SQL Server login account details must be given during setup as they are used by the jdbc driver.

Manually Creating the Mercury Application Mapping MS SQL Databases

Before creating the databases, the MS SQL Database Server must be installed according to the details in this guide. For further details on installing and configuring the MS SQL Database Server, see Chapter 2, "Installing and Configuring MS SQL Server."

Databases should be manually created prior to running the Mercury Application Mapping setup. During the setup procedure, you select the option to use existing databases and enter the relevant data about the existing databases for which you are prompted. The data you enter is written to the configuration files that are used by the JDBC driver to connect to the existing databases.

This section includes the following topics:

- "Creating the Databases" on page 48
- "Creating the Objects" on page 49
- "Shared CMDB Setup" on page 50

Creating the Databases

The database administrator should create two MS SQL databases; one for CMDB and one for CMDB History.

To create a database, you must have CREATE DATABASE permissions. To connect to an existing database, the login account with which you are connecting must be mapped to dbo in the database.

Note: Members of the sysadmin server role automatically have CREATE DATABASE permissions, and are also mapped to dbo in all databases. A database owner is automatically mapped to dbo in the database.

For details on creating databases manually, see Chapter 6, "Creating and Configuring MS SQL Server Databases."

Note: When the Mercury Application Mapping setup procedure automatically creates the databases, it uses two file groups for each database; one to hold the system tables and one for the application data.

Creating the Objects

Objects are created using SQL scripts that are run against the databases that were previously created. For details, see "Creating the Databases" on page 48.

The scripts are located in the <**Mercury Application Mapping root directory**>**\j2f\CMDB\dbscripts\ms** directory located on the Mercury Application Mapping server, and should be run from this directory. Detailed descriptions of the prerequisites and operation method of each script can be found in the **readme_ms.txt** file in this directory.

The SQL scripts require a thick database client such as ISQL. If such a client is not installed on the Mercury Application Mapping server machine, you can move all the scripts to a different machine that has the required client.

To move the scripts to a different machine:

1 Copy the <**Mercury Application Mapping root directory**>**\j2f\CMDB\dbscripts\ms** directory from the Mercury Application Mapping server machine to the machine with the required client.

2 Change the **SCRIPTS_LOCATION** parameter in the **env_cmdb_ms.bat** file, located in the scripts directory.

Creating the CMDB Objects

To create the CMDB objects, connect to the CMDB database and run the following SQL scripts:

- create_cm_tables_cmdb_ms.sql
- common_sql_dbobjects_create.sql

Creating the CMDB History Objects

To create the CMDB History objects, connect to the CMDB database and run the following SQL scripts:

- create_cm_tables_cmdb_history_ms.sql
- common_sql_dbobjects_create.sql
- mam_sql_set_system_props.sql

Shared CMDB Setup

In a shared CMDB Mercury Application Mapping installation, only the CMDB History database is created, as an existing CMDB database of a Mercury Business Availability Center system is used by Mercury Application Mapping.

To manually create the CMDB History database, refer to relevant parts in “Creating the Databases” on page 48, and “Creating the Objects” on page 49.

During a shared CMDB installation, the Mercury Application Mapping server is registered in the Mercury Business Availability Center management database tables via SQL scripts. If an existing CMDB History database is used instead of a new one, the setup procedure skips this registration step, so it must be carried out manually prior to the installation.

For the location of the script and MS SQL client requirements, see “Creating the Objects” on page 49.

Manually Registering a Shared CMDB

This section explains how to manually register a shared CMDB.

To manually register a shared CMDB:

1. Edit the **register_sql_mamserver.sql** script.
2. Locate the line **set @MAM_machine='change_test'** and replace **change_test** with the name of the Mercury Application Mapping server host.

   Note: You must use the name of the Mercury Application Mapping server host, that is, you must not use its IP address.

3. Save the file.
4. Connect to the Mercury Business Availability Center management database.
5. Run the **register_sql_mamserver.sql** script.

8

MS SQL Server Summary Checklists

This chapter contains a checklist summarizing the server and database configuration options that are supported and recommended for working with Mercury Application Mapping. It also contains a table summarizing the procedures for verifying or modifying server and database settings.

Checklist for Support and Certification

The following checklist summarizes the server and database configuration options that are supported and certified for working with Mercury Application Mapping:

Subject	**MS SQL Server 2000**	
	Supported	**Recommended**
Instances	Default, Single	
Authentication Mode	Mixed	
Collation	Case-Insensitive, Accent-Sensitive, Kana-Insensitive, Width-Insensitive	Windows Locale, with the following options selected: Latin1_General, Accent-Sensitive
Network Libraries	**Server:** TCP/IP and Named Pipes **Client:** TCP/IP and Named Pipes	**Server:** TCP/IP **Client:** TCP/IP
Server Configuration Options	Defaults, unless instructed otherwise	
Data File Properties	Manual file growth, or FILEGROWTH less than or equal to 100 MB	FILEGROWTH: ~30-100 MB
Collation Database Property	Server default	
Database Options	Defaults, unless instructed otherwise	
Recovery Model	Any	Full

Verifying and Modifying Server and Database Settings

The following table summarizes the procedures for verifying or modifying server and database settings:

Server/Database Setting	How to Verify/Modify the Setting
Default Instance	In the operating system's Services applet, a default MS SQL Server instance appears as **MSSQLServer**, while a named instance appears as **MSSQL$INSTNAME**. Ensure that MS SQL Server is installed as a default instance.
Authentication Mode	In the MS SQL Server Enterprise Manager, right-click the server, choose **Properties**, and click the **Security** tab. Select **Mixed Mode**.
Collation Settings	Run: sp_helpsort **Required results:** Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Network Libraries	**Server:** Run **svrnetcn.exe**; **Client:** Run **cliconfg.exe**. **Supported:** TCP/IP and Named Pipes for both the server and client. **Recommended:** Only TCP/IP for both the server and client.
View or change server configuration options	➤ To allow the viewing of all options, run: **EXEC sp_configure 'show advanced options', 1 reconfigure with override** To view the current values run: **EXEC sp_configure** ➤ To change a setting, run: **EXEC sp_configure** '**<option>**', **<value>** Some options take effect only after you run **reconfigure with override**, while others require restarting the MSSQLServer service. See MS SQL Server 2000 Books Online for details.

Server/Database Setting	How to Verify/Modify the Setting
Check whether the user has CREATE DATABASE permissions	Log in to the Query Analyzer with the login you want to check, and run the following: **USE master** **IF PERMISSIONS() & 1 = 1** **PRINT 'User has CREATE DATABASE permissions'** **ELSE** **PRINT 'User does not have CREATE DATABASE permissions'**
Check whether the user is dbo in the database	Log in to the Query Analyzer with the login you want to check. Change the context of the database to the required database, and run the following: **SELECT USER_NAME()**
Verify the database owner	Run: **EXEC sp_helpdb <database name>**
Data and log file destination directory is not compressed (only in NTFS)	Right click the directory, choose **Properties**, and then **Advanced**. Verify that the **Compression** check box is cleared.
Database and database file properties (including recovery model and collation properties)	➤ To view the database and database file properties, run: **EXEC sp_helpdb <database name>** ➤ To change the database properties: **ALTER DATABASE <database name> SET <option> <value>** To change the database file properties: **ALTER DATABASE <database> MODIFY FILE (name = <filename>, <property> = <value>)** You can also view or change these properties from the Database Properties dialog box in the Enterprise Manager.
MS SQL Server service pack version and edition	To verify the version and edition of your MS SQL Server service pack, refer to the following URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;q321185

Part III

Deploying and Maintaining the Oracle Server Database

9

Installing Oracle Database on a Windows Platform

This chapter explains how to install Oracle database on a Windows platform.

About Installing Oracle

Mercury Application Mapping 6.5. is certified with Oracle database release 10.2.0.1 (10gR2).

Oracle release 9.2.0.6 is also supported.

Note:

- Oracle software and database instance should be installed as a prerequisite before installing Mercury Application Mapping server.
- Oracle 10g/9i installation CD-ROMs are not supplied as part of the Mercury Application Mapping distribution package and should be obtained prior to installing Mercury Application Mapping.
- For Oracle 10g, ensure that you have the Oracle 10g Release 2 Installation CD-ROM for Windows platform.
- For Oracle 9i, ensure that you have all three of the 9i installation CD-ROMs, as well as the 9.2.0.6 patch set for Windows platform.

This section includes the following topics:

- "Oracle 10.2.0.1 (10gR2)" on page 60
- "Oracle 9.2.0.6" on page 61

Oracle 10.2.0.1 (10gR2)

To work with Oracle 10.2.0.1 (10gR2), install Oracle using one of the following procedures:

- **Mercury installation.** You can use Mercury installation scripts with your software to install the software and/or database instance according to the required software options, and to the standard recommended configuration for Mercury Application Mapping server instance.

  For details about Oracle 10g Release 2 installation, see "Installing Oracle 10g Release 2" on page 62.

- **Regular installation.** Install the Oracle software and the database instance manually.

  Make sure all database parameters and file sizes are specified according to the provided guidelines. For details, see "Oracle Server Configuration Guidelines" on page 137.

Note: Oracle software and database instance should be installed as a prerequisite before installing the Mercury Application Mapping server.

Oracle 9.2.0.6

To work with Oracle 9.2.0.6, install Oracle using one of the following procedures:

- If Oracle 9i is already installed and you have an active database, install the Oracle 9.2.0.6 patch.

  upgrade to Oracle 9.2.0.6 – Upgrade to Oracle 9.2.0.6 if you installed Oracle 9i using the provided guidelines, and you only want to install Oracle9i release 2 (9.2.0.6) patch set. For details about upgrading to Oracle 9.2.0.6, see "Upgrading an Existing Database to Oracle 9.2.0.6" on page 106.

- If Oracle 9i is not yet installed, install it using one of the following procedures:

  - **Mercury installation.** You can use Mercury installation scripts with your software to install both the software and the database instance according to the required software options, and to the standard recommended configuration for Mercury Application Mapping server instance.

    For details of the Oracle 9i installation, see "Performing Oracle 9i Silent Installation" on page 81.

- **Regular installation.** Installing the Oracle software and the database instance manually.

  Ensure that all database parameters and file sizes are specified according to the provided guidelines. For details, see “Oracle Server Configuration Guidelines” on page 137.

  For details of the Oracle 9i installation, see “Performing Oracle 9i Regular Installation” on page 91.

Installing Oracle 10g Release 2

This section explains how to install Oracle 10g Release 2 database using the setup procedure provided by Mercury.

The procedure includes the following steps:

1. Loading the configuration files. For details, see “Loading Oracle Silent Installation Configuration Files” on page 63
2. Editing the configuration with the unique names and paths of the setup. For details, see “Editing the Configuration Files” on page 63
3. Installing the database software. For details, see “Installing the Oracle Software” on page 65
4. Installing the database instance. For details, see “Installing the Oracle Database Instance” on page 66
5. Starting the Oracle Listener. For details, see “Starting the Oracle Listener” on page 81
6. Checking the installation. For details, see “Checking the Installation” on page 81

This section includes the following topics:

- “Loading Oracle Silent Installation Configuration Files” on page 63
- “Editing the Configuration Files” on page 63
- “Installing the Oracle Software” on page 65
- “Installing the Oracle Database Instance” on page 66

- "Starting the Oracle Listener" on page 81
- "Checking the Installation" on page 81

Loading Oracle Silent Installation Configuration Files

Load the installation files to the target machine. The files are located in the **dbTools\10gsilentOra** folder on the Mercury Application Mapping 6.5 Setup Windows CD-ROM.

To load Oracle silent installation configuration files:

1. Create a folder named **SilentOra** under the D: drive of the target machine.
2. Copy the entire **10gsilentOra** directory from the installation CD-ROM into the **SilentOra** folder you created.
3. Create a folder named **Disk1** under the root of the **D:** drive. This folder will include the Oracle 10g installation CD-ROM (700MB). To copy the CD-ROM to another drive see "Editing the Configuration Files" on page 63.
4. Copy the contents of the Oracle 10g installation CD-ROM into the following directory: <**Oracle Silent Installation root directory**>**Disk1**.

Editing the Configuration Files

Edit the parameters of the configuration files. Make sure you complete the following procedures, if relevant, before starting the next stage.

This section includes the following topics:

- "Editing the Relevant Configuration Files if Disk1 is Not Copied to the D: Drive" on page 64
- "Editing the Relevant Configuration Files if you Chose the Non-Default Oracle Home" on page 64
- "Editing the Listener Configuration Files" on page 65

Editing the Relevant Configuration Files if Disk1 is Not Copied to the D: Drive

Edit the files described in this section only if you did not copy Disk1 to the D: drive.

To edit the relevant configuration files if Disk1 is not copied to the D: drive:

1. In the **d:\SilentOra\Inst-Oracle.bat** file, in the **d:\Disk1\Setup** section, replace the **D:** drive with the drive you copied Disk1 to.
2. In the **D:\SilentOra\custom.rsp** file, in the **#EDIT** section, in the **FROM_LOCATION** parameter, replace the **D:** drive with the drive you copied Disk1 to.

Editing the Relevant Configuration Files if you Chose the Non-Default Oracle Home

Edit the files described in this section only if you chose a non-default Oracle home (the default is: **d:\Oracle\ora10gR2**).

To edit the relevant configuration files if you chose the non-default Oracle home:

1. In the **d:\SilentOra\custom.rsp** file, in the **#EDIT** section, in the **ORACLE_HOME** parameter, replace the **D:** drive with the drive you selected for the Oracle home. For example, H:\Oracle\ora 10gR2).
2. In the **D:\SilentOra\Inst-DBCA.bat** file, in the **#EDIT** section, in the **ORACLE_HOME** parameter, replace the **D:** drive with the drive you selected for the Oracle home. For example, H:\Oracle\ora 10gR2).
3. In the **D:\SilentOra\listener.ora** file, in the **#EDIT** section, in the **ORACLE_HOME** parameter, replace the **D:** drive with the drive you selected for the Oracle home. For example, H:\Oracle\ora 10gR2).

Editing the Listener Configuration Files

Edit the Listener Configuration files described in this section.

To edit the Listener Configuration files:

1 In the **D:\SilentOra\listener.ora** file (Oracle Listener configuration file), edit the following parameters:

- In the **HOST** parameter, replace the existing **host_name** with the host_name or ip_address of the target machine.
- In the **GLOBAL_DBNAME=CMDB** parameter, replace the Service Name with the required service name (if it is not the default CMDB).
- In the **SID_NAME=CMDB** parameter, replace SID with the required SID name (if it is not CMDB).

Installing the Oracle Software

Install Oracle 10g Release 2 software before installing the database instance.

If the software is already installed and you need to install a Mercury Business Availability Center/Mercury Application Mapping database, skip to "Installing Oracle Database Instance" on page 90.

To install the Oracle software:

1 Make sure there is no previous installation of Oracle Database Server. If Oracle Database Server has been installed in the past, you must uninstall it. For details, see below.

2 Run the **Inst-Oracle.bat** file, located in **d:\SilentOra**.

3 In the **c:\Program Files\Oracle\Inventory\logs** folder, wait until the **silentInstall<time>.log** appears.

Note: As long as the installation is running, **javaw.exe** appears in the Task Manager.

4 Ensure that the following line appears in the file:

The installation of Database 10g was successful.

To uninstall Oracle Database Server:

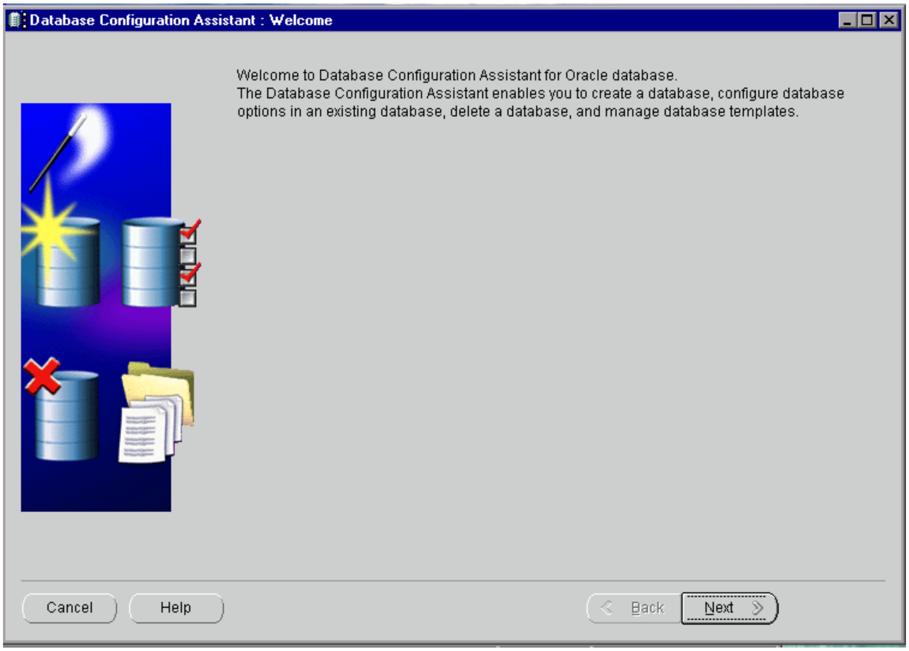
1. Open Oracle Installer by running **setup.exe** from **<Oracle Silent Installation root directory>\Disk1** directory.
2. Click **Deinstall Products**.
3. Select all the available products.
4. Click **Remove** and wait until the uninstall procedure completes.

Installing the Oracle Database Instance

Install the Oracle database instance.

