

HP Universal CMDB

for the Windows and Solaris operating systems

Software Version: 8.04

Database Guide

Document Release Date: February 2010

Software Release Date: February 2010



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 Notices

© Copyright 2005 - 2010 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Intel®, Pentium®, and Intel® Xeon™ are trademarks of Intel Corporation in the U.S. and other countries.

Java™ is a US trademark of Sun Microsystems, Inc.

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

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

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 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

Table of Contents

| | |
|------------------------------------|----------|
| Welcome to This Guide | 9 |
| How This Guide Is Organized | 9 |
| Who Should Read This Guide | 10 |
| Getting More Information | 10 |

PART I: INTRODUCING THE DATABASE ENVIRONMENT

| | |
|---------------------------------------------------------------------------|-----------|
| Chapter 1: Introduction to Preparing the Database Environment..... | 13 |
| Databases in Use - Overview | 13 |
| HP Universal CMDB Sizing | 14 |
| Hardware Sizing..... | 15 |

PART II: DEPLOYING AND MAINTAINING THE MICROSOFT SQL SERVER DATABASE

| | |
|------------------------------------------------------------------------------|-----------|
| Chapter 2: Overview of Microsoft SQL Server Deployment..... | 19 |
| About Microsoft SQL Server Deployment | 19 |
| System Requirements | 20 |
| Chapter 3: Installing and Configuring Microsoft SQL Server 2000 | 23 |
| Installing Microsoft SQL Server | 23 |
| Configuring Microsoft SQL Server | 32 |
| Chapter 4: Installing and Configuring Microsoft SQL Server 2005 | 37 |
| Workflow for Microsoft SQL Server 2005 Deployment | 37 |
| Installation Prerequisites | 39 |
| Notes and Limitations | 41 |
| Checklist for Support and Certification | 41 |
| Installing Microsoft SQL Server 2005 | 42 |
| Configuring Microsoft SQL Server 2005 | 49 |
| Verifying and Modifying Server and Database Settings..... | 51 |

| | |
|-----------------------------------------------------------------------------------------------|------------|
| Chapter 5: Manually Creating the Microsoft SQL Server Databases | 57 |
| Overview of the Microsoft SQL Server Databases | 57 |
| Manually Creating the HP Universal CMDB Microsoft SQL Server Databases..... | 58 |
| Chapter 6: Using Windows Authentication to Access Microsoft SQL Server Databases | 61 |
| Enabling HP Universal CMDB to Work with Windows Authentication..... | 61 |
| Chapter 7: Creating and Configuring Microsoft SQL Server 2000 Databases | 65 |
| Creating Databases | 65 |
| Configuring Databases | 70 |
| Chapter 8: Creating and Configuring Microsoft SQL Server 2005 Databases | 77 |
| Creating Databases | 77 |
| Configuring Databases | 83 |
| Chapter 9: Maintaining Microsoft SQL Server Databases | 91 |
| Backing Up Databases | 91 |
| Database Integrity and Fragmentation | 95 |
| Monitoring Databases | 109 |
| Database Maintenance – References | 114 |
| Chapter 10: Microsoft SQL Server Sizing Guidelines | 115 |
| Server Configuration Options | 115 |
| Data File Property Settings | 116 |
| The tempdb Database Settings | 117 |

PART III: DEPLOYING AND MAINTAINING THE ORACLE SERVER DATABASE

| | |
|-------------------------------------------------------------------------------|------------|
| Chapter 11: Overview of Oracle Server Deployment | 121 |
| About Oracle Server Deployment..... | 121 |
| System Requirements | 122 |
| Chapter 12: Manually Creating the Oracle Server Database Schemas | 127 |
| Overview of the Oracle Database Schemas | 127 |
| Prerequisites..... | 128 |
| Creating Schema Default and Temporary Tablespaces..... | 129 |
| Creating Schema User Permissions | 129 |
| Creating Schemas and Schema Objects | 130 |

| | |
|-------------------------------------------------------------------|------------|
| Chapter 13: Maintaining an Oracle Server Database | 133 |
| Database Maintenance and Tuning | 133 |
| Oracle Database Backup and Recovery | 147 |
| Chapter 14: Oracle Server Configuration and Sizing | |
| Guidelines | 151 |
| Oracle Parameter Sizing..... | 152 |
| Oracle Tablespaces..... | 156 |
| Oracle Tablespace Settings | 158 |
| Using RAID Configuration | 161 |
| Chapter 15: Configuring the Oracle Client for HP Universal | |
| CMDB | 163 |
| Oracle Client Versions and Operating System Platforms | 163 |
| Oracle Client Installation | 164 |
| Oracle Client Configuration | 164 |
| Chapter 16: Oracle Summary Checklist | 167 |
| Checklist for HP Universal CMDB Support and Certification | 168 |
| Oracle Server and Client Requirements | 174 |

PART IV: APPENDIXES

| | |
|----------------------------------------------------------------------|------------|
| Appendix A: HP Universal CMDB Database Scripts | 177 |
| About HP Universal CMDB Database Scripts | 177 |
| Database Utilities–Location | 177 |
| Database Utilities–Database Clients | 178 |
| Database Utilities–Readme File | 178 |
| Appendix B: Database Schema Verification | 179 |
| About the Verification Process | 179 |
| Running the Verification Process..... | 181 |
| Creating Database Users for the Verify Procedure | 183 |
| Appendix C: Support for Oracle Real Application Cluster | 185 |
| About Oracle Real Application Cluster (RAC)..... | 185 |
| Client Side Configuration for Oracle RAC | 187 |
| Server Side Configuration | 190 |
| Create/Connect to an HP Universal CMDB Database | 191 |
| Index | 193 |

Table of Contents

Welcome to This Guide

This guide describes how to deploy and maintain Microsoft SQL Server and Oracle Server databases for use with HP Universal CMDB.

This chapter includes:

- ▶ How This Guide Is Organized on page 9
- ▶ Who Should Read This Guide on page 10
- ▶ Getting More Information on page 10

How This Guide Is Organized

The guide contains the following parts:

Part I Introducing the Database Environment

Describes the types of databases used with HP Universal CMDB.

Part II Deploying and Maintaining the Microsoft SQL Server Database

Describes how to install and configure Microsoft SQL Server 2000/2005, how to create and configure databases on Microsoft SQL Server 2000/2005, and how to maintain databases.

Part III Deploying and Maintaining the Oracle Server Database

Describes how to install Oracle Server, how to create and configure databases on Oracle Server, how to set up the HP Universal CMDB database environment for Oracle, how to configure the Oracle Client, and how to maintain databases.

Part IV Appendixes

Describes the HP Universal CMDB database scripts and the procedure for verifying your database schema.

Who Should Read This Guide

This guide is intended for the following users of HP Universal CMDB:

- HP Universal CMDB administrators
- Database administrators

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

Getting More Information

For a complete list of all online documentation included with HP Universal CMDB, additional online resources, information on acquiring documentation updates, and typographical conventions used in this guide, see the *HP Universal CMDB Deployment Guide* PDF.

Part I

Introducing the Database Environment

1

Introduction to Preparing the Database Environment

This chapter contains information on the types of databases used with HP Universal CMDB.

This chapter includes:

- Databases in Use - Overview on page 13
- HP Universal CMDB Sizing on page 14
- Hardware Sizing on page 15

Databases in Use - Overview

To work with HP Universal CMDB, you must set up the following databases:

- **CMDB.** 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.
- **CMDB History database.** For storage of changes, over time, of the CMDB configuration items (CIs). Users can view CI changes as well as view snapshots.
- **CMDB Foundations database.** For storage of information on various application components such as Infrastructure Settings data. It also includes information on which service is running on which server.

You can set up CMDB, CMDB history, and CMDB Foundations 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.”

If you are working with an Oracle Server database, refer to Part III, “Deploying and Maintaining the Oracle Server Database.”

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

Note:

- ▶ If HP Universal CMDB is installed on a Solaris platform, the databases must be set up on an Oracle Server.
 - ▶ 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 “Working in Non-English Locales” in *Reference Information*.
-

HP Universal CMDB Sizing

HP Universal CMDB database configuration requirements are dependent on the amount of data generated by HP Universal CMDB. A small HP Universal CMDB deployment requires a database that can hold about 100,000 CMDB objects and links. A large HP Universal CMDB deployment requires a database containing one million or more CMDB objects and links.

Hardware Sizing

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 |
|------------|---------------------------------------|----------------------------------------|
| Standard | 1 dual core | Minimum: 2G RAM Recommended: 4G RAM |
| Large | Minimum 2 dual core or 1 quad core | Minimum: 4G RAM Recommended: 8G RAM |

For Microsoft SQL software sizing requirements, see “Software Requirements” on page 21.

For Oracle software sizing requirements, see “Software Requirements” on page 123.

Part II

Deploying and Maintaining the Microsoft SQL Server Database

2

Overview of Microsoft SQL Server Deployment

You can set up CMDB, CMDB history, and Foundations databases on Microsoft SQL Server. This chapter describes the following topics related to deploying Microsoft SQL Servers for use with HP Universal CMDB:

This chapter includes:

- ▶ About Microsoft SQL Server Deployment on page 19
- ▶ System Requirements on page 20

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 2000, see “Installing and Configuring Microsoft SQL Server 2000” on page 23.

For details on installing and configuring Microsoft SQL Server 2005, see “Installing and Configuring Microsoft SQL Server 2005” on page 37.

▶ **Create databases on Microsoft SQL Server to store CMDB, CMDB history, and Foundations data.**

For details on creating Microsoft SQL 2000 database, see “Creating and Configuring Microsoft SQL Server 2000 Databases” on page 65.

For details on creating Microsoft SQL 2005 database, see “Creating and Configuring Microsoft SQL Server 2005 Databases” on page 77.

Note: You can allow HP Universal CMDB to create databases for you, or you can create databases yourself using the CREATE DATABASE statement or Microsoft SQL Server management tools.

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

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” on page 20
- ▶ “Software Requirements” on page 21

Hardware Requirements

For HP Universal CMDB hardware sizing guidelines, see “Hardware Sizing” on page 15.

Software Requirements

The following table describes the Microsoft SQL Servers supported for working with HP Universal CMDB:

| Database Release | | | Microsoft Data Access Components (MDAC) on HP Universal CMDB servers |
|----------------------------------------------|---------|----------------|--------------------------------------------------------------------------------------------|
| Version | BIT Set | Service Pack | |
| Microsoft SQL Server 2005 Enterprise Edition | 32 BIT | Service Pack 3 | 2.8 SP1 Refresh – Installed automatically with your HP Universal CMDB server installation. |
| Microsoft SQL Server 2005 Enterprise Edition | 64 BIT | Service Pack 3 | |
| Microsoft SQL Server 2000 Enterprise Edition | 32 BIT | Service Pack 4 | 2.5; 2.52; 2.61; 2.62; 2.7 SP1 Refresh |

Note:

- Only supported service packs should be installed, with latest patches.
 - Consult the Microsoft documentation for supported platforms.
 - To check which version of MDAC is installed on your machine, download and run the component checker tool from:
<http://www.microsoft.com/downloads/details.aspx?FamilyID=8f0a8df6-4a21-4b43-bf53-14332ef092c9&displaylang=en>.
-

Examples of Tested Deployments

The following table details those deployment environments that were tested by HP.

| Database Release | | | Operating System | Microsoft Data Access Components (MDAC) on HP Universal CMDB servers |
|----------------------------------------------|---------|----------------|------------------------------------------------|--------------------------------------------------------------------------------------------|
| Version | BIT Set | Service Pack | | |
| Microsoft SQL Server 2005 Enterprise Edition | 32 BIT | Service Pack 3 | Windows 2003 Enterprise Edition Service Pack 2 | 2.8 SP1 Refresh – Installed automatically with your HP Universal CMDB server installation. |
| Microsoft SQL Server 2005 Enterprise Edition | 64 BIT | Service Pack 3 | Windows 2003 Enterprise Edition Service Pack 2 | |

3

Installing and Configuring Microsoft SQL Server 2000

This chapter contains information on the Microsoft SQL Server 2000 installation procedure and configuration settings.

This chapter includes:

- Installing Microsoft SQL Server on page 23
- Configuring Microsoft SQL Server on page 32

Installing Microsoft SQL Server

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

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

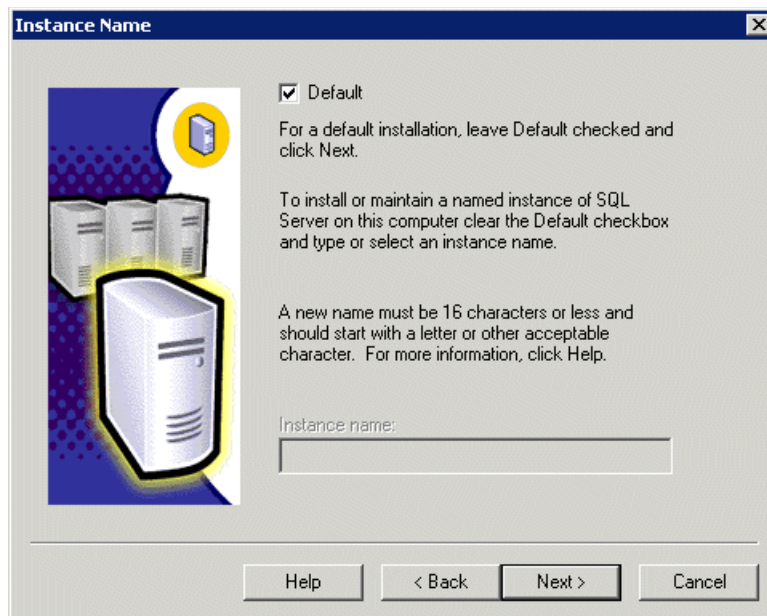
- Instance Name Dialog Box
- Setup Type Dialog Box
- Select Components Dialog Box
- Services Accounts Dialog Box
- Authentication Mode Dialog Box

- Collation Settings Dialog Box
- Network Libraries Dialog Box

Instance Name Dialog Box

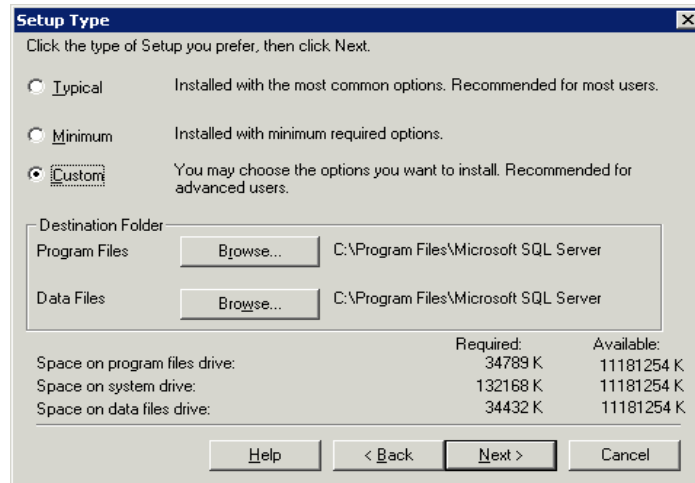
You can install multiple instances of Microsoft SQL Server 2000 on the same machine. Each instance has its own Microsoft SQL Server and Microsoft SQL Agent services, and is completely independent of the other instances.

