

HP Universal CMDB

For the Windows, Red Hat Enterprise Linux, and Oracle Enterprise Linux operating systems

Software Version: 10.01

Database Guide

Document Release Date: December 2012

Software Release Date: December 2012



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© Copyright 2002 - 2012 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

<http://www.hp.com/go/hpsoftwaresupport>

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

Database Guide	1
Contents	5
Introducing the Database Environment	8
Introduction to Preparing the Database Environment	9
Databases in Use Overview	9
HP Universal CMDB Sizing	9
Hardware Requirements	10
Deploying and Maintaining the Microsoft SQL Server Database	11
Overview of Microsoft SQL Server Deployment	12
About Microsoft SQL Server Deployment	12
System Requirements	12
Manually Creating the Microsoft SQL Server Databases	14
Overview of the Microsoft SQL Server Databases	14
Manually Creating the HP Universal CMDB Microsoft SQL Server Database	14
Installing and Configuring Microsoft SQL Server	16
Workflow for Microsoft SQL Server Deployment	16
Installation Prerequisites	17
Notes and Limitations	17
Checklist for Support and Certification	18
Installing Microsoft SQL Server	19
Configuring Microsoft SQL Server	20
Verifying and Modifying Server and Database Settings	21
Creating and Configuring the Microsoft SQL Server Database	25
Creating the Database	25
Configuring the Database	28
Maintaining Microsoft SQL Server Databases	33
Backing Up Databases	33

Database Integrity and Fragmentation	35
Using Windows Authentication to Access the Microsoft SQL Server Database	41
Enabling HP Universal CMDB to Work with Windows Authentication	41
Deploying and Maintaining the Oracle Server Database	44
Overview of Oracle Server Deployment	45
About Oracle Server Deployment	45
System Requirements	45
Manually Creating the Oracle Server Database Schemas	47
Overview of the Oracle Database Schema	47
Prerequisites	47
Creating Schema Default and Temporary Tablespaces	48
Creating Schema User Permissions	48
Creating Schemas and Schema Objects	48
Oracle Server Configuration and Sizing Guidelines	49
Oracle Parameter Settings	49
Oracle Tablespaces	52
Locally Managed Tablespaces	52
Oracle Tablespace Settings	52
Using RAID Configuration	55
Maintaining an Oracle Server Database	57
Database Maintenance and Tuning	57
Oracle Database Backup and Recovery	62
Configuring the Oracle Client for HP Universal CMDB	65
Oracle Client Versions and Operating System Platforms	65
Oracle Client Installation	65
Oracle Client Configuration	65
Oracle Summary Checklist	67
Checklist for HP Universal CMDB Support and Certification	67
Oracle Server and Client Requirements	69
Setting Oracle Initialization Parameters	69
Appendixes	70
Appendix A: Support for Oracle Real Application Cluster	71

About Oracle Real Application Cluster (RAC)	71
Single Client Access Name	72
Client Side Configuration for Oracle RAC	72
Server Side Configuration	75
Recommendation for the HP Universal CMDB Database	75
Create/Connect to an HP Universal CMDB Database	75

Introducing the Database Environment

Chapter 1

Introduction to Preparing the Database Environment

This chapter includes:

Databases in Use Overview	9
HP Universal CMDB Sizing	9
Hardware Requirements	10

Databases in Use Overview

To work with HP Universal CMDB, you must set up the CMDB database. The CMDB database is used for storage of configuration information that is gathered from the various HP Universal CMDB and third-party applications and tools. This information is used when building HP Universal CMDB views.

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

If you are working with a Microsoft SQL Server database, refer to Part II, "Deploying and Maintaining the Microsoft SQL Server Database" on page 11.

If you are working with an Oracle Server database, refer to Part III, "Deploying and Maintaining the Oracle Server Database" on page 44.

The appendixes contain additional information that is pertinent to both Microsoft SQL Server and Oracle Server databases.

Note:

- Database servers must be set to the same time zone, daylight savings settings and time as the HP Universal CMDB servers.
- For details on working in a non-English language HP Universal CMDB environment, see the section on non-English locales in the interactive *HP Universal CMDB Deployment Guide*.

HP Universal CMDB Sizing

HP Universal CMDB database configuration requirements are dependent on the amount of data, as well as on the runtime load, generated by HP Universal CMDB. A small HP Universal CMDB deployment requires a database that can hold up to 1.5 million CMDB objects and links. A large HP Universal CMDB deployment requires a database containing 1.5 million or more CMDB objects and links.

Hardware Requirements

The following table describes the hardware (CPU and memory) requirements recommended for the HP Universal CMDB Oracle or Microsoft SQL database server:

Deployment	Number of Processors	Physical Memory
Small	1 dual core	Minimum: 4G RAM Recommended: 8G RAM
Large	Minimum 2 dual core or 1 quad core	Minimum: 8G RAM Recommended: 16G RAM and up

Deploying and Maintaining the Microsoft SQL Server Database

Chapter 2

Overview of Microsoft SQL Server Deployment

This chapter includes:

About Microsoft SQL Server Deployment	12
System Requirements	12

About Microsoft SQL Server Deployment

To deploy Microsoft SQL Server for use with HP Universal CMDB, you must perform the following procedures:

- **Install and configure Microsoft SQL Server.**

For details on installing and configuring Microsoft SQL Server, see ["Installing and Configuring Microsoft SQL Server"](#) on page 16.

- **Create databases on Microsoft SQL Server.**

You can create HP Universal CMDB databases manually or you can use the Server and Database Configuration utility to create the databases. For details on manually creating databases for HP Universal CMDB, see ["Manually Creating the Microsoft SQL Server Databases"](#) on page 14.

For details on creating a Microsoft SQL database, see ["Creating and Configuring the Microsoft SQL Server Database"](#) on page 25.

System Requirements

This section describes the system requirements for working with Microsoft SQL Server in conjunction with HP Universal CMDB.

This section includes the following topics:

["Hardware Requirements"](#) below

["Software Requirements"](#) below

Hardware Requirements

For HP Universal CMDB hardware sizing guidelines, see ["Hardware Requirements"](#) on page 10. For Microsoft SQL Server hardware requirements, refer to the installation guide for your Microsoft SQL Server release for your operating system.

Software Requirements

For details on supported versions of Microsoft SQL Server, see the Server Database Requirements

section of the HP Universal CMDB Support Matrix.

Chapter 3

Manually Creating the Microsoft SQL Server Databases

This chapter includes:

Overview of the Microsoft SQL Server Databases	14
Manually Creating the HP Universal CMDB Microsoft SQL Server Database	14

Overview of the Microsoft SQL Server Databases

HP Universal CMDB uses the configuration management database (CMDB) for its persistency.

During the HP Universal CMDB setup, the new database can be set up automatically by the setup procedure, or an existing database can be used. An existing database can either be created manually in advance (for example, due to organization security restrictions), or can be created by a previous installation of the same release of HP Universal CMDB.

For details about installing the HP Universal CMDB server, see the interactive *HP Universal CMDB Deployment Guide*.

Manually Creating the HP Universal CMDB Microsoft SQL Server Database

Before creating the database, you must install the Microsoft SQL Database Server according to the details in this guide. For further details on installing and configuring the Microsoft SQL Database Server, see "Installing and Configuring Microsoft SQL Server" on page 16.

If the HP Universal CMDB database is manually created prior to running the HP Universal CMDB setup, during the setup procedure, you select the option to use an existing database and enter the relevant data about the existing database 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 database.

Creating the Database

The database administrator should create a Microsoft SQL Server database for the CMDB data. No scripts are necessary to create the objects to populate this database.

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 the database manually, see ["Creating and Configuring the Microsoft SQL Server Database"](#) on page 25.

Note: When the HP Universal CMDB setup procedure automatically creates the database, it uses two file groups for each database; one to hold the system tables and one for the application data.

Chapter 4

Installing and Configuring Microsoft SQL Server

This chapter includes:

Workflow for Microsoft SQL Server Deployment	16
Installation Prerequisites	17
Notes and Limitations	17
Checklist for Support and Certification	18
Installing Microsoft SQL Server	19
Configuring Microsoft SQL Server	20
Verifying and Modifying Server and Database Settings	21

Workflow for Microsoft SQL Server Deployment

To deploy Microsoft SQL Server for use with HP Universal CMDB, perform the following steps:

1. **Review Microsoft SQL Server sizing guidelines**

For details, see "Database File Layout" on page 26.

2. **Review installation prerequisites**

For details, see "Installation Prerequisites" on next page.

3. **Review server and database checklists**

These checklists summarize the server and database configuration options supported and recommended for HP Universal CMDB.

For details, see "Checklist for Support and Certification" on page 18.

4. **Install Microsoft SQL Server Enterprise edition**

For details, see "Installing Microsoft SQL Server" on page 19.

5. **Configure Microsoft SQL Server**

For details, see "Configuring Microsoft SQL Server" on page 20.

6. **Create HP Universal CMDB databases on Microsoft SQL Server**

For details, see "Creating the Database" on page 25.

7. **Configure HP Universal CMDB databases**

For details, see "Configuring the Database" on page 28.

8. Verify Microsoft SQL Server and databases

For details, see ["Verifying and Modifying Server and Database Settings"](#) on page 21.

9. Set up Windows authentication

For details, see ["Using Windows Authentication to Access the Microsoft SQL Server Database"](#) on page 41. This step is only relevant if you have selected Windows authentication instead of SQL Server authentication.

Installation Prerequisites

The following prerequisites must be met before installing Microsoft SQL Server:

• Database file placement

For improved performance, it is recommended to spread out HP Universal CMDB databases among several disks or RAID systems.

• Memory

There must be at least 2 GB of RAM.

• User Accounts

- If you plan to perform activities outside the local machine (such as file copying to or from different servers, backing up to a shared remote repository, and so forth), you must have a domain account available for Microsoft SQL Server services.
- If you want to install HP Universal CMDB using the installation wizard, you must provide a user account that has database creator privileges. If you are going to create the database manually, supply HP Universal CMDB with a user account that is a part of **db_datareader**, **db_datawriter** and **db_ddladmin** in the HP Universal CMDB database.