To install the Oracle Database instance:

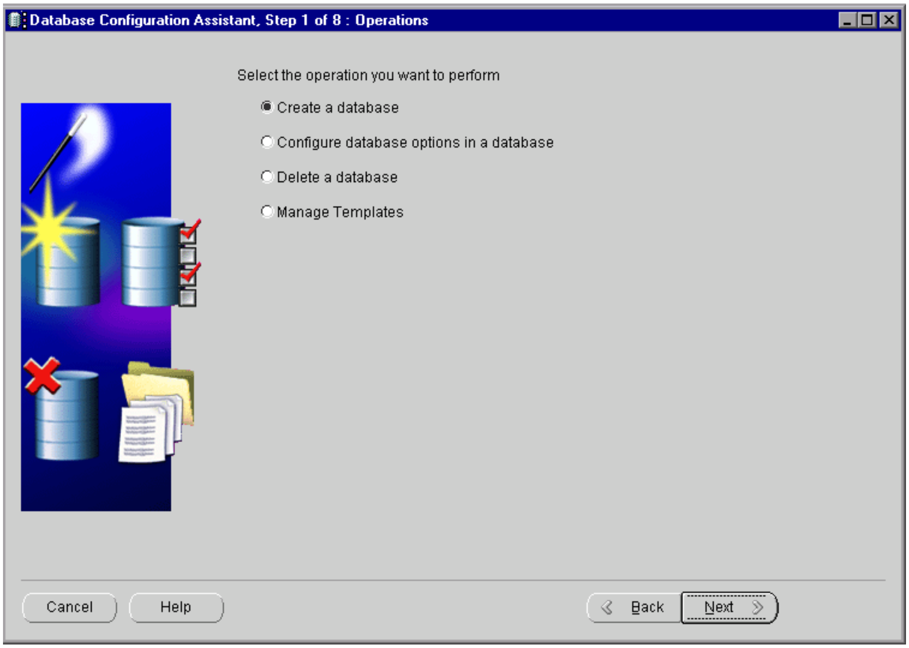
1. Open a DOS command prompt.
2. Run the **Inst-DBCA.bat** command, located in **d:\SilentOra** to open the following page:



Database Configuration Assistant : Welcome

Welcome to Database Configuration Assistant for Oracle database.
The Database Configuration Assistant enables you to create a database, configure database options in an existing database, delete a database, and manage database templates.

3 Click **Next** to open the following page:



Select the operation you want to perform

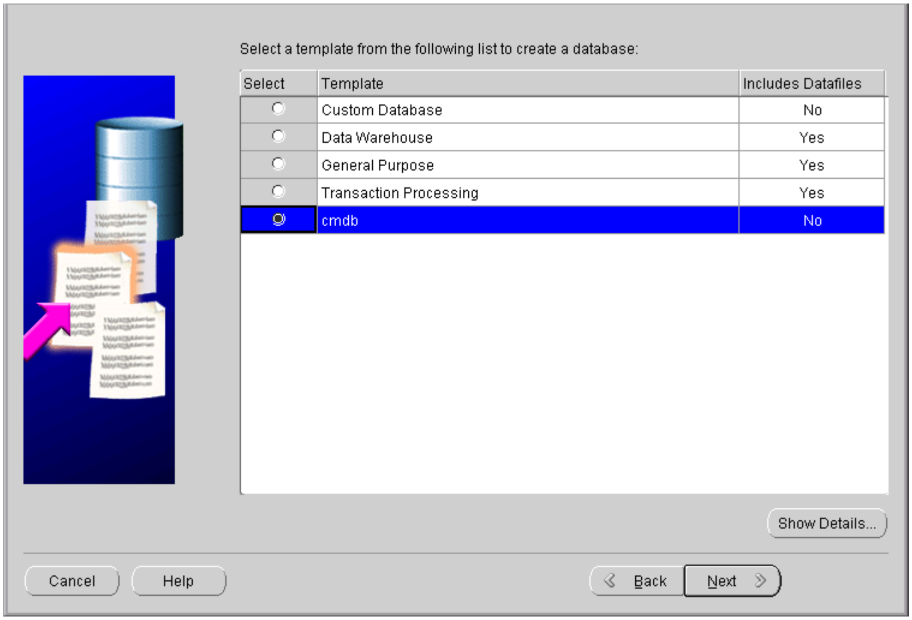
● Create a database

○ Configure database options in a database

○ Delete a database

○ Manage Templates

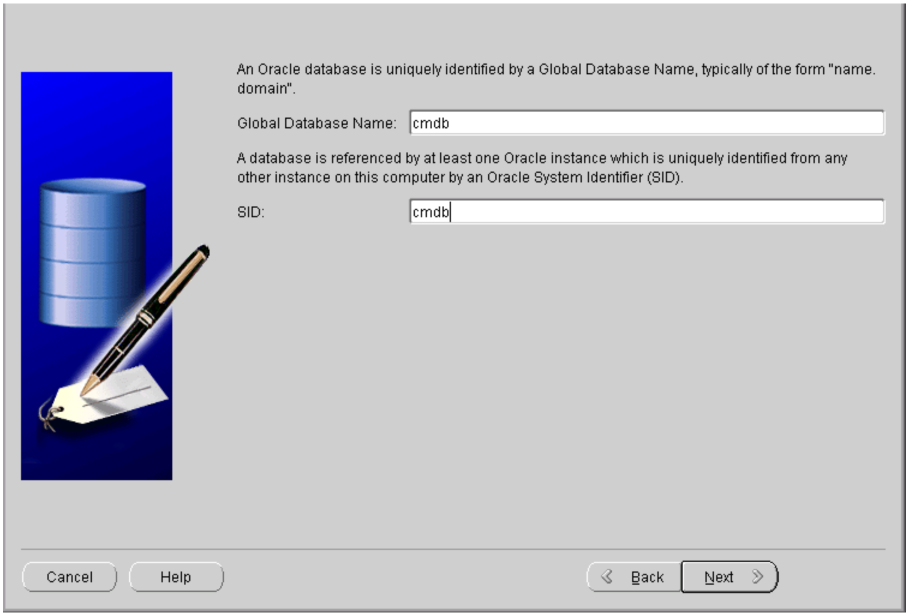
4 Select **Create a Database** and click **Next** to open the following page:



Select a template from the following list to create a database:

Select	Template	Includes Datafiles
○	Custom Database	No
○	Data Warehouse	Yes
○	General Purpose	Yes
○	Transaction Processing	Yes
●	cmdb	No

5 Select **cmdb** and click **Next** to open the following page:



An Oracle database is uniquely identified by a Global Database Name, typically of the form "name.domain".

Global Database Name: cmdb

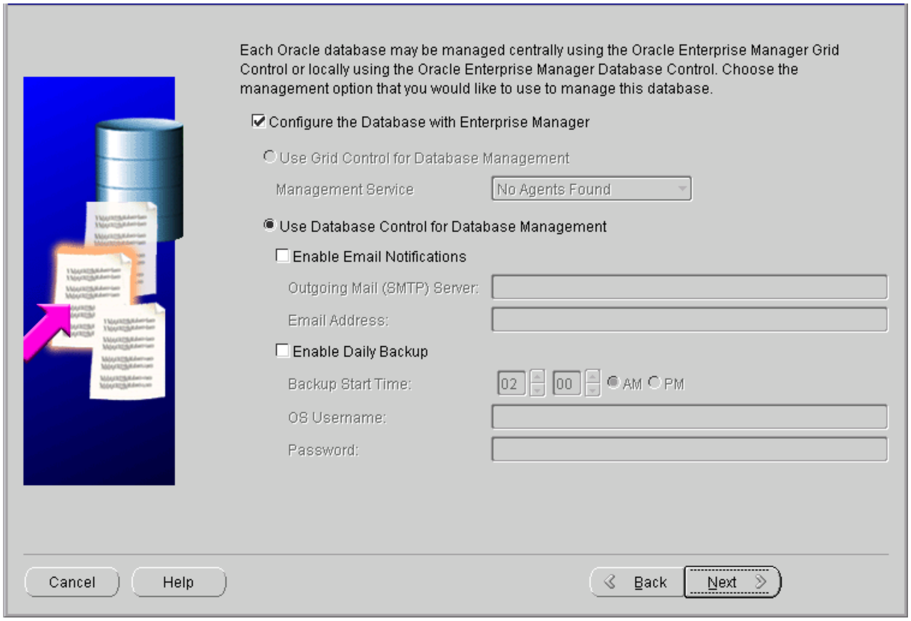
A database is referenced by at least one Oracle instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID).

SID: cmdb

The page displays the default SID and Global Database Name (Service Name).

6 Change the default SID and Global Database Name according to the names you selected earlier in “Editing the Listener Configuration Files” on page 65. It is recommended to enter the same name in both boxes.

7 Click **Next** to open the following page:



Each Oracle database may be managed centrally using the Oracle Enterprise Manager Grid Control or locally using the Oracle Enterprise Manager Database Control. Choose the management option that you would like to use to manage this database.

☑ Configure the Database with Enterprise Manager

○ Use Grid Control for Database Management

Management Service: No Agents Found

● Use Database Control for Database Management

□ Enable Email Notifications

Outgoing Mail (SMTP) Server:

Email Address:

□ Enable Daily Backup

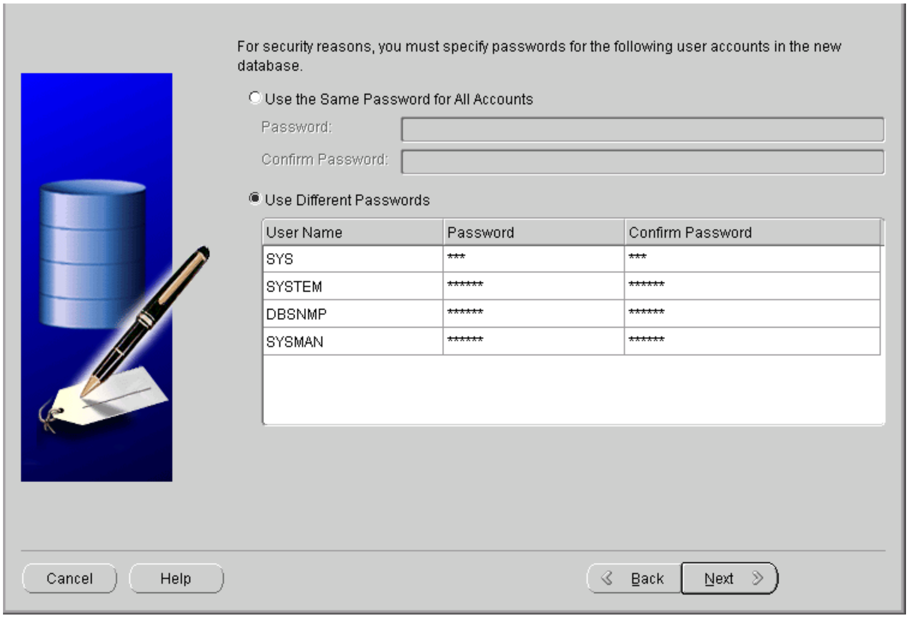
Backup Start Time: 02 00 ● AM ○ PM

OS Username:

Password:

8 Select **Configure the Database with Enterprise Manager** and **Use Database Control for Database Management.**

9 Click **Next** to open the following page:



For security reasons, you must specify passwords for the following user accounts in the new database.

○ Use the Same Password for All Accounts

Password:

Confirm Password:

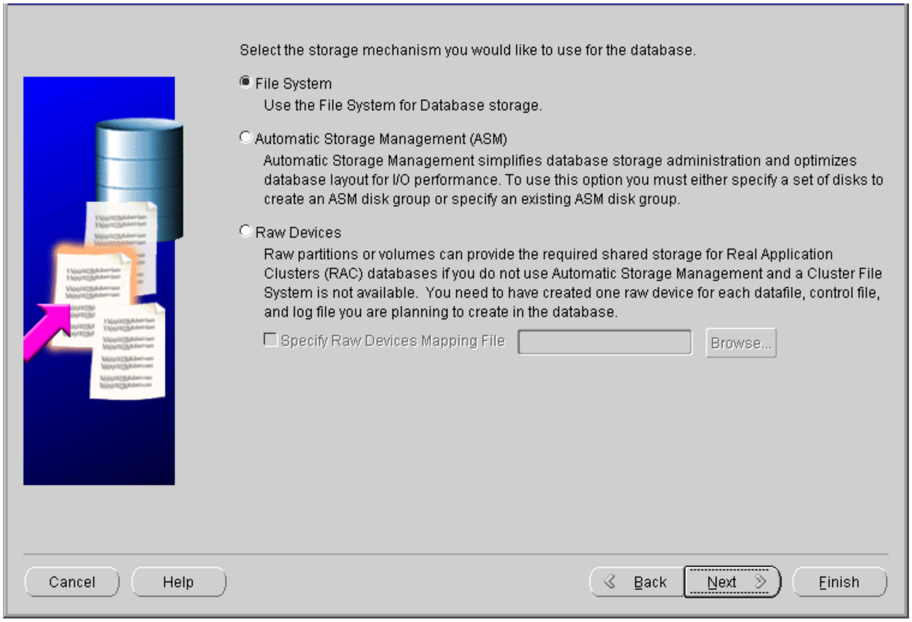
● Use Different Passwords

User Name	Password	Confirm Password
SYS	***	***
SYSTEM	******	******
DBSNMP	******	******
SYSMAN	******	******

The page displays the database credentials. Choose non-default passwords for the **SYS** and **SYSTEM** super users.

The **SYSMAN** and **DBSNMP** are also privileged users in Oracle 10g Release 2; make sure you choose non-default passwords for those users as well.

10 Click **Next** to open the following page:



Select the storage mechanism you would like to use for the database.

● File System

Use the File System for Database storage.

○ Automatic Storage Management (ASM)

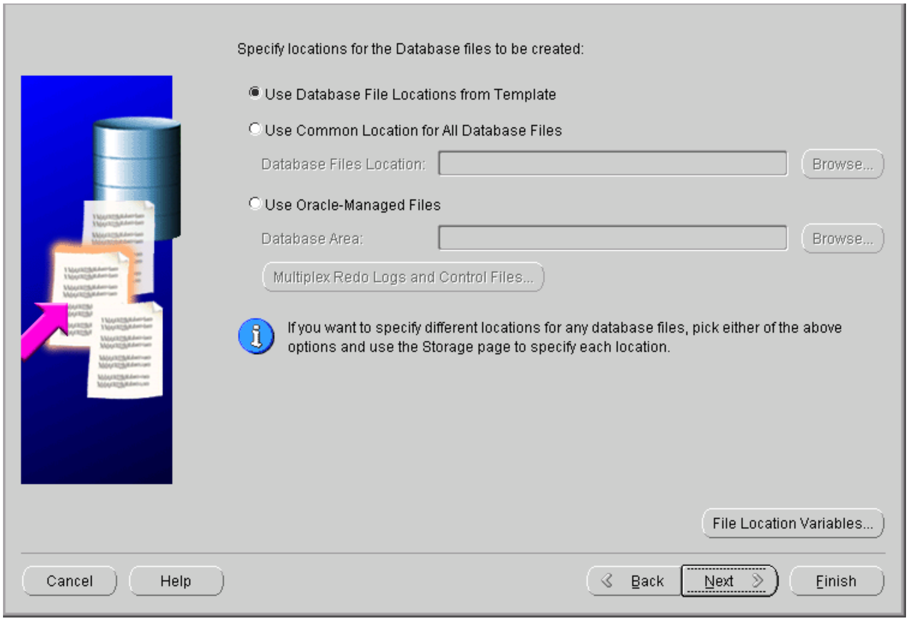
Automatic Storage Management simplifies database storage administration and optimizes database layout for I/O performance. To use this option you must either specify a set of disks to create an ASM disk group or specify an existing ASM disk group.

○ Raw Devices

Raw partitions or volumes can provide the required shared storage for Real Application Clusters (RAC) databases if you do not use Automatic Storage Management and a Cluster File System is not available. You need to have created one raw device for each datafile, control file, and log file you are planning to create in the database.

□ Specify Raw Devices Mapping File

11 Select **File System** and click **Next** to open the following page:



Specify locations for the Database files to be created:

● Use Database File Locations from Template

○ Use Common Location for All Database Files

Database Files Location:

○ Use Oracle-Managed Files

Database Area:

If you want to specify different locations for any database files, pick either of the above options and use the Storage page to specify each location.

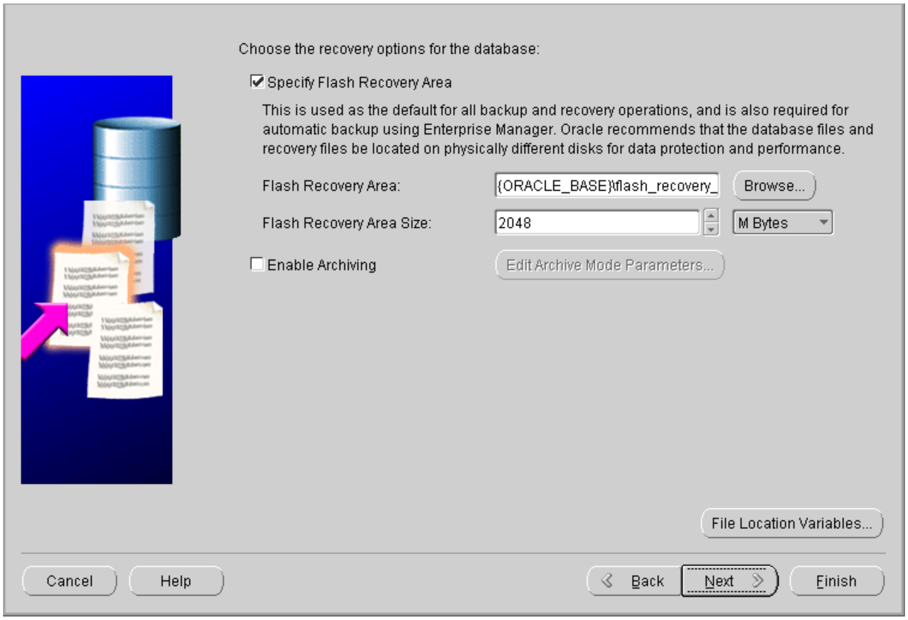
Browse...
Browse...
Multiplex Redo Logs and Control Files...
File Location Variables...
Cancel
Help
Back
Next
Finish

12 If you do not want to use the default location for the data files (for example, if you have a NETAPP drive to store Oracle files), select **Use Common Location for All Database Files**, and enter the new location in the **Database Files Location** box.

Note: Individual file locations can be modified at the storage stage. For details, see step 17 on page 77.

To locate the data files in the default location \<**ORACLE_BASE**>**oradata**\<**DB_NAME**>\, continue to the next step.

13 Click **Next** to open the following page:



Choose the recovery options for the database:

☑ Specify Flash Recovery Area

This is used as the default for all backup and recovery operations, and is also required for automatic backup using Enterprise Manager. Oracle recommends that the database files and recovery files be located on physically different disks for data protection and performance.

