# Hewlett Packard Enterprise
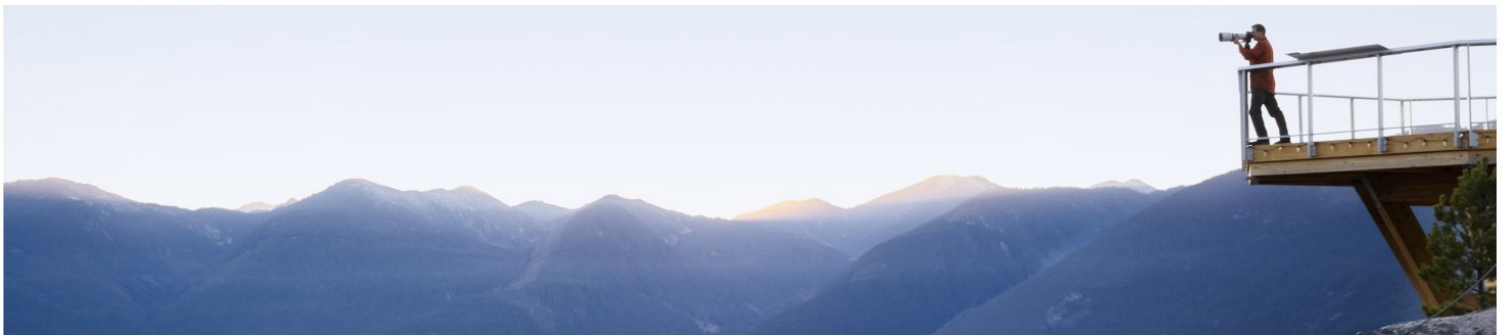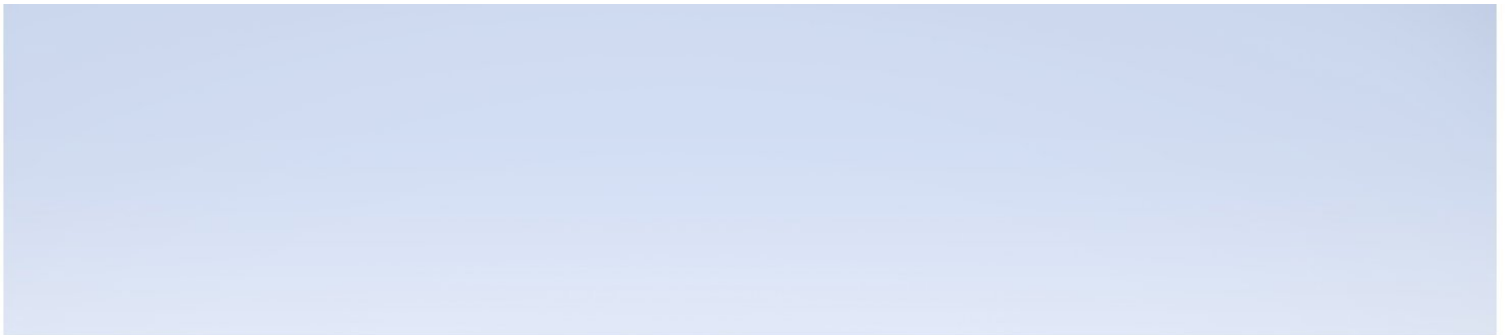
# Application Performance Management

Version 9.40, Released August 2017

# Database Guide

Published September2017

## Legal Notices

### Warranty

The only warranties for Hewlett Packard Enterprise 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. HPE 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 HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright Notice

© Copyright 2005-2017 Hewlett Packard Enterprise Development LP

### Trademark Notices

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

AMD and the AMD Arrow symbol are trademarks of Advanced Micro Devices, Inc.

Google™ and Google Maps™ are trademarks of Google Inc.

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

iPod is a trademark of Apple Computer, Inc.

Java is a registered trademark of Oracle and/or its affiliates.

Microsoft®, Windows®, Windows NT®, Windows Server® and Windows Vista™ are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle and/or its affiliates.

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: https://softwaresupport.hpe.com/group/softwaresupport/search-result?keyword=.

This site requires an HPE Passport account. If you do not have one, click the **Create an account** button on the HPE Passport Sign in page.

## Support

Visit the HPE Software Support website at: **https://softwaresupport.hpe.com**

This website provides contact information and details about the products, services, and support that HPE Software offers.

HPE Software 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 website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HPE 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 HPE Passport user and sign in. Many also require a support contract. To register for an HPE Passport ID, go to **https://softwaresupport.hpe.com** and click **Register**.

To find more information about access levels, go to:
**https://softwaresupport.hpe.com/web/softwaresupport/access-levels**

## HPE Software Integrations and Solutions

Visit the Integrations and Solutions Catalog at **https://softwaresupport.hpe.com/km/KM01702731** to explore how the products in the HPE Software catalog work together, exchange information, and solve business needs.

# Contents

# Part I: Introducing the Database Environment

# Chapter 1: Introduction to Preparing the Database Environment

This chapter contains information on the types of databases used with HPE Application Performance Management (APM).

This chapter includes:

- "Database in Use Overview" below
- "Application Performance Management Sizing" on the next page
- "Hardware Requirements" on the next page

## Database in Use Overview

To work with APM, you must set up the following databases:

- **Management database.** For storage of system-wide and management-related metadata for the APM environment. APM requires one management database. You can create this database manually, or byusing the Server and Database Configuration utility.
- **Profile database(s).** For storage of raw and aggregated measurement data obtained from the APM 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 **Manage Profile Databases** page, accessible from **Admin > Platform > Setup and Maintenance**.
- **RTSM.** The Run-time Service Model contains a database for storage of configuration information that is gathered from the various APM and third-party applications and tools. This information is used when building APM views.
- **Analytics Database(s).** For storage of raw and aggregated metrics data. Each Microsoft SQL Server database or Oracle user schema can host up to 1,000,000 metrics. You can create up to five SQL Server databases or Oracle user schemas.