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



Setup Type Dialog Box

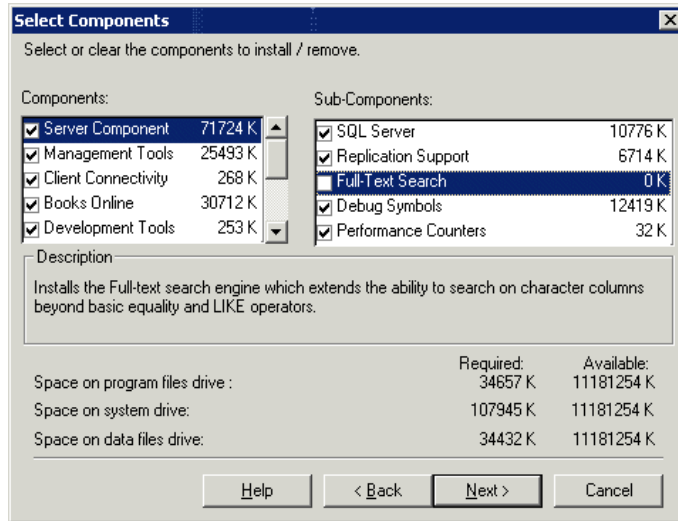
In the Setup Type dialog box, you must select the **Custom** option. If you do not select this option, you cannot view the Component Selection, Collation Settings, and Network Libraries dialog boxes, in which you must change certain default options that are not appropriate for working with Microsoft SQL Server in conjunction with HP Universal CMDB.



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

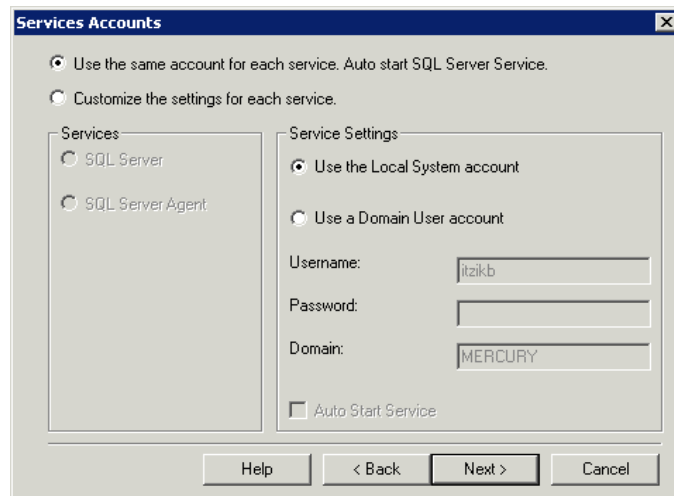
Select Components Dialog Box

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



Services Accounts Dialog Box

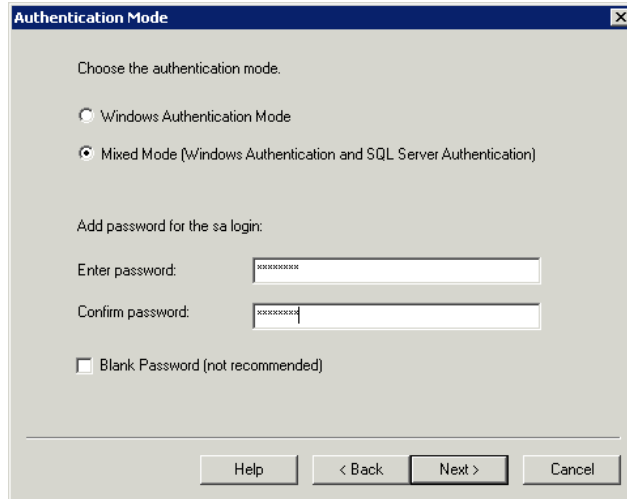
In the Services Accounts dialog box, you select the services and enter the appropriate service settings for your Microsoft SQL Server. If you want the Microsoft SQL Server service (MSSQLServer) to perform activities outside the local machine—for example, backing up files to a shared network directory—select **Use a Domain User account** and specify the user name, password, and domain of a user that is a member of the local machine's administrator group, and that has the appropriate permissions for network resources. If all Microsoft SQL Server activities are limited to the local machine, select **Use the Local System account**, which provides administrative privileges to the Microsoft SQL Server service for the local machine only.



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

Authentication Mode Dialog Box

In the Authentication Mode dialog box, you select the type of authentication you want Microsoft SQL Server to use. HP Universal CMDB works with both Windows authentication and Microsoft SQL Server authentication (recommended). To enable Microsoft SQL Server authentication, select **Mixed Mode** and provide a complex password for the **sa** login.



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

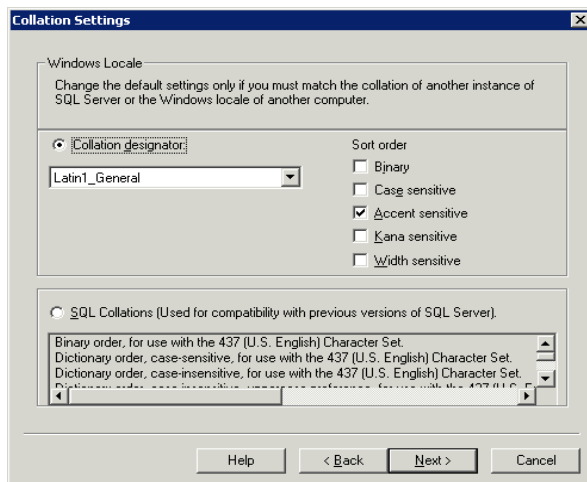
The sa user has system administrator privileges and can therefore perform all actions within the Microsoft SQL Server. Similarly, the sa user can perform all operating system/network actions in the context of the MSSQLServer service account, by using the xp_cmdshell extended procedure.

Collation Settings Dialog Box

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

Select one of the following two options:

- ▶ **Windows Locale.** Select this option only if you do not need to create backward compatibility with previous versions of Microsoft SQL Server (for example, for replication purposes). If you select this option, set the following for HP Universal CMDB certification:
 - ▶ **Collation Designator.** The language for regular character data types (char, varchar, text). Select the default option, **Latin1_General**, to support English or select a different language if you plan to store non-English data in your database. The selected language should match the language of the Windows operating system.
 - ▶ **Sort Order.** HP Universal CMDB does not support binary sort order and case sensitivity with Microsoft SQL Server 2000. The binary check box should be cleared. Accent, kana and width sensitivity can be selected according to the relevant data language requirements.
- ▶ **SQL Collations.** Select this option if the current version of Microsoft SQL Server must be compatible with previous versions, for example, if you are replicating data between servers.



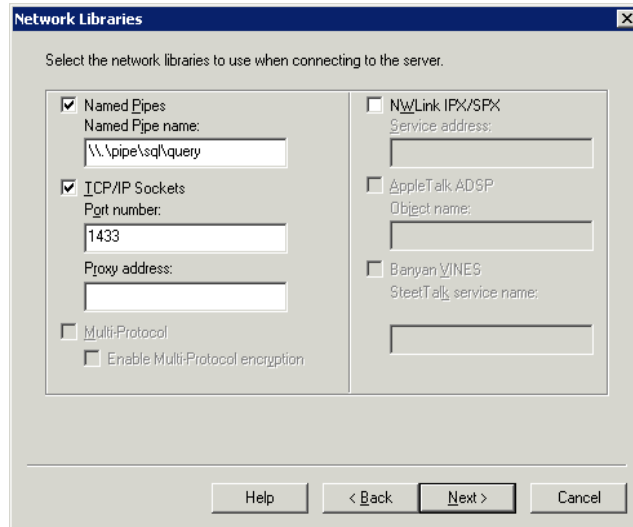
The above settings affect only the system databases and serve as the default settings for user databases. Databases can have different collation settings from the server's default settings, and a table column can have different collation settings from the database's default settings. Because of the flexibility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

Important: 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.

Network Libraries Dialog Box

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

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



You can configure the client machine to connect to Microsoft SQL Server using both TCP/IP and Named Pipes. However, for HP Universal CMDB certification, you should configure the client machine to connect to Microsoft SQL Server using TCP/IP only.

Note: All network libraries support Windows authentication and SSL data encryption.

Configuring Microsoft SQL Server

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

- ▶ “Service Configuration Options” on page 32
- ▶ “Server Configuration Options” on page 32

Service Configuration Options

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

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

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

Server Configuration Options

Most server configuration options are dynamically configured by Microsoft SQL Server (usually set to 0). For HP Universal CMDB certification, you should not change the default options unless you are instructed to do so by HP Software Support.

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

The following table describes the configuration options available in Microsoft SQL 2000, their default settings, and the settings required for HP Universal CMDB certification:

| Configuration Option | Default | HP Universal CMDB Certification in Microsoft SQL Server 2000 |
|----------------------------------|---------------|-----------------------------------------------------------------|
| affinity mask | 0 | Default |
| allow updates | 0 | Default |
| awe enabled (Enterprise Edition) | 0 | Default, unless the server needs to access 4 to 64 GB of memory |
| c2 audit mode | 0 | Default |
| cost threshold for parallelism | 5 | Default |
| cursor threshold | -1 | Default |
| default full-text language | 1033 | Default |
| default language | 0 | Default |
| fill factor | 0 | Default |
| index create memory | 0 | Default |
| lightweight pooling | 0 | Default |
| locks | 0 | Default |
| max async IO | 32 | N/A |
| max degree of parallelism | 0 | Default |
| max server memory | 2,147,483,647 | Default |
| max text repl size | 65,536 | Default |
| max worker threads | 255 | Default |
| media retention | 0 | Default |
| min memory per query | 1024 | Default |
| min server memory | 0 | Default |
| Using Nested Triggers | 1 | Default |

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

You can view all of the above configuration options by running the following:

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

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

For large installations, you may need to set the **awe enabled** option. For more information, see Chapter 10, “Microsoft SQL Server Sizing Guidelines.”

Note: 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, you should not change the default memory settings. You should allow the Microsoft SQL Server to manage memory dynamically (except when you configure **awe enabled** support).

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

4

Installing and Configuring Microsoft SQL Server 2005

This chapter contains information on the Microsoft SQL Server 2005 installation procedure and configuration settings.

This chapter includes:

- ▶ Workflow for Microsoft SQL Server 2005 Deployment on page 37
- ▶ Installation Prerequisites on page 39
- ▶ Notes and Limitations on page 41
- ▶ Checklist for Support and Certification on page 41
- ▶ Installing Microsoft SQL Server 2005 on page 42
- ▶ Configuring Microsoft SQL Server 2005 on page 49
- ▶ Verifying and Modifying Server and Database Settings on page 51

Workflow for Microsoft SQL Server 2005 Deployment

To deploy Microsoft SQL Server 2005 for use with HP Universal CMDB, perform the following steps:

1 Review Microsoft SQL Server sizing guidelines

For details, see “Microsoft SQL Server Sizing Guidelines” on page 115.

2 Review installation prerequisites

For details, see “Installation Prerequisites” on page 39.

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 41.

4 Install Microsoft SQL Server 2005 Enterprise edition

For details, see “Installing Microsoft SQL Server 2005” on page 42.

5 Configure Microsoft SQL Server 2005

For details, see “Configuring Microsoft SQL Server 2005” on page 49.

6 Create CMDB, CMDB History, and Foundations databases on Microsoft SQL Server 2005

For details, see “Creating Databases” on page 77.

7 Configure CMDB, CMDB History, and Foundations databases

For details, see “Configuring Databases” on page 83.

8 Verify Microsoft SQL Server 2005 and databases

For details, see “Verifying and Modifying Server and Database Settings” on page 51.

9 Set up Windows authentication

For details, see “Using Windows Authentication to Access Microsoft SQL Server Databases” on page 61. 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 2005:

► Software

| Software | Version Needed |
|-----------------------------------------|-------------------|
| Microsoft Windows Installer | 3.1 or later |
| Microsoft Data Access Components (MDAC) | 2.8, SP1 or later |

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

► Operating System

| SQL Server 2005, Enterprise Edition | Operating System | Service Pack Needed |
|-------------------------------------|--------------------------------------------|---------------------|
| 32-bit | Windows 2000 Server | SP4 |
| | Windows 2000 Advanced Server | SP4 |
| | Windows 2003 Server | SP1 |
| | Windows 2003 Enterprise Edition | SP1 |
| 64-bit | Windows 2003 64-bit X64 Standard Edition | SP1 |
| | Windows 2003 64-bit X64 Enterprise Edition | SP1 |

► Free disk space

There must be at least 1.4 GB free space on the operating system drive. This amount of disk space is needed during the installation. Most of this space is released when the installation is finished.

➤ **Database file placement**

For improved performance, it is recommended to spread out the databases (CMDB, CMDB History, Foundations) among several disks or raid systems.

➤ **Memory**

There must be at least 1 GB of RAM.

➤ **Backup**

If you are already have an instance of SQL Server on the computer where SQL Server 2005 is to be installed, back up your current installation of SQL Server.

➤ **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 2005 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 databases manually, supply HP Universal CMDB with a user account that is a part of **db_datareader**, **db_datawriter** and **db_ddladmin** in each HP Universal CMDB database.

➤ **Miscellaneous**

- Verify that the disk where Microsoft SQL Server 2005 is to be installed is not compressed.
- Disable antivirus software and any other applications that may interfere while installing Microsoft SQL Server 2005.

Notes and Limitations

- ▶ If you already have Microsoft SQL 2000 installed as the default instance on the machine that you want to install Microsoft SQL 2005, you can not install Microsoft SQL 2005 as a default instance but only as a named instance.

You can install only one instance as the default instance.

- ▶ 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 2005 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.

Checklist for Support and Certification

The following checklist summarizes the server and database configuration options that are supported and certified for working with HP Universal CMDB:

| Subject | Microsoft SQL Server 2005 | |
|---------------------|---------------------------|-------------|
| | Supported | Recommended |
| Instances | Default, Single | |
| Authentication Mode | Mixed | |

| Subject | Microsoft SQL Server 2005 | |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Supported | Recommended |
| Collation | Case-Insensitive. HP Universal C MDB 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. |
| 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 2005

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.