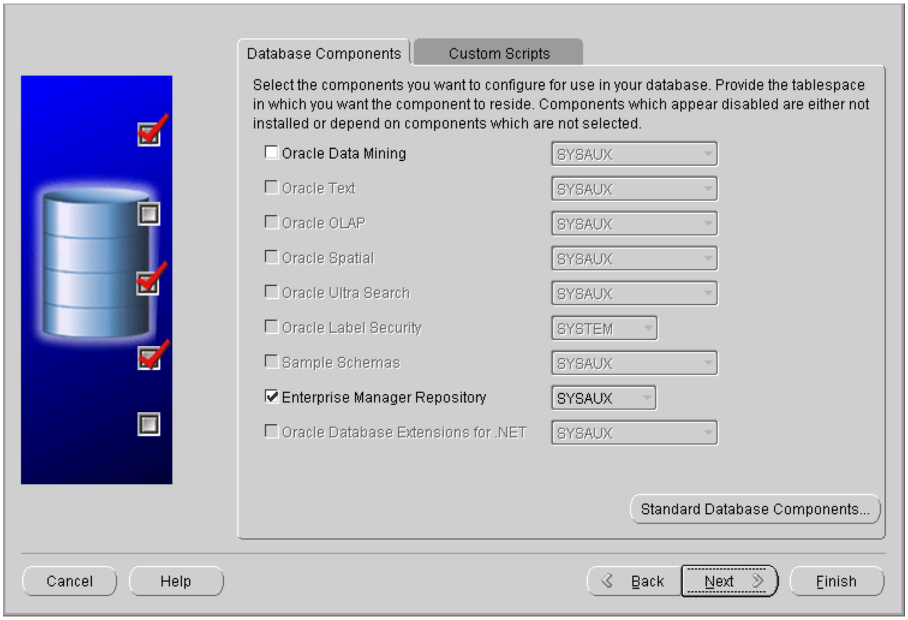
Flash Recovery Area: {ORACLE_BASE}\flash_recovery_

Flash Recovery Area Size: 2048 M Bytes

☐ Enable Archiving

14 Select **Specify Flash Recovery Area**, but do not modify the defaults in the **Flash Recovery Area** and **Flash Recovery Area Size** boxes.

15 Click **Next** to open the following page:

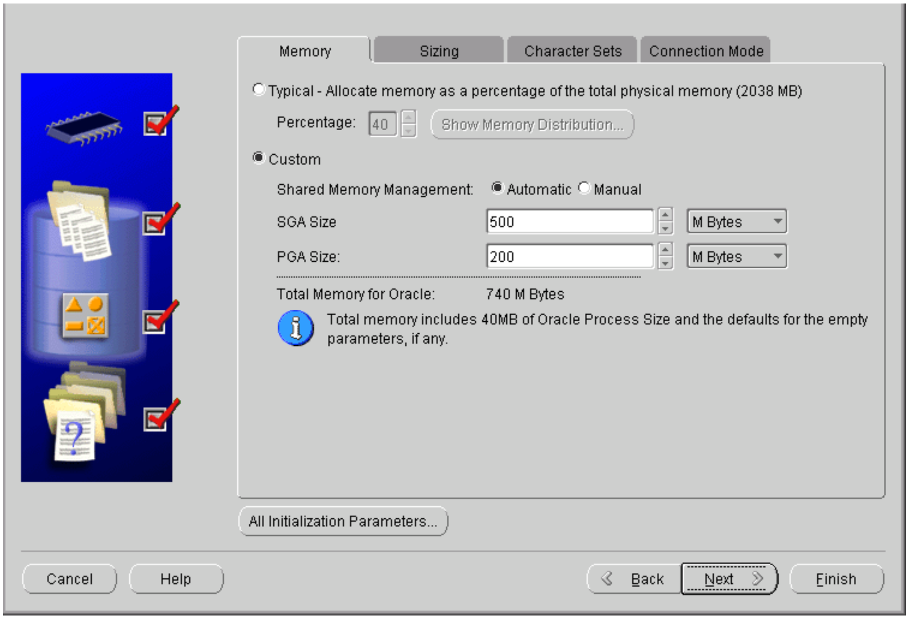


Database Components | Custom Scripts

Select the components you want to configure for use in your database. Provide the tablespace in which you want the component to reside. Components which appear disabled are either not installed or depend on components which are not selected.

□ Oracle Data Mining	SYSAUX
□ Oracle Text	SYSAUX
□ Oracle OLAP	SYSAUX
□ Oracle Spatial	SYSAUX
□ Oracle Ultra Search	SYSAUX
□ Oracle Label Security	SYSTEM
□ Sample Schemas	SYSAUX
☑ Enterprise Manager Repository	SYSAUX
□ Oracle Database Extensions for .NET	SYSAUX

16 Click **Next** to open the following page. The page displays all the initialization parameters for the database instance:



○ Typical - Allocate memory as a percentage of the total physical memory (2038 MB)

Percentage: 40

● Custom

Shared Memory Management: ● Automatic ○ Manual

SGA Size 500 M Bytes

PGA Size: 200 M Bytes

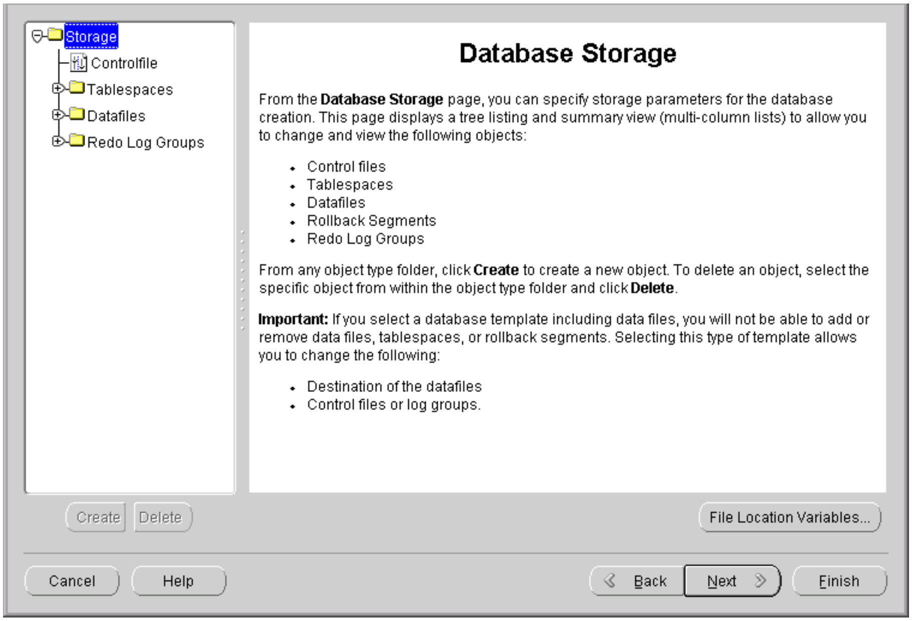
Total Memory for Oracle: 740 M Bytes

Total memory includes 40MB of Oracle Process Size and the defaults for the empty parameters, if any.

A pre-defined template is used to reflect the minimum recommended environment needed to run the Mercury Application Mapping database.

Note: Experienced DBAs can modify the template values according to the site requirements. SGA Size and PGA Size template sizes are smaller than the minimum recommendation for a typical Mercury Application Mapping deployment to accommodate smaller environments. Make sure the parameters are resized according to the configuration guidelines. For details, see “Oracle Server Configuration Guidelines” on page 137.

17 Click **Next** to accept the parameters defined in the template and to open the Database Storage page. The page displays the storage configuration information.



Database Storage

From the **Database Storage** page, you can specify storage parameters for the database creation. This page displays a tree listing and summary view (multi-column lists) to allow you to change and view the following objects:

- Control files
- Tablespaces
- Datafiles
- Rollback Segments
- Redo Log Groups

From any object type folder, click **Create** to create a new object. To delete an object, select the specific object from within the object type folder and click **Delete**.

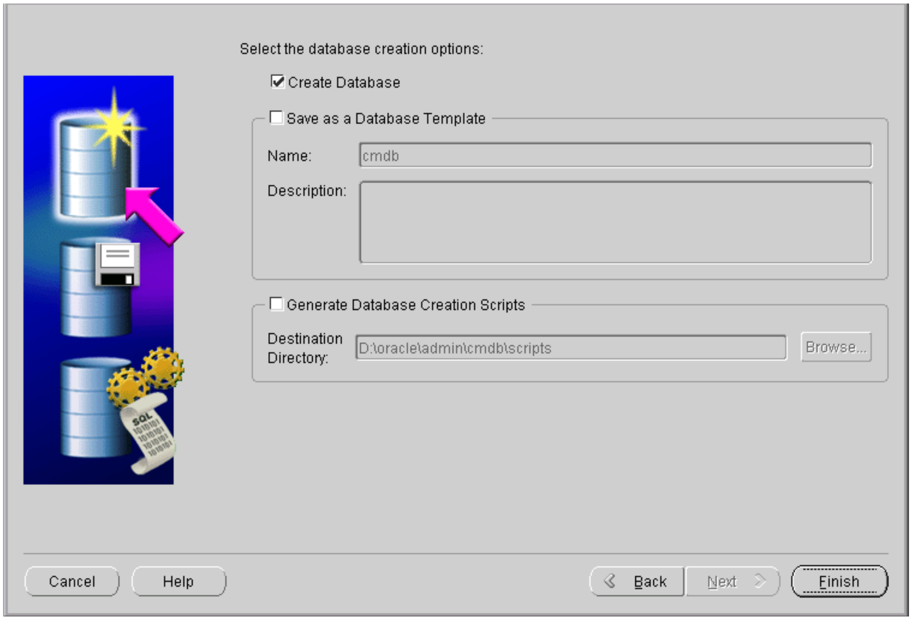
Important: If you select a database template including data files, you will not be able to add or remove data files, tablespaces, or rollback segments. Selecting this type of template allows you to change the following:

- Destination of the datafiles
- Control files or log groups.

A pre-defined template is used to reflect the recommended environment needed to run the Mercury Application Mapping database.

If you are an experienced DBA, you can modify the storage parameters and the location of a specific Tablespace; for example, you could move the Temporary Tablespace to another location.

18 Click **Next** to accept the parameters defined in the template or to save the changes you have made, and to open the following page.



Select the database creation options:

☑ Create Database

□ Save as a Database Template

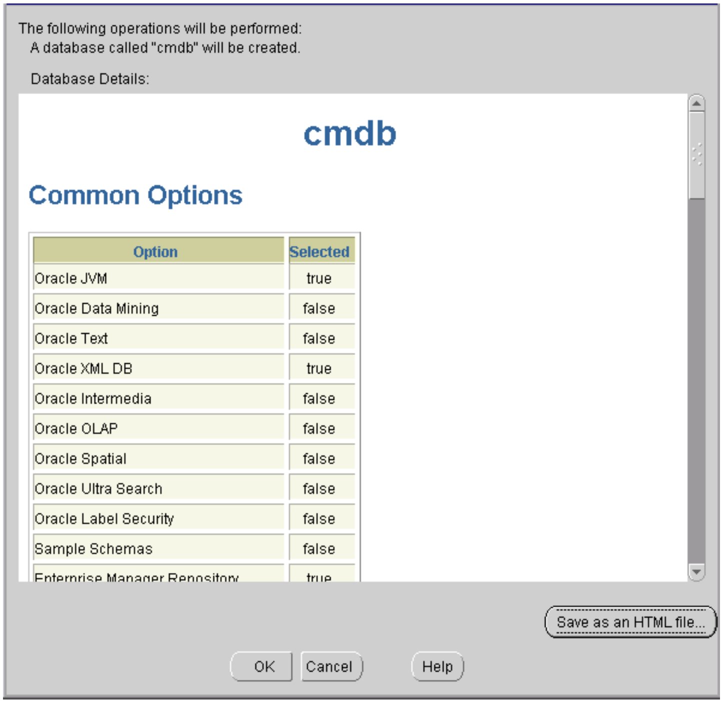
Name: cmdb

Description:

□ Generate Database Creation Scripts

Destination Directory: D:\oracle\admin\cmdb\scripts

19 Press **Finish** to open the following page:



The following operations will be performed:
A database called "cmdb" will be created.

Database Details:

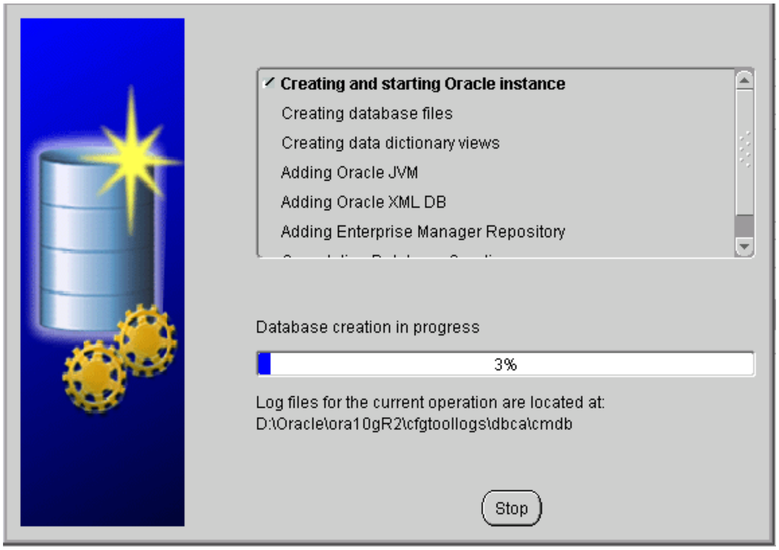
cmdb

Common Options

Option	Selected
Oracle JVM	true
Oracle Data Mining	false
Oracle Text	false
Oracle XML DB	true
Oracle Intermedia	false
Oracle OLAP	false
Oracle Spatial	false
Oracle Ultra Search	false
Oracle Label Security	false
Sample Schemas	false
Enterprise Manager Repository	true

20 Click **OK**.

21 Wait until setup has finished.



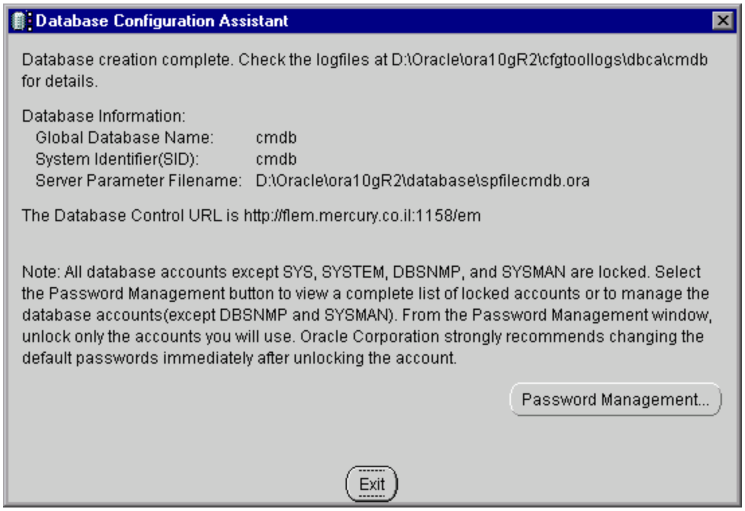
✓ **Creating and starting Oracle instance**
Creating database files
Creating data dictionary views
Adding Oracle JVM
Adding Oracle XML DB
Adding Enterprise Manager Repository

Database creation in progress

3%

Log files for the current operation are located at:
D:\Oracle\ora10gR2\cfgtoollogs\dbca\cmdb

22 The Database creation complete page opens:



Database Configuration Assistant

Database creation complete. Check the logfiles at D:\Oracle\ora10gR2\cfgtoollogs\dbca\cmdb for details.

```
Database Information:
  Global Database Name:      cmdb
  System Identifier(SID):    cmdb
  Server Parameter Filename: D:\Oracle\ora10gR2\database\spfilecmdb.ora
```

The Database Control URL is http://flem.mercury.co.il:1158/em

Note: All database accounts except SYS, SYSTEM, DBSNMP, and SYSMAN are locked. Select the Password Management button to view a complete list of locked accounts or to manage the database accounts(except DBSNMP and SYSMAN). From the Password Management window, unlock only the accounts you will use. Oracle Corporation strongly recommends changing the default passwords immediately after unlocking the account.

23 Click **Exit**.

24 Verify that the output **Database opened** appears in the following log: **<Oracle_home>\cfgtoollogs\dbca\<SID>\PostDBCreation.log**.

Starting the Oracle Listener

Start the Oracle Listener before you check the installation.

To start the Oracle Listener:

1. Open a DOS window.
2. Go to the **<Oracle_home_directory>\bin** directory.
3. Enter **lsnrctl start** and press **Enter**.

Checking the Installation

Check the installation.

To check the installation:

1. Make sure that the following services have been created:
   - ➤ **OracleService<SID>**
   - ➤ **Oracle<Home_Name>TNSListener**
2. Open a DOS command prompt and issue the following command to check whether the database is available:
   sqlplus system/<system_password>@SID

Performing Oracle 9i Silent Installation

Mercury Application Mapping uses an Oracle Enterprise Database server for application persistency.

This section explains how to install both Oracle software and database on Microsoft platforms, as preparation for Mercury Application Mapping installation.

The procedure includes the following steps:

1. Preparing the installation files. For details, see “Loading Oracle Silent Installation Configuration Files” on page 82.

2 Editing the configuration and setup files with the unique names and paths of the setup. For details, see "Editing Oracle Silent Installation Configuration Files" on page 83.

3 Installing the database software. For details, see "Installing the Oracle Software" on page 86.

4 Installing the patch set. For details, see "Installing the Oracle Patch Set" on page 87.

5 Installing the database instance. For details, see "Installing Oracle Database Instance" on page 90.

6 Starting the Oracle Listener. For details, see "Starting the Oracle Listener" on page 90.

7 Checking the installation. For details, see "Checking the Installation" on page 90.

This section includes the following topics:

- "Loading Oracle Silent Installation Configuration Files" on page 82
- "Editing Oracle Silent Installation Configuration Files" on page 83
- "Installing the Oracle Software" on page 86
- "Installing the Oracle Patch Set" on page 87
- "Installing Oracle Database Instance" on page 90
- "Starting the Oracle Listener" on page 90
- "Checking the Installation" on page 90

Loading Oracle Silent Installation Configuration Files

Load the installation files to the target machine. The files are located in the **dbTools\9iSilentOra** folder on the Mercury Application Mapping 6.5 Setup Windows CD-ROM.