• Miscellaneous

- Verify that the disk where Microsoft SQL Server is to be installed is not compressed.
- Disable antivirus software and any other applications that may interfere while installing Microsoft SQL Server.

Notes and Limitations

- It is strongly recommended that no significant processes other than a single Microsoft SQL Server installation be installed on the server that hosts the HP Universal CMDB databases.

When Microsoft SQL Server is the sole significant process on the machine, do not change the default memory settings. Allow the Microsoft SQL Server to manage memory dynamically, except when you configure **awe enabled** support.

- Microsoft SQL Server 64-bit versions include support for extended systems, also known as Windows on Windows (WOW64). WOW64 is a feature of 64-bit editions of Microsoft Windows that allows 32-bit applications to execute natively in 32-bit mode.

Applications function in 32-bit mode even though the underlying operating system is running on the 64-bit platform.

- When working with Microsoft SQL Server, the CMDB collation and the SQL Server collation must be the same in order for an upgrade to run properly.

Checklist for Support and Certification

Information is provided in this section for both supported and certified Microsoft SQL Server options.

The certified options are recommended for working with HP Universal CMDB. Certified options are rigorously tested by HP quality assurance personnel. Supported options are those options for which HP quality assurance personnel have successfully performed basic tests.

Microsoft SQL Server		
Subject	Supported	Recommended
Instances	Default, Single	
Authentication Mode	Mixed, Windows	
Collation	Case-insensitive. HP Universal CMDB does not support binary sort order and case sensitivity. Only case-insensitive order with a combination of accent, kana, or width settings is supported .	Use the Collation Settings dialog box to select the collation. Do not select the binary check box. Accent, kana, and width sensitivity should be selected according to the relevant data language requirements. The selected language must be the same as the OS Windows regional settings language.
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

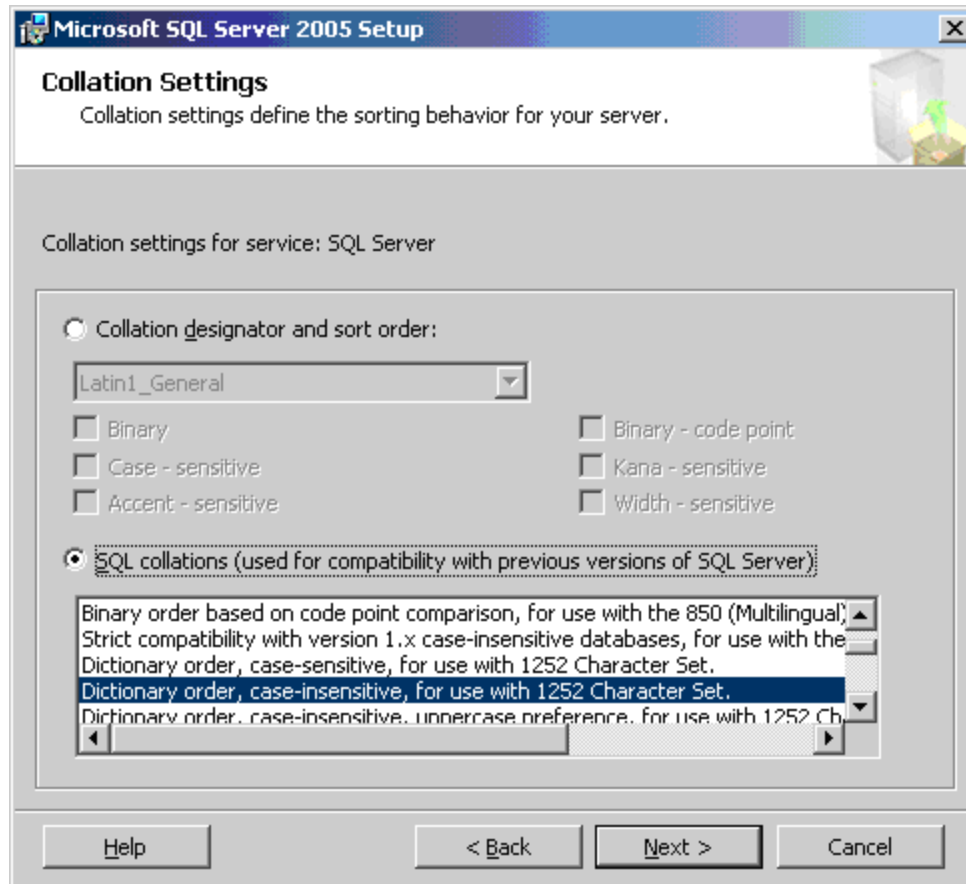
Installing Microsoft SQL Server

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. Selecting the default options may, in some cases, negatively affect the Microsoft SQL Server's performance. Select the following options in the installation dialog boxes:

- **Feature Selection dialog box configuration.**
 - Remove **Full Text Search** from the list because HP Universal CMDB does not use this indexing search feature
 - Under **Destination Folder**, make sure that the **Data Files** directory is stored on a fault-tolerant disk system, for example, RAID 1. Even though these system databases are fairly small, they are essential for the operation of Microsoft SQL Server.
- **Instance Name dialog box configuration.** For details on named instances, see "[Port Allocation Options for Named Instances](#)" on page 23. You access a default instance by specifying the server name or IP address.
- **Service Account dialog box configuration.**
 - If all Microsoft SQL Server activities are outside the local machine (for example, file copying to or from a different server, backing up to shared remote repository, replication with other servers, ActiveX script job steps, CmdExec job steps, and so forth), choose **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 Microsoft SQL Server activities are limited to the local machine, choose **Use the built-in System account** and select **Local system**. This selection provides SQL Server administrative privileges on the local machine only.
- **Authentication Mode dialog box.** HP Universal CMDB works with both Windows authentication and with Microsoft SQL Server authentication (recommended). To enable Microsoft SQL Server authentication, do the following:
 - a. Choose **Mixed Mode (Windows Authentication and SQL Server Authentication)**.
 - b. Enter the password for user **sa**.

Note: To further secure your Microsoft SQL Server, it is important to enter a password.

- **Collation Settings dialog box.** Follow the recommendations found in the checklist in "[Checklist for Support and Certification](#)" on previous page.



The above settings affect only the system databases and serve as the default settings for the user database. The database 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 Microsoft SQL Server, you can restore or attach a database that has different collation settings.

Caution: 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 Microsoft 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.

- Install the latest service pack for Microsoft SQL Server available at the Microsoft Download Center site at <http://www.microsoft.com/downloads/details.aspx?FamilyID=cb6c71ea-d649-47ff-9176-e7cac58fd4bc&DisplayLang=en>.

Configuring Microsoft SQL Server

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

This section includes the following topics:

"Service Configuration Options" on next page"

"Server Configuration Options" below

Service Configuration Options

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

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

If the dynamic port option is used for Microsoft SQL Server instances, make sure the SQL Server browser service, as well as all the SQL Server instance services, are in automatic mode.

Server Configuration Options

Most server configuration options are dynamically configured by Microsoft SQL Server. For HP Universal CMDB certification, you should not change the default options unless you are instructed to do so by HP Software Support.

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 Microsoft SQL Server instance and a named instance appear as SQL Server (Instance_Name).
Authentication Mode	In the Microsoft SQL Server Management Studio, right-click the server, choose Properties , and click the Security tab. Select SQL Server and Windows Authentication mode (Mixed Mode) .
Collation Settings	Run the following command: sp_helpsort
Network Libraries	<p>On the Server, select Start > Programs > Microsoft SQL Server <release number> > Configuration Tools > Configuration Manager. At SQL Native Client, choose client protocols and ensure the selected protocol is in the enabled state.</p> <ul style="list-style-type: none"> • Supported. Shared memory, TCP/IP, and Named Pipes for both the server and client. • Recommended. Only TCP/IP for both the server and client.

Server/Database Setting	How to Verify/Modify the Setting
View or change server configuration options	<ul style="list-style-type: none"> To allow the viewing of all options, use: EXEC sp_configure 'show advanced options', 1 reconfigure with override To view the current values, use: EXEC sp_configure To change a setting, use: EXEC sp_configure '<option>', <value> <p>Some options take effect only after you run reconfigure with override, while others require restarting the MSSQLServer service. See Microsoft SQL Server 2005 Books Online for details.</p>
Check whether the user who is going to create the HP Universal CMDB databases has CREATE DATABASE permissions	<p>Log in to the Microsoft SQL Server Management Studio with the login you want to check, and run the following:</p> <pre>USE master IF PERMISSIONS() & 1 = 1 PRINT 'User has CREATE DATABASE permissions' ELSE PRINT 'User does not have CREATE DATABASE permissions'</pre>
Check whether the HP Universal CMDB database user has enough permissions in the database	<ol style="list-style-type: none"> Log in to the Microsoft SQL Server Management Studio with the user name you want to check. Change the context of the database to the required database. Open a new query and do the following for each database: <pre>select case when IS_MEMBER ('db_owner')=1 or IS_SRVROLEMEMBER ('sysadmin')=1 or (IS_MEMBER ('db_ddladmin') = 1 and IS_MEMBER ('db_datareader')=1 and IS_MEMBER ('db_datawriter')=1 and IS_MEMBER ('db_denydatareader')=0 and IS_MEMBER ('db_denydatawriter')=0) then 'User has enough permissions' else 'User does not have enough permissions' end</pre>
Data and log file destination directory is not compressed (only in NTFS)	<p>Right-click the directory, choose Properties, and then Advanced. Verify that the Compression check box is cleared.</p>

Server/Database Setting	How to Verify/Modify the Setting
Database and database file properties (including recovery model and collation properties)	<ul style="list-style-type: none"> • To view the database and database file properties, run: <code>EXEC sp_helpdb <database name></code> • To change the database properties, run: <code>ALTER DATABASE <database name> SET <option> <value></code> • To change the database file properties, run: <code>ALTER DATABASE <database> MODIFY FILE (name = <filename>, <property> = <value>)</code> <p>You can also view or change these properties from the Database Properties dialog box in the Enterprise Manager.</p>
Microsoft SQL Server service pack version and edition	<p>Log into Management Studio and make the following query:</p> <pre>select @@version</pre>

Port Allocation Options for Named Instances

There are two options for assigning the port allocation of a named instance in SQL Server: **dynamic port** and **static port**.

