

HP Business Availability Center

for the Windows and Solaris operating systems

Software Version: 8.05

Database Guide

Document Release Date: September 2010

Software Release Date: September 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 Business Availability Center Sizing	15
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 Database	55
Overview of the Microsoft SQL Server Databases	55
Manually Creating the HP Business Availability Center Microsoft SQL Server Databases.....	56
Chapter 6: Using Windows Authentication to Access Microsoft SQL Server Databases	61
Enabling HP Business Availability Center 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: Setting Up the Database Environment.....	127
Creating HP Business Availability Center Schemas Manually	128
Profile User Schemas	132
HP Business Availability Center Oracle Schema Privileges.....	133
Chapter 13: Maintaining an Oracle Server Database	135
Database Maintenance and Tuning	135
Oracle Database Backup and Recovery	149

Chapter 14: Oracle Server Configuration and Sizing	
Guidelines	153
Oracle Parameter Sizing.....	154
Oracle Tablespaces.....	158
Oracle Tablespace Settings	160
Using RAID Configuration	164
Chapter 15: Configuring the Oracle Client for HP Business	
Availability Center	167
Oracle Client Versions and Operating System Platforms	167
Oracle Client Installation	168
Oracle Client Configuration	168
Chapter 16: Oracle Summary Checklist	171
Checklist for HP Business Availability Center Support and Certification	172
Oracle Server and Client Requirements	179

PART IV: APPENDIXES

Appendix A: Database Consolidation and Distribution	183
Advantages of Database Consolidation	183
Reasons for Database Distribution	184
Appendix B: Loader Management	185
Overview of Loader Management	186
Persistency Directories.....	187
Troubleshooting Loader Problems	189
Logs.....	190
Changing the Default Logging Level	192
Statistics	192
Appendix C: Data Partitioning and Purging	195
About Data Partitioning and Purging	195
Appendix D: Data Storage Recommendations	197
Creating Microsoft SQL Server File Groups.....	197
Appendix E: Database Schema Verification	203
About the Verification Process	203
Running the Verification Process.....	205
Creating Database Users for the Verify Procedure	209

Appendix F: Backing Up and Restoring Monitor Administration	
Configuration Data	211
Overview of Backing Up and Restoring Configuration Data.....	212
Basic Backup and Restore	213
Basic Backup on Replicated Server and Restore	214
LDIF Backup and Restore	214
Replication.....	217
Starting and Stopping OpenLDAP	220
Backup and Restore Examples.....	222
Appendix G: Support for Oracle Real Application Cluster	227
About Oracle Real Application Cluster (RAC).....	227
Client Side Configuration for Oracle RAC	229
Server Side Configuration	232
Create/Connect to One of the HP Business Availability Center Databases.....	234
Index	235

Welcome to This Guide

This guide describes how to deploy and maintain Microsoft SQL Server and Oracle Server databases for use with HP Business Availability Center.

Note: This guide is not relevant to HP Software-as-a-Service customers.

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 Business Availability Center.

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 Business Availability Center database environment for Oracle, how to configure the Oracle Client, and how to maintain databases.

Part IV Appendixes

Describes the situations in which you should consolidate all HP Business Availability Center data in a single database and the situations in which you should distribute the data among several different databases. Also contains recommendations for creating Microsoft SQL Server file groups and Oracle tablespaces, information on working with the Purging Manager, the procedure for verifying your database schema, information about HP Business Availability Center loaders, and the procedures for backing up and restoring LDAP databases for System Availability Management configuration data.

Who Should Read This Guide

This guide is intended for the following users of HP Business Availability Center:

- ▶ HP Business Availability Center 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 Business Availability Center, additional online resources, information on acquiring documentation updates, and typographical conventions used in this guide, see the *HP Business Availability Center 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 Business Availability Center.

This chapter includes:

- ▶ Databases in Use - Overview on page 13
- ▶ HP Business Availability Center Sizing on page 15
- ▶ Hardware Sizing on page 15

Databases in Use - Overview

To work with HP Business Availability Center, you must set up the following databases:

- ▶ **Management database.** For storage of system-wide and management-related metadata for the HP Business Availability Center environment. HP Business Availability Center requires one management database. You can create this database manually, or by using the Server and Database Configuration utility.
- ▶ **Profile database(s).** For storage of raw and aggregated measurement data obtained from the HP Business Availability Center data collectors. Although only one profile database is required, you can store profile data in multiple databases, if required. You can create profile databases manually, or by using the **Database Management** page, accessible from **Admin > Platform > Setup and Maintenance**.

- ▶ **Business Process Insight Database.** For storage of the Business Process Models created using the Business Process Insight Modeler.
- ▶ **CMDB.** For storage of configuration information that is gathered from the various HP Business Availability Center and third-party applications and tools. This information is used when building HP Business Availability Center views. The CMDB also contains the object repositories used to define configuration items and Key Performance Indicator (KPI)s.
- ▶ **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.

You can set up management, profile, CMDB, CMDB history, and BPI 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 Business Availability Center 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 Business Availability Center servers.
 - ▶ For details on working in a non-English language HP Business Availability Center environment, see “Working in Non-English Locales” in *Reference Information*.
-

HP Business Availability Center Sizing

HP Business Availability Center database configuration requirements are dependent on the amount of data generated by HP Business Availability Center.

The following table describes the database size, based on the amount of data generated by the various database elements:

Database Element	HP Business Availability Center Deployment		Remarks
	Standard	Large	
EPS	100	400 or more	
Logged-in HP Business Availability Center users	30	100 or more	
CMDB Objects and Links	100,000	1 million or more	
Database size	approximately 30 GB	approximately 600 GB or greater	

Hardware Sizing

The following table describes the hardware (CPU and memory) requirements recommended for the HP Business Availability Center 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 management, profile, CMDB, CMDB history, and BPI databases on Microsoft SQL Server. This chapter describes the following topics related to deploying Microsoft SQL Servers for use with HP Business Availability Center:

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 Business Availability Center, 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 management, profile, CMDB, CMDB history, and BPI 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 Business Availability Center 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 Business Availability Center 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 Business Availability Center.

This section includes the following topics:

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

Hardware Requirements

For HP Business Availability Center hardware sizing guidelines, see “Hardware Sizing” on page 15.

Software Requirements

The following table describes the Microsoft SQL Servers supported for working with HP Business Availability Center:

Database Release			Microsoft Data Access Components (MDAC) on HP Business Availability Center servers
Version	BIT Set	Service Pack	
Microsoft SQL Server 2008 Enterprise Edition	32 BIT	Service Pack 1	2.8 SP1 Refresh – Installed automatically with your HP Business Availability Center server installation.
Microsoft SQL Server 2008 Enterprise Edition	64 BIT	Service Pack 1	
Microsoft SQL Server 2005 Enterprise Edition	32 BIT	Service Pack 3	2.8 SP1 Refresh – Installed automatically with your HP Business Availability Center server installation.
Microsoft SQL Server 2005 Enterprise Edition	64 BIT	Service Pack 3	

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 Business Availability Center 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 Business Availability Center server installation.
Microsoft SQL Server 2008 Enterprise Edition	32 BIT	Service Pack 1	Windows 2008 Enterprise Edition Service Pack 1	
Microsoft SQL Server 2008 Enterprise Edition	64 BIT	Service Pack 1	Windows 2008 Enterprise Edition Service Pack 1	

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:

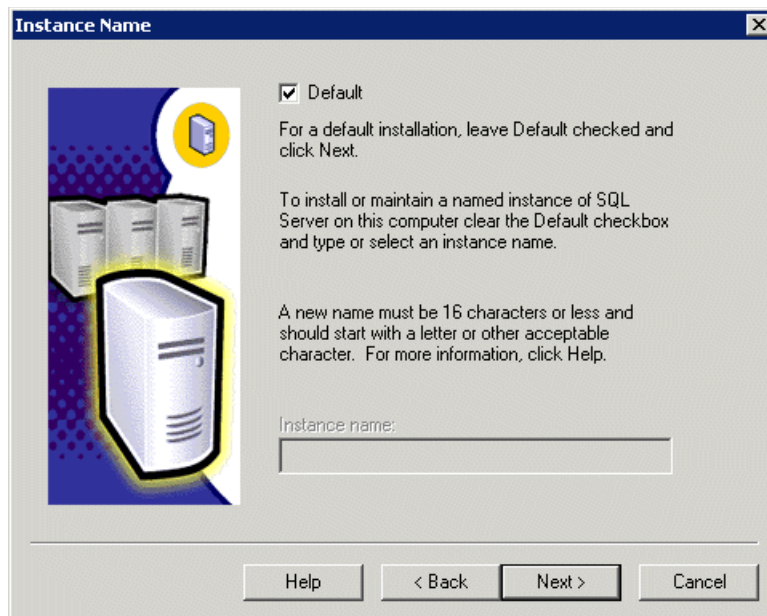
- Instance Name Dialog Box
- Setup Type Dialog Box
- Select Components Dialog Box
- Services Accounts Dialog Box
- Authentication Mode Dialog Box

- Collation Settings Dialog Box
- Network Libraries Dialog Box

Instance Name Dialog Box

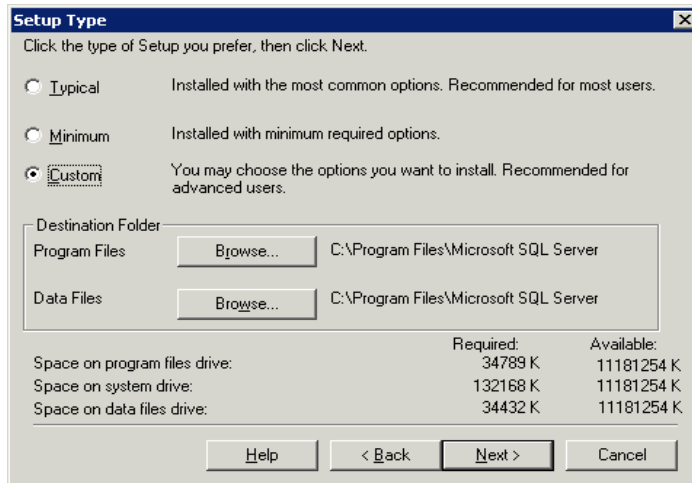
You can install multiple instances of Microsoft SQL Server 2000 on the same machine. Each instance has its own Microsoft SQL Server and Microsoft SQL Agent services, and is completely independent of the other instances.

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

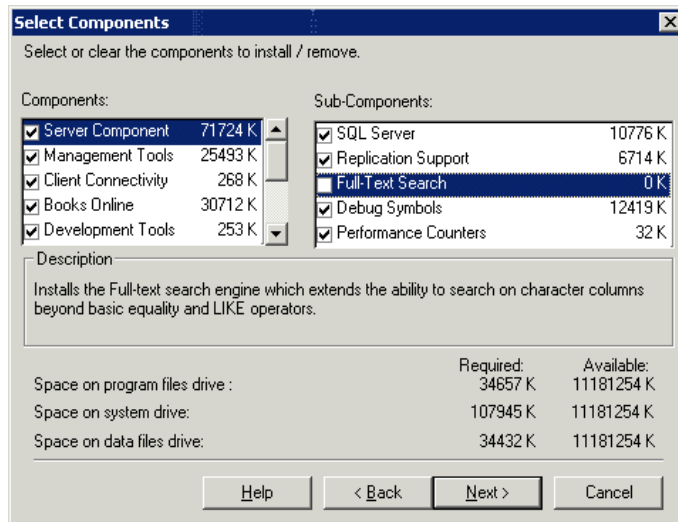
In the Setup Type dialog box, you must select the **Custom** option. If you do not select this option, you cannot view the Component Selection, Collation Settings, and Network Libraries dialog boxes, in which you must change certain default options that are not appropriate for working with Microsoft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



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.