To load Oracle Silent Installation Configuration files:

1 Create a folder named **SilentOra** under **d:** drive of the target machine.

2 Copy the entire **9isilentOra** directory from the installation CD-ROM into the SilentOra directory.

3. Create a folder for each CD-ROM.

4. Copy the contents of each one of the Oracle 9i three installation CD-ROMs to the corresponding directory in the temporary location, as described below.

   ➤ copy **DISK1** to <**Oracle Silent Installation root directory**>**Disk1**

   ➤ copy **DISK2** to <**Oracle Silent Installation root directory**>**Disk2**

   ➤ copy **DISK3** to <**Oracle Silent Installation root directory**>**Disk3**

   ➤ copy the 9.2.0.6 patch directory to <**Oracle Silent Installation root directory**>**Patch**

Editing Oracle Silent Installation Configuration Files

Edit the parameters of the Oracle Silent Installation configuration files. You must complete this procedure before starting the next stage.

To edit Oracle silent installation configuration files:

1. Open the **common.rsp** file, which contains general data and file locations, and edit it as follows:

   ➤ Enter the Oracle installation directory in <**Oracle_Home**>. For example, **c:****Oracle****ora92**.

   ➤ Enter the logical name for the Oracle installation in <**Oracle_Home_Name**>. The default is **ora92**.

   ➤ Enter <**Oracle Silent Installation root directory**>**Disk1****stage****products.jar** (the location of the Oracle installation) in **[SESSION]** <**From_Location**>.

   ➤ Enter <**Oracle Silent Installation root directory**>**Disk2** (the location for the CD-ROM 2 of the Oracle installation) in **[SESSION]** <**Location_For_Disk2**>.

   ➤ Enter <**Oracle Silent Installation root directory**>**Disk3** (the location for CD3 of the Oracle installation) in **[SESSION]** <**Location_For_Disk3**>.

2 Open the **custom.rsp** file, which contains the parameters needed to install the Oracle software, and edit it as follows:

- Enter **d:\SilentOra\common.rsp** in **[INCLUDE]** <**FILE1**>.
- enter **d:\SilentOra\netca.rsp** in **[SESSION] [oracle.networking. netca 9.2.0.1.0]** <**s_responseFileName**>.

3 Open the **dbca.rsp** file, which contains the parameters for Oracle database configuration, and edit it as follows:

- Enter **d:\SilentOra\common.rsp** in **[INCLUDE]** <**FILE1**>.
- Enter **SKAZAL** in **[CREATEDATABASE]** <**GDBNAME**>. Change the default to a name containing up to seven characters.
- Enter **SKAZAL** in **[CREATEDATABASE]** <**SID**>. Change the default to a name containing up to seven characters.

4 Open the **netca.rsp** file, which contains network configuration parameters, and edit it as follows:

- Enter **{“TCP;<HostName>;1521”}** in **[SESSION]** <**NSN_PROTOCOLS**> where: <**HostName**> is either the name of the Host computer or its IP address.

5 Open the **listener.ora** file that is located in **d:\SilentOra** and edit it as follows:

- Enter the name or IP address of the host computer in <**HOST NAME**>.
- Enter the Oracle installation directory as defined in step 1 on page 83 in <**ORACLE_HOME**>.

- Enter the name of the database as defined in step 3 on page 84 in <**ORACLE_SID**>. The following code is included in **listener.ora** :