To install Microsoft SQL Server 2005:

- 1 Run <MS SQL Server root directory>\x86\setup\setup.exe setup utility.

- 2 The End User License Agreement dialog box opens. Read the agreement and select **I accept the licensing terms and conditions** check box if you agree to the terms. Click **Next**.

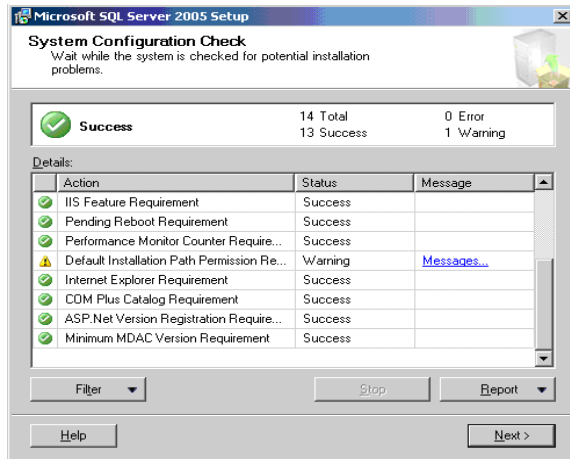
If you do not agree to the licensing terms, the installation process exits.

- 3 The Installing Prerequisites dialog box opens. If needed, the setup utility installs the following:

- Microsoft Windows .NET Framework 2.0
- Microsoft SQL Server Native Client
- Microsoft SQL Server setup support files

- 4 The setup utility checks the system configuration.

In the **status** column you should see only **Success** or **Warning** status. There should be no **Failure** status. If any items have **Warning** status, click in the message column to see the cause.



If necessary, correct any problems and rerun the setup utility.

Click **Next** to continue.

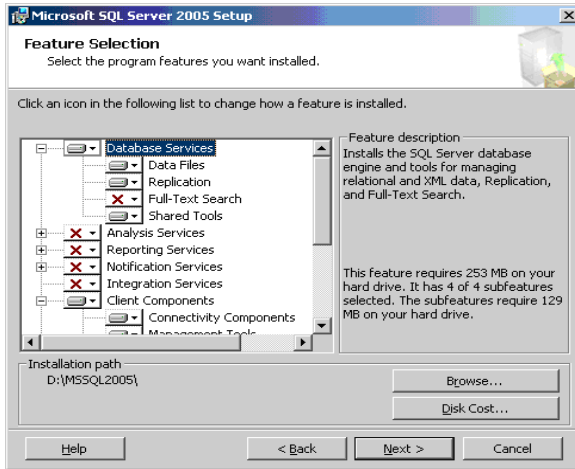
- 5 The Registration Information dialog box opens. The Product Key is automatically taken from the operating system. Accept the default information and click **Next**.

6 The Components to Install dialog box opens. Select the following check boxes:

- **SQL Server Database Services**
- **Workstation components, Books Online and development tools**

Click **Advanced**.

7 The Feature Selection dialog box opens and displays the features that you selected. A red **X** marks features that were not selected.



8 Under Database Services, do the following:

- a remove **Full Text Search** from the list since HP Universal CMDB does not use this indexing search feature
- b set the directory where data files, executables, and user and system database files are stored

Important: The installation path of each of the chosen features (Database Services, Client Components, and Documentation) must be different than the path of the system databases. For example, the directory for Microsoft SQL server executables must not be the directory where the Microsoft SQL system databases are stored.

These settings cannot be changed later on. At a later stage you can relocate tempdb files (using the ALTER DATABASE <tempdb> MODIFY FILE command) and user database files (using the detach, move, attach commands), but you cannot move **master**, **msdb**, and **model** files. Although these system database files are small, they are essential to Microsoft SQL Server. It is recommended to store these files on a fault-tolerant disk.

c Under Destination Folder, do the following:

- ▶ Set the Program Files directory where Microsoft SQL Server executables are stored
- ▶ Set the Data Files directory where system databases and user databases that do not have a specified file location are stored.

Be 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.

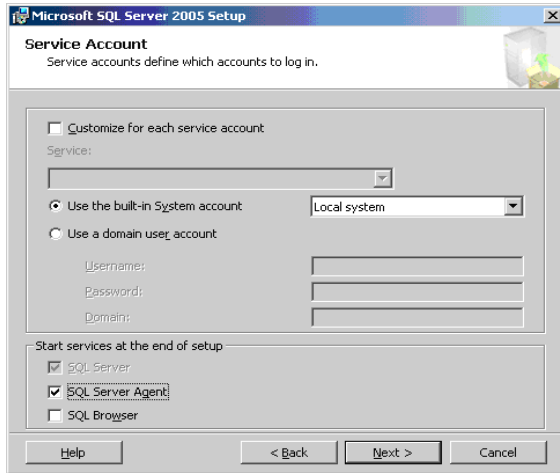
d Click **Next**.

- 9** The Instance Name dialog opens. You can install multiple instances of Microsoft SQL Server 2005 on the same machine. Each instance is completely independent of the other instances. There can be only one instance named as the default instance. All other instances must be installed as named instances. For details on named instances, see “Port Allocation Options for Named Instances” on page 54.

You access a default instance by specifying the server name or IP address.

Click **Next**.

- 10** The Service Account dialog box opens. In the **Start services at the end of setup** pane, select **SQL Server Agent** check box.



In the Services Accounts dialog box, you select the services and enter the appropriate service settings for your Microsoft SQL Server by doing either of the following:

- ▶ 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.

Click **Next**.

- 11** The Authentication Mode dialog box opens. In the Authentication Mode dialog box, you select the type of authentication you want Microsoft SQL Server to use.

HP Universal CMDB works with both Windows authentication and with Microsoft SQL Server authentication (recommended). To enable Microsoft SQL Server authentication, do the following:

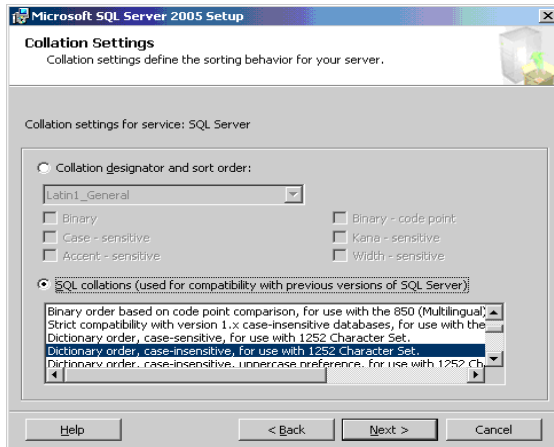
- a** Choose **Mixed Mode (Windows Authentication and SQL Server Authentication)**.
- b** Enter the password for user **sa**.

Note: To further secure your Microsoft SQL Server 2005, it is important to enter a password.

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

Click **Next**.

- 12** The Collation Settings dialog box opens. In the Collation Settings dialog box, you set the language, dictionary/binary order, and sensitivity of character data types. Follow the recommendations found in the checklist in “Checklist for Support and Certification” on page 41.



The above settings affect only the system databases and serve as the default settings for user databases. Databases can have different collation settings from the server’s default settings, and a table column can have different collation settings from the database’s default settings. Because of the flexibility in collation management in Microsoft SQL Server 2005, you can restore or attach a database that has different collation settings.

Important: 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.

Click **Next**.

- 13** The Ready to Install dialog box opens listing the components to be installed. Click **Install** to proceed with the installation or click **Back** to change any selections.
- 14** A Setup Progress dialog box opens to show each product's installation status. Click **Next**.
- 15** The Microsoft SQL Server 2005 Setup message box opens with installation summary information.
- 16** Install the latest service pack for Microsoft SQL Server 2005 available at the Microsoft site at <http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=ae7387c3-348c-4faa-8ae5-949fdfbe59c4>.
- 17** Restart the machine with the newly-installed Microsoft SQL Server 2005.

For information on modifying or verifying the authentication mode after installation, see “Verifying and Modifying Server and Database Settings” on page 51.

Configuring Microsoft SQL Server 2005

This section describes the service and server options you can configure once you have installed Microsoft SQL Server 2005:

This section includes the following topics:

- “Service Configuration Options” on page 49
- “Server Configuration Options” on page 50

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 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.

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

Note:

- ▶ The parameters **Agent XPs** and **xp_cmdshell** are set to **1** (enabled) to configure Microsoft SQL Server 2005 as much like Microsoft SQL Server 2000 as possible.
 - ▶ **SQL Mail XPs** and **Web Assistant Procedures** are not supported in later versions of Microsoft SQL Server.
-

To view the available configuration options:

In Management Studio, use the following command to verify your instance configuration:

```
EXEC sp_configure
```

Note: This command returns a result set with five columns. The **run_value** column shows the current parameter value while the **config_value** shows the configured value (the values may differ if the RECONFIGURE command was not run).

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 2005 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 Required results: Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive |

| Server/Database Setting | How to Verify/Modify the Setting |
|-----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Network Libraries | <p>On the Server, select Start > Programs > Microsoft SQL Server 2005 > Configuration Tools > Configuration Manager.</p> <p>At SQL Native Client.. choose client protocols and ensure that the selected protocols are 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. |
| View or change server configuration options | <ul style="list-style-type: none"> ▶ To allow the viewing of all options, do: EXEC sp_configure 'show advanced options', 1 reconfigure with override ▶ To view the current values, do: EXEC sp_configure ▶ To change a setting, do: 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 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> |

| Server/Database Setting | How to Verify/Modify the Setting |
|-------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Check whether the HP Universal CMDB database user has enough permissions in the database</p> | <ol style="list-style-type: none"> 1 Log in to the Microsoft SQL Server Management Studio with the user name you want to check. 2 Change the context of the database to the required database. 3 Open a new query and do the following in each database: <pre style="margin-left: 20px;">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> |
| <p>Data and log file destination directory is not compressed (only in NTFS)</p> | <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: EXEC sp_helpdb <database name> ▶ To change the database properties, run: ALTER DATABASE <database name> SET <option> <value> ▶ To change the database file properties, run: ALTER DATABASE <database> MODIFY FILE (name = <filename>, <property> = <value>) You can also view or change these properties from the Database Properties dialog box in the Enterprise Manager. |
| Microsoft SQL Server service pack version and edition | Log into Management Studio and make the following query: <pre>select @@version</pre> The result for SQL 2005 SP1 should start with: Microsoft SQL Server 2005 - 9.00.2047.00 |

Port Allocation Options for Named Instances

When working with a named instance in SQL Server, there are two options for the instance's port allocation: dynamic port and static port.

- ▶ **Dynamic Port.** When working with a dynamic port, a new port is assigned 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 \

- ▶ **Static Port.** When working with a static port, a specific port is assigned 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 2005 > 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 2005 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`.

5

Manually Creating the Microsoft SQL Server Databases

This chapter describes the manual creation of the Microsoft SQL Server databases required by HP Universal CMDB.

This chapter includes:

- Overview of the Microsoft SQL Server Databases on page 57
- Manually Creating the HP Universal CMDB Microsoft SQL Server Databases on page 58

Overview of the Microsoft SQL Server Databases

HP Universal CMDB uses the following databases for its persistency:

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

During the HP Universal CMDB setup, new databases can be set up automatically by the setup procedure, or already existing databases can be used. Existing databases 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 “Introduction to HP Universal CMDB” in the *HP Universal CMDB Deployment Guide* PDF.

Manually Creating the HP Universal CMDB Microsoft SQL Server Databases

Before creating the databases, the Microsoft SQL Database Server must be installed according to the details in this guide. For further details on installing and configuring the Microsoft SQL Database Server, see “Configuring Microsoft SQL Server” on page 32 for Microsoft SQL Server 2000 and “Configuring Microsoft SQL Server 2005” on page 49 for Microsoft SQL Server 2005.

If the HP Universal CMDB databases are manually created prior to running the HP Universal CMDB setup, during the setup procedure, you select the option to use existing databases and enter the relevant data about the existing databases for which you are prompted. The data you enter is written to the configuration files that are used by the JDBC driver to connect to the existing databases.

This section includes the following topics:

- ▶ “Creating the Databases” on page 58
- ▶ “Creating the Objects” on page 59

Creating the Databases

The database administrator should create three Microsoft SQL Server databases; one for CMDB Foundations, one for CMDB, and one for CMDB History.

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

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

For details on creating databases manually, see Chapter 7, “Creating and Configuring Microsoft SQL Server 2000 Databases” for Microsoft SQL Server 2000 and Chapter 8, “Creating and Configuring Microsoft SQL Server 2005 Databases” for Microsoft SQL Server 2005.

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

Creating the Objects

Objects are created using SQL scripts that are run against the databases that were previously created. For details, see “Creating the Databases” on page 58.

If you are running Microsoft SQL Server 2005, the script is located in the <HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\ms_2005 directory. Run the script manually using Management Studio or SQLCMD. If you are running Microsoft SQL Server 2000, the script is located in the <HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\ms_2000 directory. Run the script manually using Query Analyzer or OSQL.

Creating the CMDB Objects

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

```
create_cm_tables_cmdb_ms.sql
```

Creating the CMDB History Objects

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

```
create_cm_tables_cmdb_history_ms.sql
```

Creating the CMDB Foundations Objects

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

```
foundations_sql_dbobjects_create.sql
```

It is strongly recommended that you run the database schema verify program after creating the schemas to verify that your database is configured properly. For information on the verification process, see Appendix B, “Database Schema Verification.”

6

Using Windows Authentication to Access Microsoft SQL Server Databases

Unless configured otherwise, HP Universal CMDB uses Microsoft SQL Server authentication to access 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 on page 61

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 any of the HP Universal CMDB databases (CMDB, CMDB History, and CMDB Foundations).

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 UCMDB 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 Server and Database 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” on page 62
- ▶ “Launching the UCMDB Server service with a Windows User” on page 62
- ▶ “Creating or Connecting to a Microsoft SQL Database Using Windows Authentication” on page 64

Configuring Microsoft SQL Server to Use Windows Authentication

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

In the **Server Roles** tab, select **System Administrators** and click **OK**.