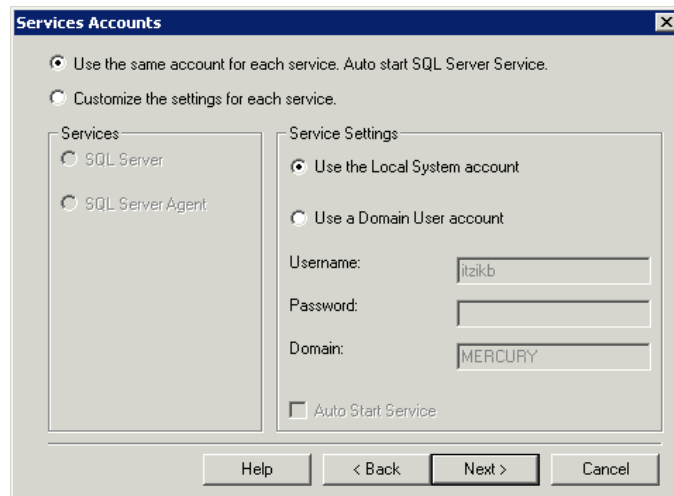
Select Components Dialog Box

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



Services Accounts Dialog Box

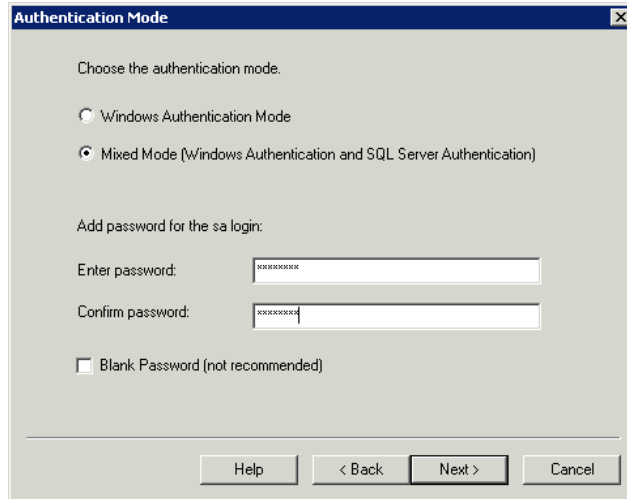
In the Services Accounts dialog box, you select the services and enter the appropriate service settings for your Microsoft SQL Server. If you want the Microsoft SQL Server service (MSSQLServer) to perform activities outside the local machine—for example, backing up files to a shared network directory—select **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 Business Availability Center 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.

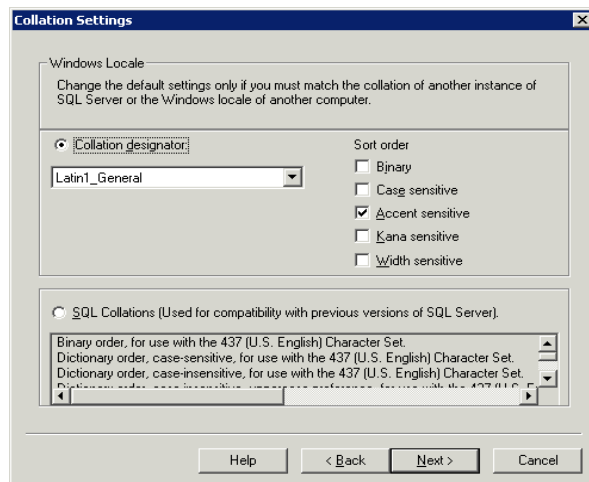
The sa user has system administrator privileges and can therefore perform all actions within the Microsoft SQL Server. Similarly, the sa user can perform all operating system/network actions in the context of the MSSQLServer service account, by using the xp_cmdshell extended procedure.

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 Business Availability Center 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 Business Availability Center 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.



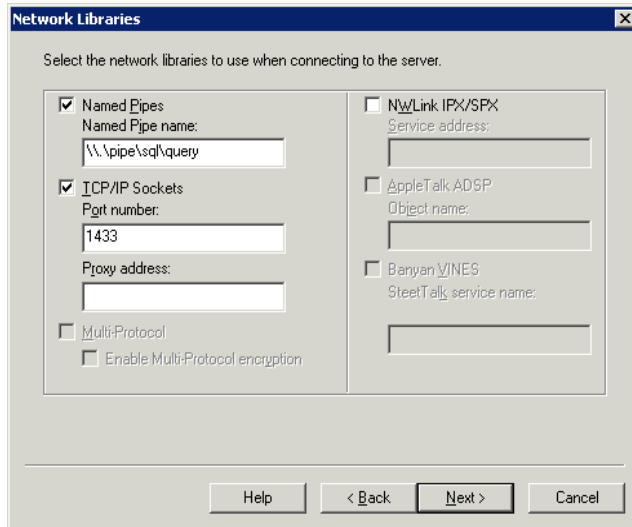
The above settings affect only the system databases and serve as the default settings for user databases. Databases can have different collation settings from the server's default settings, and a table column can have different collation settings from the database's default settings. Because of the flexibility 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 Business Availability Center. For HP Business Availability Center 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 Business Availability Center 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 Business Availability Center 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 Business Availability Center certification:

Configuration Option	Default	HP Business Availability Center 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 Business Availability Center 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 Business Availability Center 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 Business Availability Center, 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 Business Availability Center.

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 management, profile, CMDB, CMDB History, and BPI databases on Microsoft SQL Server 2005

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

7 Configure management, profile, CMDB, CMDB History, and BPI 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 (management, profile, CMDB, CMDB History, BPI) 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 Business Availability Center 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 Business Availability Center with a user account that is a part of **db_datareader**, **db_datawriter** and **db_ddladmin** in each HP Business Availability Center 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 Business Availability Center 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 Business Availability Center:

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

Subject	Microsoft SQL Server 2005	
	Supported	Recommended
Collation	Case-Insensitive. HP Business Availability Center 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**.

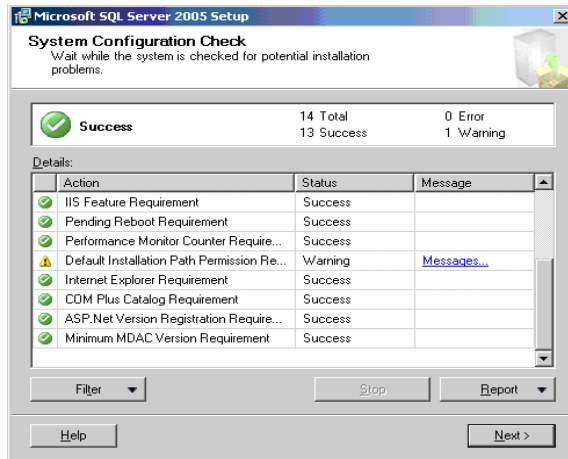
If you do not agree to the licensing terms, the installation process exits.

- 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.

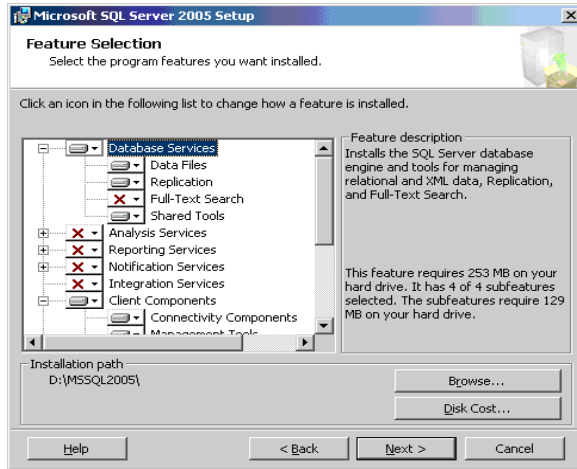
- 5 The Registration Information dialog box opens. The Product Key is automatically taken from the operating system. Accept the default information and click **Next**.

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 Business Availability Center 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.

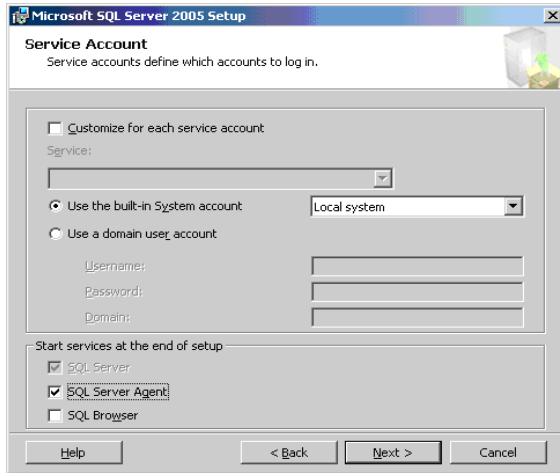
d Click **Next**.

- 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 53.

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 Business Availability Center works with both Windows authentication and with Microsoft SQL Server authentication (recommended). To enable Microsoft SQL Server authentication, do the following:

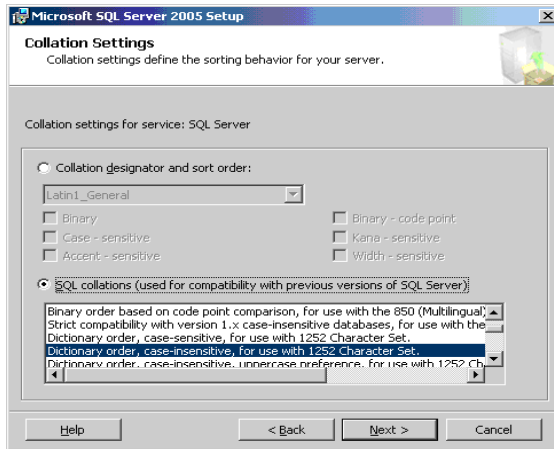
- a** Choose **Mixed Mode (Windows Authentication and SQL Server Authentication)**.
- b** Enter the password for user **sa**.

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 Business Availability Center 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
Network Libraries	On the Server, select Start > Programs > Microsoft SQL Server 2005 > Configuration Tools > Configuration Manager . At SQL Native Client.. choose client protocols and ensure that the selected protocols are in the enabled state. <ul style="list-style-type: none"> ▶ Supported Shared memory, TCP/IP, and Named Pipes for both the server and client. ▶ Recommended Only TCP/IP for both the server and client.

Server/Database Setting	How to Verify/Modify the Setting
View or change server configuration options	<ul style="list-style-type: none"> ▶ To allow the viewing of all options, 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 Business Availability Center databases has CREATE DATABASE permissions	<p>Log in to the Microsoft SQL Server Management Studio with the login you want to check, and run the following:</p> <pre>USE master IF PERMISSIONS() & 1 = 1 PRINT 'User has CREATE DATABASE permissions' ELSE PRINT 'User does not have CREATE DATABASE permissions'</pre>
Check whether the HP Business Availability Center database user has enough permissions in the database	<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>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>

Server/Database Setting	How to Verify/Modify the Setting
Data and log file destination directory is not compressed (only in NTFS)	Right click the directory, choose Properties , and then Advanced . Verify that the Compression check box is cleared.
Database and database file properties (including recovery model and collation properties)	<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>) <p>You can also view or change these properties from the Database Properties dialog box in the Enterprise Manager.</p>
Microsoft SQL Server service pack version and edition	<p>Log into Management Studio and make the following query:</p> <pre>select @@version</pre> <p>The result for SQL 2005 SP1 should start with: Microsoft SQL Server 2005 - 9.00.2047.00</p>

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 \<instance_name>, for example, server1\inst1.

- ▶ **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 Database

This chapter describes the procedure for manually creating HP Business Availability Center databases on a Microsoft SQL Server.

This chapter includes:

- Overview of the Microsoft SQL Server Databases on page 55
- Manually Creating the HP Business Availability Center Microsoft SQL Server Databases on page 56

Overview of the Microsoft SQL Server Databases

HP Business Availability Center uses the following databases for its persistency:

- **Management.** The HP Business Availability Center management database.
- **Profile.** The HP Business Availability Center profile database.
- **CMDB.** The configuration management database.
- **CMDB History.** The configuration management history database.
- **BPI.** The Business Process Insight database.

During the HP Business Availability Center 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 Business Availability Center.

For details about installing HP Business Availability Center server, see “Introducing HP Business Availability Center Platform Deployment” in the *HP Business Availability Center Deployment Guide* PDF.

Manually Creating the HP Business Availability Center Microsoft SQL Server Databases

Once you have installed and configured Microsoft SQL Server, you create HP Business Availability Center databases for the storage of data collected by HP Business Availability Center.

You can allow HP Business Availability Center to create HP Business Availability Center databases for you using the Server and Database Configuration utility, and to create profile databases for you using the **Database Management** page in **Admin > Platform > Setup and Maintenance**.

If you created the databases in advance (see below), you can connect HP Business Availability Center to the existing databases from the Server and Database Configuration utility, and to the existing profile database from the **Database Management** page in **Admin > Platform > Setup and Maintenance**.

For information on the Server and Database Configuration utility, see “Setting Database Parameters” and “Setting User Schema Parameters” in the *HP Business Availability Center Deployment Guide* PDF.

For information on creating profile databases in HP Business Availability Center, see “Database Management - Overview” in *Platform Administration*.

Note: If you allow HP Business Availability Center to create databases for you, it is strongly recommended that you then configure the databases.

This section includes the following topics:

- “Creating the Databases” on page 57
- “Creating the Objects” on page 57

Creating the Databases

The database administrator should create five Microsoft SQL Server databases; one for Management, one for CMDDB, one for CMDDB History, one for Profile, and one for BPI.