- **Dynamic Port.** When working with a dynamic port, you assign a new port to the instance each time the instance is started. To enable clients to know the port when connecting to the instance, a service called SQL Server Browser needs to be started. The browser service listens to port 1434 and directs clients to the correct port according to the desired instance name.

You access the named instance by specifying the server name or IP address followed by `\<instance_name>`, for example, `server1\inst1`.

- **Static Port.** When working with a static port, you assign a specific port to the instance, and SQL Server Browser service is not needed in order to connect to the instance (although you can still use it).

The default for SQL Server is a dynamic port.

To work with a static port, follow the following steps:

1. Select **Start > Programs > Microsoft SQL Server > Configuration Tools > SQL Server Configuration Manager**.
2. Expand Protocols for <instance name>, and double-click **TCP/IP**.
3. In the **TCP/IP Properties** dialog box, on the **IP Addresses** tab, several IP addresses appear, in the format IP1, IP2, up to IPAll. For each address:
 - a. If the **TCP Dynamic Ports** dialog is set to 0, indicating that the database engine is listening on dynamic ports, delete the 0.
 - b. In the **TCP Port** box, enter the port number on which you want this IP address to listen, and click **OK**.
4. In the console pane, click **SQL Server Services**.
5. In the details pane, right-click SQL Server (<instance name>) and click **Restart** to stop and restart SQL Server.

You access the named instance by specifying the server name or IP address followed by the port number, for example, **server1\1435**.

Chapter 5

Creating and Configuring the Microsoft SQL Server Database

This chapter includes:

Creating the Database	25
Configuring the Database	28

Creating the Database

This section includes the following topics:

"Database Permissions" below

"Database File Layout" on next page

"System Databases" on page 28

Database Permissions

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

Note: HP Universal CMDB login accounts should be mapped to dbo in the database. 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 check whether a user has CREATE DATABASE permissions, log in to Management Studio 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.'
```

To check whether a user has enough permissions in the database, log in to Management Studio 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 command:

```
select case when IS_MEMBER ('db_owner')=1
or IS_SRVROLEMEMBER ('sysadmin')=1
or (IS_MEMBER ('db_ddladmin') = 1 and
IS_MEMBER ('db_datareader')=1 and
IS_MEMBER ('db_datawriter')=1 and
IS_MEMBER ('db_denydatareader')=0 and
IS_MEMBER ('db_denydatawriter')=0 )
```

```
then 'User has enough permissions'  
else 'User does not have enough permissions'  
end
```

Database File Layout

When you create a database, it must consist of at least one data file (with an .mdf extension) and one transaction log file (with an .ldf extension). You can optionally create additional data files (.ndf), as well as additional log files (.ldf).

To enhance performance, you may want to create several data files. Microsoft SQL Server stripes the data among the data files, so that if you do not have RAID controllers that stripe your data, you can spread the data files over several regular physical disks and, in this way, have the data striped. The log, however, is read sequentially, so that there is no performance gain in adding more log files. An additional log file should be created on a different disk when your existing log is out of disk space.

Data and Log Placement

Caution:

- It is recommended not to place data or log files on the same disk that stores the page (swap) file.
 - It is recommended that you place the data and log files on separate disk subsystems.
-
- **Log files.** Changes are not flushed to the database until they are written to the log, and the log architecture dictates serial writes, so it is advisable that there be as little interference as possible with the log activity. It is usually sufficient to place the log on a RAID 1 system because of the serial writes to the log. If you have processes reading from the log (for example, triggers accessing the inserted and deleted views which are formed from the log records or transactional replication), or several log files for different databases, consider placing the log file (s) on a RAID 0+1 (striped mirror) system.
 - **Data files.** Data files should be placed on a RAID 0+1 system for optimal performance.

File and Database Properties

When you create a database you can specify the following properties for each file (.mdf, .ndf, .ldf):

- **NAME.** The logical file name which you can use later when you want to alter one of the properties.
- **FILENAME.** The physical file path and name. Make sure the destination directory is not compressed (right-click the directory in Windows Explorer, select **Advanced**, and verify that the **Compression** check box is not selected).
- **SIZE.** The initial file size.
- **MAXSIZE.** Determines the maximum size to which the file can grow. If this argument is omitted, or if you specify **Unlimited**, the file can grow until the disk is full.
- **FILEGROWTH.** The automatic growth increment of the file. This argument can be specified as either a percentage of the existing file size, or as a fixed size.

An autogrowth operation invoked by a modification sent by a client that timed out cannot be completed successfully. This means that the next time a client sends a modification, the autogrowth process starts at the beginning and may also time out.

To avoid this problem, it is recommended that you either expand the files manually every time the database nearly reaches full capacity (for example, 20 percent free), or set the growth increment to a fixed size that takes less time to be allocated than the client's timeout setting. Using a small growth increment is not recommended because it increases file system fragmentation. On the other hand, if you use a very large increment, modifications sent by clients might incur connection timeouts while waiting for the automatic expansion to finish. For large databases, a percentage growth increment can lead to exponential growth of the database and should be avoided.

For more information on this problem, refer to Microsoft Knowledge Base Article - 305635 (<http://support.microsoft.com/kb/305635>).

The tempdb Database Settings

The frequent expansion of the tempdb system database can affect the database's performance, especially in large Microsoft 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 1 GB 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.

To ensure that there is enough disk space for the tempdb to grow during times of heavy usage (for example, when aggregating or sorting data), it is recommended that you leave at least 20 GB free disk space on the drive where the tempdb is located.

File Groups

File groups are logical groupings of data files. Each of the following objects can be placed in its own file group unit:

- A table's data
- A table's large objects (text, ntext, image columns)
- An index

Data is inserted proportionally into all files belonging to the file group in which the object is stored, according to the amount of free space in each file. The **.mdf** file is placed in a file group called **PRIMARY**, which is marked as **Default** when the database is created (the default file group for objects when no file group is specified). If you do not place other data files (**.ndf** files) in their own file groups, they are also placed in the **PRIMARY** file group. Note that you can change the **Default** file group later on.

File groups can be used for performance tuning or maintenance. For details, see Microsoft SQL Server Books Online at <http://www.microsoft.com/downloads>.

Following is an example that demonstrates how to use file groups for maintenance:

- **Partial Restoring.** Microsoft SQL Server does not support the restoration of a single table. Even if you place a single table in a file group, you cannot restore a file group to a point in time earlier than the rest of the data. Instead, you must apply all log file backups in order to

synchronize the file group with the rest of the data. Microsoft SQL Server supports partial restoration to a database with a different name. A partial restoration allows you to restore a single file group, and supports point-in-time restoration. However, you must restore the PRIMARY file group because it contains the SYSTEM tables.

To be able to restore a single table to a point in time if a logical error occurs, you need to design the file groups in your database as follows:

- Ensure that the **.mdf** file is the only file in the **PRIMARY** file group.
- Place each large table in its own file group.
- Place all small tables in a separate file group.

System Databases

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

- **tempdb.** Numerous Microsoft 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 ["The tempdb Database Settings" on previous page](#).

To move tempdb's files, use the ALTER DATABASE tempdb MODIFY FILE command, and restart Microsoft SQL Server.

- **master, msdb, model.** These databases, although crucial for the operation of Microsoft 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 HP Universal CMDB certification, place system databases on fault tolerant disks. It is recommended to use RAID 1 disks.

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

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

Configuring the Database

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" on next page](#)

["Database Configuration Options" on next page](#)

Database File Configuration

You can change certain database file properties, as well as add or drop files using either of the following methods:

- use the Properties dialog box in Management Studio
- use the ALTER DATABASE command (for details, see Microsoft SQL Server Books Online at <http://www.microsoft.com/downloads>)

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 Microsoft 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 the Database](#)" on page 25.

Database Configuration Options

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

- the Options tab in the Management Studio's Properties dialog box
- the EXEC sp_dboptions stored procedure
- the ALTER DATABASE <database name> SET command

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

The following table lists, in alphabetical order, the default configuration options, as well as the configuration settings required for HP Universal CMDB certification:

Configuration Option	Description	Default	HP Universal CMDB Certification in Microsoft SQL Server
ANSI NULL default (see note below)	Specifies whether the database columns are defined as NULL or NOT NULL, by default	Not set	Not set
ANSI PADDING	Controls the way the column stores values shorter than the defined size of the column and the way the column stores values that have trailing blanks in char, varchar, binary, and varbinary data.	OFF	ON
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 create statistics	Specifies whether missing statistics required by a query for optimization are built automatically during optimization	Set	Set
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	HP Universal CMDB Certification in Microsoft SQL Server
Auto update statistics	Specifies whether out-of-date statistics required by a query for optimization are built automatically during optimization	Set	Set
Compatibility level	The version of Microsoft SQL Server that the database appears to be (for the application)	The same version as the release installed	The same version as the release installed
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 manipulation). 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)
Recursive triggers	Specifies whether recursive triggers are supported	Not set	Not set
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
Torn page detection	Specifies whether incomplete pages can be detected	Set	Set
Truncate log on checkpoint	Automatically marks inactive portions of log for reuse on checkpoint	Not set	N/A

Configuration Option	Description	Default	HP Universal CMDB Certification in Microsoft SQL Server
Use quoted identifiers	Specifies whether the Microsoft 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

Note:

Not all ANSI options can be set using Management Studio. The ANSI database configuration options include: ANSI_NULLS, ANSI_NULL_DEFAULT, ANSI_PADDING, ANSI_WARNINGS, ARITHABORT, CONCAT_NULL_YIELDS_NULL, NUMERIC_ROUNDABORT, and QUOTED_IDENTIFIER.

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	Allows Log Backup	Allows Point-in-Time/Log Mark Restoration	Allows Backup Log when Data Crashes (Saves changes until the crash point)	Amount of Bulk Operation Logging (can affect the performance of bulk operations)
Simple	No	No	No	Minimal
Bulk Logged	Yes	No	No	Minimal
Full	Yes	Yes	Yes	Full

To check your database's properties, run the command:

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

For information on SQL databases, see Microsoft SQL Server Books Online at <http://www.microsoft.com/downloads>.

Chapter 6

Maintaining Microsoft SQL Server Databases

This chapter describes the various maintenance tasks that are recommended for HP Universal CMDB databases created on Microsoft SQL Servers, such as backing up databases, checking database integrity and handling fragmentation, and monitoring databases.

This chapter includes:

Backing Up Databases	33
Database Integrity and Fragmentation	35

Backing Up Databases

Microsoft SQL Server supports three main types of database backup: full, differential, and log. It also supports file/file group backup, which is discussed in a separate section below. In order 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 Microsoft SQL Agent jobs. The Microsoft SQL Agent (represented by the SQLServerAgent service) is installed automatically when you install Microsoft SQL Server. Ensure that the Microsoft 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 Microsoft SQL Server service account needs permission to access) or tape (only local).

This section describes:

"Full Backup" on next page

"Differential Backup" on next page

"Log Backup" on next page

"File/File Group Backup" on page 35

"Maintenance Plan" on page 35

["Transaction Log Issues" on next page](#)

Full Backup

When you perform a full database backup, all information about the backed up database is contained within the backup, including data, metadata, 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, 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.

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 next page](#).

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, ensure that the database is 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 Microsoft 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 Microsoft SQL Server Management Studio, 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, Microsoft 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 inevitably leads 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 command:

```
KILL <process id>
```

Note: In Microsoft SQL Server, 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 describes:

["Database Integrity" on next page](#)

["Understanding File System Fragmentation" on next page](#)

["Understanding Internal Fragmentation" on next page](#)

["Understanding External Fragmentation" on next page](#)

["Detecting and Handling Index Fragmentation" on page 37](#)

["Supplied Utilities for Monitoring and Rebuilding Indexes" on page 38](#)

["Distribution Statistics" on page 39](#)

["Utility to Refresh Statistics" on page 39](#)

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 Microsoft 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 Microsoft SQL Server 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. 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, to avoid small, fragmented parts, use a large value as the file growth increment. Do not use too large a value, however, because this leads to client request timeouts when the file autogrows (for more details, see "[Database Permissions](#)" on page 25). 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 on the database.

Understanding Internal Fragmentation

Internal fragmentation refers to the percentage of data contained in the pages. In environments such as the HP Universal CMDB, which are characterized by transactions that frequently insert data, internal fragmentation is sometimes initiated in anticipation of new data in indexes and can be a positive occurrence. By leaving a certain percentage of the index pages free, you can avoid page splits for a certain period of time. This is especially significant for clustered indexes, because they contain the actual data pages. You can achieve internal fragmentation by periodically rebuilding your indexes using the CREATE INDEX command, with the DROP_EXISTING and FILLFACTOR options, or the ALTER INDEX REBUILD command (online or offline) and FILLFACTOR option. The FILLFACTOR option specifies the fullness of the leaf level index pages.

Understanding External Fragmentation

As page splits occur in your indexes, new allocated pages are acquired from the database file.

Ideally, a page split should yield the allocation of a page contiguous to the one that split. However, in practice, the space contiguous to the split page is usually already occupied. The more page splits that occur, the less the index's linked list reflects the physical layout of the pages on disk, and the greater the amount of external fragmentation.

External fragmentation impacts negatively on the performance of ordered index scans because the disk arm needs to move back and forth in order to retrieve the pages from disk. Ideally, the linked list should reflect the physical layout of the pages on disk so that when an ordered index scan is performed, the disk arm moves in one direction as it retrieves the pages from disk.

You can handle external fragmentation proactively by initiating internal fragmentation and leaving a certain percentage of the leaf level index pages free, thus avoiding page splits for a certain period of time. As mentioned earlier, internal fragmentation can be achieved by periodically rebuilding your indexes using the FILLFACTOR option. You can also handle external fragmentation by checking the external fragmentation status of your indexes, and rebuilding the indexes.

Detecting and Handling Index Fragmentation

Using `sys.dm_db_index_physical_stats` to Detect Fragmentation

The dynamic management function `sys.dm_db_index_physical_stats` is used to determine the degree of fragmentation of an index (replacing the DBCC SHOWCONTIG function in releases prior to Microsoft SQL Server 2005). You can detect fragmentation in a specific index, in all indexes on a table or indexed view, in all indexes in a specific database, or in all indexes in all databases. For partitioned indexes, `sys.dm_db_index_physical_stats` also provides fragmentation information for each partition.

Fragmentation of a table occurs through the process of data modifications (INSERT, UPDATE, and DELETE statements) that are made against the table and to the indexes defined on the table. Because these modifications are not ordinarily distributed equally among the rows of the table and indexes, the fullness of each page can vary over time. For queries that scan part or all of the indexes of a table, this kind of fragmentation can cause queries to return slower.

The table syntax of `sys.dm_db_index_physical_stats` is:

```
sys.dm_db_index_physical_stats (  
  { database_id | NULL }  
  , { object_id | NULL }  
  , { index_id | NULL | 0 }  
  , { partition_number | NULL }  
  , { mode | NULL | DEFAULT }  
  )
```

Run this command as regular select statement from table, for example:

```
select * from sys.dm_db_index_physical_stats (DB_ID('<BAC_database>'),  
object_id('<Table_Name>'), NULL, NULL, 'SAMPLED')
```

The **Avg_fragmentation_in_percent** column returned by `sys.dm_db_index_physical_stats` is the logical and extent fragmentation of the index.

For more information about `sys.dm_db_index_physical_stats`, see Microsoft SQL Server 2005 Books Online at <http://www.microsoft.com/technet/prodtechnol/sql/2005/downloads/books.msp>.

Handling Fragmentation

The table column **Avg_fragmentation_in_percent** returned by **sys.dm_db_index_physical_stats** reflects the degree of fragmentation per index. Based on this value, you can determine whether to handle the fragmentation and which method to use in handling it.

Use the following rough guidelines to determine the best method to correct the fragmentation:

- **Between 5% and 30%.** Use the ALTER INDEX REORGANIZE command to reorganize the index. Index reorganization is always executed online.
- **Greater than 30%.** Use the ALTER INDEX REBUILD command to rebuild the index. Index rebuilding can be executed online or offline. Rebuild the indexes online to achieve the availability similar to that of the Reorganize option.

Very low levels of fragmentation (less than 5%) should not be addressed by either of these commands due to the negligible benefit gained from removing such a small amount of fragmentation.

For more information on defragmenting indexes, refer to "Reorganizing and Rebuilding Indexes" under Designing and Implementing Structured Storage in SQL Server Books Online (<http://technet.microsoft.com/en-us/library/ms189858.aspx>).

Note: It is strongly recommended that you create an automatic index rebuild task for the CMDB database, as CMDB database data frequently changes.

Supplied Utilities for Monitoring and Rebuilding Indexes

HP Universal CMDB provides two utilities that can be used to detect and rebuild fragmented indexes. The **rebuild_fragmented_indexes.bat** utility uses 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 usually hinders 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 database administrator.

The utilities for Microsoft SQL Server are located in the **<HP Universal CMDB root directory>\UCMDBServer\tools\dbscripts\mssql** directory on the HP Universal CMDB Server.

Utility to Rebuild All Indexes in Database

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

To run the utility, execute **rebuild_indexes.bat** with the following parameters:

- SQL Server name
- Database name
- SA password

Example:

```
CMDB\db_scripts\rebuild_indexes.bat SQL_SRVR_3 UCMDB_DB_3 ad%min52.
```

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 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.

For example,

```
CMDB\db_scripts\rebuild_fragmented_indexes.bat SQL_SRVR_3 UCMDDB_DB_3  
ad%min52 1
```

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

Microsoft SQL Server 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 columns. 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 daily 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.

Whether the **auto update statistics database** option is enabled or disabled, it is strongly recommended that you create an automatic task to update statistics for each of the HP Universal CMDB databases on a daily basis, as the data is frequently changed. The job should execute the **sp_updatestats** API against the specific database.

Utility to Refresh Statistics

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

- List tables with out of date statistics. In this mode, a list of the tables is returned, together with the commands needed to update statistics the tables at a later time.
- Update statistics on the tables. In this mode, all tables with outdated statistics are being updated.

To run the **update_statistics.bat** utility:

From **<HP Universal CMDB root directory>\UCMDDBServer\tools\dbscripts\mssql**, execute **update_statistics.bat** with the following parameters:

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

For example:

```
update_statistics.bat SQL_SRVR_3 UCMDB_DB_3 ad%min52 1
```

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

Note: For information on troubleshooting application performance with Microsoft SQL Server, refer to <http://www.microsoft.com/technet/prodtechnol/sql/2005/tsprfprb.msp>.

Chapter 7

Using Windows Authentication to Access the Microsoft SQL Server Database

Unless configured otherwise, HP Universal CMDB uses Microsoft SQL Server authentication to access the Microsoft SQL Server databases. However, Windows authentication can also be used.

This chapter describes how to enable HP Universal CMDB to use Windows authentication to access Microsoft SQL Server databases.

This chapter includes:

Enabling HP Universal CMDB to Work with Windows Authentication	41
--	----

Enabling HP Universal CMDB to Work with Windows Authentication

You can enable HP Universal CMDB to use Windows authentication instead of Microsoft SQL Server authentication to access the HP Universal CMDB database.

To enable HP Universal CMDB to use Windows authentication to access a Microsoft SQL database, you must:

- Configure the Microsoft SQL Server to use Windows authentication.
- Launch the HP Universal CMDB Server service on all the HP Universal CMDB servers with a Windows user that has the necessary permissions to access the Microsoft SQL database.
- Run the UCMDB Server Configuration utility to create or connect to a Microsoft SQL database and specify the use of Windows authentication.

This section includes the following topics:

"Configuring Microsoft SQL Server to Use Windows Authentication" below

"Launching the UCMDB Server service with a Windows User " on next page

"Creating or Connecting to a Microsoft SQL Database Using Windows Authentication" on page 43

Configuring Microsoft SQL Server to Use Windows Authentication

To configure Microsoft SQL Server to use Windows authentication, perform the following steps:

1. In the SQL Server Enterprise Manager, select **Security > Logins**, right-click and choose **New Login**.
2. Enter the desired domain account, including the domain name, in the following pattern: DOMAIN\USER (for example, MY_DOMAIN\MY_ACCOUNT).
3. In the **Server Roles** tab, select **System Administrators** and click **OK**.

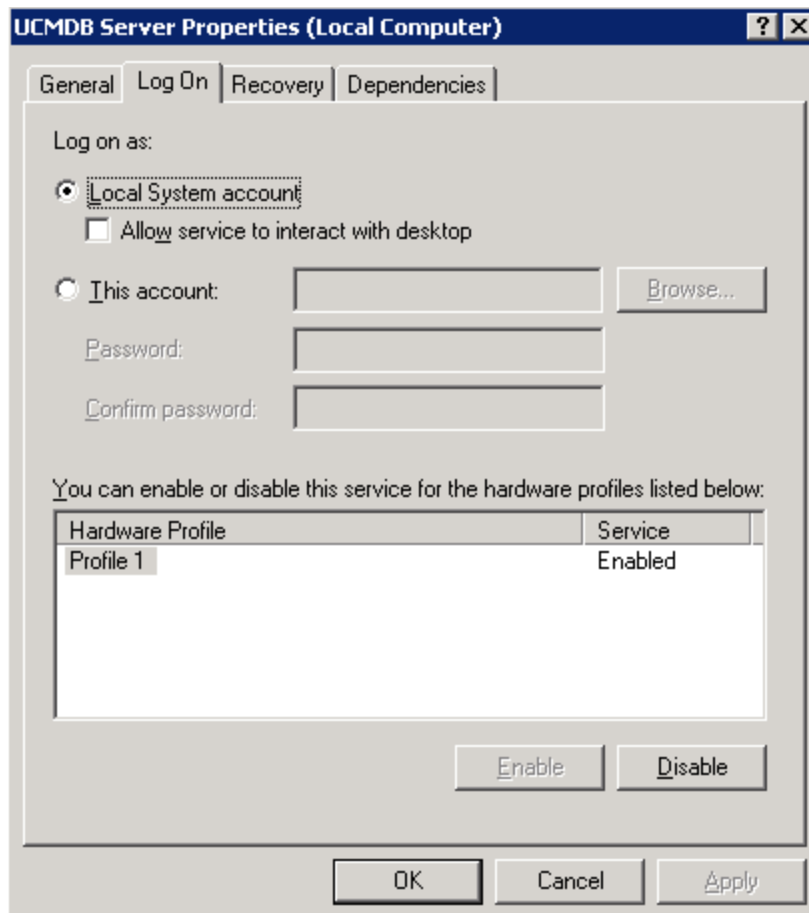
Launching the UCMDB Server service with a Windows User

By default, the UCMDB Server service is run as a system service. If you have configured your Microsoft SQL Server to use Windows authentication, you must change the user running the UCMDB Server service to the same Windows user you defined for the Microsoft SQL Server in order to enable the service user to access the database.

Note: For information on the additional permissions that the service user must have, see the section on Getting Started with UCMDB in the interactive *HP Universal CMDB Deployment Guide*.

To change the UCMDB Server service user:

1. Disable HP Universal CMDB (**Start > Programs > HP UCMDB > Stop HP UCMDB Server**).
2. In Microsoft's Services window, double-click **UCMDB_Server**. The UCMDB Server Properties (Local Computer) dialog box opens.
3. Click the **Log On** tab.



4. Select **This account** and browse to choose the user you previously defined for your Microsoft SQL Server.
5. Enter the selected user's Windows password and confirm this password.

6. Click **Apply** to save your settings and **OK** to close the dialog box.
7. Enable HP Universal CMDB (**Start > Programs > HP UCMDB > Start HP UCMDB Server**).

Creating or Connecting to a Microsoft SQL Database Using Windows Authentication

You create or connect to a database using the UCMDB Server Configuration utility. To create or connect to a Microsoft SQL database using Windows authentication, you must select this option within the UCMDB Server Configuration utility. For details on using the utility, see the section about accessing the servers in the interactive *HP Universal CMDB Deployment Guide*.

Deploying and Maintaining the Oracle Server Database

Chapter 8

Overview of Oracle Server Deployment

This chapter includes:

About Oracle Server Deployment	45
System Requirements	45

About Oracle Server Deployment

To deploy Oracle Server for use with HP Universal CMDB, perform the following procedures:

- **Install Oracle Server.**

For details on Oracle software installation, refer to the installation guide in the documentation for your specific Oracle platform. For software installation options, see ["Oracle Server Configuration and Sizing Guidelines"](#) on page 49 and ["Oracle Summary Checklist"](#) on page 67.

- **Build a database on Oracle Server to store HP Universal CMDB data.**

For instance configuration and sizing guidelines, see ["Oracle Server Configuration and Sizing Guidelines"](#) on page 49. For details on database instance installation, refer to the installation guide in the documentation for your specific Oracle platform.

- **Create one or more Oracle tablespaces to store HP Universal CMDB data.**

For details, see ["Oracle Tablespaces"](#) on page 52.

- **Create an Oracle user schema for HP Universal CMDB schemas.**

You can create HP Universal CMDB user schemas manually, or you can use the Server and Database Configuration utility to create the schemas for you. For details on creating an Oracle user schema for HP Universal CMDB, see ["Manually Creating the Oracle Server Database Schemas"](#) on page 47.

System Requirements

This section describes the system requirements for working with Oracle Server in conjunction with HP Universal CMDB.

This section includes the following topics:

["Hardware Requirements"](#) below

["Software Requirements"](#) on next page

["Oracle Instances"](#) on next page

Hardware Requirements

For HP Universal CMDB hardware sizing guidelines, see ["Hardware Requirements"](#) on page 10.

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>.

Software Requirements

For details on supported versions of Oracle Server, see the Server Database Requirements section of the HP Universal CMDB Support Matrix.

Oracle Instances

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

For HP Universal CMDB certification, do not use more than one Oracle instance. If you do use more than one instance for the HP Universal CMDB database, 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).

Chapter 9

Manually Creating the Oracle Server Database Schemas

This chapter includes:

Overview of the Oracle Database Schema	47
Prerequisites	47
Creating Schema Default and Temporary Tablespaces	48
Creating Schema User Permissions	48
Creating Schemas and Schema Objects	48

Overview of the Oracle Database Schema

HP Universal CMDB uses the configuration management database (CMDB) database schema for its persistency.

During the HP Universal CMDB setup, new schemas can be set up automatically by the setup procedure, or already existing schemas can be used. Existing schemas can either have been created manually in advance, or can have been created by a previous installation of the same release of HP Universal CMDB.

If you choose to have the schema 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 schema in advance.

For details about installing the HP Universal CMDB server, see the interactive *HP Universal CMDB Deployment Guide*.

Prerequisites

Before creating the database schema, 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 "Oracle Server Configuration and Sizing Guidelines" on page 49.

If the schema is manually created prior to running the HP Universal CMDB setup, during the setup procedure, select the option to use an existing schema and enter the relevant data about the existing schema 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 schema.

Creating Schema Default and Temporary Tablespaces

The database administrator should create at least one Temporary tablespace in the database instance hosting the HP Universal CMDB schema, and at least one tablespace to serve as the default tablespace for the HP Universal CMDB schema.

The tablespace specification should be made according to your storage policy, taking into account RAID configuration, available disks and storage devices.

For further details on tablespace file sizes and recommended settings, see "[Oracle Tablespace Settings](#)" on page 52.

Creating Schema User Permissions

The database administrator should create an Oracle schema user with the database permissions required by the HP Universal CMDB application.

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

- Roles
 - Connect
- Privileges
 - CREATE TABLE
 - CREATE VIEW
 - CREATE SEQUENCE
 - CREATE TRIGGER
 - CREATE PROCEDURE
 - UNLIMITED TABLESPACE
 - ALTER USER \${user} DEFAULT ROLE ALL
 - CREATE TYPE TO

Note: HP Universal CMDB supports any user with higher permissions. For HP Universal CMDB certification, use an Oracle user that has the exact Oracle permissions described above.

Creating Schemas and Schema Objects

The database administrator should create a schema for CMDB persistency.

Create a user schema with the permissions described in "[Creating Schema User Permissions](#)" above. No scripts are necessary to create the objects to populate these schemas.

Chapter 10

Oracle Server Configuration and Sizing Guidelines

This chapter contains guidelines for the Oracle database configuration and storage settings that should be used when working with Oracle Server and HP Universal CMDB. Note that the recommended settings differ according to the size of your HP Universal CMDB deployment. For details, see "HP Universal CMDB Sizing" on page 9.

This chapter includes:

Oracle Parameter Settings	49
Oracle Tablespaces	52
Oracle Tablespace Settings	52
Using RAID Configuration	55

Oracle Parameter Settings

The following table describes the recommended values for a number of Oracle database initialization parameters, when working with the HP Universal CMDB database server:

Parameter Name	HP Universal CMDB Deployment		Remarks
	Small	Large	
DB_BLOCK_SIZE	8K	8K-16K	Should be a multiple of the operating system block size.
DB_CACHE_ADVICE	ON	ON	For gathering statistics when tuning is required .
SGA_TARGET	1 GB	4 GB and higher	Oracle 10g. See remarks below the table.
MEMORY_TARGET	2 GB	5 GB and higher	Oracle 11g. See remarks below the table.
LOG_BUFFER	1 MB	5 MB	
DB_FILE_MULTIBLOCK_READ_COUNT	Oracle default value	Oracle default value	
PROCESSES	200	400	Add an additional 100 as a safety net

Parameter Name	HP Universal CMDB Deployment		Remarks
	Small	Large	
SESSIONS	225	445	(1.1 * PROCESSES) + 5
OPTIMIZER_INDEX_COST_ADJ parameter value	100	100	Affects performance
TIMED_STATISTICS	True	True	
LOG_CHECKPOINT_INTERVAL	0	0	
LOG_CHECKPOINT_TIMEOUT	0; or greater than or equal to 1800	0; or greater than or equal to 1800	
OPTIMIZER_MODE	ALL_ROWS	ALL_ROWS	
CURSOR_SHARING	Exact	Exact	
OPEN_CURSORS	800	800	
COMPATIBLE	The same as the release installed	The same as the release installed	
SQL_TRACE	False, True	False	
UNDO_MANAGEMENT	Auto	Auto	
UNDO_RETENTION	Oracle default value	Oracle default value	Automatic tuning is performed in Oracle 10g and up
RECYCLEBIN	Off	Off	
NLS_LENGTH_SEMANTICS	BYTE	BYTE	This parameter controls the length definition of varchar columns.

Parameter Name	HP Universal CMDB Deployment		Remarks
	Small	Large	
WORKAREA_SIZE_POLICY	AUTO	AUTO	
PGA_AGGREGATE_TARGET	400 MB	1 GB and higher	
STATISTICS_LEVEL	TYPICAL	TYPICAL	Enables tuning if required.
OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES	FALSE	FALSE	Oracle 11g. Controls Automatic Plan Capture as part of Oracle 11g SQL Management Base (SMB).
AUDIT_TRAIL	NONE	NONE	In Oracle 11g, the default is changed from none to db , which means that out-of-the-box database auditing is written to the SYS.AUD\$ audit trail table. It is advisable to change this value to none to avoid growth of the system tablespace.
CURSOR_SPACE_FOR_TIME	False	False	
USE_STORED_OUTLINES	False	False	Oracle Default
BLANK_TRIMMING	False	False	
FIXED_DATE	Not set	Not set	HP Universal CMDB uses the SYSDATE function for generating system time as part of the application process.
_PARTITION_LARGE_EXTENTS	FALSE	FALSE	Relevant for Oracle 11.2.0.2 only. When this hidden parameter is set to TRUE, it affects the size of partitions in native partitioned tables. The initial extent allocated for each partition is very large, thus causing unwanted growth of database data files.

Note the following:

- **SGA_TARGET.** Setting this 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, its 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.

- **MEMORY_TARGET.** In Oracle 11g, Automatic Memory Management enables the entire instance memory to be automatically managed and tuned by the instance. The instance memory contains the System Global Area (SGA) and the Program Global Area (PGA). In Oracle 11g, `MEMORY_TARGET` is the only required memory parameter to set, however it is recommended to set `SGA_TARGET` or `PGA_AGGREGATE_TARGET` as well to avoid frequent resizing of the SGA and PGA components. The values entered for `SGA_TARGET` and `PGA_AGGREGATE_TARGET` serve as minimum values.

Oracle Tablespaces

An Oracle tablespace is an Oracle object that is a logical container of database objects, for example, tables, indexes, and so forth. When working with HP Universal CMDB, you must create one or more dedicated default tablespaces for your HP Universal CMDB user schema. You may also want to create a dedicated temporary tablespace for HP Universal CMDB. To create a tablespace, you must provide specific operating system files that physically represent the tablespace, as well as extent parameters.

When mapping operating system files, there is an option to make the file auto-extendable. This feature is supported by HP Universal CMDB, but not certified for use with HP Universal CMDB, since it can cause the system to consume all available disk space.

Locally Managed Tablespaces

A locally managed tablespace is a feature introduced in Oracle8i. Prior to Oracle8i, all tablespaces were dictionary-managed tablespaces. A tablespace that manages its extents locally can have either uniform extent sizes, or variable extent sizes that are determined automatically by the system. When you create the tablespace, the **uniform** or **autoallocate** (system-managed) option specifies the type of allocation.

For system-managed extents, Oracle determines the optimal size of extents, with a minimum extent size of 64 KB. This is the default extent size for permanent tablespaces.

For uniform extents, you can specify an extent size, or use the default size, which is 1 MB. Temporary tablespaces that manage their extents locally can only use this type of allocation.

Note that the `NEXT`, `PCTINCREASE`, `MINEXTENTS`, `MAXEXTENTS`, and `DEFAULT STORAGE` storage parameters are not valid for extents that are managed locally.

All data and temporary tablespaces should be locally managed when working with HP Universal CMDB.

For information on locally managing temporary tablespace using `TEMPFILE`, see "[Temporary Tablespace Settings](#)" on page 54.

Oracle Tablespace Settings

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

This section includes the following topics:

"Data Tablespace Settings" below

"System Tablespace Settings" below

"Temporary Tablespace Settings" on next page

"Redo Log Settings" on next page

"Undo Segment Settings" on page 55

Data Tablespace Settings

The following table specifies the recommended sizes for HP Universal CMDB tablespaces:

HP Universal CMDB Deployment			
Tablespace	Small	Large	Remarks
CMDB	5 GB	60 GB	The specified size is a minimum requirement.

Note: The data in the table is only relevant when the following parameters are assigned their default values:

- the Oracle parameter **deferred_segment_creation** is set to **true** (only relevant for Oracle 11g R2)
- the tablespace setting **INITIAL_EXTENT** is set to **65,536**

Data Tablespace Default Storage Settings

The storage settings for data tablespaces should be:

- Locally managed tablespace
- Automatic segment space management
- Automatic local extent management

The following command can be used to create a data tablespace:

```
CREATE TABLESPACE <tablespace name> DATAFILE '<data file full path>'
REUSE SIZE <file size>
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

For example:

```
CREATE TABLESPACE APPTBS DATAFILE
'/ORADATA/ORCL/APPTBS1.DBF' REUSE SIZE 1024M
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

System Tablespace Settings

The following table specifies the recommended settings for system tablespaces:

HP Universal CMDB Deployment		
Tablespace	Small	Large
SYSTEM	2 GB	5 GB
SYSAUX	2 GB	5 GB

The System Tablespace storage default settings should be:

- Locally managed tablespace
- Segment space management:
 - SYSAUX: Automatic
 - SYSTEM: Manual
- Automatic local extent management

Temporary Tablespace Settings

The following table specifies the recommended settings for temporary tablespaces:

HP Universal CMDB Deployment			
Tablespace	Small	Large	Remarks
TEMP	1 GB	10 GB	Use multiple files with large tablespaces.
TEMP storage settings	Uniform allocation: 2 MB	Uniform allocation: 2 MB	<ul style="list-style-type: none"> • Should be locally managed (Uniform allocation). • Tablespaces should be of a temporary type (use of TEMPFILE). • Segment space management in temporary tablespaces is manual.

Redo Log Settings

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

HP Universal CMDB Deployment		
Setting	Small	Large
Redo log file size	100 MB	200 MB - 500 MB
Minimum number of groups	4	4
Minimum number of members per group	2	2

Undo Segment Settings

The following table specifies the recommended Undo settings:

HP Universal CMDB Deployment			
Setting	Small	Large	Remarks
Undo tablespace size	1 GB	10 GB	The number of segments, the minimum number of extents, and the rollback segment size (initial, next) are all set automatically by Oracle.
UNDO_MANAGEMENT parameter	AUTO		Oracle default value
UNDO_RETENTION parameter	Oracle default value		

The Undo Tablespace storage default settings should be:

- Locally managed tablespace
- Automatic segment space management
- Segment space management in undo tablespaces is manual

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.

Chapter 11

Maintaining an Oracle Server Database

This chapter describes the various maintenance and tuning procedures that are recommended for the HP Universal CMDB database created on an Oracle Server, as well as some of the available database backup and recovery methods.

This chapter includes:

Database Maintenance and Tuning	57
Oracle Database Backup and Recovery	62

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)" on next page

"Database Load Behavior" on next page

"CPU and I/O" on next page

"Oracle Alert File" on next page

"Archive Log – File System" on next page

"Tablespace Storage Space" on page 59

"Space Management" on page 59

"Collecting Statistics for Databases " on page 59

"Oracle 10g Automated Statistics Collection" on page 59

"Oracle 11g Automated Statistics Collection" on page 60

"Collecting Statistics for CMDB " on page 60

"Index Fragmentation" on page 61

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. In Oracle 11g, the parameters MEMORY_TARGET or MEMORY_MAX_TARGET cannot be greater than the shared memory file system (/dev/shm) on your operating system.

Database Load Behavior

Oracle10g AWR (Automatic Workload Repository) reports can be used to monitor performance bottlenecks and database behavior. For details, refer to Oracle Metalink Note 276103.1: PERFORMANCE TUNING USING 10g ADVISORS AND MANAGEABILITY FEATURES. AWR is an evolution of STATSPACK (8i and up) and utlbstat/utlestat.

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 AWR output to determine which particular Oracle data file is the cause of the contention and consider changing the I/O storage configuration.

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 and the storage system management tools, 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. As of Oracle 11g, the BACKGROUND_DUMP_DEST parameter is replaced by DIAGNOSTIC_DEST as part of Oracle Automatic Diagnostic Repository (ADR).

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 FIRST_TIME as "Day",  
COUNT(*) as "Number of files"  
from V$LOG_HISTORY
```

```
group by FIRST_TIME  
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.

Space Management

Free space in Oracle tablespaces is composed of newly created extents, or space that was used and freed by operations such as update and delete. If some of the free space in a tablespace is composed of extents that were used and freed, your tablespace may become fragmented. You can use the Oracle Segment Advisor (in Oracle 10g and up) to determine whether objects have unused space that can be released. Objects that are found to be fragmented can be reorganized to reclaim wasted space and compact the segment. For an overview on reclaiming wasted space, refer to the chapter *Managing Space for Schema Objects: Reclaiming Wasted Space* in the **Oracle Database Administrator's Guide** for your Oracle release.

Collecting Statistics for Databases

The HP Universal CMDB 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.

During the initial phase of HP Universal CMDB deployment, it is recommended that you collect statistics for all HP Universal CMDB objects (tables and indexes).

When working with large HP Universal CMDB environments, it is recommended that you collect statistics only for objects for which the amount of data changes significantly during the day, or for new objects that are created. Once your HP Universal CMDB system is stable, you should collect statistics once a day.

Oracle 10g Automated 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.

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

Oracle 11g Automated Statistics Collection

In Oracle 11g, the Automatic Optimizer Statistics Collection job, GATHER_STATS_JOB, is managed by the Automatic Maintenance Tasks framework instead of the Oracle Scheduler framework. Automatic Maintenance Tasks are predefined tasks that perform maintenance operations on the database. These tasks run in a maintenance window, which is a predefined time interval that is intended to occur during a period of low system load. You can manually customize maintenance windows based on the resource usage patterns of your database, or disable certain default windows from running.

According to the Oracle documentation, the preferred way to manage or change configuration of the Automatic Optimizer Statistics Collection job in Oracle 11g is to use the Automatic Maintenance Tasks screens in Enterprise Manager Database Control and Grid Control. For an overview of the Automatic Maintenance Tasks, refer to chapter *Managing Automated Database Maintenance Tasks* in the **Oracle Database Administrator's Guide**, release 11.2.

To manually collect statistics for all HP Universal CMDB objects on a database schema:

1. Log in to the HP Universal CMDB schema using SQL*Plus.
2. Run the following command:

```
Exec DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '<name of Oracle
schema>',
options => 'GATHER AUTO');
```

To collect statistics for specific schema tables and their indexes:

1. Log in to the schema using SQL*Plus.
2. For each table, collect statistics by running the following command:

```
Exec DBMS_STATS.GATHER_TABLE_STATS (ownname => '<name of Oracle
schema>', tablename
=> '<Name of table for which you want to collect statistics>',
estimate_percent => 5, cascade =>
TRUE);
```

Note:

- Cascade => True instructs the Oracle database to analyze all the indexes in the table.
- 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.

Collecting Statistics for CMDB

Unlike some databases, where queries are predefined and can be tuned according to the expected database size, the 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, such as bulk insert transactions triggered by automated DFM jobs. DFM is the process responsible for automatically detecting configuration items (CIs) and inserting them into CMDB.

Index Fragmentation

HP Universal CMDB schemas consists of Oracle B-tree indexes for enhancing searches on table columns.

It is recommended to validate the structure of the 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, nor 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 is able 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, there may be a need to rebuild some of the 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 HP Universal CMDB tables, it is recommended to rebuild any index that has more than three levels.

Index Maintenance Utility

HP Universal CMDB'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.

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.

Caution: 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. Copy the following files from the **<HP Universal CMDB root directory>\UCMDBServer\tools\dbscripts\oracle** directory to a Windows machine that has Oracle database client installed:

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

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, 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. You can lose or corrupt data 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 includes the following topics:

["Available Backup Methods" below](#)

["Oracle Recovery Manager – RMAN" on next page](#)

Available Backup Methods

This section describes the various backup methods that are available.

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 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* for your Oracle release.

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.

Note: If the Oracle Flash Recovery Area is used, then the default destination for archiving is the value set in the db_recovery_file_dest parameter.

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, HP Universal CMDB may also experience some performance degradation due to disk load.

For more information, refer to the *Oracle Backup and Recovery Guide* for your Oracle release.

Data Pump Export Import

In addition to the cold and hot physical backup methods, you can use the backup method known as Data Pump. Data Pump APIs (introduced in Oracle 10g) for moving data and metadata between databases and schemas, replaced Oracle's original export/import utilities. The original export/import utilities are no longer supported for general use as of Oracle 11g.

The Oracle Data Pump export utility dumps schema structure and contents into one or more Oracle binary files called a dump 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 Data Pump import utility. For more information on Oracle Data Pump utilities, refer to *Oracle Database Utilities* in the Oracle documentation for your release.

Note: HP Universal CMDB does not require you to use a specific backup method; however, it is recommended that your backup method accommodate HP Universal CMDB's use of more than one database user schema.

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 for your release.

Chapter 12

Configuring the Oracle Client for HP Universal CMDB

This chapter describes how to configure the Oracle Client for HP Universal CMDB database offline utilities. Installing an Oracle client on the application server is not a requirement, however it is recommended to have one to operate the various offline database scripts such as manually creating users or collecting statistics against HP Universal CMDB databases.

This chapter includes:

Oracle Client Versions and Operating System Platforms	65
Oracle Client Installation	65
Oracle Client Configuration	65

Oracle Client Versions and Operating System Platforms

The Oracle Client versions and operating system platforms that are supported and recommended for working with HP Universal CMDB are identical to the Oracle Server versions and platforms. For details, see the Oracle Server requirements table in the Server Database Requirements section of the HP Universal CMDB Support Matrix.

Oracle Client Installation

To install the Oracle Client, refer to the Oracle documentation.

If you choose the custom installation option during the installation process, ensure that you install the following components (under Oracle Client):

- Oracle Net (Including TCP/IP Adaptor)
- Oracle Database Utilities
- SQL*Plus
- Oracle Call Interface (OCI)

Oracle Client Configuration

In order to work with HP Universal CMDB, you must configure the **tnsnames.ora** file that is located in the **<ORACLE_HOME>\network\admin** directory. Make sure to specify the name or IP of the Oracle Server host machine, the Oracle Server listener port (by default, usually 1521), and the SID (by default, ORCL) or service_name. The following is an example of a **tnsnames.ora** file.

```
# TNSNAMES.ORA Network Configuration File: D:\oracle\ora81\network\admin\tnsnames.ora
# Generated by oracle configuration tools.

LONDON.MERCURY.CO.IL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = london)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = london)
    )
  )
```

It is recommended that you use the Oracle Net Configuration Assistant Oracle tool built to configure the **tnsnames.ora** file. For more information, refer to the Oracle documentation.

Ensure that the Oracle Client configuration, such as SID and port settings, matches the Oracle Server configuration. To test the connection from the Oracle Client machine to the Oracle Server machine, use the **tnsping** utility.

If you are using a RAC deployment, you can configure your tnsnames entry to work with the RAC. For an example of a RAC entry, see "[Appendix A: Support for Oracle Real Application Cluster](#)" on page 71.

Note:

- The HP Universal CMDB server accesses the Oracle Server using the JDBC thin driver. The JDBC thin driver does not support a firewall connection that is net*8/9 compliant, and therefore allows for SQL data transmission only.
- The Easy Connect Naming method eliminates the need to look up names in the **tnsnames.ora** files for TCP/IP environments. For details, refer to the Oracle Database Net Services Administrator's Guide in the Oracle documentation set of your release.

Chapter 13

Oracle Summary Checklist

This chapter contains a checklist summarizing the requirements for HP Universal CMDB support and certification.

This chapter includes:

Checklist for HP Universal CMDB Support and Certification	67
Oracle Server and Client Requirements	69
Setting Oracle Initialization Parameters	69

Note: For more information on the Oracle database configuration settings that should be used when working with Oracle Server and HP Universal CMDB, see "Oracle Server Configuration and Sizing Guidelines" on page 49

Checklist for HP Universal CMDB Support and Certification

Information is provided in this section for both supported and certified Oracle options.

The certified options are recommended for working with HP Universal CMDB. Certified options are rigorously tested by HP quality assurance personnel. Supported options are those options for which HP quality assurance personnel have successfully performed basic tests.

Option	Supported	Recommended	Remarks	For more information, see
Oracle edition	Standard, Enterprise	Enterprise		
Dedicated HP Universal CMDB server	Not necessary	Not necessary. It is recommended to dedicate an instance for HP Universal CMDB.		
Use of multiple Oracle instances	Yes	No	The configuration of all instances must match in a certified environment.	"Oracle Instances" on page 46

Option	Supported	Recommended	Remarks	For more information, see
Use of non-default port	Yes	Yes		
Undo management	Automatic; Manual	Automatic	Set UNDO_MANAGEMENT parameter to AUTO in a certified environment	
Oracle Shared Servers connection method	Yes	No	HP Universal CMDB uses a connection pool architecture. Use the dedicated server connection method in a certified environment	
Oracle replication	No full support	No		
Operating System file compression	No	No	Not supported by Oracle; Causes abnormal behavior and affects performance	
Database control files required	Greater than or equal to 2		Preferable on different disks.	
Redo log groups	Greater than or equal to 3	4	Oracle enables the software mirroring of redo log files. This is achieved by creating at least two members of redo log in each group. Members of the same group should reside on different disks.	
Character set	WE8ISO8859P1; UTF8,AL32UTF8	AL32UTF8		
OPEN_CURSORS	800	800		
Working in archive log mode	True; False	True		

Option	Supported	Recommended	Remarks	For more information, see
Autoextend option in tablespace files	Yes	No		
Locally managed data tablespace	Yes	Yes		"Locally Managed Tablespaces" on page 52
Tablespace extent management	Local uniform for TEMP tablespace	Local uniform for TEMP tablespace		"Oracle Tablespace Settings" on page 52
Automatic Segment Space Management Tablespace (ASSM)	Yes	Yes		

Oracle Server and Client Requirements

For the Oracle Server and Client versions and operating system platforms that are supported for working with HP Universal CMDB, see the Server Database Requirements section of the HP Universal CMDB Support Matrix.

Setting Oracle Initialization Parameters

For a list of recommended settings for Oracle server initialization parameters, see "Oracle Parameter Settings" on page 49.

Appendixes

Appendix A: Support for Oracle Real Application Cluster

This appendix includes:

About Oracle Real Application Cluster (RAC)	71
Single Client Access Name	72
Client Side Configuration for Oracle RAC	72
Server Side Configuration	75
Create/Connect to an HP Universal CMDB Database	75

Note: The information in this appendix applies to advanced users only.

About Oracle Real Application Cluster (RAC)

A cluster is a collection of interconnected servers that appear as one server to the end user and to applications. Oracle Real Application Cluster (RAC) is Oracle's solution for high availability, scalability, and fault tolerance. It is based on clustered servers that share the same storage.

Oracle RAC is a single Oracle database installed on a cluster of hardware servers. Each server runs an instance of the database and all the instances share the same database files.

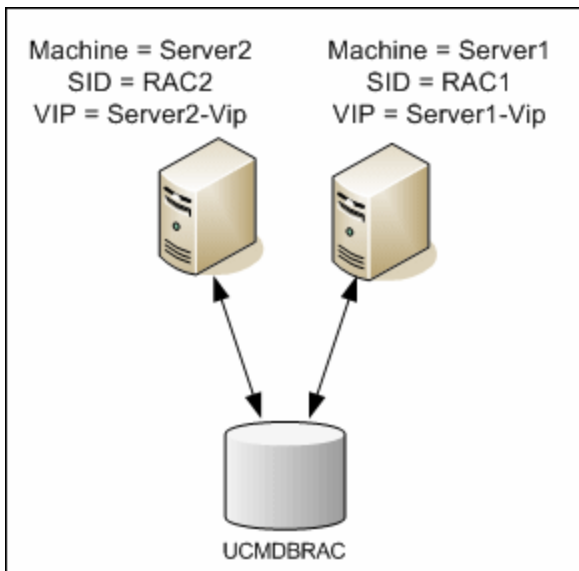
For more details about Oracle RAC, refer to the Oracle Clusterware Guide and the Oracle Real Application Clusters Administration and Deployment Guide in the Oracle documentation set of your release.

In this appendix, the following Oracle RAC example is used:

- Oracle RAC database name: UCMDBRAC
- Machine names: Server1, Server2
- On each machine there is an Oracle instance of UCMDBRAC:
 - SID on Server1: RAC1
 - SID on Server2: RAC2
- On each machine there is a virtual IP (Server1-Vip and Server2-Vip):
 - Server1-Vip is assigned to Server1
 - Server2-Vip is assigned to Server2

The virtual IP is in addition to the static IP assigned to the machine.

- The listeners on both servers are listening on the default port 1521 and support the database service UCMDBRAC.



Single Client Access Name

In release 11g, Oracle introduced the Single Client Access Name (SCAN), as a preferred access method for clients connecting to the RAC. In this method clients are not required to configure individual nodes in the RAC, rather they use a single virtual IP known as the SCAN or the SCAN VIP.

The SCAN is a single network name defined for the cluster either in your organization's Domain Name Server (DNS) or in the Grid Naming Service (GNS) that rotates between several IP addresses reflecting multiple listeners in the cluster. The SCAN eliminates the need to change clients when nodes are added to or removed from the cluster.

The SCAN and its associated IP addresses provide a stable name for clients to use for connections, independent of the nodes that make up the cluster. Database server SCAN addresses, virtual IP addresses, and public IP addresses must all be on the same subnet.

In HP Universal CMDB 10.01 when using Oracle 11g RAC, it is recommended to use the SCAN method.

Client Side Configuration for Oracle RAC

HP Universal CMDB uses DataDirect's JDBC driver to connect to regular Oracle databases and to Oracle RAC databases.

When using a RAC database, install the servers without creating the relevant UCMDB database schemas (for a description of UCMDB schemas, see ["Introduction to Preparing the Database Environment" on page 9](#)). After completing the client and server side RAC configuration, create the schemas as described in ["Create/Connect to an HP Universal CMDB Database" on page 75](#).

Make the following changes in HP Universal CMDB's configuration files before you create the management database or connect to an existing one on Oracle RAC:

1. On the UCMDB Server, create the file **<HP Universal CMDB root directory>\UCMDBServer\conf\ucmdb-tnsnames.ora**.

The format of **ucmdb-tnsnames.ora** is the same as the Oracle tnsnames.ora network configuration file:

```
<DB service name> =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <first instance virtual ip> ) (PORT = <first
instance's listener port>))
    (ADDRESS = (PROTOCOL = TCP)(HOST = <second instance virtual ip> ) (PORT =
<second instance's listener port>))
    (... entry for each instance...)
  )
  (LOAD_BALANCE = on)
  (FAILOVER = on)
)
(CONNECT_DATA =
  (SERVER = DEDICATED)
  (SERVICE_NAME = <DB service name>)
)
)
```

where:

- <DB service name> is the name of a service the listeners support. It is the same one used in the CONNECT_DATA part.
- ADDRESS_LIST contains an address entry for each node in the RAC environment. In the case of Oracle 11g using SCAN, it contains only the SCAN virtual IP. The address contains all the details needed for connecting to the node:
 - HOST contains the virtual-IP for that instance. It is important to use the virtual IP and not the static IP of the node for faster failure detection.
 - PORT is the port on which the listener is configured to listen on that given node.
 - FAILOVER set to **on** allows the driver to try to connect to another node after failing to connect to one of the nodes. The connection attempts continue until a connection is successfully established to one of the nodes or until none of the nodes can be reached.

An attempt to connect to another node takes place only if the connection to the current node fails. If the connection is refused by the node (for example, communication to the node was established, but the connection was rejected), no attempt is made to connect to another node.

Caution: Failover is for connection attempts only. If a connection fails during a database transaction, there is no failover of the transaction to another machine to continue the transaction.

- LOAD_BALANCE set to **on** instructs the driver to distribute connection requests between the nodes to prevent overloading any single node. The order in which the driver accesses the nodes is random.

- SERVER is the connection mode you want to use. There are two possible connection modes: **dedicated** and **shared**. Configure this according the Server configuration you support.
- SERVICE_NAME is the name of a service that the listeners support. It is the same one you gave in <DB service name>.

In the above example, **ucmdb-tnsnames.ora** would be configured as:

```
UCMDBRAC =
(DESCRIPTION =
(AADDRESS_LIST =
(AADDRESS = (PROTOCOL = TCP)(HOST = Server1-Vip)(PORT = 1521))
(AADDRESS = (PROTOCOL = TCP)(HOST = Server2-Vip)(PORT = 1521))
(Load_Balance = on)
(Failover = on)
)
)
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = UCMDBBSMRAC)
)
)
```

2. On the UCMDB Server, open the directory **<HP Universal CMDB root directory>\UCMDBServer\conf**. Locate the **jdbc.properties** file.

- a. Find the line starting with **cmdb.url**.
- b. Replace this with the following line:

```
cmdb.url=
jdbc:mercury:oracle:TNSNamesFile=<HP Universal CMDB root
directory>\UCMDBServer\conf\ucmdb-tnsnames.ora;TNSServerName=<SERVICE
NAME>
```

where <SERVICE NAME> is the entry in **ucmdb-tnsnames.ora**, equivalent to the RAC service name.

Note that each back slash (\) in the path of TNSNamesFile is doubled.

- c. If the file does not exist, create an empty **jdbc.properties** file under the above folder and add the following entry:

```
Oracle = ddoracle
cmdb.url = jdbc:mercury:oracle:TNSNamesFile=<HP Universal CMDB root
directory>\UCMDBServer\conf\ucmdb-tnsnames.ora;TNSServerName=<SERVICE
NAME>
```

where <SERVICE NAME> is the entry in **ucmdb-tnsnames.ora**, equivalent to the RAC service name.

- d. If your server is running the UNIX or LINUX operating system, replace all the double back slashes with single backslashes.

Server Side Configuration

In Oracle RAC, the Oracle listeners are always balancing the connection between the nodes according to either of the following algorithms:

- **Load Based (Default).** The listener redirects the connection according to the run queue length on the nodes. The least loaded node, CPU-wise, is connected.
- **Session based.** The listener balances the number of sessions between the nodes.

The first algorithm is optimized for short-lived connections, and less optimized for long-lived connections like the ones used with connection pools.

The entire load balancing is done during the connection time and not after it. This means that an established connection does not move to another node once connected.

Recommendation for the HP Universal CMDB Database

It is recommended to use the session based algorithm since HP Universal CMDB uses connection pools. One connection can be used for different purposes and has a long lifetime.

To make the Oracle listener use the session based algorithm, a listener parameter must be added to the **listener.ora** parameter file of every listener (each node has its own listener, so this change must be done on all listeners):

```
PREFER_LEAST_LOADED_NODE_<LISTENER_NAME> =OFF
```

where:

<LISTENER_NAME> is the name of the listener. The default is LISTENER_<node name>.

For example:

The default listener name on Server1 is LISTENER_SERVER1. In this case, you need to add the following to Server1's **listener.ora** file:

```
PREFER_LEAST_LOADED_NODE_LISTENER_SERVER1=OFF
```

Create/Connect to an HP Universal CMDB Database

When creating a new database schema or connecting to an existing schema in the Server and Database Configuration utility, fill all the parameters as follows:

- **Oracle Machine.** One of the Virtual IPs. In Oracle 11g, the SCAN virtual IP can be used.
- **Port.** The local listener port on the Oracle machine or the port of the SCAN listener.
- **SID.** The service name of the database.
- **Schema name and schema password.** The name and password of the existing user schema, or the name that you are giving the new user schema (for example, UCMDB_schema).

If you are creating a new user schema, you need the following additional information:

- **Admin user name and admin password (to connect as an administrator).** The name and password of a user with administrative permissions on Oracle Server (for example, a System user).
- **Default tablespace.** The name of the default tablespace you created for the user schema. For details on creating an HP Universal CMDB tablespace, see "[Manually Creating the Oracle Server Database Schemas](#)" on page 47".
- **Temporary tablespace.** The name of the temporary tablespace you assigned to the user schema. The default Oracle temporary tablespace is **temp**.

In the example, the parameters would be:

Parameter	Value
Oracle Machine	Server1-Vip
Port	1521
SID	UCMDBRAC