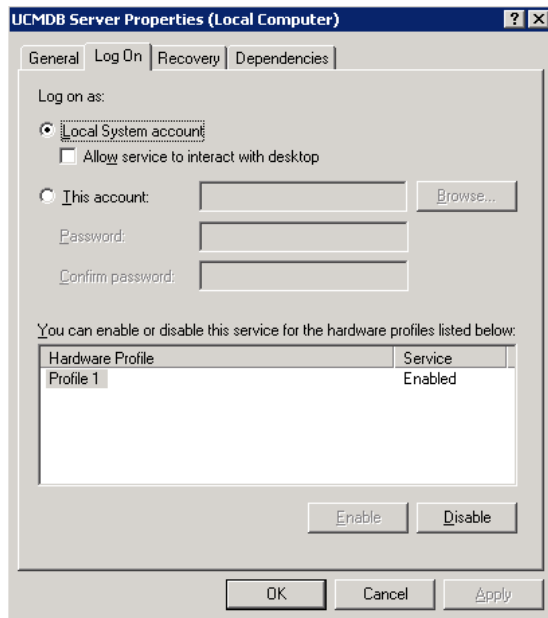
Launching the UCMDB Server service with a Windows User

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 “Changing the HP Universal CMDB Service User” in the *HP Universal CMDB Deployment Guide* PDF.

To change the UCMDB Server service user:

- 1 Disable HP Universal CMDB (**Start > Programs > UCMDB > Stop 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 > UCMDB > Start UCMDB Server**).

Creating or Connecting to a Microsoft SQL Database Using Windows Authentication

You create or connect to a database using the Server and Database Configuration utility. To create or connect to a Microsoft SQL database using Windows authentication, you must select this option within the Server and Database Configuration utility. For details on using the Server and Database Configuration utility, see “Connect to an Existing Microsoft SQL Server Database” in the *HP Universal CMDB Deployment Guide* PDF.

7

Creating and Configuring Microsoft SQL Server 2000 Databases

This chapter describes the creation and configuration of HP Universal CMDB databases on a Microsoft SQL Server 2000.

This chapter includes:

- Creating Databases on page 65
- Configuring Databases on page 70

Creating Databases

This section includes the following topics:

- “Database Permissions” on page 65
- “Database File Layout” on page 67
- “System Databases” on page 70

Database Permissions

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.

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

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

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

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

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

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

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

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

```
SELECT USER_NAME()
```

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

It is recommended that you place the data and log files on separate disk subsystems. 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.

The data files should be placed on a RAID 0+1 system for optimal performance.

Note: It is recommended not to place data or log files on the same disk that stores the page (swap) file.

File and Database Properties

When you create a database you can specify the following five 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. For more information, see Chapter 10, “Microsoft SQL Server Sizing Guidelines.”

Note: 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. In general, most systems are able to allocate 100 MB in less than 30 seconds, which is the common client timeout setting.

For more information on this problem, refer to Microsoft Knowledge Base Article - 305635 (<http://support.microsoft.com/default.aspx?scid=kb:en-us:Q305635>).

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, refer to Microsoft SQL Server 2000 Books Online at <http://www.microsoft.com/downloads/details.aspx?FamilyID=a6f79cb1-a420-445f-8a4b-bd77a7da194b&displaylang=en>.

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

- **Partial Restoring.** Microsoft SQL Server 2000 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 2000 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 Chapter 10, "Microsoft SQL Server Sizing Guidelines." 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.

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

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

Configuring Databases

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

This section includes the following topics:

- ▶ Database File Configuration
- ▶ Database Configuration Options

Database File Configuration

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

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.

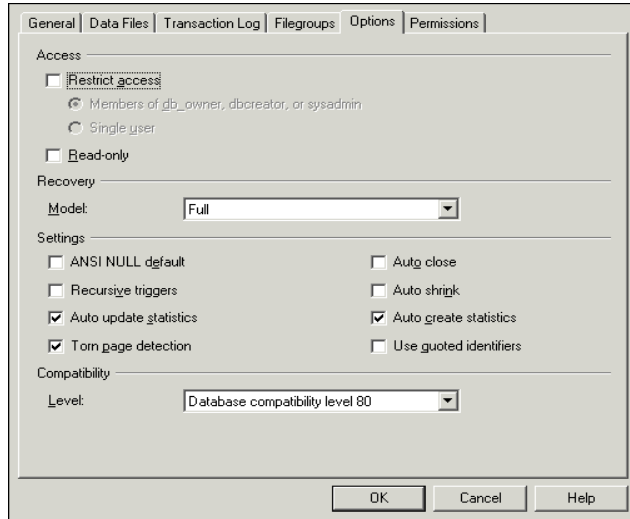
Changing File Properties

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 "Database Permissions" on page 65.

Database Configuration Options

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

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



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

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

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

| Configuration Option | Description | Default | HP Universal CMDB Certification in Microsoft SQL Server 2000 |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|---------------------------------------------------------------------------------------------------|
| Restrict access | Only single users or members of the db_owner, dbcreator, or sysadmin groups can access the database. | Not set (MULTI_USER) | MULTI_USER |
| Read only | Database is read only | Not set (READ_WRITE) | READ_WRITE |
| Recovery | The database recovery model determines the recovery capabilities by controlling the amount of bulk operation logging (such as Select into, Bulk, Insert, Create index, LOB 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) |
| 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 2000 |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------|
| Select into/bulk copy | Allows the use of minimally logged Select into/bulk copy operations | Not set | N/A |
| ANSI NULL default (see note below) | Specifies whether the database columns are defined as NULL or NOT NULL, by default | Not set | Not set |
| Recursive triggers | Specifies whether recursive triggers are supported | Not set | Not set |
| Auto update statistics | Specifies whether out-of-date statistics required by a query for optimization are built automatically during optimization | Set | Set |
| Auto create statistics | Specifies whether missing statistics required by a query for optimization are built automatically during optimization | Set | Set |
| Torn page detection | Specifies whether incomplete pages can be detected | Set | Set |

| Configuration Option | Description | Default | HP Universal CMDB Certification in Microsoft SQL Server 2000 |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Auto close | Specifies whether the database shuts down after its resources are freed and all users exit | Not set | Not set Note: If set, it may take a long time for the database to allocate resources every time a user connects, after the database is closed. |
| Auto shrink | Specifies whether the database is automatically shrunk every hour, leaving 25% of free space | Not set | Not set Note: If set, constant growth/ shrinkage may cause file system fragmentation. |
| 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 |
| Compatibility level | The version of Microsoft SQL Server that the database appears to be (for the application) | 80 | 80 |

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

The following table summarizes the characteristics of each recovery model:

| Model/ Support | 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>
```

8

Creating and Configuring Microsoft SQL Server 2005 Databases

This chapter describes the creation and configuration of HP Universal CMDB databases on a Microsoft SQL Server 2005.

This chapter includes:

- Creating Databases on page 77
- Configuring Databases on page 83

Creating Databases

This section includes the following topics:

- “Database Permissions” on page 77
- “Database File Layout” on page 78
- “System Databases” on page 82

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: 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

Important:

- ▶ 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. For more information, see “Microsoft SQL Server Sizing Guidelines” on page 115.

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.

In general, most systems are able to allocate 100 MB in less than 30 seconds, which is the common client timeout setting.

For more information on this problem, refer to Microsoft Knowledge Base Article - 305635 (<http://support.microsoft.com/default.aspx?scid=kb:en-us:Q305635>).

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 2005 Books Online at <http://www.microsoft.com/technet/prodtechnol/sql/2005/downloads/books.mspx>.

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

- **Partial Restoring.** Microsoft SQL Server 2005 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 2005 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 “Microsoft SQL Server Sizing Guidelines” on page 115.

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 Databases

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

This section includes:

- ▶ Database File Configuration
- ▶ Database Configuration Options

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 2005 Books Online at <http://www.microsoft.com/technet/prodtechnol/sql/2005/downloads/books.msp>)

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 Databases” on page 77.

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 2005 |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| ANSI NULL default (see note below) | Specifies whether the database columns are defined as NULL or NOT NULL, by default | Not set | Not set |
| Auto close | Specifies whether the database shuts down after its resources are freed and all users exit | Not set | Not set Note: If set, it may take a long time for the database to allocate resources every time a user connects, after the database is closed. |
| Auto 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 2005 |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|---------------------------------------------------------------------------------------------------|
| 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) | 90 | 90 |
| 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) |

| Configuration Option | Description | Default | HP Universal CMDB Certification in Microsoft SQL Server 2005 |
|-----------------------|------------------------------------------------------------------------------------------------------|----------------------|--------------------------------------------------------------|
| 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 |
| Select into/bulk copy | Allows the use of minimally logged Select into/bulk copy operations | Not set | N/A |
| Torn page detection | Specifies whether incomplete pages can be detected | Set | Set |

| Configuration Option | Description | Default | HP Universal CMDB Certification in Microsoft SQL Server 2005 |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------|
| Truncate log on checkpoint | Automatically marks inactive portions of log for reuse on checkpoint | Not set | N/A |
| 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 2005 Books Online at

<http://www.microsoft.com/technet/prodtechnol/sql/2005/downloads/books.msp>.

9

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 on page 91
- ▶ Database Integrity and Fragmentation on page 95
- ▶ Monitoring Databases on page 109
- ▶ Database Maintenance – References on page 114

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 page 92
- “Differential Backup” on page 93
- “Log Backup” on page 94
- “File/File Group Backup” on page 94
- “Maintenance Plan” on page 94
- “Transaction Log Issues” on page 95

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. Note that the differential backup performance of Microsoft SQL Server 2000/2005 is superior to that of Microsoft SQL Server 7.0. The difference in performance is due to the fact that Microsoft SQL Server 7.0 scans the entire database, reading each extent to check whether it was changed after the last full backup, whereas Microsoft SQL Server 2000/2005 has bitmaps in each file mapping the changed extents, and the differential backup directs the disk arms to read only the relevant extents.

After performing operations that affect large portions of data, such as index rebuilds or defragmentations, it is recommended that you perform a full backup. Otherwise, differential backups can become very large. For more information on index rebuilds and defragmentation, see “Database Integrity and Fragmentation” on page 95.

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

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

Log Backup

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

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

A log backup also marks the portion of the log that was backed up as available for reuse. In a database that is set to the full or bulk-logged recovery model, log portions that were not backed up cannot be reused. When the log is full, and 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 page 96
- ▶ “Understanding File System Fragmentation” on page 96
- ▶ “Understanding Internal Fragmentation” on page 97
- ▶ “Understanding External Fragmentation” on page 98

- “Using DBCC SHOWCONTIG” on page 98
- “Handling Fragmentation” on page 102
- “Handling CMDB Index Fragmentation” on page 105
- “Distribution Statistics” on page 107
- “Utility to Refresh Statistics” on page 108

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 65). 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 system, 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 `DBCC DBREINDEX` command. 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.

Using DBCC SHOWCONTIG

`DBCC SHOWCONTIG` enables you to check the levels of both your internal and external fragmentation. The following is the `DBCC SHOWCONTIG` syntax:

```
DBCC SHOWCONTIG
    [ ( { table_name | table_id | view_name | view_id }
      [ , index_name | index_id ]
    )
  ]
  [ WITH { ALL_INDEXES
        | FAST [ , ALL_INDEXES ]
        | TABLERESULTS [ , { ALL_INDEXES } ]
```

```

    [ , { FAST | ALL_LEVELS } ]
  }
]

```

The following is the output of DBCC SHOWCONTING run against the clustered index of the Order Details table in the Northwind sample database:

```

DBCC SHOWCONTIG scanning 'Order Details' table...
Table: 'Order Details' (325576198); index ID: 1, database ID: 6
TABLE level scan performed.
- Pages Scanned.....: 9
- Extents Scanned.....: 6
- Extent Switches.....: 5
- Avg. Pages per Extent.....: 1.5
- Scan Density [Best Count:Actual Count].....: 33.33% [2:6]
- Logical Scan Fragmentation .....: 0.00%
- Extent Scan Fragmentation .....: 16.67%
- Avg. Bytes Free per Page.....: 673.2
- Avg. Page Density (full).....: 91.68%

```

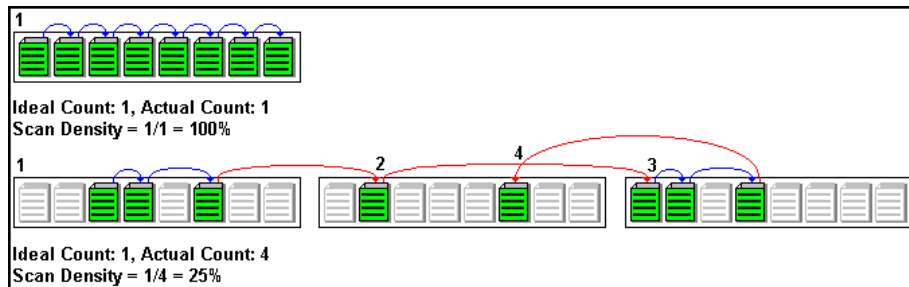
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

To determine the level of internal fragmentation, check the value of the Average Page Density, which contains the average percentage of page population. The closer the Average Page Density is to 100 percent, the less internal fragmentation there is.

To determine the level of external fragmentation, check the following values:

- **Scan Density.** The most important value to check for in order to identify external fragmentation. The scan density is the ratio between the ideal extent changes if everything is contiguously linked, and the actual extent changes. For example, if your index consumes 8000 pages, the ideal count is 1000. As the fragmentation level of your index increases and an ordered index scan switches back and forth between the extents, the actual count increases. Scan density is measured by percentage: 100 percent indicates that there is no external fragmentation, and any number less than a 100 percent indicates the level of fragmentation.

The following figure displays an example of how scan density is calculated:



- **Logical Scan Fragmentation.** The percentage of out-of-order pages returned from scanning the leaf pages of an index. An out-of-order page is one for which the next page indicated in an IAM (Index Allocation Map) is a different page than the page pointed to by the next page pointer in the leaf page. The value of logical scan fragmentation should be as low as possible—ideally 0%.
- **Extent Scan Fragmentation.** The percentage of out-of-order extents in scanning the leaf pages of an index. An out-of-order extent is one for which the extent containing the current page for an index is not physically the extent after the one containing the previous page for an index. The value of extent scan fragmentation should also be as low as possible—ideally 0%.

When to Defragment Indexes

It is recommended that you defragment indexes using the techniques in the following section when the Logical Scan Fragmentation values are equal to, or higher than, 10%. When the Logical Scan Fragmentation values are equal to, or higher than, 20%, it is strongly recommended that you defragment indexes.

Using sys.dm_db_index_physical_stats

In addition to using DBCC SHOWCONTIG in SQL 2005 to check the levels of internal and external fragmentation, you can also use `sys.dm_db_index_physical_stats`. Querying this table returns size and fragmentation information for the data and indexes of the specified table.

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 `sys.dm_db_index_physical_stats` dynamic management function shows the same information as the `DBCC SHOWCONTIG` statement shows, but with the following differences:

- ▶ `sys.dm_db_index_physical_stats` is much faster.
- ▶ `sys.dm_db_index_physical_stats` shows fragmentation in a specific index, in all indexes on a table or indexed view, in all indexes in a database, or in all indexes in all databases.

Avg_fragmentation_in_percent is the logical and extent fragmentation of the index. This parameter includes storage that spans multiple files and is more accurate than `DBCC SHOWCONTIG`.

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.mspx>.

Handling Fragmentation

Microsoft SQL Server provides several methods enabling you to handle internal and external fragmentation. You can rebuild a specific index using the `DROP_EXISTING` option of the `CREATE INDEX` statement, or you can rebuild all indexes belonging to a table by using the `DBCC DBREINDEX` statement. When the index is rebuilt, there is no external fragmentation, because all of its pages are laid on disk in the order of the linked list. Note that you can optionally specify a `fillfactor` to initiate internal fragmentation.

The above two fragmentation handling options are preferable to dropping and recreating your indexes for two reasons. First, you cannot drop an index created by a `PRIMARY KEY` or `UNIQUE` constraint. In order to drop this index, you must drop the constraint. Second, when you drop a clustered index, all nonclustered indexes are recreated in order to accommodate the changed row locators. The same thing occurs when you recreate the clustered index. If the clustered index is unique (uses the `DROP_EXISTING` option of the `CREATE INDEX` statement or `DBCC DBREINDEX`), the nonclustered indexes remain untouched when the clustered index is rebuilt.

The drawback of the above two fragmentation handling options is the effect they have on the users that access the data while the index is being rebuilt. When a clustered index is rebuilt, an exclusive lock is acquired on the table so that both the modification and retrieval of data from the table are not allowed. When a nonclustered index is rebuilt, a shared lock is acquired on the table so that data modification is not allowed. Although data retrieval is allowed, the index cannot be used by queries while it is being rebuilt.

To handle external fragmentation and allow data retrieval and modification at the same time, you can use the `DBCC INDEXDEFRAG` statement, introduced in Microsoft SQL Server 2000. The `DBCC INDEXDEFRAG` statement uses the fillfactor that was specified in the `CREATE INDEX` statement; you cannot specify a new fillfactor. Note that `DBCC INDEXDEFRAG` uses the bubble algorithm to defragment an index, swapping couples of pages until their order from left to right reflects the order of the linked list. However, it does not attempt to restructure the index in order to lay the index pages contiguously on disk. For this reason, it is recommended that you periodically rebuild your indexes using the `DROP_EXISTING` option or `DBCC DBREINDEX`.

The following table summarizes the differences between the defragmentation options:

| Option | Create Index with <code>DROP_EXISTING</code> | <code>DBCC DBREINDEX</code> | <code>DBCC INDEXDEFRAG</code> |
|--------------|------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| Locked table | Clustered – exclusive (no modification, no retrieval); Nonclustered – shared (no modification) | Clustered – exclusive (no modification, no retrieval); Nonclustered – shared (no modification) | Online – table is not locked |
| Granularity | Index | Table (all indexes) or index level | Index |
| Page sorting | Yes | Yes | Yes |

| Option | Create Index with DROP_EXISTING | DBCC DBREINDEX | DBCC INDEXDEFRAG |
|-------------------------------|---------------------------------|----------------|---------------------------------|
| Contiguity handling | Yes | Yes | No |
| Ability to specify fillfactor | Yes | Yes | No – attempts to apply original |

For more information on defragmenting indexes, refer to the Microsoft SQL Server 2000 Index Defragmentation Best Practices site (<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/ss2kidbp.mspx>). This site is applicable to SQL Server 2005 as well.

Note: Generally, indexes, including indexes on global temporary tables, can be created online. The indexes below are exceptions:

- XML index
- index on a local temporary table
- clustered index if its table contains LOB data types: image, ntext, text, varchar(max), varbinary(max), and xml
- clustered and nonclustered index defined with LOB data type columns
- initial unique clustered index on a view

You can perform concurrent online index operations on the same table only while doing the following:

- creating multiple nonclustered indexes
- reorganizing different indexes on the same table
- reorganizing different indexes while rebuilding non-overlapping indexes on the same table

All other online index operations performed at the same time fail. For example, the following scenarios fail:

- rebuilding two or more indexes concurrently on the same table
- creating a new index while rebuilding an existing index on the same table

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

Handling CMDB Index Fragmentation

HP Universal CMDB provides two utilities that can be used to detect and rebuild fragmented indexes. The utilities use the Logical scan fragmentation and Scan Density criteria to detect, and if instructed to rebuild, fragmented indexes. The operation of listing the fragmented tables has a very small impact on system performance and can be executed on line. The operation of rebuilding the indexes 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 2000 are located in the <HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\ms_2000\utils directory on the HP Universal CMDB Server. The utilities for Microsoft SQL Server 2005 are located in the <HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\ms_2005\utils 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 them.

To run the rebuild_indexes.bat 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 BAC_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 rebuild_fragmented_indexes.bat utility:

Execute **CMDB\db_scripts\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 BAC_DB_3
ad%min52 1

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

Note: You must run schemas for the CMDB databases.

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 the CMDB databases on a daily basis, as the data is frequently changed. The job should execute the **sp_updatestats** API against the specific database.

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

The following scenario warrants a manual refresh of CMDB statistics:

- ▶ **Automated Discovery tasks.** The Discovery and Dependency Mapping application is responsible for automatically detecting configuration items (CIs) and inserting them into the CMDB.

To manually refresh CMDB statistics:

- 1** In a Web browser, open:
`http://<Gateways Server machine name>:8080/jmx-console`
- 2** In the **Topaz** section, select **CMDB Dal Services**.
- 3** Under **runStatistics**, enter the customer ID. The default customer ID for an individual HP Universal CMDB system (that is, one not managed by HP Software-as-a-Service) is 1.
- 4** Under **runStatistics**, click **Invoke**. The CMDB statistics are regenerated.

Utility to Refresh Statistics

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

- ▶ List tables with out of dated 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:

Execute `CMDB\db_scripts\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, CMDDB\db_scripts\update_statistics.bat SQL_SRVR_3
BAC_DB_3 ad%min52 1

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

Monitoring Databases

This section describes the recommended system monitor counters for baseline/daily monitoring.

A baseline enables you to track the different counters during normal system activity, and learn what the counter values are when the system is operating well. In problematic situations, you can examine and compare current counter values with the values in the baseline.

The *SQL Server 2000 Operations Guide* (<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/sqlops0.mspx>) recommends tracking the following counters to create a baseline and monitor it:

| Counter | Description |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Memory - Pages/second | The number of pages read from, or written to, disk to resolve hard page faults. (Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk). This counter is designed as a primary indicator of the types of faults that cause system-wide delays, and is the sum of Memory: Pages Input/sec and Memory: Pages Output/sec . It is measured in numbers of pages, so that it can be compared to other page counts, such as Memory: Page Faults/sec , without conversion. It includes pages retrieved to satisfy faults in the file system's non-cached mapped memory files. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval. |
| Network Interface - Total Bytes/second | The number of bytes traveling over the network interface per second. If this rate begins to drop, you should investigate whether network problems are interfering with your application. |
| PhysicalDisk - Disk Transfers/second | The rate of read and write operations on the disk. You should define a counter for each physical disk on the server. |

| Counter | Description |
|---------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Processor - & Processor Time | The percentage of time that the processor is executing a non-idle thread. This counter is designed as a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the idle process in each sample interval, and subtracting this value from 100 percent. (Each processor has an idle thread that consumes cycles when no other threads are ready to run). It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval, which is calculated by monitoring the time the service was inactive, and then subtracting that value from 100 percent. After all of the processors devoted to Microsoft SQL Server have reached 100 percent utilization, it is likely that end user requests are being ignored. |
| SQLServer:Access Methods - Full Scans/second | The number of unrestricted base table or full index scans. |
| SQLServer:Buffer Manager - Buffer Cache Hit Ratio | The percentage of pages that were found in the buffer pool, without having to incur a read from disk. When this percentage is high, your server is operating at optimal efficiency (as far as disk I/O is concerned). |
| SQLServer:Databases - Log Growths (run against your application database instance) | The total number of log growths for the selected database. |
| SQLServer:Databases Application Database - Percent Log Used (run against your application database instance) | The percentage of space in the log that is in use. |

| Counter | Description |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SQLServer:Databases Application Database - Transactions/second (run against your application database instance) | The number of transactions started for the database. |
| SQLServer:General Statistics - User Connections | The number of users connected to the system. Dramatic shifts in this value should be investigated. |
| SQLServer:Latches - Average Latch Wait Time | The average latch wait time (in milliseconds) for latch requests that had to wait. If this number is high, your server may be facing contention for its resources. |
| SQLServer:Locks - Average Wait Time | The average amount of wait time (milliseconds) for each lock request that resulted in a wait. |
| SQLServer:Locks - Lock Waits/second | The number of lock requests that could not be satisfied immediately and required the caller to wait before being granted the lock. |
| SQLServer:Locks - Number of Deadlocks/second | The number of lock requests that resulted in a deadlock. |
| SQLServer:Memory Manager - Memory Grants Pending | The current number of processes waiting for a workspace memory grant. |

The following counters can help you identify hardware problems:

| Counter | Description |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Network Interface()\Packets Outbound Errors | The number of outbound packets that could not be transmitted due to errors. |
| Network Interface()\Packets Received Errors | The number of inbound packets that contained errors preventing the packets from being delivered to a higher-layer protocol. |
| Server\Errors System | The number of times that an internal server error was detected. Errors can reflect problems with logon, security, memory allocation, disk operations, transport driver interface operations, communication [such as receipt of unimplemented or unrecognized server message blocks (SMBs)], or I/O request packet stack size for the server. Many of these errors are also written to the system log and security log in the Event Viewer. The server can recover from most the errors displayed by this counter, but they are unexpected and should be reported to Microsoft Product Support Services. |

For information on creating a baseline, refer to the system monitor documentation, or to the *SQL Server 2000 Operations Guide* (<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/sqlops0.msp>).

Database Maintenance – References

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

- Troubleshooting application performance with Microsoft SQL Server:
<http://www.microsoft.com/technet/prodtechnol/sql/2005/tsprfprb.mspx>
- Performance counters:
<http://www.databasejournal.com/features/mssql/article.php/1477311#disk>
- How to perform a Microsoft SQL Server performance audit:
http://www.sql-server-performance.com/sql_server_performance_audit.asp
- Microsoft SQL Server performance tuning tips:
http://www.sql-server-performance.com/best_sql_server_performance_tips.asp

10

Microsoft SQL Server Sizing Guidelines

This chapter contains guidelines for the server and database configuration settings that should be used when working with Microsoft SQL Server and HP Universal CMDB. Note that the recommended settings generally differ according to the size of your HP Universal CMDB deployment.

This chapter includes:

- Server Configuration Options on page 115
- Data File Property Settings on page 116
- The tempdb Database Settings on page 117

Server Configuration Options

You can reconfigure the Microsoft SQL Server option **awe enabled**. When Microsoft SQL Server must access more than 4 GB of physical memory, you can use the Microsoft Windows Address Windowing Extensions (AWE) API to support up to a maximum of 64 GB.

For more information, refer to Microsoft SQL Server Books Online and the *SQL Server 2000 Operations Guide* (<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/sqlops0.msp>).

Data File Property Settings

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

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

The tempdb Database Settings

The frequent expansion of the tempdb system database can affect the database's performance, especially in large 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 500 MB and with a growth increment of 50 MB. The tempdb database should be striped across several disks, ideally on a RAID 0+1 controller. It is recommended to move the tempdb database to its own set of disks.

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.

Part III

Deploying and Maintaining the Oracle Server Database

11

Overview of Oracle Server Deployment

Note: If HP Universal CMDB is installed on a Solaris platform, the databases must be set up on an Oracle Server.

This chapter describes the following topics related to deploying Oracle Servers for use with HP Universal CMDB.

This chapter includes:

- ▶ About Oracle Server Deployment on page 121
- ▶ System Requirements on page 122

About Oracle Server Deployment

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

- ▶ **Install Oracle Server.**

For details, refer to the installation guide for your specific Oracle platform.

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

For details, refer to the installation guide for your specific Oracle platform.

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

For details, see “Oracle Tablespaces” on page 156.

► **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 127.

Information is provided in this section for both recommended and supported Oracle environments. HP Universal CMDB recommendation indicates that HP quality assurance personnel have rigorously tested the recommended environment/option. A supported environment or option means that HP quality assurance personnel have successfully performed basic tests on the environment/option.

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” on page 123
- “Software Requirements” on page 123
- “Examples of Tested Deployments” on page 124
- “Oracle Instances” on page 125

Hardware Requirements

For HP Universal CMDB hardware sizing guidelines, see “Hardware Sizing” on page 15.

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

The following table lists the Oracle servers supported for working with HP Universal CMDB.

| Database Release - Version | BIT Set |
|----------------------------------------|---------|
| Oracle 10.2.0.3 RAC Enterprise Edition | 64 BIT |
| Oracle 10.2.0.4 Enterprise Edition | 64 BIT |
| Oracle 10.2.0.4 RAC Enterprise Edition | 64 BIT |
| Oracle 9.2.0.8 Enterprise Edition | 64 BIT |
| Oracle 10.2.0.3 Enterprise Edition | 32 BIT |
| Oracle 10.2.0.3 Enterprise Edition | 64 BIT |

Note:

- It is strongly recommended to apply the latest critical Oracle patches per your operating system. For details, consult the Oracle documentation.
 - Consult the Oracle documentation for supported platforms.
 - The Oracle Partitioning option should be enabled.
-

Examples of Tested Deployments

The following table details those deployment environments that were tested by HP.

| Database Release - Version | BIT Set | Operating System |
|----------------------------------------|---------|--------------------------|
| Oracle 10.2.0.4 Enterprise Edition | 64 BIT | Linux Enterprise Edition |
| Oracle 10.2.0.4 RAC Enterprise Edition | 64 BIT | Linux Enterprise Edition |

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 databases, ensure that all the instances are configured as described in this document and that they all have the same characteristics (such as the same character set).

Note: Only the Enterprise editions of Oracle are supported by HP Universal CMDB.

12

Manually Creating the Oracle Server Database Schemas

This chapter describes the manual creation of the Oracle Server database schemas required by HP Universal CMDB.

This chapter includes:

- ▶ Overview of the Oracle Database Schemas on page 127
- ▶ Prerequisites on page 128
- ▶ Creating Schema Default and Temporary Tablespaces on page 129
- ▶ Creating Schema User Permissions on page 129
- ▶ Creating Schemas and Schema Objects on page 130

Note: It is strongly recommended that you run the database schema verify program after creating the schemas to verify that your database schemas are configured properly. For information on the verification process, see Appendix B, “Database Schema Verification.”

Overview of the Oracle Database Schemas

HP Universal CMDB uses the following database schemas for its persistency:

CMDB. The configuration management database.

CMDB History. The configuration management history database.

CMDB Foundations. The configuration management infrastructure database.

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

For details about installing the HP Universal CMDB server, see “Introduction to HP Universal CMDB” in the *HP Universal CMDB Deployment Guide* PDF.

Prerequisites

Before creating the database schemas, the Oracle Database Server must be installed according to the details in this guide. For further details on installing and configuring the Oracle Database Server, see Chapter 14, “Oracle Server Configuration and Sizing Guidelines.”

If schemas are manually created prior to running the HP Universal CMDB setup, during the setup procedure, select the option to use existing schemas and enter the relevant data about the existing schemas for which you are prompted. The data you enter is written to the configuration files that are used by the JDBC driver to connect to the existing schemas.

Creating Schema Default and Temporary Tablespaces

The database administrator should create at least one Temporary tablespace in the database instance hosting HP Universal CMDB schemas, and at least one tablespace to be served as the default tablespace for the HP Universal CMDB schemas. For details on creating an Oracle tablespace using a provided script, see “Creating an Oracle Tablespace” on page 157.

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 158.

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
- Permissions
 - CREATE TABLE
 - CREATE VIEW
 - CREATE SEQUENCE
 - CREATE TRIGGER
 - CREATE PROCEDURE
 - UNLIMITED TABLESPACE
 - ALTER USER \${user} DEFAULT ROLE ALL

Note: HP Universal CMDB supports any user with higher permissions. However, during the upgrade process, the redundant permissions should be revoked in order to pass schema verification. 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 three schemas; one for CMDB persistency, one for CMDB historical data persistency (CMDB History), and one for the Foundation schema.

Objects are created using SQL scripts which are run against the schemas that were previously created.

The scripts are located in the **<HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\oracle** directory located on the HP Universal CMDB server, and should be run from this directory.

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

To move the scripts to a different machine:

Copy the **<HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\oracle** directory from the HP Universal CMDB server machine to the machine with the required client.

This section includes the following topics:

- “Creating the CMDB Objects” on page 131
- “Creating the CMDB History Objects” on page 131
- “Creating the CMDB Foundation Objects” on page 131

Creating the CMDB Objects

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

- `create_cm_tables_cmdb.sql`

Creating the CMDB History Objects

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

- `create_history_tables_cmdb.sql`

Creating the CMDB Foundation Objects

To create the CMDB Foundation objects, connect to the CMDB Foundations schema and run the following SQL script:

- `foundations_ora_dbobjects_create.sql`

13

Maintaining an Oracle Server Database

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

This chapter includes:

- Database Maintenance and Tuning on page 133
- Oracle Database Backup and Recovery on page 147

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 page 134
- “Database Load Behavior” on page 134
- “CPU and I/O” on page 135
- “Oracle Alert File” on page 135
- “Archive Log – File System” on page 135
- “Tablespace Storage Space” on page 136
- “Dictionary-Managed Tablespace Coalescing” on page 136
- “Collecting Statistics for Databases” on page 136
- “Collecting Statistics for CMDB” on page 140
- “FREELISTS Storage Parameter” on page 143
- “Index Fragmentation” on page 144

System Global Area (SGA)

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

Database Load Behavior

In Oracle10g, an 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. For using STATSPACK in Oracle 9i, refer to *Oracle Metalink Note 94224.1: STATSPACK FAQ*.

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

CPU and I/O

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

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

Oracle Alert File

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

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

Archive Log – File System

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

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

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

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

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

Tablespace Storage Space

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

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

Dictionary-Managed Tablespace Coalescing

Free space in Oracle tablespaces is composed of newly created extents, or extents that were used and freed. If some of the free space in a tablespace is composed of extents that were used and freed, your tablespace may become temporarily fragmented. To repair fragmentation, two extents that reside next to one another can be coalesced in order to create one large extent.

To check for fragmentation, run the following query using SQL*Plus (using the system administrator account):