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 Business Availability Center 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

After creating each HP Business Availability Center database, you create objects by running SQL scripts. The object creation scripts are located in the <Gateways Server root directory>\AppServer\webapps\site.war\DataBases\SQL_Svr_DB_Utils directory.

Creating the Management Objects

To create the HP Business Availability Center management objects, connect to the management database and run the following SQL scripts:

- `management_sql_dbobjects_create.sql`
- `create_cm_tables_cmdb_ms.sql`
- `foundations_sql_dbobjects_create.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 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 Profile Objects

To create the HP Business Availability Center profile objects, connect to the profile database and run the following SQL script:

```
profile_sql_dbobjects_create.sql
```

Creating the BPI Objects

To create the Business Process Insight objects, connect to the BPI database and run the following SQL script:

```
bpi_sql_dbobjects_create.sql
```

If you are running Microsoft SQL Server 2005, run the script manually using Management Studio or SQLCMD. If you are running Microsoft SQL Server 2000, manually run the script using Query Analyzer or OSQL.

Note: Perform the above procedures only if you are an experienced Microsoft SQL Server database administrator.

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

6

Using Windows Authentication to Access Microsoft SQL Server Databases

Unless configured otherwise, HP Business Availability Center 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 Business Availability Center to use Windows authentication to access Microsoft SQL Server databases.

This chapter includes:

- ▶ Enabling HP Business Availability Center to Work with Windows Authentication on page 61

Enabling HP Business Availability Center to Work with Windows Authentication

You can enable HP Business Availability Center to use Windows authentication instead of Microsoft SQL Server authentication to access any of the HP Business Availability Center databases (management, profile, CMDB, CMDB History, and BPI).

To enable HP Business Availability Center to use Windows authentication to access a Microsoft SQL database, you must:

- ▶ Configure the Microsoft SQL Server to use Windows authentication.
- ▶ Launch the HP Business Availability Center service on all the HP Business Availability Center 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 HP Business Availability Center 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**.

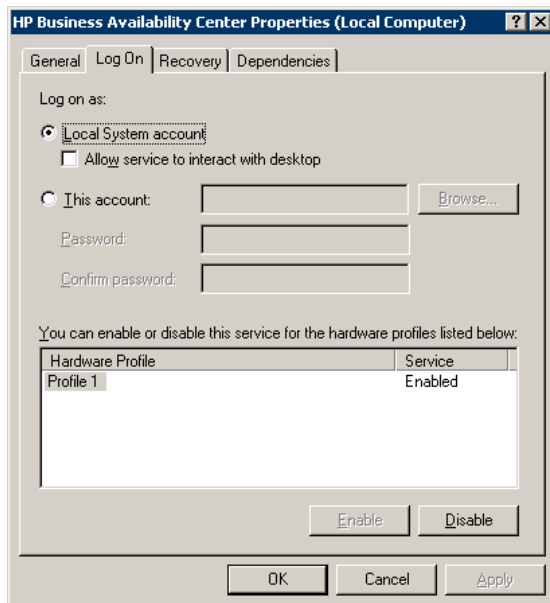
Launching the HP Business Availability Center service with a Windows User

By default, the HP Business Availability Center 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 HP Business Availability Center 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 Business Availability Center Service User” in the *HP Business Availability Center Deployment Guide* PDF.

To change the HP Business Availability Center service user:

- 1 Disable HP Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Disable Business Availability Center**).
- 2 In Microsoft's Services window, double-click **HP Business Availability Center**. The HP Business Availability Center 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 Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Enable Business Availability Center**).

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 “Setting Database Parameters on a Windows Platform” in the *HP Business Availability Center Deployment Guide* PDF.

7

Creating and Configuring Microsoft SQL Server 2000 Databases

This chapter describes the creation and configuration of HP Business Availability Center 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 Business Availability Center 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

Note: If you enabled HP Business Availability Center to create HP Business Availability Center databases for you, it is strongly recommended that you configure them using the configuration instructions in this section.

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.

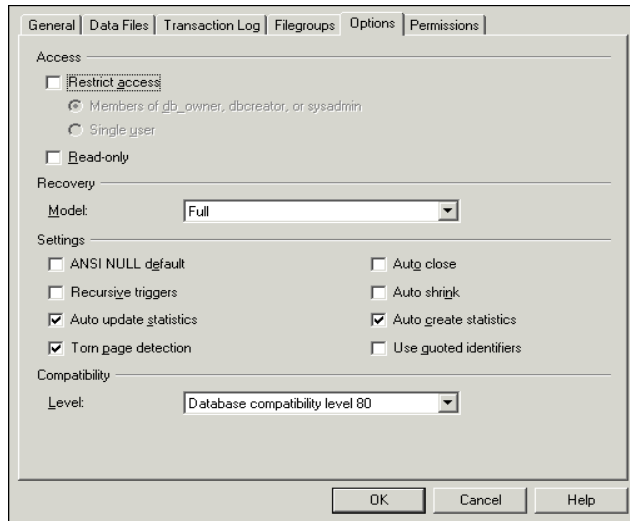
Changing File Properties

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 Business Availability Center certification:

Configuration Option	Description	Default	HP Business Availability Center 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 Business Availability Center 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 Business Availability Center 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 Business Availability Center 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 Business Availability Center 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

Important: If you enabled HP Business Availability Center to create HP Business Availability Center databases for you, it is strongly recommended that you configure them using the configuration instructions in this section.

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 Business Availability Center certification:

Configuration Option	Description	Default	HP Business Availability Center 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 Business Availability Center 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 Business Availability Center 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 Business Availability Center 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 Business Availability Center 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 (such as the management and profile 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 Business Availability Center 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%

```

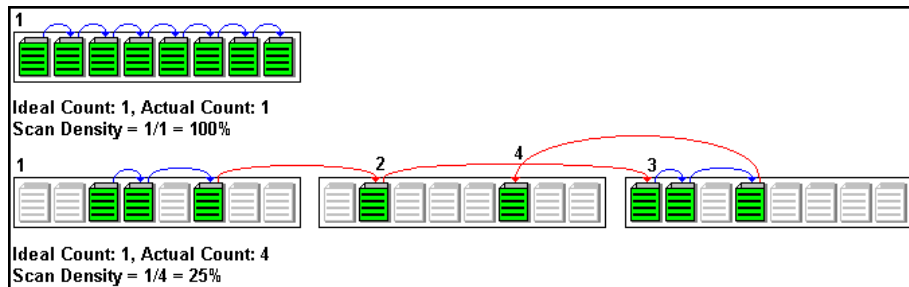
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

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.msp>.

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 DROP_EXISTING	DBCC DBREINDEX	DBCC INDEXDEFRAG
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 management, CMDB, and profile databases, as CMDB database data frequently changes, and profile database data in the clustered indexes is not always supported by timestamp, producing data fragmentations during the data load.

Handling CMDB Index Fragmentation

HP Business Availability Center 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 Business Availability Center root directory>\CMDB\dbscripts\ms_2000\utils** directory on the HP Business Availability Center Gateway Server. The utilities for Microsoft SQL Server 2005 are located in the **<HP Business Availability Center root directory>\CMDB\dbscripts\ms_2005\utils** directory on the HP Business Availability Center Gateway 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 management, CMDB, and profile 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 management, profile, and 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 scenarios warrant 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.
- ▶ **HP Business Availability Center adapters synchronization.** When you expect to load substantial data to the CMDB, you synchronize adapters from the Source Manager tab, accessed from **Admin > Universal CMDB > Source Manager**.

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 Business Availability Center 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 CMDDB\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 Business Availability Center. Note that the recommended settings generally differ according to the size of your HP Business Availability Center 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.mspx>).

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

You can set up management, profile, CMDB, CMDB history, and BPI databases on an Oracle Server.

Note: If HP Business Availability Center 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 Business Availability Center.

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 Business Availability Center, 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 Business Availability Center data.**

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

- ▶ **Create one or more Oracle tablespaces to store HP Business Availability Center data.**

For details, see “Oracle Tablespaces” on page 158.

- ▶ **Create an Oracle user schema for HP Business Availability Center schemas.**

You can create HP Business Availability Center 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 Business Availability Center, see “Creating HP Business Availability Center Schemas Manually” on page 128.

Information is provided in this section for both recommended and supported Oracle environments. HP Business Availability Center 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 Business Availability Center.

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 Business Availability Center 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 Business Availability Center.

Database Release - Version	BIT Set
Oracle 10.2 (10.2.0.3 or higher component specific release number 10.2.0.X) Enterprise Edition	64 BIT
Oracle 10.2 RAC (10.2.0.3 or higher component specific release number 10.2.0.X) Enterprise Edition	64 BIT
Oracle 10.2.0.3 Enterprise Edition	32 BIT
Oracle 11.1.0.7 Enterprise Edition	64 BIT
Oracle 11.2 (11g R2) Enterprise Edition	64 BIT
Oracle 11.2 (11g R2) RAC 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 11.2 (11g R2) 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 Business Availability Center certification, do not use more than one Oracle instance. If you do use more than one instance for the HP Business Availability Center 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 Business Availability Center.

12

Setting Up the Database Environment

This chapter describes how to create Oracle tablespaces for HP Business Availability Center user schemas and how to build management and profile user schemas. It also contains information on Oracle permissions for HP Business Availability Center user schemas.

This chapter includes:

- ▶ Creating HP Business Availability Center Schemas Manually on page 128
- ▶ Profile User Schemas on page 132
- ▶ HP Business Availability Center Oracle Schema Privileges on page 133

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

Creating HP Business Availability Center Schemas Manually

The HP Business Availability Center user schemas can be created manually or automatically using the Server and Database Configuration utility. For more information on the Server and Database Configuration utility, see “Setting Database Parameters” and “Setting User Schema Parameters” in the *HP Business Availability Center Deployment Guide* PDF.

The Server and Database Configuration utility prompts you to supply the following parameters:

- **Host name.** The name of the host machine on which Oracle Server is installed.
- **Port.** The Oracle listener port. HP Business Availability Center automatically displays the default port, **1521**.
- **SID.** The Oracle instance name that uniquely identifies the Oracle database instance being used by HP Business Availability Center.
- **Schema name and password.** The name and password of the existing user schema, or the name that you give the new user schema (for example, **BAC_MANAGEMENT**).
- **Admin user name and password.** (to connect as an administrator) The name and password of a user with administrative permissions on Oracle Server (for example, a System user).
- **Default tablespace.** The name of the dedicated default tablespace you created for the user schema. For details on creating a dedicated HP Business Availability Center tablespace, see “Oracle Tablespaces” on page 158.
- **Temporary tablespace.** The name of the temporary tablespace you assigned to the user schema. The default Oracle temporary tablespace is **temp**.

When manually creating management or profile user schemas, you must supply the following parameters:

- **Administrator user name.** The name of a user with administrative permissions on Oracle Server that is used to create the user schema (by default, the Oracle administrative user is **system**).
- **Administrator password.** The specified user’s password (by default, the Oracle administrative user’s password is **manager**).

- **User name.** A name for the user schema that you want to create. Note that you must specify a unique user name for each user schema that you create for HP Business Availability Center.
- **User password.** A password enabling you to access the user schema.
- **tnsnames entry name.** The TNS name specified in the **tnsnames.ora** file on the Oracle Client machine.
- **Default tablespace.** The name of the dedicated default tablespace you created for the user schema.
- **Temporary tablespace.** The name of a dedicated default temporary tablespace that you created for the user schema. If you did not create a dedicated temporary tablespace, specify an alternative temporary tablespace (the default Oracle temporary tablespace is called **temp**).

This section includes the following topics:

- “Creating a Management User Schema Manually” on page 129
- “Creating a CMDB User Schema Manually” on page 130
- “Creating a CMDB History User Schema Manually” on page 131
- “Creating a BPI User Schema Manually” on page 131
- “Updating License Information (for Solaris installations only)” on page 131

Creating a Management User Schema Manually

It is recommended that a certified database administrator create the management user schema manually (using your organization’s database characteristic methodology). You then connect to the management user schema later in the Server and Database Configuration utility. For more information on the Server and Database Configuration utility, see “Setting Database Parameters” and “Setting User Schema Parameters” in the *HP Business Availability Center Deployment Guide* PDF.

Note: Manually configuring a management user schema may be required if you cannot submit database administrator connection parameters over a non-secure connection.

You create the management user schema manually using the **management_ora_dbojects_create.bat** (or **management_ora_dbojects_create.sh** in UNIX installation) script, located in the <Gateways Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utils directory.

To create a management user schema manually:

Run the following command (see page 128 for an explanation of the parameters to be used) from the <Gateways Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utils directory:

```
management_ora_dbojects_create [admin_user] [admin_password]
[application_management_user] [application_management_password]
[tns_entry_name] [default_tablespace] [temporary_tablespace]
```

Creating a CMDB User Schema Manually

It is recommended that a certified database administrator create the CMDB user schema manually (using your organization's database characteristic methodology). You then connect to the CMDB user schema later in the Server and Database Configuration utility. For more information on the Server and Database Configuration utility, see "Setting Database Parameters" and "Setting User Schema Parameters" in the *HP Business Availability Center Deployment Guide* PDF.

To create a separate CMDB user schema manually:

Create a user schema with the permissions described in "HP Business Availability Center Oracle Schema Privileges" on page 133. Connect to this schema using sqlplus and run the following script located in the <Gateways Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utils directory:

```
create_cm_tables_cmdb.sql
```

Creating a CMDB History User Schema Manually

To create a separate CMDB History user schema manually:

Create a user schema with the permissions described in “HP Business Availability Center Oracle Schema Privileges” on page 133. Connect to this schema using sqlplus and run the following script located in the <Gateways Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utils directory:

```
create_history_tables_cmdb.sql
```

Creating a BPI User Schema Manually

To create a separate BPI user schema manually:

Make a copy of the `bpi_ora_dbojects_create.sql` script file located in the <Gateways Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utils directory.

Save the new file as `bpi_ora_dbojects_create_manual.sql`. Edit the `bpi_ora_dbojects_create_manual.sql` file and replace all “#” characters with the “/” character. For example:

```
END IF;
END;
#
```

would be changed to

```
END IF;
END;
/
```

Create a user schema with the permissions described in “HP Business Availability Center Oracle Schema Privileges” on page 133. Connect to this schema using sqlplus and run the `bpi_ora_dbojects_create_manual.sql` file.

Updating License Information (for Solaris installations only)

If HP Business Availability Center is installed on a Solaris platform, once you have created the management user schema, you must update the HP Business Availability Center license information.

To update HP Business Availability Center license information:

- 1 Log in to Solaris on any of the HP Business Availability Center servers as the **root** user.
- 2 Go to the **<HP Business Availability Center root directory>\scripts** directory.
- 3 Ensure that the **oracle_env** file in this directory contains the correct path to the Oracle client home.
- 4 Run the **create_license.sh** script using the parameters **<Management database user name> <Management database password> <database tns name>**. For example:

```
create_license.sh user1 password1 mymngdb
```

Profile User Schemas

Profile user schemas contain performance data collected by HP Business Availability Center, and are generally large databases. For information on the size of the profile databases, see Chapter 15, “Oracle Server Configuration and Sizing Guidelines.”

Profile user schemas can be created manually or automatically using the **Database Management** page in **Admin > Platform > Setup and Maintenance**. For information on creating profile databases automatically, see “Database Management - Overview” in *Platform Administration*.

This section includes the following topic:

- “Creating Profile User Schemas Manually” on page 132

Creating Profile User Schemas Manually

It is recommended that a certified database administrator create profile user schemas manually (using your organization’s database characteristic methodology). You then connect to these profile user schemas later from the **Database Management** page in **Admin > Platform > Setup and Maintenance**. For more information on connecting to profile user schemas from Platform Administration, see “Database Management - Overview” in *Platform Administration*.

Note: Manually configuring profile user schemas may be required if you cannot submit database administrator connection parameters over a non-secure connection.

You create the profile user schema manually using the **profile_ora_dbobjects_create.bat** (or **profile_ora_dbobjects_create.sh** in UNIX installation) script, located in the <Gateways Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utills directory.

To create a profile user schema manually:

Run the following command (see the list of parameters on page 128 that are used when creating a profile user schema) from the <Gateways Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utills directory:

```
profile_ora_dbobjects_create [admin_user] [admin_password]
[application_management_user] [application_management_password]
[tns_entry_name] [default_tablespace] [temporary_tablespace]
```

HP Business Availability Center Oracle Schema Privileges

When using an Oracle user schema, you can perform any database action that is allowed by the Oracle permission granted to the user. The following list (taken from the **oracle_user_create.sql** script located in the <Gateways Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utills directory) describes the required database permissions that must be granted to the HP Business Availability Center user. These permissions are also used by the HP Business Availability Center platform components to create a new Oracle user:

- GRANT "CONNECT" TO <HP Business Availability Center Oracle user schema>;
- GRANT CREATE INDEX TO <HP Business Availability Center Oracle user schema>;

- ▶ GRANT CREATE SEQUENCE TO <HP Business Availability Center Oracle user schema>;
- ▶ GRANT CREATE TABLE TO <HP Business Availability Center Oracle user schema>;
- ▶ GRANT CREATE TRIGGER TO <HP Business Availability Center Oracle user schema>;
- ▶ GRANT UNLIMITED TABLESPACE TO <HP Business Availability Center Oracle user schema>;
- ▶ GRANT CREATE VIEW TO <HP Business Availability Center Oracle user schema>;
- ▶ GRANT CREATE PROCEDURE TO <HP Business Availability Center Oracle user schema>;
- ▶ ALTER USER <HP Business Availability Center Oracle user schema> DEFAULT ROLE ALL;

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

13

Maintaining an Oracle Server Database

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

This chapter includes:

- Database Maintenance and Tuning on page 135
- Oracle Database Backup and Recovery on page 149

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 136
- “Database Load Behavior” on page 136
- “CPU and I/O” on page 137
- “Oracle Alert File” on page 137
- “Archive Log – File System” on page 137
- “Tablespace Storage Space” on page 138
- “Dictionary-Managed Tablespace Coalescing” on page 138
- “Collecting Statistics for Databases” on page 138
- “Collecting Statistics for CMDB” on page 141
- “FREELISTS Storage Parameter” on page 145
- “Index Fragmentation” on page 146

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 archive log 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 Business Availability Center 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 Business Availability Center deployment, it is recommended that you collect statistics for all HP Business Availability Center 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 Business Availability Center root directory>\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 Business Availability Center 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 (such as new tables and indexes created by the Partitioning and Purging Manager).

To collect statistics for all HP Business Availability Center objects on a 9i database:

- 1 Log in to the HP Business Availability Center profile 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 Business Availability Center system is stable, you should collect statistics once a day.

To collect statistics for all HP Business Availability Center objects on a 10g database:

- 1 Log in to the HP Business Availability Center profile 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 profile 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.
- **HP Business Availability Center adapters synchronization.** When you expect to load substantial data to CMDB, synchronize adapters from the Source Manager tab, accessed from Admin > 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 Business Availability Center root directory>\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 Business Availability Center 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 138.

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 Business Availability Center root directory> \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 Business Availability Center 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 Business Availability Center environment, it is recommended that you enlarge the FREELISTS storage parameter value in the HP Business Availability Center 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 profile schema, HP Business Availability Center automatically sets the FREELISTS parameter to 20. Database objects (tables and indexes) created later on by other components, such as the Partitioning and Purging Manager, 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 Business Availability Center 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 Business Availability Center tables, it is recommended to rebuild any index that has more than three levels.

Index Maintenance Utility

HP Business Availability Center'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 Business Availability Center root directory> \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 147, 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 Business Availability Center 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=driilldown&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 Business Availability Center does not require you to use a specific backup method; however, it is recommended that your backup method accommodate HP Business Availability Center'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 Business Availability Center. Note that the recommended settings differ according to the size of your HP Business Availability Center deployment.

This chapter includes:

- ▶ Oracle Parameter Sizing on page 154
- ▶ Oracle Tablespaces on page 158
- ▶ Oracle Tablespace Settings on page 160
- ▶ Using RAID Configuration on page 164

Oracle Parameter Sizing

The following table describes the recommended size for a number of parameters, when working with the HP Business Availability Center database server:

Parameter Name	HP Business Availability Center 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 Business Availability Center 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 Business Availability Center 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 Business Availability Center 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 Business Availability Center, you must create one or more dedicated default tablespaces for your HP Business Availability Center user schemas. You may also want to create a dedicated temporary tablespace for HP Business Availability Center. 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 Business Availability Center, but not certified for use with HP Business Availability Center, 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 Business Availability Center.

For information on locally managing temporary tablespace using TEMPFILE, see “Temporary Tablespace Settings” on page 162.

Creating an Oracle Tablespace

You create an HP Business Availability Center Oracle tablespace using the **oracle_tablespace_create.sh** script Oracle tablespace for a Windows installation using the **oracle_tablespace_create.bat** script. For a UNIX installation, you create the Oracle tablespace using the **oracle_tablespace_create.sh** script. Both scripts are found in <HP Business Availability Center server>\AppServer\webapps\site.war\DataBases\ORA_DB_Utils.

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 Business Availability Center installation. For more information, see “Data Tablespace Settings” on page 161.

To create an HP Business Availability Center 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 management user schema, specify a file size of at least 500 MB.
 - For a CMDB or CMDB History user schema, specify a file size of at least 1 GB.
 - For a profile user schema, specify a file size of at least 5 GB. Note that the actual size needed varies depending on the amount of data HP Business Availability Center generates and the amount of time you keep historical data.

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 161
- “Temporary Tablespace Settings” on page 162
- “Redo Log Settings” on page 162
- “Undo Segment Settings” on page 163

Data Tablespace Settings

The following table specifies the recommended sizes for HP Business Availability Center tablespaces:

Tablespace	HP Business Availability Center Deployment		Remarks
	Small	Large	
Management	500 MB	1 GB	The specified size is a minimum requirement.
Profile	30 GB	600 GB	The specified size is a minimum requirement. If you create more than one profile database, it is recommended that you create each profile database in a separate tablespace.
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.
BPI	20 MB	50 MB	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 Business Availability Center 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 Business Availability Center Deployment	
	Small	Large
Redo log file size	100 MB	200 MB

Setting	HP Business Availability Center Deployment	
	Small	Large
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 Business Availability Center 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 Business Availability Center

This chapter describes how to configure the Oracle Client for HP Business Availability Center 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 Business Availability Center databases.

This chapter includes:

- Oracle Client Versions and Operating System Platforms on page 167
- Oracle Client Installation on page 168
- Oracle Client Configuration on page 168

Oracle Client Versions and Operating System Platforms

The Oracle Client versions and operating system platforms that are supported and recommended for working with HP Business Availability Center 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 Business Availability Center, 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 Business Availability Center servers access 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 169

MDAC for Oracle Client

HP Business Availability Center 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 Business Availability Center. For HP Business Availability Center 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 Business Availability Center support and certification.

This chapter includes:

- ▶ Checklist for HP Business Availability Center Support and Certification on page 172
- ▶ Oracle Server and Client Requirements on page 179

Note: For more information on the Oracle database configuration settings that should be used when working with Oracle Server and HP Business Availability Center, see “Oracle Server Configuration and Sizing Guidelines” on page 153.

Checklist for HP Business Availability Center Support and Certification

The following table summarizes the Oracle database options that are supported and certified for working with HP Business Availability Center:

Option	Supported	Recommended	Remarks	For more information, see
Oracle edition	Server & Enterprise	Enterprise		
Dedicated HP Business Availability Center 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_MANAGEMENT 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 Business Availability Center uses a connection pool architecture.	
Oracle replication	No full support	No		
Use of SYSTEM, RBS, UNDO, TEMP tablespaces	Not allowed for HP Business Availability Center schema	Not allowed for HP Business Availability Center 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 154

Option	Supported	Recommended	Remarks	For more information, see
STATISTICS_LEVEL parameter value	Typical	Typical		“Oracle Parameter Sizing” on page 154
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		
PARTITION_VIEW_ENABLED parameter value	True; False	True	Performance boost when using the HP Business Availability Center Partition and Purging Manager. This parameter is obsolete in Oracle 10g.	
CURSOR_SPACE_FOR_TIME parameter value	True; False	False		

Option	Supported	Recommended	Remarks	For more information, see
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		
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		

Option	Supported	Recommended	Remarks	For more information, see
FIXED_DATE parameter value	Not set	Not set	HP Business Availability Center 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 163
WORKAREA_SIZE_POLICY parameter value	Manual; Auto	Auto		“Oracle Parameter Sizing” on page 154 (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 158

Option	Supported	Recommended	Remarks	For more information, see
Tablespace extent management	Dictionary; Local; Local uniform	Local automatic for management and profile schemas; Local uniform for TEMP tablespace		“Oracle Tablespace Settings” on page 160
RECYCLEBIN	Off	Off		
Automatic Segment Space Management Tablespace	Yes	Yes		
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 169
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 for working with HP Business Availability Center, see “Software Requirements” on page 123.

Part IV

Appendixes

A

Database Consolidation and Distribution

This appendix describes the situations in which you should consolidate all HP Business Availability Center data in a single database, as opposed to the situations in which you should distribute the data among several different databases.

This chapter includes:

- Advantages of Database Consolidation on page 183
- Reasons for Database Distribution on page 184

Advantages of Database Consolidation

It is generally recommended that you consolidate all data in a single database. The reasons for data consolidation include:

- **Manageability.** A consolidated database reduces administration workload because there is a single point to manage, maintain, and monitor. When working with multiple databases, you must duplicate all maintenance tasks and monitoring procedures.
- **Easier space management.** With a single database, the data is contained in a single disk or disk array and it is easier to track database growth and disk space problems. With multiple databases and distributed disks or disk arrays, you must track the growth of each database and space management becomes more complex.
- **Report limitations.** Percentile and Standard Deviation reports are only supported on a single database.

Reasons for Database Distribution

The distribution of data among several databases should be considered in the following scenarios:

- ▶ **Table size.** Activity performance against small tables (especially insert rates) is generally better than activity performance against large tables. It is recommended that tables be no larger than 1 million rows. To keep table sizes fairly small, you should use partitioning and not distribute the data among different databases. You can fine-tune the partitioning cycle to achieve optimal table sizes. However, if the database is very large (more than 80 GB), keeping table sizes no larger than 1 million rows means that the partitioning cycle would be less than a day. In this case, data for a month would require more than 30 partitions. This scenario should be avoided and you should consider distributing the data among several databases so that the partitioning cycle is not less than a day and table sizes do not exceed 1 million rows.
- ▶ **Maintenance granularity.** If you have different maintenance requirements (such as data purging and backup frequency) for different profiles, you may want to use a different database for each group of profiles.

In addition, if you want to schedule maintenance activities (such as index rebuilds) that require downtime and you do not have a sufficient window for a single maintenance task, you can distribute the data among several databases so that a single maintenance task is completed within the allotted window. Similarly, the unit of recovery is a single database. If there are profiles that you must recover quickly in the case of a system failure, you may want to consider placing these files in a separate database.

- ▶ **Physical resources.** If you want to provide different physical resources to different profiles—for example, you want to allocate fast, expensive disks to very active profiles and slower, less expensive disks to profiles with less activity—you can achieve this by distributing the profiles among different databases.

B

Loader Management

This appendix explains how HP Business Availability Center uses its loader to manage the data streaming from data collectors to the Gateway Server and on to the database. The loader system enables you to verify that data collected by HP Business Availability Center is being reported to the database. HP Software Support also uses loader information when helping to troubleshoot data problems.

This chapter includes:

- Overview of Loader Management on page 186
- Persistency Directories on page 187
- Troubleshooting Loader Problems on page 189
- Logs on page 190
- Changing the Default Logging Level on page 192
- Statistics on page 192

Overview of Loader Management

The HP Business Availability Center Loader enables efficient and persistent reporting of data from data collectors to the Gateway Server and on to the database.

The HP Business Availability Center Loader works in the following manner:

- 1** Data streams (via HTTP requests) from a data collector (Business Process Monitor, Real User Monitor, and SiteScope) to the **Web Data Entry** component, located on the **Data Entry Gate** on the Gateway Server.
- 2** The samples are extracted and verified on the Web Data Entry component.
- 3** The data samples are sent to the loader via the **bus**, with a copy of the data stored in the **Main Persistency** directory if the loader needs to be restarted.
- 4** Every 30 seconds, the loader automatically transfers data from the Persistency Directories to the database. An ack message is then sent to the **Main Persistency** directory to purge the stored data.

If the database is unavailable, the loader transfers the data to the **Recovery Persistency** directories.

If hierarchal data (from Business Process Monitor or Webtrace) does not flatten successfully, it is sent to the **Flatten Failure** directory.

- 5** If the data is not successfully transferred to the database (for example, because the data is corrupt), the loader transfers the data to the **DLQ** ("Dead Letter Queue") directory where it is stored. The DLQ is a subdirectory of the Main Persistency directory. For details on the DLQ, see "Persistency Directories" on page 187.

Note: You can attempt to transfer data in the DLQ to the database. For details on transferring data that is stored in the DLQ directories, see "Troubleshooting Loader Problems" on page 189.

Persistency Directories

The Persistency Directories assist with the transfer of data to the database. The Persistency Directories are:

1 Main

As data is sent from the loader to the database, it is copied to the **Main Persistency** directory. If the loader needs to be restarted, either because it crashed or the data has not reached the database, the data is retrieved from the Main Persistency directory.

Once the data reaches the database, the Main Persistency directory marks the stored data for purging.

Main Persistency Subdirectories

The Main Persistency directory includes the following subdirectories:

- ▶ **Current.** Contains data received from loader prior to crash.
- ▶ **Temp.** Funnels data from Current directory to Recovery Persistency directory to be sent to the database.
- ▶ **Dead Letter Queue (DLQ).** Contains data that does not successfully reach the database.

The Main Persistency Subdirectories store the data as Main Persistency Partition (.mpp) files.

2 Recovery

If data does not reach the database, either due to problems in the database or with the data itself, it gets sent to the **Recovery Persistency** directory. Once there, an ack message is sent to the Main Persistency directory indicating that it can purge its stored data.

Additionally, if the loader is restarted, data from the Main Persistency directory is sent to the Recovery Persistency directory before being sent onward to the database.

If the data continually does not reach the database, either due to being corrupted or not compatible with the existing HP Business Availability Center configuration, it gets sent from the Recovery Persistency Directory to the **DLQ** (Dead Letter Queue) in the Main Persistency Directory.

Recovery Persistency Subdirectories

The subdirectories in the Recovery Persistency Directory are called **queues**. The queues are named according to the following factors, with each factor separated by a dot (.) in the filename:

- ▶ Data Sample Type - The type of data contained in the directory.
- ▶ Destination Database Name - The name of the database.
- ▶ Database Server ID - The name of the server that the database sits on.
- ▶ Index - Indicates the origin of the subdirectory. The index has one of two possible numerical values:
 - ▶ **0**: Indicates that the directory was created on the current server.
 - ▶ **Any other number**: Indicates that the directory was created on a server other than the current server.

The Recovery Persistency Queues contain the following files:

- ▶ **Partition files (.rpp)**. Accumulate the data before it is sent on to the database.
- ▶ **Index files (.idx)**. Keeps track of the data to purge in partition files.

For the location of the Persistency directories, see “Location of Persistency Directories” on page 189.

3 Flatten Failure

The data that flows from Business Process Monitor and Webtrace is hierarchal and must be flattened before it can be sent to the Main Persistency directories. If the process of flattening is continually unsuccessful, due to problems with database connectivity, the data is sent to the **Flatten Failure** directory. The data stays there until it is successfully flattened, when it is sent to the Recovery Persistency directories.

The Flatten Failure directory stores the hierarchal data as Recovery Persistency Partition (.rpp) files.

4 Location of Persistency Directories

The location of the Persistency directories on your machine is as follows:

Main Persistency:

<Gateway Server root directory>\.persist_dir\db_loader\main

Recovery Persistency:

<Gateway Server root directory>\.persist_dir\db_loader\recovery

Flatten Failure:

<Gateway Server root directory>\.persist_dir\db_loader\flattenfailure

Troubleshooting Loader Problems

You can troubleshoot problems involving the transfer of data to the database by checking the loader log files and by checking the number of files in each loader directory.

- ▶ Check the log files for error messages. If the level of logging is set to **debug** (or **info**), all **fatal**, **error**, and **warning** messages appear in the log files. If requested by HP Software Support, you can change the default logging level. For details, see “Changing the Default Logging Level” on page 192. For a definition of HP Business Availability Center log levels, see “Log Severity Levels” in *Reference Information*.
- ▶ Check whether there is an increase in the number of files in the Recovery or Flatten Failure directories, which would indicate a persistency problem. Check the status of the database, via System Health, the LoaderStatistics.log file, or the Loader.log file (the main log file). If you do not find a database problem, contact HP Software Support. For a description of the loader directories, see “Persistency Directories” on page 187.

- ▶ A large number of unsuccessfully extracted samples in the **wdeIgnoredSamples.log** file may be an indication of a configuration problem. Check your configuration and verify that the fields of the sample appear in the meta data definition of the sample type.

Additionally, System Health provides information on the performance of the Data Loading components. For details, see “System Health User Interface” in *Platform Administration*.

Transferring Data from the Dead Letter Queue to the Database

To transfer data in the DLQ to the database:

- 1** Stop the loader.
- 2** Move files from the <Gateway Server root directory>\.persist_dir\db_loader\main\dlq folder to the <Gateway Server root directory>\.persist_dir\db_loader\main\current. Make sure to avoid using identical filenames. For example, if both the current folder and the dlq folder contain a file named part-1.mpp, rename one of them (using the same extension) before moving the files.
- 3** Start the loader.

If data transfer problems still occur, and some of the data does not reach the database, the problematic data is returned to the DLQ directory.

Logs

The logs store information about loader issues. This information is useful when debugging data transfer problems, at which time you need to change the default logging level. For details, see “Changing the Default Logging Level” on page 192.

The loader log files are located at:

<Gateway Server root directory>\log\mercury_db_loader

The log files used are:

- ▶ **loader.all.log.** The generic log which contains information contained in other Data Loading logs.
- ▶ **LoaderStatistics.log.** Contains statistical information about the loader's performance.
- ▶ **LoaderSamples.log.** Contains copies of all the data samples that pass through the loader on the way to the database.

The Web Data Entry logs store information on the data samples passing through the Web Data Entry component on the Data Entry Gate.

The Web Data Entry logs are located at:

<Gateway Server root directory>\log\mercury_wde

The log files used are:

- ▶ **wdePublishedSamples.log.** Contains information on the data samples successfully extracted from the Web Data Entry component on the Data Entry Gate.
- ▶ **wdeIgnoredSamples.log.** Contains information on the data samples that were not successfully extracted from the Web Data Entry component on the Data Entry Gate.

Note: A large number of unsuccessfully extracted samples in the **wdeIgnoredSamples.log** may be an indication of a configuration problem. Check your configuration and verify that the fields of the sample appear in the meta data definition of the sample type.

Additionally, the **wde.all.log** file contains information on all areas of the Web Data Entry Gate, and includes information contained in the aforementioned log directories.

Changing the Default Logging Level

As the loader works automatically, you generally do not need to make any changes to the loader files. You can change the level of logging, however, if requested by HP Software Support, from a reporting level to a debug level.

To change the level of logging for the loader:

- 1** Access the loader properties file, located at:
**<Core Server root
directory>\conf\core\Tools\log4j\mercury_db_loader\loader.properties**
- 2** In the loader.properties file, locate the entry **loglevel=ERROR**.

Note: By default, the loglevel parameter is set to **ERROR** in the loader.properties file.

- 3** Change the loglevel parameter setting from **ERROR** to **INFO** or **DEBUG**.
- 4** Save the file.

If you have multiple Gateway Server machines, repeat this procedure on every machine.

To change the level of logging for Web Data Entry:

- 1** Access the WDE properties file, located at:
**<Core Server root
directory>\conf\core\Tools\log4j\mercury_wde\wde.properties**
- 2** Repeat steps 2 through 4 above.

Statistics

You can view statistics on the logs that are kept on the various levels of the data flow, as follows:

1 WDE Statistics

The Web Data Entry logs contain statistics files on data that enters the Web Data Entry Gate from the Data Collectors. The directory which contains the statistics files is:

<CoreServer Root Directory>\log\mercury_wde

The files available in this directory are:

- ▶ **wdeStatistics.log:** Contains statistics on the performance of the Web Data Entry Gate. The statistics are collected over the following time frames:
 - ▶ Since the most recent WDE Gate reset. The default setting for the WDE Gate to be reset is every 24 hours.
 - ▶ Over the past 5 minutes.

These settings can be customized in the **Infrastructure Settings Manager**, located at: **Admin > Platform > Infrastructure Settings Manager > Foundations > Web Data Entry.**

The data is processed through the following actions outlined in the log file, before being sent onto the database:

- ▶ Decoding
 - ▶ Translation
 - ▶ Publishing
- ▶ **jvm_statistics.log:** Contains statistics on memory usage of the Web Data Entry Gate.

Additionally, the **wde.all.log** file contains information on all areas of the Web Data Entry Gate, and includes information contained in the aforementioned statistics directories.

2 Loader Statistics

Contains statistics files on data that enters the loader from the bus. The path for the loader statistics file is:

<CoreServer Root Directory>\log\mercury_db_loader\LoaderStatistics.log

The **LoaderStatistics.log** file monitors the log activity as follows:

- General Statistics, relating to the quantity of samples and the capacity of the loader.
- Statistics on Specific areas of BAC, such as Site Scope and BPM monitors data.

The statistics in this log are collected over the following time frames:

- Since the most recent loader reset. The default setting for the loader to be reset is once per week.
- Over the past 10 minutes.

These settings can be customized via the JMX tool, located at:

<Core Server Root Directory>/jmx-console/HtmlAdaptor?action=inspectMBean&name=Foundations%3Aservice%3DInfrastructure+Settings+Manager

The statistics in this log are processed in the following manner:

- Events Per Second (EPS)
- Overall loader statistics
- Statistics for each database
- Statistics for samples in each database

C

Data Partitioning and Purging

This appendix describes how to update the profile database table partitioning and purging policy.

This chapter includes:

- ▶ About Data Partitioning and Purging on page 195

About Data Partitioning and Purging

The data collection tables in the profile databases can grow to a very large size. Over time, this can severely degrade system performance.

HP Business Availability Center's Partition and Purging Manager splits fast-growing tables into partitions at defined time intervals. After a defined amount of time has elapsed, data in a partition is made inaccessible for use in HP Business Availability Center reports. After an additional defined amount of time, a partition is purged from the profile database.

You use the Partition and Purging Manager to instruct the platform to automatically partition historical data for later removal from profile databases.

You can set a different partitioning and purging policy for each table in each profile database, or share the same configuration for all profile databases.

Once you have configured the partitioning and purging policy for the tables in a specific profile database, policy changes made for all profile databases do not take effect on this specific database.

Note:

- ▶ To run the Partition and Purging Manager in Microsoft SQL Server 2000, you must enable it. For details, see “Partitioning and Purging Historical Data from Profile Databases” in *Platform Administration*.
 - ▶ The Partition and Purging Manager is not supported in Microsoft SQL Server standard edition.
-

The Partition and Purging Manager creates partitions in the default Oracle tablespace. These partitions automatically acquire the storage parameters of this tablespace.

D

Data Storage Recommendations

This appendix contains recommendations for creating Microsoft SQL Server file groups and Oracle tablespaces in which to store HP Business Availability Center data. This information applies to advanced users only.

This chapter includes:

- ▶ Creating Microsoft SQL Server File Groups on page 197

Creating Microsoft SQL Server File Groups

By default, HP Business Availability Center stores management database objects in two file groups and profile database objects in two file groups (by default, the **PRIMARY** file group for system tables and the **userdata001** file group for user tables). When automatically creating management and profile databases, therefore, HP Business Availability Center creates two file groups for the management database (**PRIMARY** and **userdata001**) and two file groups for the profile database (also **PRIMARY** and **userdata001**). If you are setting up a large deployment of HP Business Availability Center, it is recommended that you create additional file groups in which to store management and profile database objects.

Benefits of Multiple File Groups

Placing the system catalog in its own file group and creating several user-defined file groups for data storage has the following benefits:

- ▶ **Maintenance.** Backing up and restoring databases can be performed on a file group level.
- ▶ **Partial restoration.** You can partially restore a database on the file group level; however when you do so, you must restore the **PRIMARY** file group as well, as it contains the system catalog. By keeping only the **.mdf** file in the **PRIMARY** file group and setting another user file group as the default file group, the **PRIMARY** file group remains small and does not pose difficulties for a partial restoration. This suggested file group structure creates a single user file group, which is a step toward multiple file groups that can be created by the database administrator or by HP Business Availability Center in the future.
- ▶ **Read-only.** A file group can be marked as read-only. Because data cannot be changed, Microsoft SQL Server does not have to acquire shared locks and performance is enhanced.
- ▶ **Recovery.** If the data portion of a database crashes, you can save the changes made before the crash even if the **.mdf** file is damaged, provided the log file and master database are undamaged.
- ▶ **Performance.** Numerous tuning options are available, such as hot tables on fast disks, the separation of table data and indexes into different file groups, and so forth.
- ▶ **Flexibility and Control.** The use of multiple file groups provides the database administrator with greater flexibility and control. In addition, the use of a fixed size for database growth prevents situations in which the 10 percent default autogrowth increment in large databases cannot finish before the timeout setting of HP Business Availability Center components and rolls back database activity, causing an infinite loop of inserts.

You can create an additional file group (together with an additional Microsoft SQL Server database) by running the following scripts.

To create a database with default settings:

```
CREATE DATABASE [testdb]
```

To modify the .mdf (catalog) file SIZE parameter:

```
ALTER DATABASE [testdb]
MODIFY FILE
(
    NAME = N'testdb',
    SIZE = 5MB
)
```

To modify the .mdf (catalog) file FILEGROWTH parameter:

```
ALTER DATABASE [testdb]
MODIFY FILE
(
    NAME = N'testdb',
    FILEGROWTH = 5MB
)
```

To modify the .ldf (log) file SIZE parameter:

```
ALTER DATABASE [testdb]
MODIFY FILE
(
    NAME = N'testdb_log',
    SIZE = 10MB
)
```

To modify the .ldf (log) file FILEGROWTH parameter:

```
ALTER DATABASE [testdb]
MODIFY FILE
(
    NAME = N'testdb_log',
    FILEGROWTH = 50MB
)
```

To return the path of the database's default data directory:

```
SELECT LEFT(filename,  
            LEN(filename) - CHARINDEX(N'\', REVERSE(filename))) AS path  
FROM master.dbo.sysdatabases  
WHERE name = N'testdb'
```

Use the **'testdb'** name returned above in the following script.

To create a user file group, add a data file to it, and set it as the default file group:

```
ALTER DATABASE [testdb]  
ADD FILEGROUP USERDATA001  
  
ALTER DATABASE [testdb]  
ADD FILE  
(  
    NAME = N'testdb_Data001',  
    FILENAME = N'<path to data directory>',  
    SIZE = 10MB,  
    FILEGROWTH = 50MB  
)  
TO FILEGROUP USERDATA001
```

To set the userdata001 file group as the default file group:

```
ALTER DATABASE [testdb] MODIFY FILEGROUP USERDATA001 DEFAULT
```

Note: When creating file groups, the following configuration rules are important for proper file recovery, fault tolerance, performance, and archiving:

- You should create two separate data files, in two separate file groups: the **.mdf** file in the **PRIMARY** file group and the **.ndf** file in the **userdata001** group.
 - The **userdata001** group should be set as the default file group.
 - The **.mdf** file should be the only file in the **PRIMARY** file group that contains the system catalog. The **.mdf** file should be created with an initial **SIZE** parameter value of 5 MB and a **FILEGROWTH** parameter value of 5 MB. The other data file (**.ndf**) and log file (**.ldf**) should be created with an initial **SIZE** parameter value of 10 MB and a **FILEGROWTH** parameter value of 50 MB. In all cases, the **FILEGROWTH** parameter value should be a fixed size (not a percentage).
-

E

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 203
- ▶ Running the Verification Process on page 205
- ▶ Creating Database Users for the Verify Procedure on page 209

About the Verification Process

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

In addition to verifying the HP Business Availability Center 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 209.

Note: HP Business Availability Center 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 Business Availability Center schemas as the login. By assigning an administrator schema to perform datapump operations, you do not have to grant additional permissions to HP Business Availability Center schemas and the datapump tables are created in the administrator schema.

- ▶ If the same database server is listed in the management database using two different host names (for example, dbserver and dbserver.merc-int.com), you are prompted for connection separately for each host name. Make sure that you enter the same connection parameters each time.

Running the Verification Process

You run the database schema verify program from an HP Business Availability Center server machine.

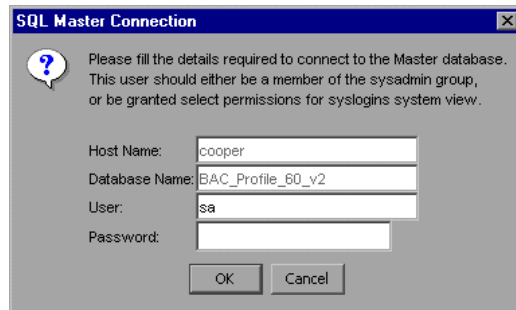
To verify a database schema:

- 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 Business Availability Center root directory>\dbverify\bin directory and run the **run_db_verify.bat** file.
 - ▶ If you are running the verify program from a Solaris platform, make sure that the DISPLAY environment variable is set. Open an X-terminal window, move to the <HP Business Availability Center root directory>/dbverify/bin directory, and type the following:


```
./run_db_verify.sh
```

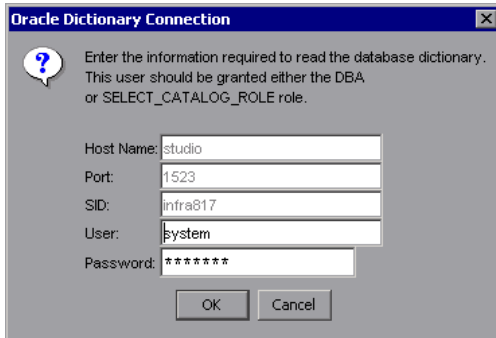
The database verify program starts.

- 2 Specify the details required to connect to the appropriate database:
 - ▶ In the SQL Master Connection dialog box, specify the details required to connect to the Master database.



In the **User** and **Password** boxes, type the user name and password of a user with permissions for the database. (The User box displays the default Microsoft SQL Server administrator user name, **sa**; by default, there is no password.) Click **OK**.

- ▶ In the Oracle Dictionary Connection dialog box, specify the details required to connect to the Oracle database.

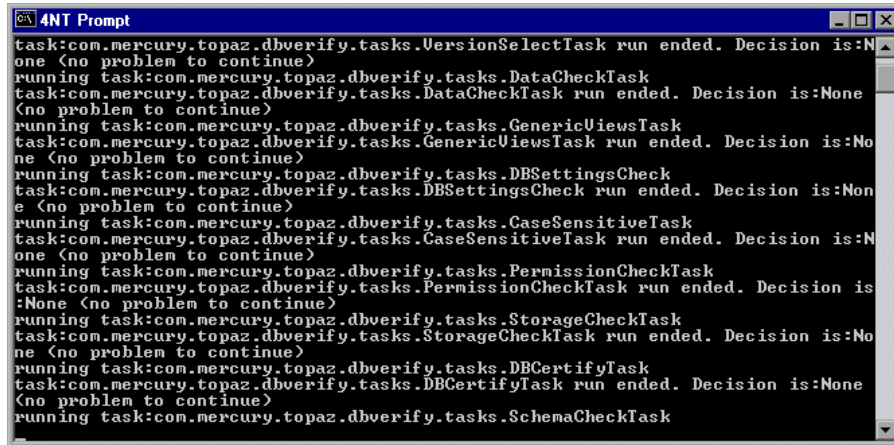


In the **User** and **Password** boxes, type the user name and password of a user with permissions for the database, and click **OK**.

Note:

- ▶ You are prompted for connection data for each different server on which your HP Business Availability Center databases reside.
 - ▶ 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 verify program to operate. For details on how to create this user, see “Creating Database Users for the Verify Procedure” on page 209.
 - ▶ In order to connect using Windows authentication, leave the user and password fields blank. The user entered must be given the appropriate permissions directly, not via a Windows group.
-

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

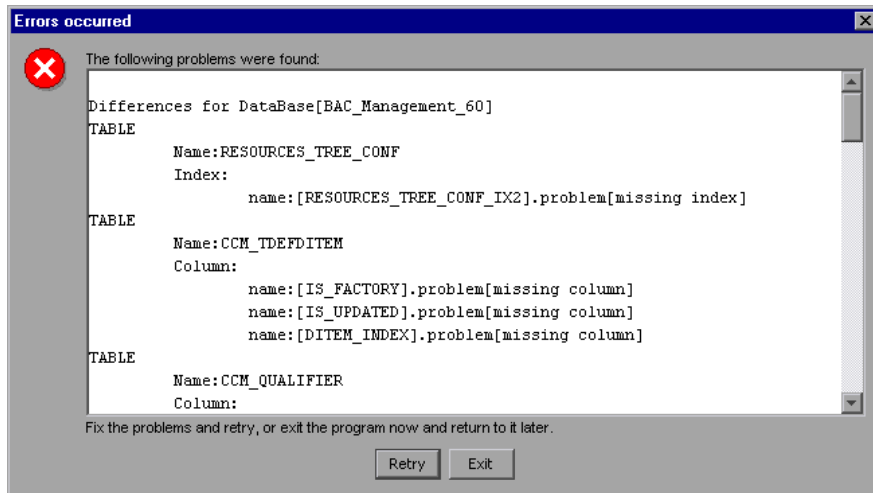