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = <HOST NAME>)(PORT
= 1521))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = <ORACLE_HOME>)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = <ORACLE_SID>)
      (ORACLE_HOME = <ORACLE_HOME>)
      (SID_NAME = <ORACLE_SID>)
    )
```

6 Open the **Inst-Oracle.bat** file, which is the batch file used to install the Oracle software. Update the full path for the **setup.exe** file and the **custom.rsp** file (if it is incorrect) in the following line: **D:\Disk1\setup -responseFile D:\silentora\custom.rsp -silent -noconsole.**

7 Open the **Inst-DBCA.bat** file, which is the batch file used to install the Oracle database, and edit it as follows:

- Enter the full path defined for the Oracle Installation directory as defined in step 1 on page 83 in <**Oracle_Home**>.

Installing the Oracle Software

This section explains how to install the Oracle software.

To install the Oracle software:

1. Make sure there is no previous installation of Oracle Database Server. If Oracle Database Server has been installed in the past, you must uninstall it. For details, see below.
2. Run the **Inst-Oracle.bat** file, located in **d:\SilentOra**.
3. In the **c:\Program Files\Oracle\Inventory\logs** folder, wait until the **silentInstall<time>.log** appears.

Note: As long as the installation is running, **javaw.exe** appears in the Task Manager.

---

4. Ensure that the following line appears in the file:

   The installation of Oracle9i Database was successful.

To uninstall Oracle Database Server:

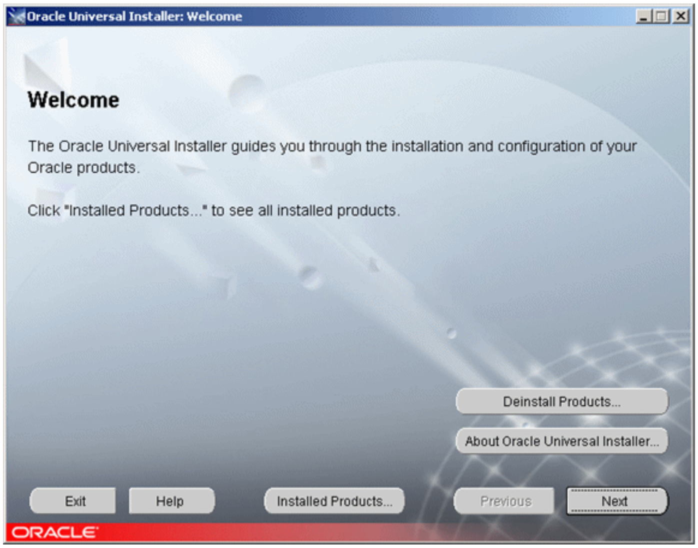
1. Open Oracle Installer by running **setup.exe** from <**Oracle Silent Installation root directory**>**Disk1** directory.
2. Click **Deinstall Products**.
3. Select all the available products.
4. Click **Remove** and wait until the uninstall procedure completes.
5. If you installed a database with the same SID (Instance Identifier) in the previous installation, you must remove the service otherwise the installation will fail. To remove the service:
   - open the DOS prompt
   - move to the <**Oracle Silent Installation root directory**> path
   - enter **Instsrv.exe OracleService<SID> remove** to remove the service

Installing the Oracle Patch Set

This section explains how to install the Oracle patch set.

To install the Oracle patch set:

1 To run the Oracle Universal Installer, run **setup.exe.** located in \<**Oracle Silent Installation root directory**>**Patch.** The Welcome page is displayed.



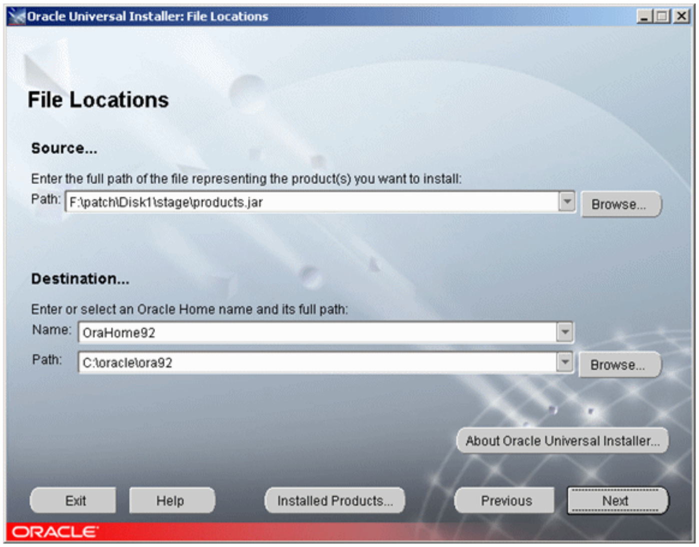
Oracle Universal Installer: Welcome

Welcome

The Oracle Universal Installer guides you through the installation and configuration of your Oracle products.

Click "Installed Products..." to see all installed products.

2 Click **Next** to open the Specify File Locations page.



Oracle Universal Installer: File Locations

File Locations

Source...

Enter the full path of the file representing the product(s) you want to install:

Path: F:\patch\Disk1\stage\products.jar

Destination...

Enter or select an Oracle Home name and its full path:

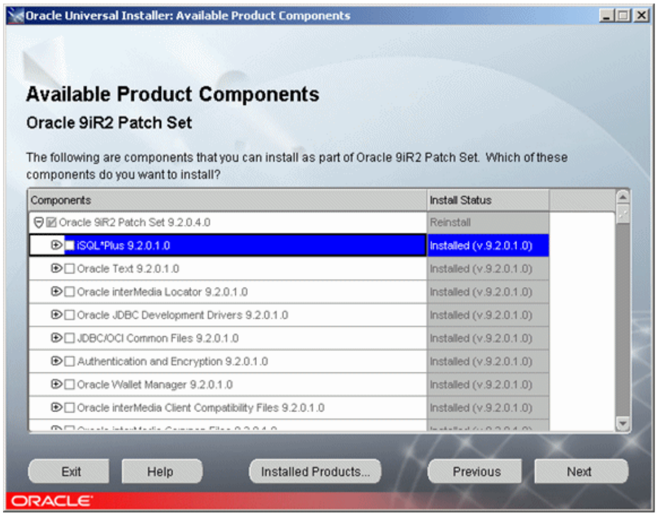
Name: OraHome92

Path: C:\oracle\ora92

See the following table for a description of the fields:

Field	**Description**
Path (under **Source**)	The path to the file that contains the names of products you need to install.
Name (under **Destination**)	The logical name of the Oracle installation as defined in the **common.rsp** file. For details, see “Editing Oracle Silent Installation Configuration Files” on page 83.
Path (under **Destination**)	The location of the Oracle installation directory as defined in the **common.rsp** file. For details, see “Editing Oracle Silent Installation Configuration Files” on page 83.

3 Click **Next** to display the list of installed components to which the patch is applied in the Available Product Components page.


Oracle Universal Installer: Available Product Components

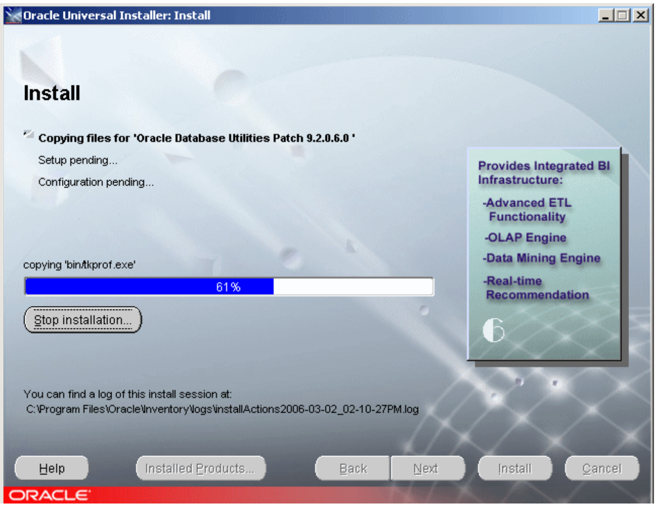
Available Product Components

Oracle 9iR2 Patch Set

The following are components that you can install as part of Oracle 9iR2 Patch Set. Which of these components do you want to install?

Components	Install Status
☑ Oracle 9iR2 Patch Set 9.2.0.4.0	Reinstall
□ iSQL*Plus 9.2.0.1.0	Installed (v.9.2.0.1.0)
□ Oracle Text 9.2.0.1.0	Installed (v.9.2.0.1.0)
□ Oracle interMedia Locator 9.2.0.1.0	Installed (v.9.2.0.1.0)
□ Oracle JDBC Development Drivers 9.2.0.1.0	Installed (v.9.2.0.1.0)
□ JDBC/OCI Common Files 9.2.0.1.0	Installed (v.9.2.0.1.0)
□ Authentication and Encryption 9.2.0.1.0	Installed (v.9.2.0.1.0)
□ Oracle Wallet Manager 9.2.0.1.0	Installed (v.9.2.0.1.0)
□ Oracle interMedia Client Compatibility Files 9.2.0.1.0	Installed (v.9.2.0.1.0)

4 Click **Next** to install the patch and click **Install** to complete the installation.


Oracle Universal Installer: Install

Install

Copying files for 'Oracle Database Utilities Patch 9.2.0.6.0 '

Setup pending...

Configuration pending...

copying 'bin/tkprof.exe'

61 %

You can find a log of this install session at:
C:\Program Files\Oracle\Inventory\logs\installActions2006-03-02_02-10-27PM.log

5 The following message is displayed for a successful installation: **The installation of Oracle 9iR2 Patch Set was successful.**

6 Click **Exit** to close the installation window.

Installing Oracle Database Instance

This section explains how to install the Oracle database instance.

To install the Oracle Database instance:

1. Open a DOS command prompt.
2. Run the **Inst-DBCA.bat** command, located in **d:\SilentOra**.
3. Open the following log:
   <Oracle-dir>\admin\<SID>\create\PostDBCreation.log
4. Make sure the following output appears:
   Database opened

Starting the Oracle Listener

You must start the Oracle Listener before you check the installation.

To start the Oracle Listener:

1. Open a DOS window.
2. Go to the **<Oracle directory>\<home>\bin** directory.
3. Enter **lsnrctl start** and press **Enter**.

Checking the Installation

This section explains how to check the installation.

To check the installation:

1. Make sure that the following services have been created:
   - **OracleService<SID>**
   - **Oracle<Name>TNSListener**
2. Open a DOS command prompt and issue the following command to check whether the database is available: **sqlplus system/manager@SID**.

Performing Oracle 9i Regular Installation

You install the Oracle database in a regular way as a preparation for Mercury Application Mapping server installation as follows:

1. Launch Oracle Universal Installer interactively from the Oracle 9.2.0.1 installation CD-ROM to install both the software and the database instance.
2. Install the Oracle 9.2.0.6 patch set.

This section includes:

- "Launching Oracle Universal Installer" on page 92
- "Configuring the Oracle Services" on page 104
- "Changing Oracle SYS and SYSTEM Passwords" on page 105
- "Installing the Oracle 9.2.0.6 Patch Set" on page 106

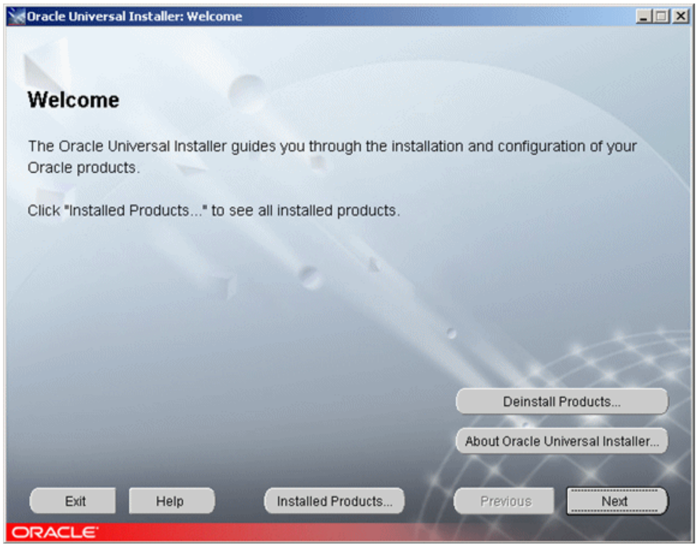
Launching Oracle Universal Installer

Run the Oracle Universal Installer as explained in this section.

To launch Oracle Universal Installer:

1 Launch the Oracle Universal Installer using the **setup.exe** file located in Oracle 9.2.0.1 installation CD-ROM 1.

The Oracle installation begins and displays the first installation page.



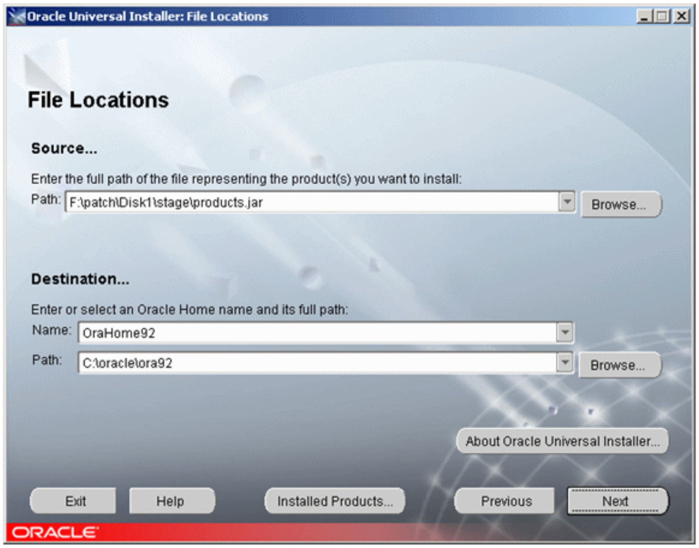
Oracle Universal Installer: Welcome

Welcome

The Oracle Universal Installer guides you through the installation and configuration of your Oracle products.

Click "Installed Products..." to see all installed products.

2 Click **Next** to display the File Locations page.



Oracle Universal Installer: File Locations

File Locations

Source...

Enter the full path of the file representing the product(s) you want to install:

Path: F:\patch\Disk1\stage\products.jar

Destination...

Enter or select an Oracle Home name and its full path:

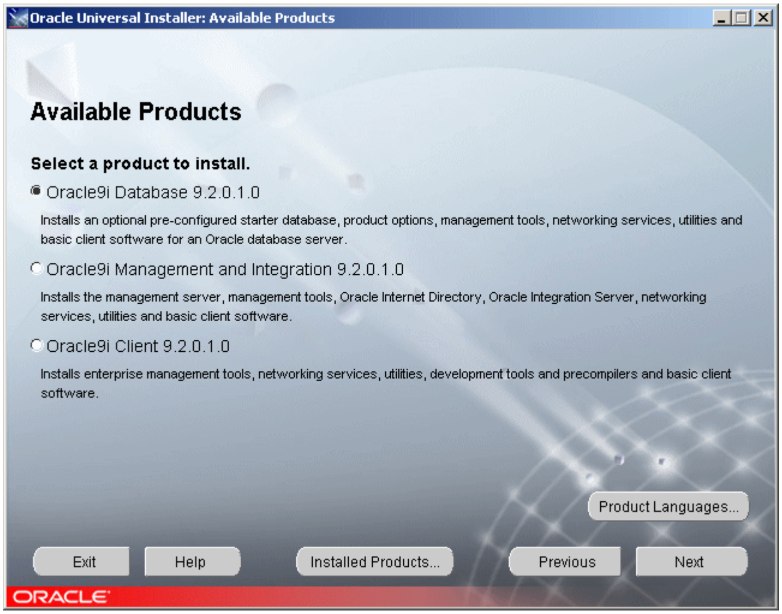
Name: OraHome92

Path: C:\oracle\ora92

See the following table for a description of the fields.

Field	Description
Path (under **Source**)	The location of the file that contains the names of products you need to install. Choose an Oracle root directory and keep the default file path that is given.
Name (under **Destination**)	The logical name of the Oracle installation. Keep the default value that is given.
Path (under **Destination**)	The location of the Oracle installation directory. Choose a default directory.

3 Click **Next** to open the Available Products page.



Oracle Universal Installer: Available Products

Available Products

Select a product to install.

● Oracle9i Database 9.2.0.1.0

Installs an optional pre-configured starter database, product options, management tools, networking services, utilities and basic client software for an Oracle database server.

○ Oracle9i Management and Integration 9.2.0.1.0

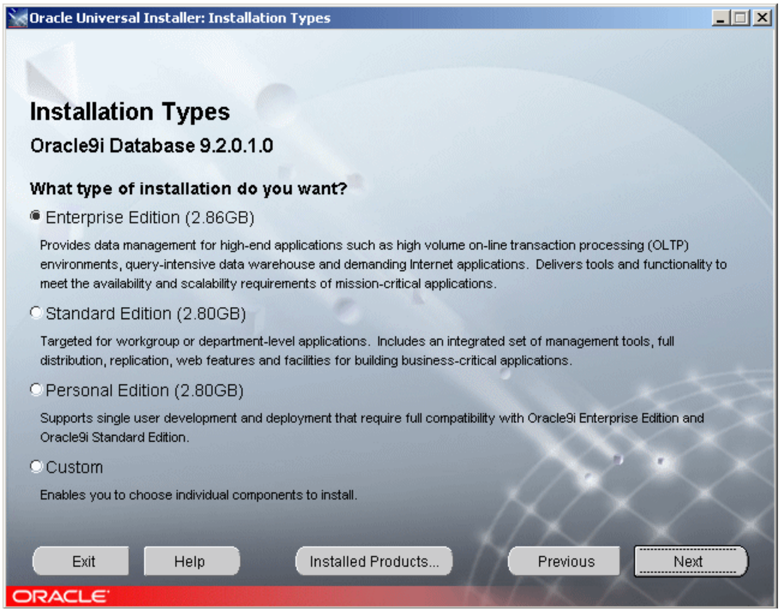
Installs the management server, management tools, Oracle Internet Directory, Oracle Integration Server, networking services, utilities and basic client software.

○ Oracle9i Client 9.2.0.1.0

Installs enterprise management tools, networking services, utilities, development tools and precompilers and basic client software.

Select **Oracle9i Database 9.2.0.1.0**.

4 Click **Next** to open the Installation Types page.



Installation Types

Oracle9i Database 9.2.0.1.0

What type of installation do you want?

● Enterprise Edition (2.86GB)

Provides data management for high-end applications such as high volume on-line transaction processing (OLTP) environments, query-intensive data warehouse and demanding Internet applications. Delivers tools and functionality to meet the availability and scalability requirements of mission-critical applications.

○ Standard Edition (2.80GB)

Targeted for workgroup or department-level applications. Includes an integrated set of management tools, full distribution, replication, web features and facilities for building business-critical applications.

○ Personal Edition (2.80GB)

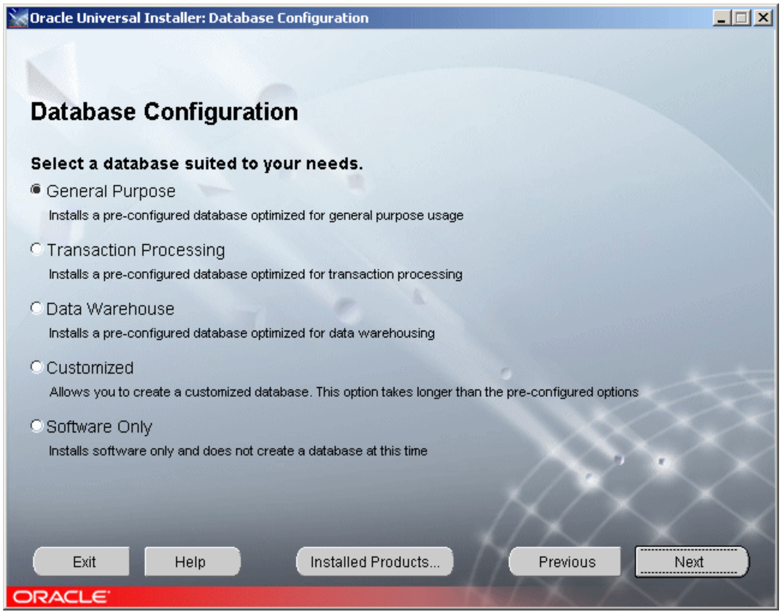
Supports single user development and deployment that require full compatibility with Oracle9i Enterprise Edition and Oracle9i Standard Edition.

○ Custom

Enables you to choose individual components to install.

Select **Enterprise Edition**.

5 Click **Next** to open the Database Configuration page.



Oracle Universal Installer: Database Configuration

Database Configuration

Select a database suited to your needs.

● General Purpose
Installs a pre-configured database optimized for general purpose usage

○ Transaction Processing
Installs a pre-configured database optimized for transaction processing

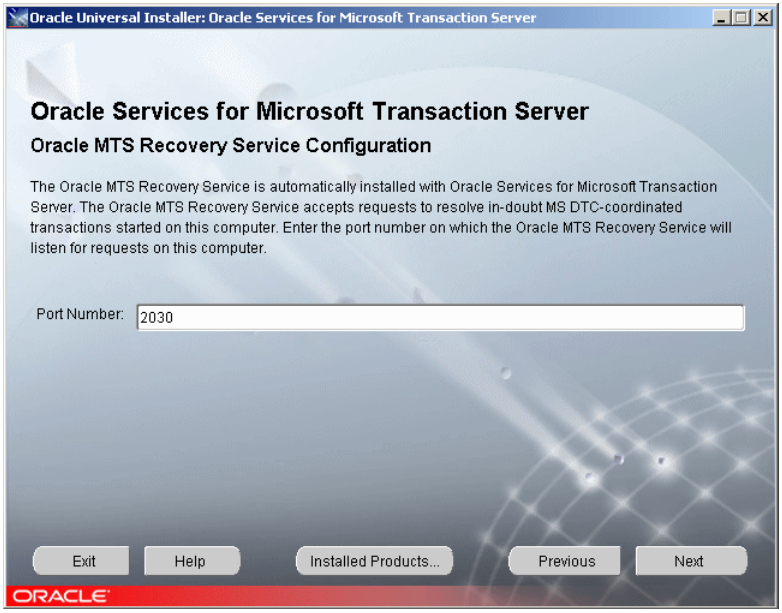
○ Data Warehouse
Installs a pre-configured database optimized for data warehousing

○ Customized
Allows you to create a customized database. This option takes longer than the pre-configured options

○ Software Only
Installs software only and does not create a database at this time

Select **General Purpose**.

6 Click **Next** to open the Oracle Services for Microsoft Transaction page.



Oracle Universal Installer: Oracle Services for Microsoft Transaction Server

Oracle Services for Microsoft Transaction Server

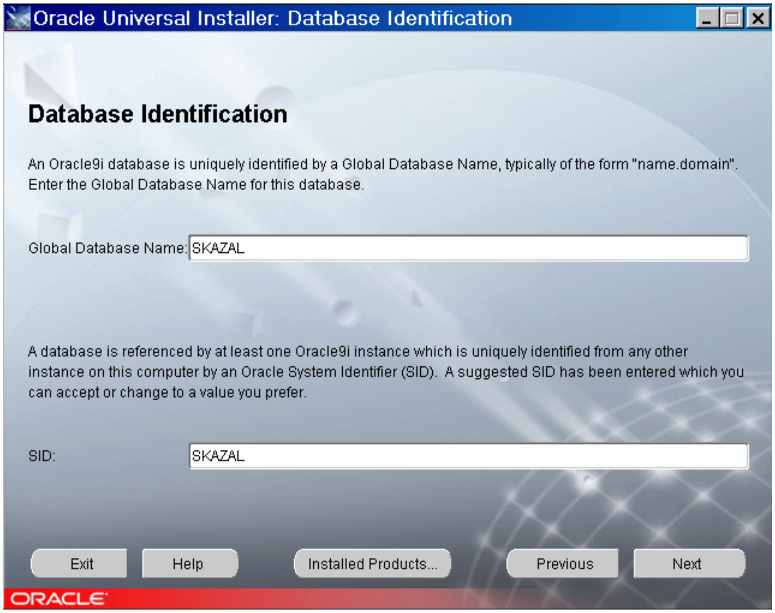
Oracle MTS Recovery Service Configuration

The Oracle MTS Recovery Service is automatically installed with Oracle Services for Microsoft Transaction Server. The Oracle MTS Recovery Service accepts requests to resolve in-doubt MS DTC-coordinated transactions started on this computer. Enter the port number on which the Oracle MTS Recovery Service will listen for requests on this computer.

Port Number: 2030

Enter the port number on which the Oracle MTS Recovery Service will listen for requests on this computer.

7 Click **Next** to open the Database Identification page.



Oracle Universal Installer: Database Identification

Database Identification

An Oracle9i database is uniquely identified by a Global Database Name, typically of the form "name.domain". Enter the Global Database Name for this database.

Global Database Name: SKAZAL

A database is referenced by at least one Oracle9i instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID). A suggested SID has been entered which you can accept or change to a value you prefer.

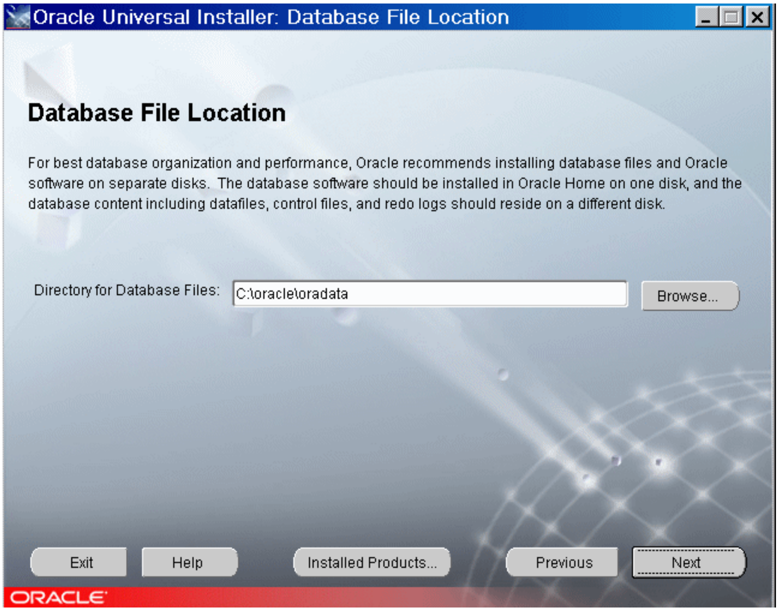
SID: SKAZAL

➤ In the **Global Database Name** box, enter the Global Database name for the Oracle database.

➤ In the **SID** box, enter the database SID to which you want Mercury Application Mapping to connect.

Note: Both Database Identification names on this installation page must be identical. In addition, the Database Identification names used must be identical to the Oracle SID you used when installing Mercury Application Mapping.

8 Click **Next** to open the Database File Location page.



Oracle Universal Installer: Database File Location

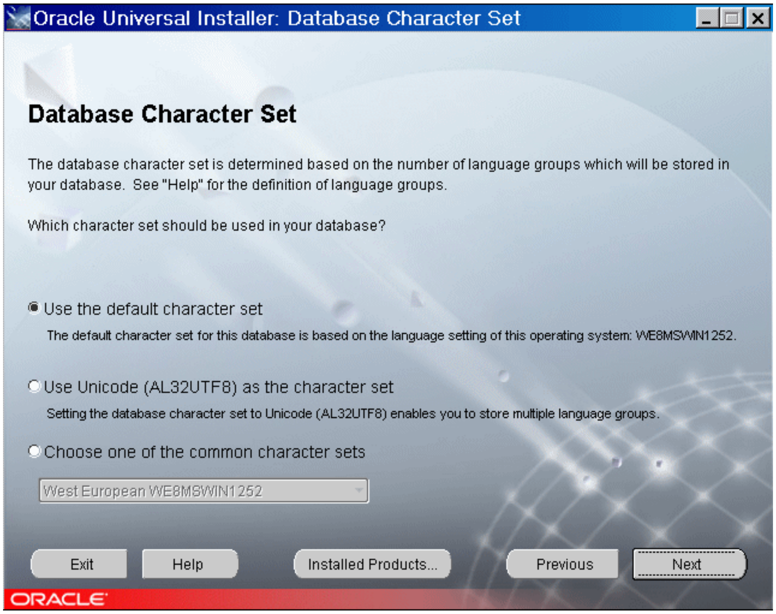
Database File Location

For best database organization and performance, Oracle recommends installing database files and Oracle software on separate disks. The database software should be installed in Oracle Home on one disk, and the database content including datafiles, control files, and redo logs should reside on a different disk.

Directory for Database Files: C:\oracle\oradata

In the **Directory for Database Files** box, specify where the Oracle database is to be installed.

9 Click **Next** to open the Database Character Set page.



Oracle Universal Installer: Database Character Set

Database Character Set

The database character set is determined based on the number of language groups which will be stored in your database. See "Help" for the definition of language groups.

Which character set should be used in your database?

● Use the default character set

The default character set for this database is based on the language setting of this operating system: WE8MSWIN1252.

○ Use Unicode (AL32UTF8) as the character set

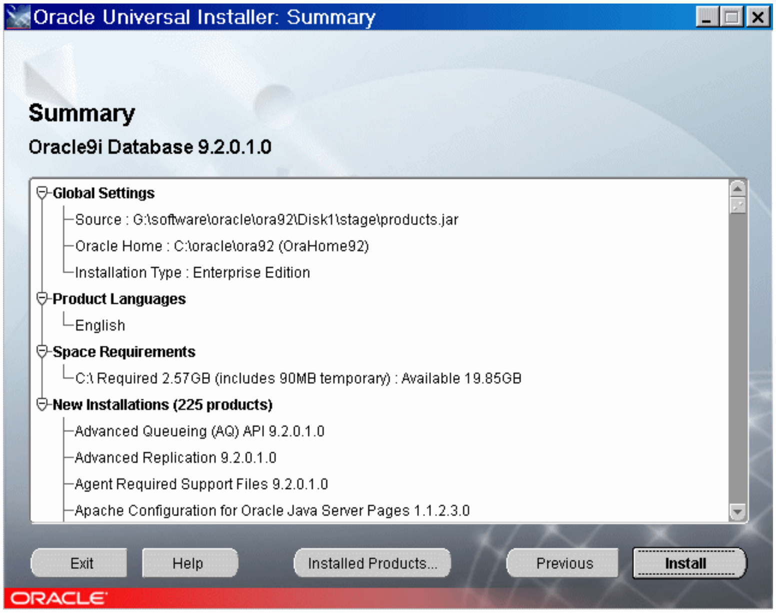
Setting the database character set to Unicode (AL32UTF8) enables you to store multiple language groups.

○ Choose one of the common character sets

West European WE8MSWIN1252

Select **Use the default character set.**

10 Click **Next** to open the Summary page listing the installation options that you have selected.



Oracle Universal Installer: Summary

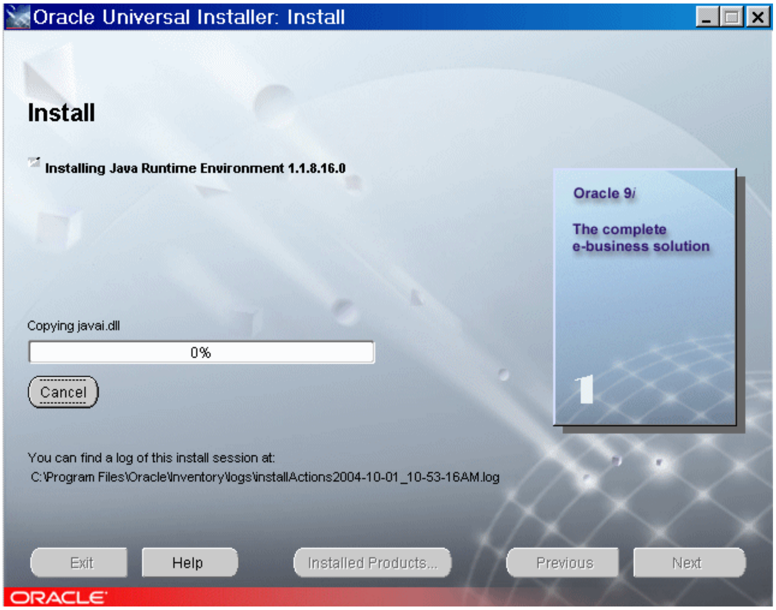
Summary

Oracle9i Database 9.2.0.1.0