```
SELECT A.TABLESPACE_NAME, COUNT(*) BLOCK_CASES
      FROM DBA_FREE_SPACE A, DBA_FREE_SPACE B
      WHERE A.TABLESPACE_NAME = B.TABLESPACE_NAME
      AND A.FILE_ID = B.FILE_ID
      AND A.BLOCK_ID+A.BLOCKS = B.BLOCK_ID
GROUP BY A.TABLESPACE_NAME
/
```

This query returns a list of tablespaces that require coalescing. Although the Oracle SMON process automatically performs coalescing, it may do so infrequently. It is therefore recommended that you coalesce tablespace extents using the ALTER TABLESPACE <tablespace name> COALESCE; command. This task is not needed when using locally managed tablespaces.

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).

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

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

An example of changing the job to a schedule running every night at midnight can be found in the `change_gather_stats_job.bat` database script located at **<HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\oracle\utils**.

Note: This script should be used with caution and should only be run by the system administrator.

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.

To collect statistics for all HP Universal CMDB objects on a 9i database:

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

```
Exec DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '<name of Oracle schema>', estimate_percent => 20, cascade => TRUE);
```

Once your HP Universal CMDB system is stable, you should collect statistics once a day.

To collect statistics for all HP Universal CMDB objects on a 10g database:

- 1 Log in to the HP Universal CMDB relevant 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>', tabname => '<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.

Alternatively, you can collect statistics for each table and index using the analyze command.

To collect statistics for tables and indexes using the analyze command:

- 1** Log in to the relevant schema using SQL*Plus.
- 2** For each table, collect statistics using the `ANALYZE TABLE <table_name> ESTIMATE STATISTICS SAMPLE <x> ROWS;` query. It is recommended that "x" be approximately five percent of the records in the table.
- 3** For each index, collect statistics using the `ANALYZE INDEX <index name> COMPUTE STATISTICS;` query.

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

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

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

Collecting Statistics for CMDB

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

- ▶ **Automated Discovery and Dependency Mapping (DDM) jobs.** DDM is the process responsible for automatically detecting configuration items (CIs) and inserting them into CMDB.

Statistics Collection Guidelines for Oracle 9i

The following guidelines are applicable to Oracle 9i.

Running a Daily Job

To create a daily job to run statistics on CMDB, use the **create_statistics_job.bat** script located in the **<HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\oracle\utils** directory. A thick database client tool (SQLPLUS) is required to run the script. If you need to move the script to a different machine that has SQLPLUS installed, copy the directory in which the script is located to the required machine. The script is run with the following command:

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

where:

- ▶ **schema.** The name of the CMDB schema user of the schema for which you are installing the statistics job.
- ▶ **password.** 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.
- ▶ **hour.** The hour of the day to run the statistics job. Accepted values are 0-23 with a default value of midnight.

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

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

Running a Manual Statistics Refresh

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

- ▶ Use the **runStatistics** JMX:
 - ▶ In a Web browser, open **http://<Gateways Server machine name>:8080/jmx-console**.
 - ▶ In the **Topaz** section, select **CMDB Dal Services**.
 - ▶ Under **runStatistics**, enter the customer ID. The default customer ID for an individual HP Universal CMDB system (that is, one not managed by HP Software-as-a-Service) is 1.
 - ▶ Under **runStatistics**, click **Invoke**. The CMDB statistics are regenerated.
- ▶ Manually run the daily job using an Oracle client:
 - ▶ Connect to the CMDB schema via SQLPLUS.
 - ▶ Run the query **Select job from user_jobs where upper(what) like '%GATHER_SCHEMA_STATS%'**;
 - The query output is the job number.
 - ▶ Run the job with the call **Exec dbms_job.run (<job number>)** ;
- ▶ Connect to the CMDB schema and execute the PL/SQL block **begin DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '<name of Oracle schema>', cascade => TRUE) ; end;**

Statistics Collection Guidelines for Oracle 10g

The following guidelines are applicable to Oracle 10g.

Running a Daily Job

From Oracle 10g and later, there is no need to define a dedicated, daily job against the CMDB schema as Oracle 10g includes an automated job for statistics collection. For details, see “Collecting Statistics for Databases” on page 136.

Running an Hourly Job

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

If you do not manually refresh statistics for each major data change, it is recommended to schedule a Scheduler job running every hour for refreshing stale statistics against the CMDB schema. You execute an automatic, hourly refresh of statistics by executing the PL/SQL block **begin**

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

The provided database script, `create_CMDB_Stats_Job.Bat`, located in the `<HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\oracle\utils` directory, creates the needed job scheduled to run every hour. The job is owned by SYSTEM user and operates against a defined CMDB schema.

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

Running a Manual Statistics Refresh

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

- Use the **runStatistics JMX**:

- In a Web browser, open **http://<Gateways Server machine name>:8080/jmx-console**.
- In the **Topaz** section, select **CMDB Dal Services**.
- Under **runStatistics**, enter the customer ID. The default customer ID for an individual HP Universal CMDB system (that is, one not managed by HP Software-as-a-Service) is 1.
- Under **runStatistics**, click **Invoke**. The CMDB statistics are regenerated.

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

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

FREELISTS Storage Parameter

In a distributed HP Universal CMDB environment, it is recommended that you enlarge the FREELISTS storage parameter value in the HP Universal CMDB database tables and their indexes from the default value of 1 to a value of 20. Enlarging the FREELISTS storage parameter value prevents data block waits.

Use SQL*Plus to change the FREELISTS parameter to a value of 20.

To change the FREELISTS parameter in a table:

Use the following command:

```
alter table <table name> storage ( freelists 20 );
```

To change the FREELISTS parameter in an index:

Use the following command:

```
alter index <index name> storage ( freelists 20 );
```

When creating large tables in the schema, HP Universal CMDB automatically sets the FREELISTS parameter to 20. Database objects (tables and indexes) created later on by other components are created with the default Oracle FREELISTS value of 1. It is recommended that you track these objects and set their FREELISTS parameter values to 20.

Note: The FREELISTS parameter applies only to objects created in tablespaces whose segment space is manually managed.

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.

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

To run the index maintenance utility:

- 1** Copy the following files from the \<HP Universal CMDB root directory>\UCMDBServer\j2f\CMDB\dbscripts\oracle\utils 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 on page 146, 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 describes:

- Available Backup Methods
- Oracle Recovery Manager – RMAN

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 a the point in time in the past at which the database snapshot was taken.

For more information, refer to the *Oracle Backup and Recovery Guide* (http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96519&remark=dri1ldown&word=Backup).

Hot Backup

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

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

For more information, refer to the *Oracle Backup and Recovery Guide* (http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96519&remark=dri1ldown&word=Backup).

Export

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

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

For more information, refer to the Export/Import section of *Oracle Utilities* (http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96652&remark=dri1ldown&word=Export).

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

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

(http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96566&remark=drilldown&word=RMAN) and

Oracle Recovery Manager Reference

(http://otn.oracle.com/pls/db92/db92.show_toc?partno=a96565&remark=drilldown&word=RMAN) for Oracle 9i and

Backup and Recovery Advanced User's Guide (http://download-east.oracle.com/docs/cd/B19306_01/backup.102/b14191/toc.htm) and

Backup and Recovery Reference (http://download-east.oracle.com/docs/cd/B19306_01/backup.102/b14194/toc.htm) for Oracle 10g.

14

Oracle Server Configuration and Sizing Guidelines

This chapter contains guidelines for the Oracle database configuration 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.

This chapter includes:

- ▶ Oracle Parameter Sizing on page 152
- ▶ Oracle Tablespaces on page 156
- ▶ Oracle Tablespace Settings on page 158
- ▶ Using RAID Configuration on page 161

Oracle Parameter Sizing

The following table describes the recommended size for a number of 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 | See remarks below the table |
| DB_CACHE_SIZE | 1 GB | 4 GB | See remarks below the table. For Oracle 10g, it is recommended that you use the SGA_TARGET parameter. |
| DB_CACHE_ADVICE | ON | ON | For gathering statistics when tuning is required |
| SHARED_POOL_SIZE | Oracle 9i: 80 MB Oracle 10g: 200 MB | Oracle 9i: 112 MB Oracle 10g: 300 MB | For Oracle 10g, it is recommended that you use the SGA_TARGET parameter |
| SGA_TARGET | 1 GB | 4 GB | Available in Oracle 10g only |
| LOG_BUFFER | 1 MB | 5 MB | |
| DB_FILE_MULTIBLOCK_READ_COUNT | 16 | 32 | |
| PROCESSES | 200 | 400 | Add an additional 100 as a safety net |
| SESSIONS | 225 | 445 | (1.1 * PROCESSES) + 5 |

| Parameter Name | HP Universal CMDB Deployment | | Remarks |
|-------------------------|-------------------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Small | Large | |
| SORT_AREA_SIZE | 1 MB | 2 MB | It is recommended that you use the PGA_AGGREGATE_TARGET parameter instead. This parameter is reserved for backward compatibility and shared server mode. In order for it to work, the WORKAREA_SIZE_POLICY parameter must be set to MANUAL. |
| SORT_AREA_RETAINED_SIZE | Equal to SORT_AREA_SIZE value | Equal to SORT_AREA_SIZE value | See remarks for the SORT_AREA_SIZE parameter. |
| HASH_AREA_SIZE | 3 MB | 6 MB | Is equal to 3 times the SORT_AREA_SIZE value. See remarks for the SORT_AREA_SIZE parameter. |

| Parameter Name | HP Universal C MDB Deployment | | Remarks |
|----------------------|-------------------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Small | Large | |
| WORKAREA_SIZE_POLICY | AUTO | AUTO | The automatic PGA memory management mode applies only to work areas allocated by dedicated Oracle Servers. The size of work areas allocated by shared Oracle Servers is still controlled by the previous *_AREA_SIZE parameters because these work areas are allocated mainly in SGA and not in PGA. |
| PGA_AGGREGATE_TARGET | 400 MB | 800 MB | |
| STATISTICS_LEVEL | TYPICAL | TYPICAL | Enables tuning if required. |

Note the following:

- ▶ **DB_BLOCK_SIZE.** DB_BLOCK_SIZE must be a multiple of the operating system block size.
- ▶ **DB_CACHE_SIZE.** The above recommendations are for very large implementations. Regardless of HP Universal CMDB certification, you must have the following additional hardware requirements in order to configure this parameter:
 - ▶ **Solaris hardware.** In 32-bit operating systems, there is a shared memory limit of 1.75 GB for all instances. If you want to use 4 GB of shared memory, you must use a 64-bit operating system and 64-bit Oracle RDBMS.
 - ▶ **Windows 2000.** In order to support and use SGA that is larger than 2 GB, you must enable a specific Windows and Oracle configuration (VLM). For more information, refer to the Microsoft and Oracle online documentation.
- ▶ **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.

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 schemas. 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.

This section describes:

- Locally Managed Tablespaces
- Creating an Oracle Tablespace

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 159.

Creating an Oracle Tablespace

You create an HP Universal CMDB Oracle tablespace for a Windows installation using the **oracle_tablespace_create.bat** script in <HP Universal CMDB root directory>\UCMDBServer\J2F\cmdb\dbscripts\oracle\utils.

For a UNIX installation, you create the Oracle tablespace using the **oracle_tablespace_create.sh** script in <HP Universal CMDB root directory>\UCMDBServer\J2F\cmdb\dbscripts\unix.

Note: This script is a basic script for tablespace creation that contains one data file. You can edit this file according to the size of your HP Universal CMDB installation. For more information, see “Data Tablespace Settings” on page 158.

To create an HP Universal CMDB Oracle tablespace:

Run the following command from the directory in which the **oracle_tablespace_create.bat** or **oracle_tablespace_create.sh** script is located:

```
oracle_tablespace_create [admin_user] [admin_password] [tns_entry_name]
[tablespace_name] [file_name] [file_size]
```

- **[admin_user]**. Name of user with administrative permissions on Oracle Server.
- **[admin_password]**. Password of specified user.
- **[tns_entry_name]**. The TNS name specified in the **tnsnames.ora** file on the local Oracle Client.
- **[tablespace_name]**. Name of tablespace to create.

- **[file_name]**. File name to be created, including full path to file.
- **[file_size]**. File size; use M for MB and K for KB.
 - For a CMDB or CMDB History user schema, specify a file size of at least 1 GB.

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” on page 158
- “Temporary Tablespace Settings” on page 159
- “Redo Log Settings” on page 160
- “Undo Segment Settings” on page 160

Data Tablespace Settings

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

| Tablespace | HP Universal CMDB Deployment | | Remarks |
|--------------|------------------------------|-------|----------------------------------------------|
| | Small | Large | |
| CMDB | 1 GB | 20 GB | The specified size is a minimum requirement. |
| CMDB History | 1 GB | 20 GB | The specified size is a minimum requirement. |

Data Tablespace Default Storage Settings

The storage settings for data tablespaces should be:

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

Temporary Tablespace Settings

The following table specifies the recommended settings for temporary tablespaces:

| Tablespace | HP Universal CMDB Deployment | | Remarks |
|-----------------------|------------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Small | Large | |
| 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). ▶ Tablespace should be of a temporary type (use of TEMPFILE). |

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

Redo Log Settings

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

| Setting | HP Universal CMDB Deployment | |
|-------------------------------------|------------------------------|--------|
| | Small | Large |
| Redo log file size | 100 MB | 200 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:

| Setting | HP Universal CMDB System Profile | | Remarks |
|----------------------------|----------------------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------|
| | Small | Large | |
| 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 values. |
| UNDO_ RETENTION parameter | Oracle default value | | |

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 |

| RAID | Type of RAID | Control File | Database File | Redo Log File/ Temporary | Archive File |
|------|-------------------------------|--------------|----------------------------------------------------------|-----------------------------|--------------|
| 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.
-

15

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 on page 163
- Oracle Client Installation on page 164
- Oracle Client Configuration on page 164

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 “Software Requirements” on page 123.

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.

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.

This section includes the following topics:

- ▶ “MDAC for Oracle Client” on page 165

MDAC for Oracle Client

HP Universal CMDB uses Microsoft MDAC components to connect to the database. MDAC is installed by default on a Windows 2000 operating system.

Note: MDAC versions 2.5, 2.52, 2.61, 2.62, and 2.7 SP1 Refresh are all supported by HP Universal CMDB. For HP Universal CMDB certification, ensure that MDAC version 2.7 SP1 Refresh is installed on the Oracle Client machine.

16

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 on page 168
- ▶ Oracle Server and Client Requirements on page 174

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 151.

Checklist for HP Universal CMDB Support and Certification

The following table summarizes the Oracle database options that are supported and certified for working with HP Universal CMDB:

| Option | Supported | Recommended | Remarks | For more information, see |
|---------------------------------------------------------|----------------------|-------------|---------------------------------------------------------------------------|--------------------------------|
| Oracle edition | Server & Enterprise | Enterprise | | |
| Dedicated HP Universal CMDB server | Not necessary | Necessary | | |
| Use of multiple Oracle instances | Yes | No | The configuration of all instances must match in a certified environment. | “Oracle Instances” on page 125 |
| Use of non-default port | Yes | Yes | | |
| Undo management | Automatic; Manual | Automatic | Set UNDO_ MANAGEMEN T parameter to AUTO in certified environment | |
| Manual undo management + Public rollback segments | Yes | No | | |

| Option | Supported | Recommended | Remarks | For more information, see |
|--------------------------------------------|------------------------------------------|------------------------------------------|---------------------------------------------------------------------------|---------------------------|
| Oracle multi-threaded server (MTS) | Yes | No | HP Universal CMDB uses a connection pool architecture. | |
| Oracle replication | No full support | No | | |
| Use of SYSTEM, RBS, UNDO, TEMP tablespaces | Not allowed for HP Universal CMDB schema | Not allowed for HP Universal CMDB schema | | |
| Windows and UNIX file compression | No | No | Not supported by Oracle; Causes abnormal behavior and affects performance | |
| Database control files required | Greater than or equal to 2 | 3 | Preferable on different disks. | |

| Option | Supported | Recommended | Remarks | For more information, see |
|------------------------------------------|-----------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|
| 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 | | |
| OPTIMIZER_INDEX_COST_ADJ parameter value | 100 | 100 | Affects performance | |
| HASH_JOIN_ENABLED parameter value | True; False | True | This parameter is obsolete in Oracle 10g | |
| TIMED_STATISTICS parameter value | True; False | True | | |
| DB_CACHE_ADVICE parameter value | Off; On | On | | “Oracle Parameter Sizing” on page 152 |

| Option | Supported | Recommended | Remarks | For more information, see |
|--------------------------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------|---------|---------------------------------------|
| STATISTICS_LEVEL parameter value | Typical | Typical | | “Oracle Parameter Sizing” on page 152 |
| LOG_CHECKPOINT_INTERVAL parameter value | Greater than or equal to 0 | 0 | | |
| LOG_CHECKPOINT_TIMEOUT parameter value | Greater than or equal to 0 | 0; or greater than or equal to 1800 | | |
| OPTIMIZER_MODE parameter value | Oracle 9i: Choose Oracle 10g: ALL_ROWS | Oracle 9i: Choose Oracle 10g: ALL_ROWS | | |
| CURSOR_SPACE_FOR_TIME parameter value | True; False | False | | |
| CURSOR_SHARING parameter value | Exact | Exact | | |
| LOG_BUFFER parameter value | greater than 1 MB | greater than 1 MB | | |
| USE_STORED_OUTLINES parameter value | No | No | | |
| OPEN_CURSORS parameter value | 800 | 800 | | |
| COMPATIBLE parameter value | 9.2.06 | 10.2.0.1 | | |

| Option | Supported | Recommended | Remarks | For more information, see |
|-----------------------------------|----------------------|----------------------|------------------------------------------------------------------------------------------------------------|-------------------------------------|
| SQL_TRACE parameter value | True; False | False | | |
| Working in archive log mode | True; False | True | | |
| LOG_ARCHIVE_START parameter value | True; False | True | Only when working archive log mode is set to "True" | |
| BLANK_TRIMMING parameter value | False | False | | |
| FIXED_DATE parameter value | Not set | Not set | HP Universal CMDB uses the SYSDATE function for generating system time as part of the application process. | |
| SPIN_COUNT parameter value | Yes | No | | |
| UNDO_MANAGEMENT parameter value | Manual; Auto | Auto | | |
| UNDO_RETENTION parameter value | Oracle default value | Oracle default value | Automatic tuning is performed in Oracle 10g | "Undo Segment Settings" on page 160 |

| Option | Supported | Recommended | Remarks | For more information, see |
|-----------------------------------------------|----------------------------------|-----------------------------------|---------|-------------------------------------------------------------------------------------|
| WORKAREA_SIZE_POLICY parameter value | Manual; Auto | Auto | | “Oracle Parameter Sizing” on page 152 (in particular, the SORT_AREA_SIZE parameter) |
| Autoextend option in tablespace files | Yes | No | | |
| Locally managed data tablespace | Yes | Yes | | “Locally Managed Tablespaces” on page 156 |
| Tablespace extent management | Dictionary; Local; Local uniform | Local uniform for TEMP tablespace | | “Oracle Tablespace Settings” on page 158 |
| RECYCLEBIN | Off | Off | | |
| Automatic Segment Space Management Tablespace | Yes | Yes | | |

| Option | Supported | Recommended | Remarks | For more information, see |
|----------------------|----------------------------------------|-----------------|-----------------------------------------------------------------------|--------------------------------------|
| MDAC version | 2.5; 2.52; 2.61; 2.62; 2.7 SP1 Refresh | 2.7 SP1 Refresh | Set three registry settings according to the Oracle Client installed. | “MDAC for Oracle Client” on page 165 |
| NLS_LENGTH_SEMANTICS | BYTE | BYTE | This parameter controls the length definition of varchar columns. | |

Oracle Server and Client Requirements

For the Oracle Server and Client versions and operating system platforms that are supported and certified for working with HP Universal CMDB, see “Software Requirements” on page 123.

Part IV

Appendixes

A

HP Universal CMDB Database Scripts

This chapter describes the HP Universal CMDB database script.

This chapter includes:

- ▶ About HP Universal CMDB Database Scripts on page 177
- ▶ Database Utilities–Location on page 177
- ▶ Database Utilities–Database Clients on page 178
- ▶ Database Utilities–Readme File on page 178

About HP Universal CMDB Database Scripts

A directory of database scripts is deployed on the HP Universal CMDB server to be used for various offline administrative tasks against the HP Universal CMDB databases.

For example, scripts are available to check index fragmentation and to handle statistics in HP Universal CMDB schemas.

Database Utilities–Location

The locations of the database utilities are:

- ▶ Oracle scripts – <HP Universal CMDB root directory>\UCMDBServer\J2F\cmdb\dbscripts\oracle
- ▶ SQL Server 2000 scripts – <HP Universal CMDB root directory>\UCMDBServer\J2F\cmdb\dbscripts\ms_2000

- ▶ SQL Server 2005 scripts – <HP Universal CMDB root directory>\UCMDBServer\J2F\cmdb\dbscripts\ms_2005

Database Utilities–Database Clients

The database scripts require a thick database client tool: SQLPLUS for Oracle, ISQL for SQL Server 2000, and sqlcmd for SQL Server 2005.

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

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

Database Utilities–Readme File

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

B

Database Schema Verification

This appendix describes how to verify your database schemas to help ensure that your database schemas are configured properly.

This chapter includes:

- ▶ About the Verification Process on page 179
- ▶ Running the Verification Process on page 181
- ▶ Creating Database Users for the Verify Procedure on page 183

About the Verification Process

You verify the CMDB, CMDB history, and Foundations schemas using the database schema verify program. The verification process does not involve downtime for your system.

In addition to verifying the HP Universal CMDB database schemas, the verification process also determines if there is a later schema version to which you can upgrade. If a newer version of the database schemas is detected, the verification process notifies you of possible lengthy operations that could slow down a future upgrade.

During the course of the verification process, you need to specify the user name and password, which is required for certain (read-only) tests to be performed. If you do not want to supply your DBA account user name and password, you can create a user name with the minimum privileges required for the verify program to operate. For details on how to create this user, see “Creating Database Users for the Verify Procedure” on page 183.

Note: HP Universal CMDB does not store the user credentials entered in the verification process.

Notes and Limitations

The following notes and limitations apply when running the database schema verify program:

- If you are running the database verify program on Oracle 10g schemas and use Oracle datapump utilities to import or export the target schemas, ensure that you do not have any active datapump jobs running against the target schemas.

If you have datapump tables in the target schemas, they should be dropped prior to running the database schema verification program.

It is recommended to assign an administrator schema to perform datapump operations and not to use HP Universal CMDB schemas as the login. By assigning an administrator schema to perform datapump operations, you do not have to grant additional permissions to HP Universal CMDB schemas and the datapump tables are created in the administrator schema.

Running the Verification Process

You run the database schema verify program from an HP Universal CMDB server machine.

To verify a database schema:

- 1** Run the verify program according to your operating platform:
 - If you are running the verify program from a Windows platform, Go to the **<HP Universal CMDB root directory>\UCMDBServer\j2f\dbverify\bin** directory and run the **cmdbDbVerify.bat** file.
 - If you are running the verify program from a Solaris platform, open an X-terminal window, move to the **<HP Universal CMDB root directory>/UCMDBServer/j2f/dbverify/bin** directory, and type the following:


```
./cmdbDbVerify.sh
```

The database verify program starts.

- 2** Specify the details required to connect to the appropriate database:
 - In Microsoft SQL Server, it is not necessary to specify any parameters.
 - If the Foundation, CMDB, and History schemas reside on Oracle:

In most cases, when all three schemas reside on the same Oracle instance, you should run **cmdbDbUpgrade** with just two parameters – the user name of a database user with admin privileges (for example, **system**) and the password.

However, if the Foundation, CMDB, and History schemas reside on different Oracle instances, you should specify the admin user name and password for each Oracle instance where the schema is located, in the following order:

- Foundation schema
- CMDB schema
- History schema

Note: If you do not want to supply your database administrator account user name and password, you can create a user name with the minimum privileges required for the **cmdbDbUpgrade** tool to operate.

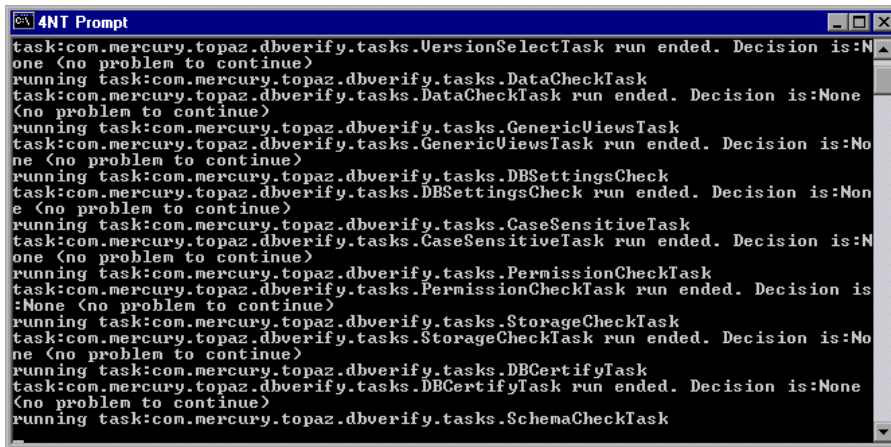
To create the user:

For Oracle Server:

```
CREATE USER dbv_read IDENTIFIED BY admin;  
GRANT SELECT_CATALOG_ROLE TO dbv_read;  
GRANT CONNECT TO dbv_read;
```

You must run this script as a system user.

- 3 The database verify program performs database verification. You can view the progress of the verify process in a command prompt window.