```

task:com.mercury.topaz.dbverify.tasks.VersionSelectTask 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

```

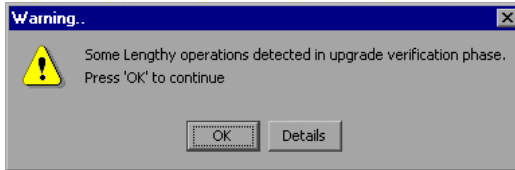
- 4 If problems occur during the database verification, a dialog box is displayed listing the errors.



Either fix the problems found and click **Retry**, or click **Exit** and rerun the database schema verify program at a later date. If you are unable to fix the problems, contact HP Software Support for assistance.

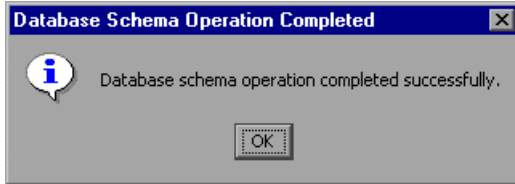
You can view a log file of the errors located in the <HP Business Availability Center root directory>\dbverify\log directory.

- 5 If the verification process detected a later version of the database schemas to which you can upgrade, and encountered lengthy operations that could slow down the upgrade process, resulting in longer downtime, a dialog box is displayed.



Click **OK** to continue, or **Details** to display the estimated times that the lengthy operations detected could take during an upgrade.

- 6 If the database verification is successful, the following confirmation message is displayed.



- 7 Click **OK**.

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.

F

Backing Up and Restoring Monitor Administration Configuration Data

This appendix describes the backup and restore procedures for System Availability Management configuration data. Establishing dedicated backup and restore processes is a fundamental task in a high-availability implementation.

This chapter includes:

- Overview of Backing Up and Restoring Configuration Data on page 212
- Basic Backup and Restore on page 213
- Basic Backup on Replicated Server and Restore on page 214
- LDIF Backup and Restore on page 214
- Replication on page 217
- Starting and Stopping OpenLDAP on page 220
- Backup and Restore Examples on page 222

Note: For details on setting up automatic replication and failover procedures for the LDAP server, refer to HP Software Self-solve knowledge base article 42154.

Overview of Backing Up and Restoring Configuration Data

Monitor Administration configuration data is stored in an LDAP database (Lightweight Directory Access Protocol). By default this database resides in **<HP Business Availability Center server root directory>\openldap\bdb\id*.***. Any backup and restore processes should be performed on this directory.

This section includes the following sections about backup and restore methods and about LDAP database locations:

- ▶ “Backup and Restore Methods” on page 212
- ▶ “LDAP Database Location” on page 212

Backup and Restore Methods

Choose between the following methods for backing up and restoring your Monitor Administration data files:

- ▶ Basic Backup and Restore
- ▶ Basic Backup on Replicated Server and Restore
- ▶ LDIF Backup and Restore

LDAP Database Location

You can check the location of the LDAP database used by HP Business Availability Center to store Monitor Administration data.

To verify the location of the LDAP database that is used for storing the Monitor Administration data:

- 1** Access **Admin > Platform > Setup and Maintenance > Infrastructure Settings**.
- 2** Select the **Foundations** context.
- 3** Choose **Monitor Administration** from the list.
- 4** Locate the **Monitor Administration Data Storage Location** entry.

Basic Backup and Restore

This is the basic backup and restore method and is the preferred method. The LDAP service must not be running during the physical copying of the configuration database, to prevent the possibility of cached data and file locking.

This section includes the following topics:

- “Backup” on page 213
- “Restore” on page 214

Backup

It is good practice to back up the data at least once daily. Store the data on a high-availability server where the files are stored in directories with a date-time stamp name such that multiple historical archives can be preserved. Another technique is to back up to a specific high-availability location using a rotating day of the week, for example, `..\\backups\Sunday`, `..\\backups\Monday`, and so forth.

It is also considered good practice to back up before and after any major upgrade or delete operation.

To manually back up your LDAP database:

- 1** Stop the LDAP service by stopping HP Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Disable Business Availability Center**).
- 2** Copy the contents of the directory `<HP Business Availability Center server root directory>\openldap\bdb` to the backup media. You can do this manually or by running a batch file. For an example of a backup file, see “backup.bat” on page 222.
- 3** Start the LDAP service by re-starting HP Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Enable Business Availability Center**).

Restore

To manually restore your LDAP database:

- 1 Stop the LDAP service by stopping HP Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Disable Business Availability Center**).
- 2 Delete all the files in the **<HP Business Availability Center server root directory>\openldap\bdb** directory. (It is good practice to copy these files to a temporary location before deleting them.)
- 3 Copy the files from the appropriate source to the appropriate target media. You can do this manually or by running a batch file. For an example of a restore file, see “restore.bat” on page 223.
- 4 Start the LDAP service by re-starting HP Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Enable Business Availability Center**).

Basic Backup on Replicated Server and Restore

This method is identical to the Basic Backup and Restore procedure, except that the Basic Backup process is performed on the replicated server file system. It is the most highly recommended solution for high-availability.

The advantage is that the primary service (including HP Business Availability Center) does not need to be shut down for the duration of the operation. Furthermore, the replicated server acts as a cold-backup in the event of a failure of the primary server.

LDIF Backup and Restore

This method uses the LDAP Data Interchange Format (LDIF) representation to extract the contents of an LDAP database to an external, textual, representation.

Note that, with OpenLDAP 2.1.4 or higher, using the back-bdb backend (the default) you can use `slapcat.exe` and `db_dump.exe` while `slapd.exe` is running. This is because the back-bdb backend has page level locking.

This section includes the following topics:

- “Backup” on page 215
- “Restore” on page 216

Backup

You can choose to backup the LDAP database using the LDIF method.

To back up the data to LDIF format:

Note: If you are running OpenLDAP 2.1.4 or higher, you can skip steps 1 and 2 below.

- 1** Stop the LDAP service by stopping HP Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Disable Business Availability Center**).
- 2** From a command window (with sufficient rights to read the files), change the working directory to the **<HP Business Availability Center server root directory>/openldap** directory.
- 3** Invoke the following command:

```
slapcat -f slapd.conf -b "E=SSEnterprise" -f filename
```

Where:

filename is the target LDIF file, for example, **JUL0404.LDIF**.

For an example of a backup file, see “backupLDIF.bat” on page 224.

- 4** Start the LDAP service by re-starting HP Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Enable Business Availability Center**).

Restore

It is excellent practice to test the following process on a staging or test LDAP server.

To restore the data to a new LDAP database:

Note: If you are running OpenLDAP 2.1.4 or later, you can skip steps 1 and 2 below.

- 1** Stop the LDAP service by stopping HP Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Disable Business Availability Center**).
- 2** From a command window (with sufficient rights to read the files), change the working directory to the **<HP Business Availability Center server root directory>/openldap** directory.
- 3** Delete all the files in the **bdb** directory (it is good practice to copy these to a temporary location before deleting).
- 4** Invoke the following command:

```
slapadd -f slapd.conf -b "E=SSEnterprise" -f filename
```

Where:

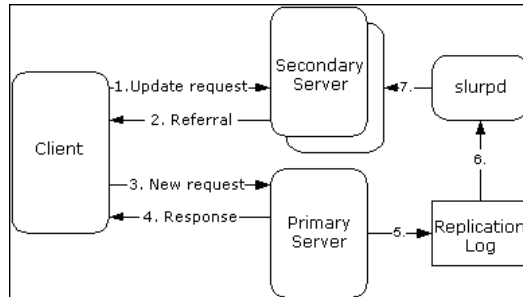
filename is the target LDIF file, for example, **JUL0404.LDIF**.

For an example of a restore file, see “restoreLDIF.bat” on page 225.

- 5** Start the LDAP service by re-starting HP Business Availability Center (**Start > Programs > HP Business Availability Center > Administration > Enable Business Availability Center**).

Replication

OpenLDAP provides replication services (see slurpd at <http://www.openldap.org/software/man.cgi?query=slurpd>). You can use these services to maintain a cold-backup of all information on your primary server (the host machine) replicated to a secondary machine. You should consult with the HP Business Availability Center system administrator before using these services.



This section contains the following topics:

- ▶ Establishing Replication
- ▶ Modifying the Nanny Supervisor Process for Replicated Environments

Establishing Replication

The following procedure assumes that HP Business Availability Center is running on a primary machine that also has an LDAP server installed on it, and that a secondary machine is running the secondary LDAP server.

To establish a replicated OpenLDAP service:

- 1** Stop HP Business Availability Center.
- 2** Zip the **openldap** directory (<HP Business Availability Center server root directory>\openldap).
- 3** Transfer the **openldap.zip** file to the secondary machine and unzip it.
- 4** Add a registry entry as needed.

For details on adding a registry entry (when running openldap as a service on a non-standard port), see the next section.

- 5 Install the **slapd** service on the secondary LDAP server:

```
cd c:\openldap
slapd install
```

- 6 Edit the **slapd.conf** file running on the primary LDAP server (located in the <HP Business Availability Center server root directory>\openldap directory) and add or uncomment the following lines:

```
# Replication MASTER
Replica uri=ldap://slaveHost:9389 binddn="E=SSEnterprise"
bindmethod=simple credentials=f11pp3r
repllogfile ./replicate.log
```

Note: **slaveHost** is the name of the LDAP server running on the secondary machine.

- 7 Edit the **slapd.conf** file running on the secondary LDAP server (located in the **openldap** directory) and add or uncomment the following line:

```
#Replication SLAVE
updatedn "E=SSEnterprise"
updateref ldap://mercuryAMhostname:9389 minServer installation Change
```

Note: **mercuryAMhostname** is the name of the HP Business Availability Center primary machine.

- 8 Locate the file **replica.log** in the <HP Business Availability Center server root directory>\openldap directory on the primary machine. If one is not present, create the file.
- 9 Install **slurpd** on the primary machine:

```
cd c:\HPBAC\openldap
slurpd install
```

- 10 Start HP Business Availability Center.
- 11 Start the LDAP server running on the secondary machine.
- 12 Verify that replication is successful:
 - Add a template container to the enterprise.
 - Connect to the secondary machine with an LDAP browser on port 9389, using the credentials: e=SSEnterprise password: fl1pp3r.
 - Verify that the template container you added is present in the LDAP server on the secondary machine. There should be an LDAP node on the LDAP server for the object you created.

Modifying the Nanny Supervisor Process for Replicated Environments

On HP Business Availability Center machines running both **slapd** and the **slurpd** replication service, you should modify the files which control the Nanny supervisor process.

To modify the files controlling the Nanny supervisor process:

- 1 Modify the file <HP Business Availability Center server root directory>\openldap\openldap_service_recover.bat: add the following line:

```
net start openldap-slurpd
```

- 2 Create the batch file **openldap_stop**, add the following lines, and save it in the **openldap** directory:

```
net stop openldap-slapd
net stop openldap-slurpd
```

- 3 Modify the file <HP Business Availability Center server root directory>\launch_service\dat\nanny\a_openldap.nanny: Change the **stop_nt** line to run the **openldap_stop** batch file, to stop the OpenLDAP services. For example,

```
stop_nt="d:\HPBAC/openldap/openldap_stop"
```

Starting and Stopping OpenLDAP

By default, OpenLDAP starts each time HP Business Availability Center is started. Furthermore, OpenLDAP performs a data integrity test automatically at startup. To prevent HP Business Availability Center from starting OpenLDAP automatically, you must remove the **a_openldap.nanny** file in the **<HP Business Availability Center server root directory>\launch_services\dat** directory, or you must remove its .nanny extension.

Note: If OpenLDAP is brought down in a non-orderly manner, it may not restart correctly, which affects all HP Business Availability Center functionality. When running HP Business Availability Center on a Solaris machine, you must perform a database recovery before restarting HP Business Availability Center. For details, see the following procedure.

To stop and start OpenLDAP on a Windows or Solaris platform, you perform the procedure described in the following sections:

- “OpenLDAP Running on a Windows Platform” on page 220
- “OpenLDAP Running on a Solaris Platform” on page 221

OpenLDAP Running on a Windows Platform

OpenLDAP is run as an NT service on Windows server machines. Even if HP Business Availability Center is stopped, OpenLDAP continues to run as a service, as it may be serving multiple Gateways Server machines in the configuration. If the Gateways Server is restarted and the service is already running, the service continues to run.

You can assess if the OpenLDAP service is running and other properties of the service by using the services management tool (**Start > Control Panel > Administrative Tools > Services**).

To stop and start OpenLDAP running on a Windows platform:

Issue the command:

```
net stop openldap-slapd  
net start openldap-slapd
```

You can manually remove OpenLDAP as a service by changing directory:

```
cd \<HP Business Availability Center home directory>\openldap
```

and issuing the command:

```
slapd remove
```

The service can be manually reinstalled by entering:

```
slapd install
```

A registry key is required (and is automatically created at installation) to run OpenLDAP as a service on a non-standard (9389) port:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\OpenLDAP\Parameters]  
"Urls"="ldap://:9389"
```

If the port is changed, the HP Business Availability Center configuration needs to be updated to reflect the change.