```
Global Settings
  Source : G:\software\oracle\ora92\Disk1\stage\products.jar
  Oracle Home : C:\oracle\ora92 (OraHome92)
  Installation Type : Enterprise Edition
Product Languages
  English
Space Requirements
  C:\ Required 2.57GB (includes 90MB temporary) : Available 19.85GB
New Installations (225 products)
  Advanced Queueing (AQ) API 9.2.0.1.0
  Advanced Replication 9.2.0.1.0
  Agent Required Support Files 9.2.0.1.0
  Apache Configuration for Oracle Java Server Pages 1.1.2.3.0
```

Click **Install**.

11 The Install page indicates that the installation procedure is currently being performed.



Oracle Universal Installer: Install

Install

Installing Java Runtime Environment 1.1.8.16.0

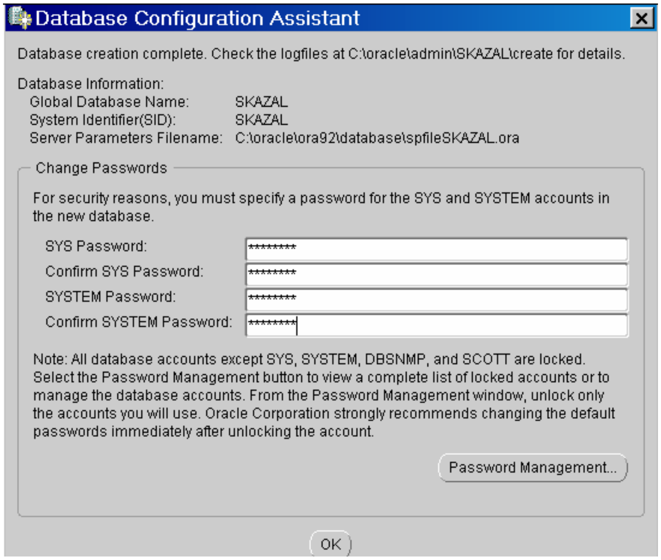
Copying javai.dll

0%

You can find a log of this install session at:
C:\Program Files\Oracle\Inventory\logs\installActions2004-10-01_10-53-16AM.log

Note: The installation process takes about thirty minutes.

12 When installation is complete, click **Next** to specify a password for the SYS and SYSTEM accounts in the new database.



Database Configuration Assistant

Database creation complete. Check the logfiles at C:\oracle\admin\SKAZAL\create for details.

Database Information:
Global Database Name: SKAZAL
System Identifier(SID): SKAZAL
Server Parameters Filename: C:\oracle\ora92\database\spfileSKAZAL.ora

Change Passwords

For security reasons, you must specify a password for the SYS and SYSTEM accounts in the new database.

SYS Password: *********
Confirm SYS Password: *********
SYSTEM Password: *********
Confirm SYSTEM Password: *********

Note: All database accounts except SYS, SYSTEM, DBSNMP, and SCOTT are locked. Select the Password Management button to view a complete list of locked accounts or to manage the database accounts. From the Password Management window, unlock only the accounts you will use. Oracle Corporation strongly recommends changing the default passwords immediately after unlocking the account.

13 Enter the SYSTEM password: <**new_system_password**> in the following fields:

➤ **SYSTEM Password**

➤ **Confirm SYSTEM password**

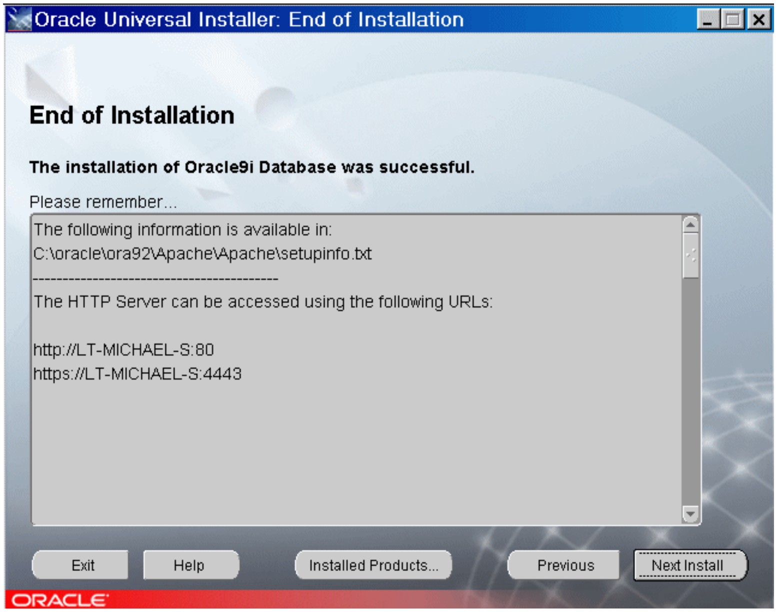
14 Enter the SYS password: <**new_sys_password**> in the following fields:

➤ **SYS Password**

➤ **Confirm SYS Password**

Note: It is strongly recommended to change the default password for Oracle SYS and SYSTEM super users. When the installation is complete, click **Next** to specify non-default passwords for the SYS and SYSTEM accounts in the new database. Remember to replace the default passwords when installing Mercury Application Mapping in the appropriate screens. For details, see "Changing Oracle SYS and SYSTEM Passwords" on page 105.

15 Click **OK**. The End of Installation page opens informing you that the installation was successful.



End of Installation

The installation of Oracle9i Database was successful.

Please remember...

```
The following information is available in:
C:\oracle\ora92\Apache\Apache\setupinfo.txt
-------------------------------------------------
The HTTP Server can be accessed using the following URLs:

http://LT-MICHAEL-S:80
https://LT-MICHAEL-S:4443
```

16 Click **Exit** to close the Oracle Universal Installer.

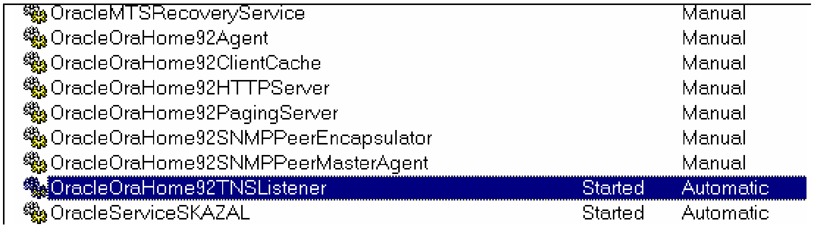
Configuring the Oracle Services

After you have finished installing the Oracle Database, you configure the Oracle services.

To configure the Oracle services:

1. Click **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Services.**
2. Set **OracleOraHome92TNSListener** to **Automatic.**
3. Set **OracleServicesSKAZAL** to **Automatic.**

4 Set all the other Oracle services to **Manual**.



OracleMTSRecoveryService		Manual
OracleOraHome92Agent		Manual
OracleOraHome92ClientCache		Manual
OracleOraHome92HTTPServer		Manual
OracleOraHome92PagingServer		Manual
OracleOraHome92SNMPPeerEncapsulator		Manual
OracleOraHome92SNMPPeerMasterAgent		Manual
OracleOraHome92TNSListener	Started	Automatic
OracleServiceSKAZAL	Started	Automatic

5 Restart the machine.

6 Open **Services** again to verify that **OracleOraHome92TNSListener** and **OracleServicesSKAZAL** are still set to automatic.

Changing Oracle SYS and SYSTEM Passwords

It is strongly recommended to change the default passwords for Oracle SYS and SYSTEM super users. If you did not change the default SYS and SYSTEM passwords during setup, you can do it manually as explained in this section.

To change Oracle SYS and SYSTEM Passwords:

1 Select **Start** > **Programs** > **Accessories** > **Command Prompt** to open a Command Prompt window.

2 From the command line:

- type **sqlplus system/<system_current_password>@<DB_alias>** and press **Enter**
- type **alter user system identified by <new_password_system>;** and press **Enter**
- type **alter user sys identified by <new_password_sys>;** and press **Enter** where:
  - **<system_current_password>** is the latest password for the SYSTEM user; the installation default password is **MANAGER**.
  - **<DB_alias>** is the identifier used to connect to an Oracle Server, as per the entry in <**ORACLE_home_directory**>**network****tnsnames.ora**. When using silent setup, the db_alias is identical to the SID.
  - **<new_password_system>** is the password to use for SYSTEM.
  - **<new_password_sys>** is the password to use for SYS.

Installing the Oracle 9.2.0.6 Patch Set

The next step is to install the Oracle 9.2.0.6 Patch Set. For details, see "Upgrading an Existing Database to Oracle 9.2.0.6" on page 106.

Note: The suggested procedure is the basic typical installation. Make sure all the parameters and files are configured according to the provided guidelines. For details, see "Oracle Server Configuration Guidelines" on page 137.

Upgrading an Existing Database to Oracle 9.2.0.6

If you used a Mercury Application Mapping database with Oracle 9.2.0.1 or Oracle 9.2.0.4, you must upgrade the database to the patch set 9.2.0.6.

To upgrade an existing database to Oracle 9.2.0.6:

1. It is strongly advised to commit any unsaved database transactions before upgrading the database.
2. Close the database instance.
3. Stop the Oracle instance service and the Oracle Listener service. A message may be issued to request that you stop the Windows services specified in the message.
4. Follow the steps in "Installing the Oracle Patch Set" on page 87.
5. Re-start both the Oracle instance service and the Oracle Listener service.
6. Update the database catalog tables with the new patch details. This is done by running the **run_patch_upgrade.bat** batch file.

   ➤ Copy **patch_catalog.sql** and **run_patch_upgrade.bat** from the **dbTools\9iSilentOra** folder on the Mercury Application Mapping 6.5 Setup Windows CD-ROM, to your local drive.

   ➤ Open a DOS command prompt and run the following command from the directory into which you copied the **run_patch_upgrade.bat** file:

   run_patch_upgrade <sys_password> <db_alias>

where <**sys_password**> is the SYS user password (of the Oracle SYS user) and <**db_alias**> is the database identifier as specified in **tnsnames.ora** in the following location:
<Oracle_home_directory>/network/admin

The database catalog update can take a few minutes.

The database is now ready.

10

Manually Creating the Oracle Server Database Schemas

This chapter describes the manual creation of the Oracle Server database schemas required by Mercury Application Mapping.

Overview of the Oracle Database Schemas

Mercury Application Mapping uses the following database schemas for its persistency:

CMDB. The configuration management database.

CMDB History. The configuration management history database.

During the Mercury Application Mapping setup, new schemas and/or tablespaces can be set up automatically by the setup procedure, or already existing schemas and tablespaces can be used. Existing schemas and tablespaces can either have been created manually in advance, or can have been created by a previous installation of the same release of Mercury Application Mapping.

If you choose to have the schemas and/or tablespaces set up automatically by the setup procedure, you must have administrator permissions. Usually, the System account credentials are used. If you are unable to submit database administrator connection parameters (for example, due to organization security restrictions), manually set up the schemas in advance.

For details about installing the Mercury Application Mapping server, see "Introduction to Installation" in *Mercury Application Mapping Installation Guide*.

Prerequisites

Before creating the database schemas, the Oracle Database Server must be installed according to the details in this guide. For further details on installing and configuring the Oracle Database Server, see Chapter 9, "Installing Oracle Database on a Windows Platform," and Chapter 12, "Oracle Server Configuration Guidelines."

Schemas should be manually created prior to running the Mercury Application Mapping setup. During the setup procedure, you select the option to use existing schemas and enter the relevant data about the existing schemas for which you are prompted. The data you enter is written to the configuration files that are used by the JDBC driver to connect to the existing schemas. You can leave the Administrator password with a wrong value (empty values are not permitted) to avoid exposing it.

Note: In Mercury Application Mapping 6.5, a connectivity check using the Administrator credentials is performed against both the CMDB and CMDB History schemas. If an incorrect password has been entered, the check will fail and you must click **Ignore** to proceed with the setup.

Creating Schema Default and Temporary Tablespaces

The database administrator should create at least one Temporary tablespace in the database instance hosting Mercury Application Mapping schemas, and at least one tablespace to be served as the default tablespace for the Mercury Application Mapping schemas. For further details on tablespace file sizes and recommended settings, see "Oracle File Sizing" on page 140.

Mercury Application Mapping setup supports up to two tablespaces per schema (data and index); however, it is possible to specify one tablespace for both schemas, or one tablespace for each schema to hold both tables and indexes data. The tablespace specification should be made according to your storage policy, taking into account RAID configuration, available disks and storage devices.

Creating Schema User Permissions

The database administrator should create an Oracle schema user with the database permissions required by the Mercury Application Mapping application.

The following database permissions must be granted to Mercury Application Mapping Oracle schema users. These permissions are also used by the Mercury Application Mapping installer to create a new Oracle user:

- Roles
  - Connect
  - Resource

- Permissions
  - CREATE ANY INDEX
  - CREATE ANY TABLE
  - CREATE ANY VIEW
  - CREATE ANY SEQUENCE
  - CREATE ANY TRIGGER
  - CREATE ANY PROCEDURE
  - CREATE DATABASE LINK

Note: Mercury Application Mapping supports any user with higher permissions. For Mercury Application Mapping certification, use an Oracle user that has the exact Oracle permissions described above.

Creating Schemas and Schema Objects

The database administrator should create two schemas; one for CMDB persistency and one for CMDB historical data persistency (CMDB History).

Objects are created using SQL scripts which are run against the schemas that were previously created. For details, see "Creating Schema Default and Temporary Tablespaces" on page 111.

The scripts are located in the **<Mercury Application Mapping root directory>\j2f\CMDB\dbscripts\oracle** directory located on the Mercury Application Mapping server, and should be run from this directory. Detailed descriptions of the prerequisites and operation method of each script can be found in the **readme.txt** file in this directory.

The SQL scripts require an Oracle 10g or 9i client with SQLPLUS client tool. If these are not installed on the Mercury Application Mapping server machine, you can move all the scripts to a different machine that has the required client.

To move the scripts to a different machine:

1. Copy the <**Mercury Application Mapping root directory**>**\j2f\CMDB\dbscripts\oracle** directory from the Mercury Application Mapping server machine to the machine with the required client.
2. Change the **SCRIPTS_LOCATION** parameter in the **env_cmdb.bat** file, located in the scripts directory.

This section includes the following topics:

- "Creating the CMDB Objects" on page 113
- "Creating the CMDB History Objects" on page 113

Creating the CMDB Objects

To create the CMDB objects, connect to the CMDB schema and run the following SQL scripts:

- create_cm_tables_cmdb.sql
- common_ora_dbobjects_create.sql

Creating the CMDB History Objects

To create the CMDB History objects, connect to the CMDB History schema and run the following SQL scripts:

- create_history_tables_cmdb.sql
- common_ora_dbobjects_create.sql
- mam_ora_set_system_props.sql

Shared CMDB Setup

In a shared CMDB Mercury Application Mapping installation, only the CMDB History database is created, as an existing CMDB database of a Mercury Business Availability Center system is used by Mercury Application Mapping.

To manually create the CMDB History database, refer to relevant parts in “Creating Schema Default and Temporary Tablespaces” on page 111 and “Creating Schemas and Schema Objects” on page 112.

During a shared CMDB installation, the Mercury Application Mapping server is registered in the Mercury Business Availability Center management database tables via SQL scripts. If an existing CMDB History database is used instead of creating a new one, the installer will skip this registration step, so it must be carried out manually prior to the installation.

For the location of the script, and Oracle client requirements, see “Creating Schemas and Schema Objects” on page 112.

Manually Registering a Shared CMDB

This section explains how to manually register a shared CMDB

To manually register a shared CMDB:

1 Connect to the Mercury Business Availability Center management schema user.

2 Run the **register_ora_mamserver.sql** script with the <**MAM Server**> parameter. For example:

```
register_ora_mamserver.sql <MAM Server>
```

<MAM Server> is the name of the host on which the Mercury Application Mapping server is installed. Do not use the IP address of the host.

11

Maintaining an Oracle Server Database

This chapter describes the various maintenance and tuning procedures that are recommended for Mercury Application Mapping databases created on Oracle Servers, as well as the available database backup and recovery methods.

Oracle Server Deployment

You can set up CMDB databases on an Oracle Server.

To deploy Oracle Server for use with Mercury Application Mapping, see Chapter 9, “Installing Oracle Database on a Windows Platform.”

To install Mercury Application Mapping schemas against the Oracle server, see “Single Machine (Typical) Deployment” in *Mercury Application Mapping Installation Guide*.

This section includes the following topics:

- “Hardware Requirements” on page 116
- “Oracle Server Requirements” on page 116
- “Oracle Client Requirements” on page 117

Hardware Requirements

For Mercury Application Mapping hardware sizing guidelines, see Chapter 12, “Oracle Server Configuration Guidelines.”

For Oracle hardware requirements, refer to the installation guide for your specific Oracle platform. Additional information is also available in the Oracle software distribution media as well as the online Oracle documentation. For Oracle documentation, refer to: http://otn.oracle.com/documentation/index.html

Oracle Server Requirements

Mercury Application Mapping supports both UNIX and Windows database servers. The following table describes the subset of the Oracle Server certification matrix that is supported and certified for working with Mercury Application Mapping.

Component	Supported		Recommended	
	Version/Edition	**Service Pack**	**Version/Edition**	**Service Pack**
Windows Operating System	Windows 2000 Server/Advanced Server	Service Pack 4	Windows 2003 Server standard / enterprise	Service Pack 1
Sun Solaris Operating System	Solaris 9		Solaris 8, 10	
Oracle	Oracle 9.2.0.6		Oracle 10.2.0.1	

Note:

- Solaris environments are 64-bit only.
- Oracle environments for Oracle 9i are 32-bit on all platforms and for Oracle 10g, 32-bit for windows and 64-bit for UNIX.

For information on Oracle Client software requirements, see "Oracle Client Versions and Operating System Platforms" on page 104.

Oracle Instances

You can install more than one Oracle instance on a machine, using the same Oracle database engine.

For Mercury Application Mapping certification, do not use more than one Oracle instance. If you do use more than one instance for the Mercury Application Mapping databases, ensure that all the instances are configured as described in this document and that they all have the same characteristics (such as the same character set).

Oracle Client Requirements

The following table describes the Oracle Client versions and operating system platforms that are supported and certified for working with Mercury Application Mapping.

Component	Supported		Recommended	
	Version/Edition	Service Pack	Version/Edition	Service Pack
Windows Operating System	Windows 2000 Server/Advanced Server	Service Pack 4	Windows 2003 Server standard / enterprise	Service Pack 1
Sun Solaris Operating System	Solaris 9		Solaris 8, 10	
Oracle	Oracle 9.2.0.6		Oracle 10.2.0.1	

Database Maintenance and Tuning

Poor database performance can be caused by the faulty configuration of the instance and database, or by abnormal resource consumption of an Oracle transaction, user, or process. It is essential for the database administrator to proactively monitor resource consumption, and correct any abnormalities before performance is affected.

Note: Memory, CPU, and I/O are the three most common system resources consumed by Oracle.

There are a number of third-party tools that you can use to monitor database behavior and assist you in identifying bottlenecks in your system. Use the following guidelines to help you.

This section includes the following topics:

- System Global Area (SGA)
- Database Load Behavior
- CPU and I/O
- Oracle Alert File
- Archive Log – File System
- Tablespace Storage Space
- Collecting Statistics for CMDB
- Statistics Collection Guidelines for Oracle 9i
- Statistics Collection Guidelines for Oracle 10g
- CMDB Index Fragmentation

System Global Area (SGA)

Always configure your SGA to fit physical memory and avoid using swapping. It is recommended that you not set the SGA for more than 70 percent of system physical memory, leaving enough memory for additional system and client processes.

Database Load Behavior

Run **ustlbstat/utlestat** (or STATSPACK) regularly to monitor the database behavior. In Oracle10g, an AWR snapshot is created regularly (every hour by default) and reserved for 7 days. An AWR report can be generated and used in the same way as a STATSPACK report. For additional information on running and interpreting the output you receive, refer to *Oracle Metalink Note 62161.1: BSTAT/ESTAT* (http://metalink.oracle.com/metalink/plsql/showdoc?db=Not&id=62161.1) or *Oracle Metalink Note 94224.1: STATSPACK FAQ* (http://metalink.oracle.com/metalink/plsql/showdoc?db=Not&id=94224.1).

It is also recommended that you monitor I/O load on the system to identify I/O contention. Once you determine which disk is most loaded, you can use the utlbstat/utlestat output to determine which particular Oracle data file is the cause of the contention and consider the relocation of data files.

CPU and I/O

It is recommended that you monitor the CPU and file system, which are the main resources consumed by the database server. CPU usage should not exceed 70 percent and the I/O wait should not be higher than 10 percent.

You can use **perfmon** on Windows, or **top** in UNIX, to monitor the above resources.

Oracle Alert File

Oracle registers abnormal events in the **alert.log** file, whose location is defined by the BACKGROUND_DUMP_DEST parameter.

It is recommended that you check this file regularly to identify abnormalities that should be corrected, for example, ORA-XXXXX errors.

Archive Log – File System

When using the archivelog mode, monitor your ARCHIVE_DUMP_DEST location for disk usage. These files should be backed up and deleted regularly to leave sufficient disk space for new archive files.

The archive file is usually the same size as the redo log file. To determine the size of a redo log file, use the operating system command or the following query:

```
SQL> select GROUP#, BYTES
        from V$LOG;
```

To determine the number of archive files generated over a period of time, for example, a day, you can use the following query after the system is stable:

```
SQL> alter session set NLS_DATE_FORMAT = 'DD-MON-YYYY';
SQL> select TO_DATE(TO_CHAR(FIRST_TIME,'DD-MON-YYYY')) as “Day”,
        COUNT(*) as "Number of files"
        from V$LOG_HISTORY
        group by TO_CHAR(FIRST_TIME,'DD-MON-YYYY')
        order by 1 asc;
```

Tablespace Storage Space

To avoid space errors caused by data growth, monitor your tablespace usage regularly.

If you run out of space in one of your tablespaces, you can add one or more data files to it by using the ALTER TABLESPACE <tablespace name> ADD DATAFILE… command.

Collecting Statistics for CMDB

The Mercury Application Mapping platform is planned and built to work with the Oracle Cost Base Optimizer. For the Optimizer to work properly, you must periodically collect statistics for all schema tables.

Once your Mercury Application Mapping system is stable, statistics should be collected at least once a day during maintenance hours from the CMDB and CMDB History schemas.

Note: Collecting statistics is a resource-consuming operation that can take a long time. It is therefore recommended that you collect statistics during special maintenance hours.

To obtain the list of tables, use the SELECT TABLE_NAME FROM USER_TABLES query.

To obtain the list of schema indexes, use the SELECT INDEX_NAME FROM USER_INDEXES query.

Unlike some databases, where queries are predefined and can be tuned according to the expected database size, CMDB database constructs queries dynamically, according to Pattern Views defined against its data model. This necessitates accurate statistics at all times. In addition to running a daily job to update statistics for CMDB, it is recommended that you manually refresh statistics if major changes to the CMDB schema objects have occurred, usually caused by bulk insert transactions.

The following scenario **Automated Discovery tasks**, warrants a manual refresh of CMDB statistics: Discovery Manager is the process responsible for automatically detecting configuration items (CIs) and inserting them into CMDB.

For details about the guidelines for the collection of statistics for Oracle 9i, see “Statistics Collection Guidelines for Oracle 9i” on page 122.

For details about the guidelines for the collection of statistics for Oracle 9i, see “Statistics Collection Guidelines for Oracle 10g” on page 125.

Statistics Collection Guidelines for Oracle 9i

The following guidelines are applicable to Oracle 9i.

Running a Daily Job

To create a daily job to run statistics on CMDB schemas (both CMDB and CMDB history) use the **create_statistics_job.bat** script located in the **<Mercury Application Mapping server name>\J2F directory\CMDB\dbscripts\oracle** directory. For details, see "Mercury Application Mapping Database Script" on page 149.

The script is run with the following command:

```
create_statistics_job.bat <schema> <password> <db alias> <hour>
```

where:

- **schema.** The name of the CMDB schema user of the schema for which you are installing the statistics job.
- **password. T**he password for the database schema user.
- **db alias.** The database alias for connecting to the target database as specified in the **tnsnames.ora** Oracle client file. Ensure that there is an entry in the tnsnames.ora file for the target server.
- **hour.** The hour of the day to run the statistics job. Accepted values are 0-23 with a default value of midnight.

Note: If you prefer not to use the provided script to generate an automatic, daily job you can execute the following, manual command:

```
Begin DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '<name of Oracle schema>', cascade => TRUE) ; end;
```

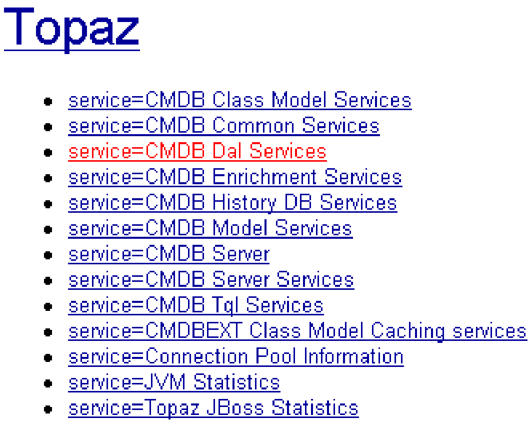
This API gathers statistics on all tables and their related indexes, even if statistics are the same as the previous call.

The job should be scheduled for a maintenance hour when the system is not heavily loaded (for example, every night at midnight).

Running a Manual Statistics Refresh

When major data changes have been carried out on CMDB, you refresh the statistics by one of the following methods:

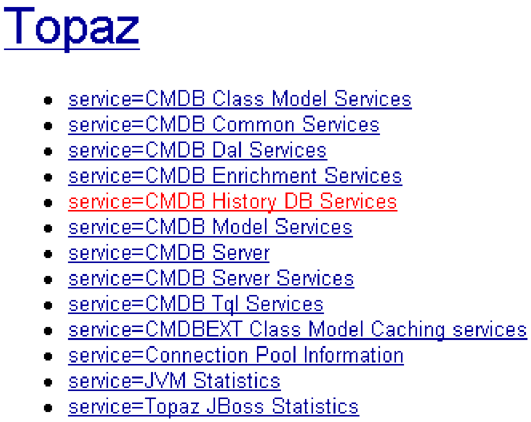
- Use the **runStatistics** JMX:
  - In a Web browser, open **http://<Mercury Application Mapping server name>:8080/jmx-console**.
  - In the **Topaz** section, select **CMDB Dal Services** to refresh the CMDB schema.



Topaz

- service=CMDB Class Model Services
- service=CMDB Common Services
- service=CMDB Dal Services
- service=CMDB Enrichment Services
- service=CMDB History DB Services
- service=CMDB Model Services
- service=CMDB Server
- service=CMDB Server Services
- service=CMDB Tql Services
- service=CMDBEXT Class Model Caching services
- service=Connection Pool Information
- service=JVM Statistics
- service=Topaz JBoss Statistics

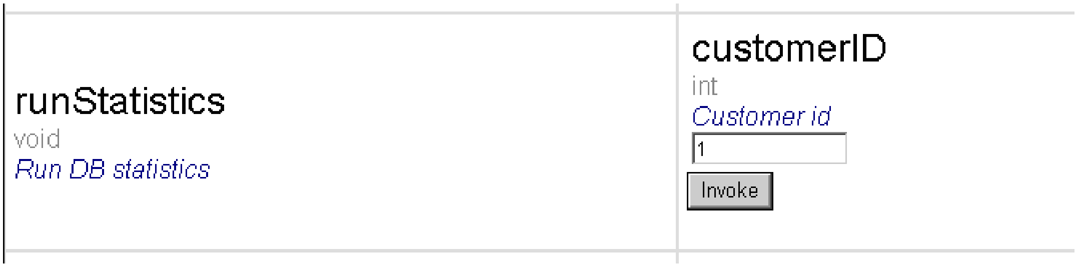
To refresh the CMDB History schema, use the **CMDB History DB Services:**



Topaz

- service=CMDB Class Model Services
- service=CMDB Common Services
- service=CMDB Dal Services
- service=CMDB Enrichment Services
- service=CMDB History DB Services
- service=CMDB Model Services
- service=CMDB Server
- service=CMDB Server Services
- service=CMDB Tql Services
- service=CMDBEXT Class Model Caching services
- service=Connection Pool Information
- service=JVM Statistics
- service=Topaz JBoss Statistics

- Under **runStatistics**, enter the customer ID number in the **customerID** box. The default customer ID for an individual Mercury Application Mapping system (that is, one not managed by Mercury Managed Services) is 1:



runStatistics
void
Run DB statistics

customerID
int
Customer id
1

- click **Invoke**. The CMDB statistics are regenerated.

- Manually run the daily job using an Oracle client:
  - Connect to the CMDB schema via SQLPLUS.
  - Run the query **Select job from user_jobs where upper(what) like '%GATHER_SCHEMA_STATS%';**.

    The query output is the job number.
  - Run the job with the call **Exec dbms_job.run (<job number>) ;**.
- Connect to the CMDB schema and execute the PL/SQL block **begin DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '<name of Oracle schema>' , cascade => TRUE) ; end;**.

Note: If major data changes have taken place, you must also refresh the CMDB History schema, using one of the manual statistics refresh methods.

Statistics Collection Guidelines for Oracle 10g

The following guidelines are applicable to Oracle 10g.

Running a Daily Job

From Oracle 10g and later, there is no need to define a dedicated daily job against the CMDB schema as Oracle 10g includes an automated job for statistics collection.

Oracle 10g has an automated job for statistics collection of all database schemas as part of using 10g Scheduler APIs. The automated job is the **GATHER_STATS_JOB** which is owned by the **SYS** super user. The job collects stale (inaccurate) statistics at a predefined time (Maintenance Window). The job only refreshes statistics for objects with empty or stale statistics thereby avoiding scanning unnecessary data as was the case in Oracle 9i.

The Maintenance Window comprises the WEEKNIGHT_WINDOW (with the job starting at 10:00 PM Monday-Friday), and the WEEKEND_WINDOW (with the job starting at 12:00 AM on Saturday), with no job scheduled for Sunday. If you need to collect statistics at other maintenance times that better suit your system, your database administrator can change the schedule using the Oracle Enterprise Manager console. For an overview of the Oracle Scheduler, refer to the chapter "Overview of Scheduler Concepts" in the Oracle Database Administrator's Guide in the Oracle 10g documentation set.

An example of changing the job to a schedule running every night at midnight can be found in the change_gather_stats_job.bat database script. This script should be used cautiously and only by the system administrator.

For details, see "Mercury Application Mapping Database Script" on page 149.

Running an Hourly Job

In Oracle 10g, object changes are automatically monitored and it is possible to operate the **DBMS_STATS.GATHER_SCHEMA_STATS** API with the **GATHER AUTO** method collecting only missing or stale statistics (that is, when statistics no longer represent the object accurately, usually when the object's data change by 10% or more).

If you do not manually refresh statistics for each major data change, it is recommended to schedule a Scheduler job running every hour for refreshing stale statistics against the CMDB schema. You execute an automatic, hourly refresh of statistics by executing the PL/SQL block **begin DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '<name of Oracle schema>', options => 'GATHER AUTO') ; end;**.

Note: From Oracle 10g and on, it is strongly recommended to use Oracle Scheduler job APIs instead of DBMS_JOB APIs for job automation.

The provided database script **create_CMDB_Stats_Job.bat** creates the needed job scheduled to run every hour, it is owned by SYSTEM user and operates against CMDB schema only.

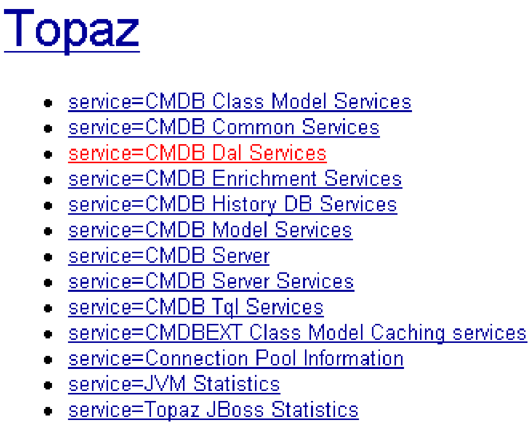
For details, see "Mercury Application Mapping Database Script" on page 149.

Running a Manual Statistics Refresh

When major data changes have been carried out on CMDB, you refresh the statistics by one of the following methods:

- ➤ Use the **runStatistics** JMX:
  - ➤ In a Web browser, open **http://<Mercury Application Mapping server name>:8080/jmx-console**.

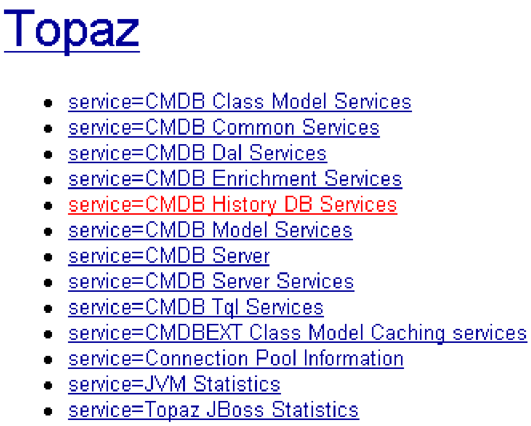
➤ In the **Topaz** section, select **CMDB Dal Services** to refresh the CMDB schema.



Topaz

- service=CMDB Class Model Services
- service=CMDB Common Services
- service=CMDB Dal Services
- service=CMDB Enrichment Services
- service=CMDB History DB Services
- service=CMDB Model Services
- service=CMDB Server
- service=CMDB Server Services
- service=CMDB Tql Services
- service=CMDBEXT Class Model Caching services
- service=Connection Pool Information
- service=JVM Statistics
- service=Topaz JBoss Statistics

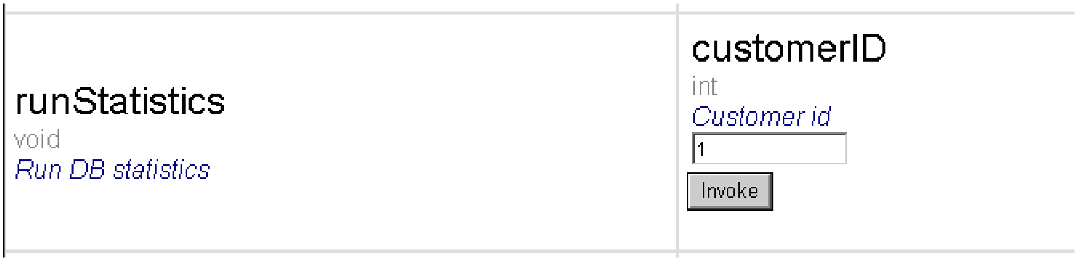
To refresh the CMDB History schema, use the **CMDB History DB Services**:



Topaz

- service=CMDB Class Model Services
- service=CMDB Common Services
- service=CMDB Dal Services
- service=CMDB Enrichment Services
- service=CMDB History DB Services
- service=CMDB Model Services
- service=CMDB Server
- service=CMDB Server Services
- service=CMDB Tql Services
- service=CMDBEXT Class Model Caching services
- service=Connection Pool Information
- service=JVM Statistics
- service=Topaz JBoss Statistics

- ➤ Under **runStatistics**, enter the customer ID number in the **customerID** box. The default customer ID for an individual Mercury Application Mapping system (that is, one not managed by Mercury Managed Services) is 1:



customerID: 1

- ➤ click **Invoke**. The CMDB statistics are regenerated.

Note:

The JMX utility checks the database release and in the case of an Oracle 10g database, runs the statistics in the **GATHER AUTO** method.

If major data changes have taken place, refresh the CMDB History schema as well, using one of the manual statistics refresh methods.

- ➤ Connect to the CMDB schema and execute the PL/SQL block **begin DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '<name of Oracle schema>' , options => GATHER AUTO) ; end;**.

CMDB Index Fragmentation

The CMDB schema consists of Oracle B-tree indexes for enhancing searches on table columns.

It is recommended to validate the structure of the CMDB schema indexes periodically (at least every week for active systems), and if necessary to rebuild the indexes found to be fragmented.

The main reasons for indexes becoming fragmented are:

- **Row deletes.** When rows in a table are deleted, Oracle index nodes are not physically deleted, and neither are the entries removed from the index. Rather, Oracle logically deletes the index entry and leaves dead nodes in the index tree, where they may be reused if another adjacent entry is required. However, when large numbers of adjacent rows are deleted, it is highly unlikely that Oracle will have an opportunity to reuse the deleted leaf rows. In addition to wasting space, large volumes of deleted leaf nodes cause index scans to take more time.

  Over time, following row deletes from schema tables (both explicitly by users, and implicitly by the CMDB engine), there may be a need to rebuild some of the CMDB schema indexes.

- **Index height.** The height of an index refers to the maximum number of levels encountered within the index. As the number of levels in an index increases, more block reads are needed when searching the index. When a large amount of rows are added to a table, Oracle may create additional levels of an index to accommodate the new rows, thereby causing an index to reach four levels, although only in those areas of the index tree where massive inserts have occurred. While Oracle indexes can support many millions of entries in three levels, any Oracle index that has four or more levels can benefit from rebuilding.

  For CMDB tables, it is recommended to rebuild any index that has more than three levels.

Index Maintenance Utility

Mercury Application Mapping's index maintenance utility (**maintain_indexes.bat**) can be used to identify and rebuild indexes that have more than three levels, or that have 100,000 values or more with 10% of deleted values. For details, see "Mercury Application Mapping Database Script" on page 149.

You can set a flag when running the utility to instruct it to rebuild indexes identified as being fragmented automatically, although it is recommended that you rebuild indexes manually.

When run, the utility produces a log file (**index_stats.log**) that contains the following entries:

- An alphabetical list of indexes that were identified as candidates for rebuilding. For each index listed, statistics are shown such as the height of the index and the percentage of deleted rows.
- Rebuild commands for each index listed that can be used to rebuild the indexes manually.

The utility also creates a table called **TEMP_STATS** in the target schema that contains all the indexes and their related statistics (not only the indexes listed as candidates for rebuilding). The table remains in the schema until it is manually dropped to enable inspection of the results at a later stage.

Warning: The index maintenance utility is resource intensive, as it analyzes all indexes in the schema. It can also cause locks on database objects or skip indexes that are locked by other sessions. It is recommended to run the index maintenance utility during maintenance hours only.

To run the index maintenance utility:

1. If you do not have an Oracle client on the Mercury Application Mapping server, copy the following files from the \<**Mercury Application Mapping server>\J2F\CMDB\dbscripts\oracle\utils** directory on a Windows machine that has Oracle database client installed:

   - **maintain_indexes.bat**
   - **maintain_indexes.sql**

   If you have an Oracle client on the Mercury Application Mapping server, run the scripts from the dbscripts directory. For details, see “Mercury Application Mapping Database Script” on page 149.

2. On the machine to which you copied the files, open a DOS command window and move to the location in which you copied the files.

3 Run the index maintenance utility with the following command:

maintain_indexes.bat <schema> <password><db alias> (rebuild flag)

where:

- **schema.** The name of the database schema user of the schema for which you are running the utility.
- **password.** The password for the database schema user.
- **db alias.** The db alias for connecting to the target database as specified in the **tnsnames.ora** file. Ensure that there is an entry in the tnsnames.ora file for the target server.
- **rebuild flag.** The flag to instruct the utility to rebuild indexes automatically. Set the flag to **0** if you don't want the utility to rebuild indexes automatically and to **1** if you do. The default setting is **0**.

When the index maintenance utility has finished running, check the **index_stats.log** file in the directory into which you copied the files in step 1 on page 130 for the list of indexes that are candidates for rebuilding, and the rebuild commands to be used.

Note: The execution time of the index maintenance utility depends on the size of the indexes and the load on the system when being run.

Oracle Database Backup and Recovery

Your backup strategy is tested when a failure occurs and data is lost. Data can be lost or corrupted in several ways, such as a logical application error, an instance failure that prevents Oracle from starting, or a media failure caused by a disk crash. In addition to your scheduled backups, it is important to perform a backup when the database structure changes (for example, when a data file is added to the database), or before you upgrade your software or hardware.

When choosing a backup strategy, consider several factors, such as the system workload, the usage schedule, the importance of the data, and the hardware environment of the database.

Oracle backups can be performed using scripts executing SQL commands combined with operating system commands to copy files, or using Oracle RMAN (Recovery Manager) commands.

It is recommended that you maintain updated records of backups performed on your database so that you can use them for recovery on demand. If you are using RMAN, catalog information is available from the catalog.

This section describes:

- Available Backup Methods
- Oracle Recovery Manager – RMAN

Available Backup Methods

This section describes the various backup methods that are available.

This section includes the following topics:

- "Cold Backup" on page 133
- "Hot Backup" on page 133
- "Export" on page 134

Cold Backup

Cold backup, also known as offline backup, is a database level backup. It normally requires that the database be shut down before the backup is started. The amount of downtime is dependent on the database size, the backup media (disk or tape), the backup software, and the hardware in use.

Once the instance is down, all its data files, log files, control files, and configuration files should be copied either to disk or other media. After the copy is complete, the instance can be restarted.

This backup method enables recovery to a the point in time in the past at which the database snapshot was taken.

For more information, refer to the *Oracle Backup and Recovery Guide* (http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96519&remark=drilldown&word=Backup).

Hot Backup

Hot backup, also known as online backup, enables you to run a backup while the instance is running and users are connected to the database. This backup method is a tablespace backup level and requires the database to operate in archivelog mode, which enables Oracle to track changes over time by generating redo log file copies called archive files. The generated archive files are written to the archive destination specified by the LOG_ARCHIVE_DEST (or LOG_ARCHIVE_DEST_NN) parameter in the instance parameter files. Other related archiving parameters are LOG_ARCHIVE_FORMAT and LOG_ARCHIVE_START.

Once you start the backup, all the data files, control files, archive files, and configuration files should be copied either to disk or other media. This method enables recovery to any point in time. Note that working in archivelog mode requires additional disk space to contain incremental archive files, which can influence database performance. During the backup process, Mercury Application Mapping may also experience some performance degradation due to disk load.

For more information, refer to the *Oracle Backup and Recovery Guide* (http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96519&remark=drilldown&word=Backup).

Export

In addition to the cold and hot physical backup methods, you can use the logical backup method known as export.

The export utility dumps schema structure and contents into an Oracle structured file. This method can be used to transfer data between two schemas in the same database, or between two separate Oracle databases. To load exported data back into the database, use the import utility.

For more information, refer to the Export/Import section of *Oracle Utilities* (http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96652&remark=drilldown&word=Export).

In Oracle 10g, the Oracle Data Pump utility can be used for exporting data. For more information, refer to the Oracle Data Pump page in the Oracle Utilities section of the Oracle Web site (http://download-east.oracle.com/docs/cd/B19306_01/server.102/b14215/part_dp.htm#i436481).

Oracle Recovery Manager – RMAN

Recovery Manager (RMAN) is a generic Oracle tool that enables you to back up and restore your target database. When working with RMAN, you can choose to work with the RMAN catalog schema. The catalog is managed within the Oracle schema and stores information on the registered database structure and backups performed using RMAN. It can be queried to produce backup reports and copy availability. A single catalog can manage backup information from one or more target databases.

The RMAN catalog is usually placed on a different database instance than the operational database and has a backup strategy of its own. It need only be available during the backup or recovery process.

The RMAN tool can be used in conjunction with third-party backup software for a complete backup and recovery solution.

The following are some advantages of RMAN:

- Minimizes backed up data by compressing backed up files to exclude empty data blocks, thereby saving time and space.
- Supports incremental backups.

- Supplies the user with backup status reporting ability.
- Supports parallel backup and recovery processes when possible.
- Can be used with a third-party backup media tool.

For more information on RMAN, refer to the Oracle Recovery Manager User's Guide (http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96566&remark=drilldown&word=RMAN) and Oracle Recovery Manager Reference (http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96565&remark=drilldown&word=RMAN) for Oracle 9i; and Backup and Recovery Advanced User's Guide (http://download-east.oracle.com/docs/cd/B19306_01/backup.102/b14191/toc.htm) and Backup and Recovery Reference (http://download-east.oracle.com/docs/cd/B19306_01/backup.102/b14194/toc.htm) for Oracle 10g.

12

Oracle Server Configuration Guidelines

This chapter contains guidelines for the Oracle database configuration settings that should be used when working with Oracle Server and Mercury Application Mapping. Note that the recommended settings differ according to the size of your Mercury Application Mapping deployment.

Oracle Parameter Sizing

The information provided in this section is best suited for environments with less than 1 million CIs in the CMDB and less than 1 GB of tablespace usage for each CMDB database (CMDB and History CMDB). For larger environments, some of the parameters should be resized.

The following table describes the recommended size for a number of parameters, when working with the Mercury Application Mapping database server deployment:

Parameter Name	Deployment	Remarks
DB_BLOCK_SIZE	8192	
DB_CACHE_SIZE	500MB	For Oracle 10g, it is recommended that you use the SGA_TARGET parameter.
DB_CACHE_ADVICE	ON	For gathering statistics when tuning is required
SHARED_POOL_SIZE	**Oracle 9i**: 80 MB **Oracle 10g**: 200 MB if SGA_TARGET is set, then use 80MB.	For Oracle 10g, it is recommended that you use the SGA_TARGET parameter
SGA_TARGET	800 MB	Available in Oracle 10g only (see remarks for SGA_TARGET)
DB_FILE_MULTIBLOCK_READ_COUNT	16	
PROCESSES	200	Add an additional 100 as a safety net
SESSIONS	225	(1.1 * PROCESSES) + 5
Compatible	➤ Oracle 9i: 9.2.0.6 ➤ Oracle 10g: 10.2.0.1	

Parameter Name	Deployment	Remarks
SORT_AREA_SIZE	1 MB	It is recommended that you use the PGA_AGGREGATE_TARGET parameter instead. This parameter is reserved for backward compatibility and shared server mode. In order for it to work, the WORKAREA_SIZE_POLICY parameter must be set to MANUAL.
SORT_AREA_RETAINED_SIZE	Equal to SORT_AREA_SIZE value	See remarks for the SORT_AREA_SIZE parameter.
HASH_AREA_SIZE	3 MB	Is equal to 3 times the SORT_AREA_SIZE value. See remarks for the SORT_AREA_SIZE parameter.
RECYCLEBIN	Off	Parameter available when working with Oracle 10g only.
WORKAREA_SIZE_POLICY	AUTO	The automatic PGA memory management mode applies only to work areas allocated by dedicated Oracle Servers. The size of work areas allocated by shared Oracle Servers is still controlled by the previous *_AREA_SIZE parameters because these work areas are allocated mainly in SGA and not in PGA.
PGA_AGGREGATE_TARGET	400 MB	
STATISTICS_LEVEL	TYPICAL	Enables tuning if required.
OPEN_CURSORS	800	
UNDO_MANAGEMENT	AUTO	
UNDO_RETENTION	Oracle default	

Note:

- Setting the **SGA_TARGET** parameter configures Oracle to automatically determine the size of the buffer cache (db_cache_size), shared pool (shared_pool_size), large pool (large_pool_size), java pool (java_pool_size), and streams pool (streams_pool_size).
- The value configured for SGA_TARGET sets the total size of the SGA components.
- When SGA_TARGET is set (that is, it's value is not 0), and one of the above pools is also set to a non-zero value, the pool value is used as the minimum value for that pool.

Oracle File Sizing

This section describes the file sizing guidelines for tablespaces, temporary tablespaces, data tablespaces, redo logs, and rollbacks.

This section includes the following topics:

- "Mercury Application Mapping Tablespace Settings" on page 140
- "Temporary Tablespace Settings" on page 142
- "Redo Log Settings" on page 143
- "Undo Segment Settings" on page 143

Mercury Application Mapping Tablespace Settings

The Mercury Application Mapping Server default setup creates the Tablespaces.

You might want to create dedicated Mercury Application Mapping tablespaces prior to the Mercury Application Mapping server setup if you have special storage configuration at your database site.

Tablespaces Default Setup

The Mercury Application Mapping server default setup creates the tablespaces that are needed for the Mercury Application Mapping server schemas. For details, see "Installation Procedure with Oracle Server" in *Mercury Application Mapping Installation Guide*.

As part of the Mercury Application Mapping Server setup, the following dedicated tablespaces are created:

- **CMDBDATA.** For CMDB tables.
- **CMDBINDEX.** For CMDB indexes.
- **CMDBHISTORYDATA.** For CMDB History tables.
- **CMDBHISTORYINDEX.** For CMDB History indexes.

The data files are created by default in the Oracle directory according to the release default location.

Both the names and files locations can be changed during setup according to the customer's environment.

The files initial size is 500 MB with auto extend on. You can switch off auto extend on after the setup and resize the tablespaces.

Allow at least 1 GB for each tablespace if auto extend is switched off, for an average number of CIs in the CMDB. If the CMDB is expected to hold more than 1 million CIs, tablespaces should be larger. It is strongly recommended that the local DBA creates Mercury Application Mapping tablespaces before installing the Mercury Application Mapping server. For details, see below.

Creating Tablespaces in Advance

The main reason you would want to create dedicated Mercury Application Mapping tablespaces in advance (prior to the Mercury Application Mapping server setup) is the special storage configuration in each database site. This configuration is based on available disk devices and/or RAID configuration.

For instance at a site with good RAID configuration, one large Tablespace is sufficient, while at a site with local disks (without RAID), it is recommended to split the data files amongst the available disks and to create at least two Tablespaces per schema.

For the recommended RAID configuration, see "Using RAID Configuration" on page 144.

To use this option, make sure that you override the names of the default Tablespaces and data files in the Mercury Application Mapping database installation screens, with the Tablespaces and data files you have created. For details, see "Single Machine (Typical) Deployment" in *Mercury Application Mapping Installation Guide*.

To share a Tablespace between two schemas, or for both table data and index data, just fill the same Tablespace name in the relevant Tablespace Name boxes.

In addition, tablespaces used in Mercury Application Mapping should:

- be locally managed
- have Automatic Segment space management

Note: There are additional size settings for temporary and undo tablespaces. For more information, see "Temporary Tablespace Settings" on page 142 and "Undo Segment Settings" on page 143.

Temporary Tablespace Settings

Temporary tablespace should be available before setting up the Mercury Application Mapping server according to the following settings. For details, see "Installation Procedure with Oracle Server" in *Mercury Application Mapping Installation Guide*:

Tablespace	Size	Remarks
TEMP	1 GB	
TEMP storage settings	Uniform allocation: 2 MB	➤ Should preferably be locally managed (Uniform allocation, automatic segment space management). ➤ Tablespace should be of a temporary type (use of TEMPFILE).

Note: If the WORKAREA_SIZE_POLICY parameter is set to MANUAL, the initial and next extent temporary tablespace storage space should be a multiple of the SORT_AREA_SIZE parameter (greater than 1).

Redo Log Settings

The following table specifies the recommended sizes for redo log files:

Setting	Size
Redo log file size	100 MB
Minimum number of groups	3
Minimum number of members per group	2

Note: Increase the sizes in large implementations of Mercury Application Mapping.

Undo Segment Settings

The following table specifies the recommended sizes for rollback segments:

Setting	Size	Remarks
Undo tablespace size	1 GB	The number of segments, the minimum number of extents, and the rollback segment size (initial, next) are all set automatically by Oracle.

Setting	Size	Remarks
UNDO_MANAGEMENT parameter	AUTO	Oracle default values.
UNDO_RETENTION parameter	Oracle default value	

Note: Increase the sizes in large implementations of Mercury Application Mapping.

Using RAID Configuration

The use of RAID is transparent to Oracle. All the features specific to RAID configurations are handled by the operating system and not by Oracle.

The use of RAID devices differs according to the Oracle file type. Data files and archive logs can be placed on RAID devices, since they are accessed randomly. Redo logs should not be put on RAID devices, since they are accessed sequentially and performance is enhanced by having the disk drive head near the last write location. However, mirroring of redo log files is strongly recommended by Oracle.

RAID is much easier to use than the Oracle techniques for data placement and striping.

Note the following RAID configuration recommendations:

- RAID usually impacts write operations more than read operations. This is especially true where parity needs to be calculated (RAID 3, RAID 5, and so forth).
- You can place online or archived redo log files on RAID 1 devices. Do not use RAID 5. In addition, place TEMP tablespace data files on RAID 1 devices, instead of RAID 5, because the streamed write performance of distributed parity (RAID 5) is not as good as that of simple mirroring (RAID 1).

- Swap space can be used on RAID devices without affecting Oracle.

The following table describes the RAID devices and types to be used with each Oracle file type:

RAID	Type of RAID	Control File	Database File	Redo Log File/ Temporary	Archive File
0	Striping	Avoid	OK	Avoid	Avoid
1	Shadowing	OK	OK	Recommended	Recommended
0+1	Striping + Shadowing	OK	Recommended	Avoid	Avoid
3	Striping with static parity	OK	Avoid when this data file involves heavy write operation	Avoid	Avoid
5	Striping with rotating parity	OK	Avoid when this data file involves heavy write operation	Avoid	Avoid

Note:

- RAID 0 does not provide protection against failures. It requires a strong backup strategy.
- RAID 0+1 is recommended for database files because it avoids hot spots and provides the best possible performance during a disk failure. The disadvantage of RAID 0+1 is its costly configuration.
- Use the highest RPM disks for temporary/redo logs. Use as many controllers as you can in the array, and ensure that you place the redo log groups on different controllers.

Recommended Character Settings

The character set should be: AL32UTF8.

Part IV

Appendix

A

Mercury Application Mapping Database Script

This chapter describes the Mercury Application Mapping database script.

About Mercury Application Mapping Database Script

A directory of database scripts is deployed on the Mercury Application Mapping server to be used for various offline administrative tasks against the Mercury Application Mapping databases.

For example, scripts are available to check index fragmentation and to handle statistics in Mercury Application Mapping schemas.

Database Utilities–Location

The locations of the database utilities are:

- Oracle scripts – **<Mercury Application Mapping server>\J2F\cmdb\dbscripts\oracle**
- SQL server scripts – **<Mercury Application Mapping server>\J2F\cmdb\dbscripts\ms**

Database Utilities–Database Clients

The database scripts require thick database client tool: SQLPLUS for Oracle and ISQL for SQL server.

If you do not have a thick database client tool on the Mercury Application Mapping server machine, then you can easily move all the scripts to a different machine with the appropriate client by the following procedure:

- Step 1 - Copy the ms directory (when working with an SQL server) or oracle directory (when working with an Oracle server) to a machine with database client installation.
- Step 2 - Change SCRIPTS_LOCATION in the environment file: env_cmdb.bat for Oracle and env_cmdb_ms.bat for SQL server.

Database Utilities–Readme File

A detailed description of prerequisites and operation method of each utility can be found in the readme.txt file available in the DB scripts path. (Readme.txt for Oracle, Readme_ms.txt for SQL server).

Index