```
task:com.mercury.topaz.dbverify.tasks.UersionSelectTask run ended. Decision is:None (no problem to continue)  
running task:com.mercury.topaz.dbverify.tasks.DataCheckTask  
task:com.mercury.topaz.dbverify.tasks.DataCheckTask run ended. Decision is:None (no problem to continue)  
running task:com.mercury.topaz.dbverify.tasks.GenericViewsTask  
task:com.mercury.topaz.dbverify.tasks.GenericViewsTask run ended. Decision is:None (no problem to continue)  
running task:com.mercury.topaz.dbverify.tasks.DBSettingsCheck  
task:com.mercury.topaz.dbverify.tasks.DBSettingsCheck run ended. Decision is:None (no problem to continue)  
running task:com.mercury.topaz.dbverify.tasks.CaseSensitiveTask  
task:com.mercury.topaz.dbverify.tasks.CaseSensitiveTask run ended. Decision is:None (no problem to continue)  
running task:com.mercury.topaz.dbverify.tasks.PermissionCheckTask  
task:com.mercury.topaz.dbverify.tasks.PermissionCheckTask run ended. Decision is:None (no problem to continue)  
running task:com.mercury.topaz.dbverify.tasks.StorageCheckTask  
task:com.mercury.topaz.dbverify.tasks.StorageCheckTask run ended. Decision is:None (no problem to continue)  
running task:com.mercury.topaz.dbverify.tasks.DBCertifyTask  
task:com.mercury.topaz.dbverify.tasks.DBCertifyTask run ended. Decision is:None (no problem to continue)  
running task:com.mercury.topaz.dbverify.tasks.SchemaCheckTask
```

- 4 If problems occur during the database verification, they appear in the command prompt window.

Fix the problems found and rerun the database schema verify program. If you are unable to fix the problems, contact HP Software Support for assistance.

You can view a log file of the errors located in the <HP Universal CMDB root directory>\UCMDBServer\j2f\log directory.

- 5 If the database verification is successful, a message is displayed in the command prompt window and in the log.

Note: For details on database verification troubleshooting issues, see the HP Software Self-solve knowledge base (h20230.www2.hp.com/selfsolve/documents). To enter the knowledge base, you must log in with your HP Passport ID.

Creating Database Users for the Verify Procedure

When running the database schema verify and upgrade utility, you are prompted to supply a user name and password that can access the master database. You can create users with minimum privileges by running one of the following scripts.

For Microsoft SQL Server

```
set nocount on
use master
GO
sp_addlogin @loginame ='dbv_read',@passwd = '<pass>'
GO
sp_adduser @loginame = 'dbv_read', @name_in_db = 'dbv_read'
go
grant select on syslogins to dbv_read
go
set nocount off
```

Note: You must run this script as an **sa** user.

For Oracle Server

```
CREATE USER dbv_read IDENTIFIED BY admin;  
GRANT SELECT_CATALOG_ROLE TO dbv_read;  
GRANT CONNECT TO dbv_read;
```

Note: You must run this script as a system user.

C

Support for Oracle Real Application Cluster

This appendix contains the configuration that need to be done for HP Universal CMDB to work with Oracle Real Application Cluster. This information applies to advanced users only.

This chapter includes:

- About Oracle Real Application Cluster (RAC) on page 185
- Client Side Configuration for Oracle RAC on page 187
- Server Side Configuration on page 190
- Create/Connect to an HP Universal CMDB Database on page 191

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.

To use Oracle RAC, your database version must be Oracle 10.2.0.1 or later.

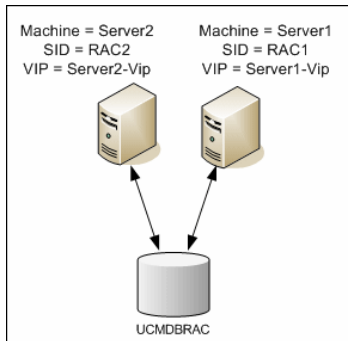
For more details about Oracle RAC, see the Oracle documentation site (http://www.oracle.com/pls/db102/to_toc?pathname=rac.102%2Fb14197%2Ftoc.htm).

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.



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 UCMDDB database schemas (for a description of UCMDDB schemas, see “Introduction to Preparing the Database Environment” on page 13). 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 191.

Make the following changes in HP Universal CMDB's configuration files:

- 1 On all servers, create the file < HP Universal CMDB root directory>\UCMDDBServer\j2f\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. The address contains all the details needed for connecting 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.

Important: 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, ucldb-tnsnames.ora would be configured as:

```
UCMDBRAC =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = Server1-Vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = Server2-Vip)(PORT = 1521))
    (LOAD_BALANCE = on)
    (FAILOVER = on)
  )
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = UCMDBRAC)
  )
)
```

2 On all servers, edit the file <HP Universal CMDB root directory>\UCMDBServer\j2f\conf\jdbc.drivers.properties.

a Find the section named **#ddoracle**. In this section, there is a line for `ddoracle.url` similar to:

```
ddoracle.url=jdbc:mercury:oracle://${host}:${port};sid=${sid}
```

b Replace this line with either of the following lines (depending on the operating system):

➤ **Windows.**

```
ddoracle.url=jdbc:mercury:oracle:TNSNamesFile=<UCMDB_HOME>
\\UCMDBServer\j2f\conf\ucmdb-tnsnames.ora;TNSServerName=${sid}
```

Note that each back slash (\) in the path of <UCMDB_HOME> is doubled.

➤ **UNIX.**

```
ddoracle.url=jdbc:mercury:oracle:TNSNamesFile=<UCMDB_HOME>
/UCMDBServer/j2f/conf/ucmdb-tnsnames.ora;TNSServerName=${sid}
```

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.

This section includes the following topics:

- ▶ “Recommendation for HP Universal CMDB Database” on page 190

Recommendation for 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 described in UCMDB Server Configuration in the *HP Universal CMDB Deployment Guide* PDF except for the following:

| Parameter | Needed Value |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Oracle Machine | One of the Virtual IPs. |
| Port | The port of the listener on the node with virtual IP from the Oracle Machine. |
| SID | The service name of the database. Important: This must be the same as <SERVICE_NAME> entered in ucmdb_tnsnames.ora . |

In the example, the parameters would be:

| Parameter | Value |
|----------------|-------------|
| Oracle Machine | Server1-Vip |
| Port | 1521 |
| SID | UCMDBRAC |

Index

A

- alert file, Oracle Server 135
- archive file, Oracle Server 135
- authentication
 - enabling HP Universal CMDB authentication 61
- Authentication Mode dialog box 28

B

- backup
 - Microsoft SQL Server databases 91
 - Oracle Server databases 147

C

- checklists
 - Microsoft SQL Server 2005 support and certification 41
 - Oracle Server support and certification 168
- CMDB database fragmentation, Microsoft SQL Server 102
- CMDB index fragmentation, Microsoft SQL Server 105
- CMDB index fragmentation, Oracle Server 144
- Coalition Settings dialog box 29
- CPU, Oracle Server 135
- creating schema objects 130
- creating schemas 130

D

- data file property settings, Microsoft SQL Server 116
- data placement

- Microsoft SQL 2000 Server 67
- Microsoft SQL Server 2005 79
- database
 - configuration options on Microsoft SQL Server 2000 72
 - configuration options on Microsoft SQL Server 2005 84
 - creating users 183
 - file configuration on Microsoft SQL Server 2000 71
 - file configuration on Microsoft SQL Server 2005 83
 - file layout on Microsoft SQL Server 2000 67
 - file layout on Microsoft SQL Server 2005 78
 - fragmentation on Microsoft SQL Server 96
 - integrity on Microsoft SQL Server 96
 - load behavior on Oracle Server 134
 - maintenance on Microsoft SQL Server 91
 - maintenance on Oracle Server 133
 - Microsoft SQL Server maintenance references 114
 - monitoring on Microsoft SQL Server 109
 - permissions on Microsoft SQL Server 2000 65
 - permissions on Microsoft SQL Server 2005 77
 - properties on Microsoft SQL Server 2000 68
 - properties on Microsoft SQL Server 2005 80
 - requirements 13
 - schema verification 179

Index

- database configuration
 - Microsoft SQL Server 2000 70
 - Microsoft SQL Server 2005 83
- database configuration options
 - Microsoft SQL Server 2000 72
 - Microsoft SQL Server 2005 84
- database file configuration
 - Microsoft SQL Server 2000 71
 - Microsoft SQL Server 2005 83
- database manual creation
 - Microsoft SQL Server 58
- database schemas overview
 - Microsoft SQL Server 57
 - Oracle Server 127
- database utilities
 - database clients 178
 - location 177
 - readme file 178
- databases
 - HP Universal CMDB 13
- dbverify utility 181
- default tablespaces
 - creating 129
- deployment
 - Microsoft SQL Server 19, 37
 - Oracle Server 121
- distribution statistics
 - Microsoft SQL Server 107

F

- file groups
 - Microsoft SQL Server 2000 69
 - Microsoft SQL Server 2005 81
- file layout
 - Microsoft SQL Server 2000 67
 - Microsoft SQL Server 2005 78
- file properties
 - Microsoft SQL Server 2000 68
 - Microsoft SQL Server 2005 80
- fragmentation
 - Microsoft SQL Server database 96, 102

H

- hardware sizing 15

- HP Universal CMDB
 - database script 177
 - databases 13
 - enabling for Windows authentication 61
- HP Universal CMDB database sizing 14

I

- index fragmentation
 - Microsoft SQL Server CMDB database 105
- index fragmentation, CMDB
 - Oracle Server 144
- installation
 - Microsoft SQL Server 2000 23
 - Microsoft SQL Server 2005 42
 - Oracle Client 164
- installing Microsoft SQL Server 23
- instance
 - Oracle Server 125
- Instance Name dialog box 24

L

- locally managed tablespace
 - creation on Oracle Server 157
- log placement
 - Microsoft SQL Server 2000 67
 - Microsoft SQL Server 2005 79

M

- MDAC
 - Oracle Client 165
 - Oracle Server 165
- Microsoft SQL Server
 - CMDB index fragmentation 105
 - collecting statistics 94
 - database backup 91
 - database fragmentation 96, 102
 - database integrity 96
 - database maintenance 91
 - database maintenance references 114
 - database monitoring 109
 - database schemas overview 57
 - deployment overview 19, 37

- distribution statistics 107
- installing 23
- manual creation of databases 58
- refreshing statistics 108
- sizing guidelines 115
- system requirements 20
- Microsoft SQL Server 2000
 - adding data files 71
 - changing data file properties 71
 - configuring 32
 - configuring databases 70
 - data placement 67
 - database configuration options 72
 - database file configuration 71
 - database file layout 67
 - database permissions 65
 - database properties 68
 - dropping data files 71
 - file groups 69
 - file properties 68
 - log placement 67
 - permissions 65
 - system databases 70
- Microsoft SQL Server 2005
 - adding data files 83
 - changing data file properties 83
 - checklist for support and certification 41
 - configuring databases 83
 - data placement 79
 - database configuration options 84
 - database file configuration 83
 - database file layout 78
 - database permissions 77
 - database properties 80
 - dropping data files 83
 - file groups 81
 - file properties 80
 - installing 42
 - log placement 79
 - modifying settings 51
 - permissions 77
 - system databases 82
 - verifying settings 51

N

- Network Libraries dialog box 31

O

Oracle

- alert file 135
- instances 125
- optimizing query performance 136, 140
- Recovery Manager (RMAN) 149
- tablespace settings 158

Oracle Client

- configuring 164
- installation 164
- MDAC 165
- system requirements 163

Oracle Server

- checklist for support and certification 168
- CMDB index fragmentation 144
- collecting statistics for CMDB 140
- CPU 135
- creating schema user permissions 129
- database backup 147
- database maintenance 133
- database schemas overview 127
- deployment overview 121
- dictionary-managed tablespace
 - coalescing 136
- Input/Output 135
- instances 125
- parameter sizing 152
- RAID configuration 161
- requirements 174
- sizing guidelines 151
- system requirements 122
- tablespace maintenance 136
- tablespaces 156

- Oracle Server Real Application Cluster
 - see Real Application Cluster 185

P

- parameter sizing

- Oracle Server 152

Index

permissions

- Microsoft SQL Server 2000 65
- Microsoft SQL Server 2005 77

Q

- query performance on Oracle Server, optimizing 136, 140

R

- RAID configuration, Oracle Server 161

Real Application Cluster

- client configuration 187
- connect to or create Management database 191
- introduction 185
- server configuration 190

Recovery Manager (RMAN)

- Oracle Recovery Manager 149

refreshing statistics

- Microsoft SQL Server 108

S

schema objects

- creating 130

schema user permissions

- creating 129

schemas

- creating 130

- Select Components dialog box 26

server configuration options

- Microsoft SQL Server 115
- Microsoft SQL Server 2000 32
- Microsoft SQL Server 2005 50

service configuration options

- Microsoft SQL Server 2000 32
- Microsoft SQL Server 2005 49

- Setup Type dialog box 25

SID 164

sizing guidelines

- Microsoft SQL Server 115
- Oracle Server 151

statistics

- collecting for CMDB in Oracle Server 140

- collecting in Microsoft SQL Server 94

system databases

- Microsoft SQL Server 2000 70
- Microsoft SQL Server 2005 82

System Global Area (SGA)

- Oracle Server 134

system requirements

- Microsoft SQL Server 20
- Oracle Client 163
- Oracle Server 122

T

- tablespace settings, Oracle Server 158

tablespaces

- Oracle Server 156

tablespaces, Oracle Server

- maintenance 136

tempdb database

- Microsoft SQL Server 2000 70
- Microsoft SQL Server 2005 82
- settings on Microsoft SQL 117

temporary tablespaces

- creating 129

tnsnames.ora

- location and configuring 164

transaction log

- backup 95
- Microsoft SQL Server 2000 67
- Microsoft SQL Server 2005 78

U

users

- creating for database schema verification 183

V

verification

- database schemas 179
- Microsoft SQL Server 2005 settings 51

W

Windows authentication

- enabling HP Universal CMDB 61