OpenLDAP Running on a Solaris Platform

OpenLDAP runs on Solaris as the **slapd** process. Stopping HP Business Availability Center stops the **slapd** process automatically. To manually stop the slapd process, issue the **KILL_SLAPD** script (which issues the **pkill** command). To start the process, run the **RUN_SLAPD** process.

Backup and Restore Examples

This section contains the following examples:

- backup.bat
- restore.bat
- backupLDIF.bat
- restoreLDIF.bat

backup.bat

The following batch file can be run on a Windows system to back up a Monitor Administration LDAP database:

```
@echo on
rem back.bat
rem
rem This batch file will back up a Monitor Administration database
rem
if "%1"==" " goto usage
if not exist %1 md %1
xcopy bdb\*. * %1
goto done
:usage
echo backup [DATAFILE]
echo Where:
echo [DATAFILE] is the path of the file to back up to
echo Example: backup \\backups\mar0404\
:done
```

restore.bat

The following batch file can be run on a Windows system to restore a Monitor Administration LDAP database:

```
@echo on
rem restore.bat
rem
rem This batch file will copy a Monitor Administration database from a
rem previous backup.bat database copy
rem
if "%1"==" " goto usage
cd bdb
if not exist hold md hold
cd hold
echo Y | del *.*
cd ..
xcopy *.* hold
echo Y | del *.*
cd ..
xcopy %1 bdb
goto done
:usage
echo restore [DATAFILE]
echo Where:
echo [DATAFILE] is the path of the file to restore
echo Example: restore \\backups\mar0404.ldif
:done
```

backupLDIF.bat

The following batch file can be run on a Windows system to back up a Monitor Administration database to an LDAP Interchange Format textual file:

```
@echo off
rem back.bat
rem
rem This batch file will back up a Monitor Administration database to an
rem LDAP Interchange Format textual file
rem
if "%1"==" " goto usage
slapcat -f slapd.conf -b "E=SSEnterprise" -l %1
goto done
:usage
echo backup [DATAFILE]
echo Where:
echo [DATAFILE] is the path of the file to back up to
echo Example: backup \\backups\mar0404.ldif
:done
```


restoreLDIF.bat

The following batch file can be run on a Windows system to restore a Monitor Administration database from a previously created LDIF file:

```
@echo on
rem restore.bat
rem
rem This batch file will restore a Monitor Administration database from a
rem previous backup.bat created LDIF file
rem
if "%1"==" " goto usage
cd *ldb
if not exist hold md hold
cd hold
echo Y | del *.*
cd ..
xcopy *.* hold
echo Y | del *.*
cd ..
slapadd -f slapd.conf -b "E=SSEnterprise" -l %1
goto done
:usage
echo restore [DATAFILE]
echo Where:
echo [DATAFILE] is the path of the file to restore
echo Example: restore \\backups\mar0404.ldif
:done
```


G

Support for Oracle Real Application Cluster

This appendix contains the configuration that need to be done for HP Business Availability Center 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 227
- ▶ Client Side Configuration for Oracle RAC on page 229
- ▶ Server Side Configuration on page 232
- ▶ Create/Connect to One of the HP Business Availability Center Databases on page 234

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.

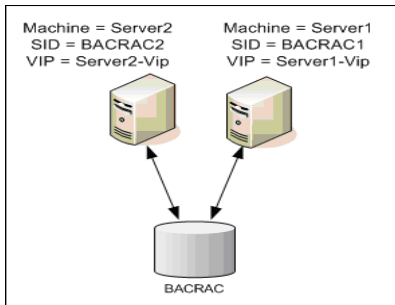
For more details about Oracle RAC, see the Oracle documentation site (http://www.oracle.com/pls/db102/to_toc?pathname=rac.102%2Fb14197%2Ftoc.htm).

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

- Oracle RAC database name: BACRAC
- Machine names: Server1, Server2
- On each machine there is an Oracle instance of BACRAC:
 - SID on Server1: BACRAC1
 - SID on Server2: BACRAC2
- 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 BACRAC.



Client Side Configuration for Oracle RAC

HP Business Availability Center uses DataDirect's JDBC driver to connect to regular Oracle databases and to Oracle RAC databases.

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

- 1 On all servers, create the file < HP Business Availability Center root directory>\conf\bac-tnsnames.ora.**

The format of **bac-tnsnames.ora** is the same as `tnsnames.ora`:

```
<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, bac-tnsnames.ora would be configured as:

```
BACRAC =
(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 = BACRAC)
  )
)
```

2 On all servers, edit the file <HP Business Availability Center root directory>\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=<BAC_HOME>
\conf\bac-tnsnames.ora;TNSServerName=${sid}
```

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

► **UNIX.**

```
ddoracle.url=jdbc:mercury:oracle:TNSNamesFile=<BAC_HOME>
/conf/bac-tnsnames.ora;TNSServerName=${sid}
```

3 On all servers, edit the file < HP Business Availability Center root directory>\dbverify\conf\jdbc.drivers.properties.

Change this file the same way you changed < HP Business Availability Center root directory>/conf/ jdbc.drivers.properties in the previous step.

- 4 On all servers, edit the file < HP Business Availability Center root directory>\dbverify\conf\bpi_process_repository\template\bia-model-repository-db.oracle.properties.

Modify the `jdbc.connection.url` setting to include the same information that you included in the `bac-tnsnames.ora` file in step 1.

For example:

```
jdbc.connection.url jdbc:oracle:thin@(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 = BACRAC)))
```

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 Business Availability Center Database” on page 233

Recommendation for HP Business Availability Center Database

It is recommended to use the session based algorithm since HP Business Availability Center 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 One of the HP Business Availability Center Databases

When creating a new database schema or connecting to an existing schema in the Server and Database Configuration utility, fill all the parameters as you usually would 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 bac_tnsnames.ora .

In the example, the parameters would be:

Parameter	Value
Oracle Machine	Server1-Vip
Port	1521
SID	BACRAC

Index

A

- alert file, Oracle Server 137
- archive file, Oracle Server 137
- authentication
 - enabling HP Business Availability Center for Windows authentication 61
- Authentication Mode dialog box 28

B

- backup
 - Microsoft SQL Server databases 91
 - Oracle Server databases 149

C

- checklists
 - Microsoft SQL Server 2005 support and certification 41
 - Oracle Server support and certification 172
- CMDB database fragmentation, Microsoft SQL Server 102
- CMDB index fragmentation, Microsoft SQL Server 105
- CMDB index fragmentation, Oracle Server 146
- Coalition Settings dialog box 29
- CPU, Oracle Server 137

D

- data file property settings, Microsoft SQL Server 116
- data loaders 185
 - changing default logging level 192

- managing data 185
 - troubleshooting 189
- data management 185
- data partitioning 195
- data placement
 - Microsoft SQL 2000 Server 67
 - Microsoft SQL Server 2005 79
- data purging 195
- data storage 197
- database
 - advantages of consolidation 183
 - configuration options on Microsoft SQL Server 2000 72
 - configuration options on Microsoft SQL Server 2005 84
 - creating users 209
 - creation on Microsoft SQL Server 56
 - distribution 184
 - 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 136
 - maintenance on Microsoft SQL Server 91
 - maintenance on Oracle Server 135
 - Microsoft SQL Server maintenance references 114
 - monitoring on Microsoft SQL Server 109

Index

- 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
- purging data from 195
- requirements 13
- schema verification 203
- setting on Oracle Server 127
- 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 schemas overview
 - Microsoft SQL Server 55
- databases
 - HP Business Availability Center 13
- dbverify utility 205
- deployment
 - Microsoft SQL Server 19, 37
 - Oracle Server 121
- distribution
 - databases 184
- distribution statistics
 - Microsoft SQL Server 107

F

- file groups
 - Microsoft SQL Server 197
 - 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 Business Availability Center
 - databases 13
 - enabling for Windows authentication 61
- HP Business Availability Center database
 - sizing 15

I

- index fragmentation
 - Microsoft SQL Server CMDB database 105
- index fragmentation, CMDB
 - Oracle Server 146
- installation
 - Microsoft SQL Server 2000 23
 - Microsoft SQL Server 2005 42
 - Oracle Client 168
- installing Microsoft SQL Server 23
- instance
 - Oracle Server 125
- Instance Name dialog box 24

L

- LDAP
 - backing up and restoring monitor
 - administration configuration data 211
 - backup and restore examples 222
 - backup and restore methods 212
 - backup.bat 222
 - backuptdif.bat 224
 - basic backup and restore 213
 - basic backup on replicated server and restore 214
 - database location 212
 - modifying nanny process for replicated environments 219
 - replication 217
 - restore.bat 223

- restoreldif.bat 225
- starting and stopping openLDAP 220
- LDIF backup and restore 214
- loaders. *See* data loaders
- locally managed tablespace
 - creation on Oracle Server 159
- log level
 - changing default for data loaders 192
- log placement
 - Microsoft SQL Server 2000 67
 - Microsoft SQL Server 2005 79

M

- management database
 - definition 13
- management user schema 128
- MDAC
 - Oracle Client 169
 - Oracle Server 169
- Microsoft SQL Server
 - CMDB index fragmentation 105
 - collecting statistics 94
 - creating databases 56
 - data storage 197
 - database backup 91
 - database fragmentation 96, 102
 - database integrity 96
 - database maintenance 91
 - database maintenance references 114
 - database monitoring 109
 - database schemas overview 55
 - deployment overview 19, 37
 - distribution statistics 107
 - installing 23
 - 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 84
 - 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

- openLDAP
 - running on a Solaris platform 221
 - running on a Windows platform 220
- Oracle
 - alert file 137
 - database environment 127
 - instances 125
 - optimizing query performance 141
 - Recovery Manager (RMAN) 151

Index

- tablespace settings 160
- Oracle Client
 - configuring 168
 - installation 168
 - MDAC 169
 - system requirements 167
- Oracle schema
 - privileges for Application Management 133
- Oracle Server
 - checklist for support and certification 172
 - CMDB index fragmentation 146
 - collecting statistics for CMDB 141
 - collecting statistics for profile databases 138
 - CPU 137
 - data storage 197
 - database backup 149
 - database maintenance 135
 - deployment overview 121
 - dictionary-managed tablespace
 - coalescing 138
 - Input/Output 137
 - instances 125
 - parameter sizing 154
 - RAID configuration 164
 - requirements 179
 - sizing guidelines 153
 - system requirements 122
 - tablespace maintenance 138
 - tablespaces 158
- Oracle Server Real Application Cluster
 - see Real Application Cluster 227
- P**
 - parameter sizing
 - Oracle Server 154
 - permissions
 - Microsoft SQL Server 2000 65
 - Microsoft SQL Server 2005 77
 - privileges, Application Management Oracle schema 133
 - profile databases
 - collecting statistics with Oracle Server 138
 - definition 13
 - profile user schemas 132
 - purging data from database 195
 - Purging Manager 195
- Q**
 - query performance on Oracle Server, optimizing 141
- R**
 - RAID configuration, Oracle Server 164
 - Real Application Cluster
 - client configuration 229
 - connect to or create Management database 234
 - introduction 227
 - server configuration 232
 - Recovery Manager (RMAN)
 - Oracle Recovery Manager 151
 - refreshing statistics
 - Microsoft SQL Server 108
- S**
 - Select Components dialog box 26
 - server and database configuration utility 56
 - 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 168
 - sizing guidelines
 - Microsoft SQL Server 115
 - Oracle Server 153
 - statistics
 - collecting for CMDB in Oracle Server 141
 - collecting for profile databases in Oracle Server 138

- collecting in Microsoft SQL Server 94
- system databases
 - Microsoft SQL Server 2000 70
 - Microsoft SQL Server 2005 82
- System Global Area (SGA)
 - Oracle Server 136
- system requirements
 - Microsoft SQL Server 20
 - Oracle Client 167
 - Oracle Server 122

T

- tablespace settings, Oracle Server 160
- tablespaces
 - Oracle Server 158
- tablespaces, Oracle Server
 - maintenance 138
- tempdb database
 - Microsoft SQL Server 2000 70
 - Microsoft SQL Server 2005 82
 - settings on Microsoft SQL 117
- tnsnames.ora
 - location and configuring 168
- transaction log
 - backup 95
 - Microsoft SQL Server 2000 67
 - Microsoft SQL Server 2005 78
- troubleshooting
 - data loaders 189

U

- user schema
 - management 128
 - profile 132
- users
 - creating for database schema
 - verification 209

V

- verification
 - database schemas 203
 - Microsoft SQL Server 2005 settings 51

W

- Windows authentication
 - enabling HP Business Availability
 - Center 61