  **Note:** Historical data will be lost (for example, SLM data, Alerts, SiS data, and KPI Over Time) if you change the default profile schema.

You can set up APM 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, see "Deploying and Maintaining the Microsoft SQL Server Databases" on page 11.

If you are working with an Oracle Server database, see "Deploying and Maintaining the Oracle Server Database" on page 43.

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

  **Note:**

- Database servers must be set to the same time zone, daylight savings settings and time as the APM servers.

- For security reasons, HPE recommends that database administrators change the default administrator user name and password according to their organization password policy.

- For details on working in a non-English language APM environment, see "Working in Non-English Locales" in the Platform Administration Guide. If you are working with a Microsoft SQL Server database, note that in non-English language APM environments the database and APM system must be set up in the same language and support is only available in that language.

- For details on the data partitioning and purging, see "Partitioning and Purging Historical Data from Databases" in the APM Platform Administration Guide.

# Application Performance Management Sizing

APM database configuration requirements are dependent on the amount of data, as well as on the runtime load, generated by APM. Database size is also affected by the purging policy defined for the different services. For example, the Profile database size is affected by the Partition Manager purging rules per data sample.

The following table can help determine database size according to the capacity of the main system elements consuming database resources:

| Database Element | APM Deployment | |
| --- | --- | --- |
| | Standard | Large |
| EPS (events per second) | 100 | 400 or more |
| Logged-in APM users | 25 | 75 or more |
| CMDB Objects and Links | 100,000 | 1 million or more |

# Hardware Requirements

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

| Deployment | Number of Processors | Physical Memory |
| --- | --- | --- |
| Small | 4 CPU cores | Minimum: 4 GB Recommended: 8 GB RAM |
| Large | Minimum 8 CPU cores | Minimum: 8 GB Recommended: 24 GB RAM and up |

For Microsoft SQL software requirements, see "Software Requirements" on page 12.

For Oracle software requirements, see "Software Requirements" on page 45.

# Part II: Deploying and Maintaining the Microsoft SQL Server Databases

# Chapter 2: Microsoft SQL Server Deployment Overview

You can use Microsoft SQL Server for deploying APM databases. This chapter describes the following topics related to deploying Microsoft SQL Servers for use with APM:

This chapter includes:

- "About Microsoft SQL Server Deployment" on page 14
- "System Requirements" below

## About Microsoft SQL Server Deployment

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

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

  For details on installing and configuring Microsoft SQL Server, see "Installing and Configuring Microsoft SQL Server" on page 18.

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

  You can create APM databases using the Server and Database Configuration utility.

  For details on creating Microsoft SQL database, see "Creating and Configuring Microsoft SQL Server Databases" on page 26.

## System Requirements

This section describes the system requirements for working with Microsoft SQL Server in conjunction with APM.

## Hardware Requirements

For APM hardware sizing guidelines, see "Hardware Requirements" on page 10. For Microsoft SQL Server hardware requirements, see the installation guide for your Microsoft SQL Server release for your operating system.

## Software Requirements

The following table lists the Microsoft SQL Servers supported for working with APM. A supported option means that HPE quality assurance personnel have successfully performed basic tests on that option.

| Database Release | | |
|---|---|---|
| **Version** | **System Type** | **Service Pack** |
| Microsoft SQL Server 2014 Enterprise Edition - with failover clustering | 64 BIT | — |
| Microsoft SQL Server 2014 Enterprise Edition | 64 BIT | — |
| Microsoft SQL Server 2014 Developer Edition | 64 BIT | — |
| Microsoft SQL Server 2012 Enterprise Edition - with failover clustering | 64 BIT | 1,2 |
| Microsoft SQL Server 2012 Enterprise Edition | 64 BIT | 1,2 |
| Microsoft SQL Server 2012 Developer Edition | 64 BIT | 1,2 |
| Microsoft SQL Server 2008 R2 Enterprise Edition - with failover clustering | 64 BIT | 1,2 |
| Microsoft SQL Server 2008 R2 Enterprise Edition | 64 BIT | 1, 2 |
| Microsoft SQL Server 2008 Enterprise Edition | 32 BIT | 2, 3 |
| Microsoft SQL Server 2008 Enterprise Edition | 64 BIT | 2, 3 |

**Note:**

- Only supported service packs should be installed. Patches newer than the installed service pack are also supported.
- Consult the Microsoft documentation for supported platforms.
- Failover clustering is supported with all APM databases. To configure failover clustering with APM, in the Setup and Database Configuration Utility enter the cluster server name as the host name. No extra configuration is required.

# Examples of Tested Deployments

The following table details the deployment environments that have been rigorously tested by HPE quality assurance personnel.

| Database Release | | | |
| --- | --- | --- | --- |
| **Version** | **System Type** | **Service Pack** | **Operating System** |
| Microsoft SQL Server 2014 Enterprise Edition - with failover clustering System | 64 BIT | — | Windows 2012 R2 Enterprise Edition (64-bit) |
| Microsoft SQL Server 2012 Enterprise Edition - with failover clustering | 64 BIT | Service Pack 1 | Windows 2012 Enterprise Edition (64-bit) |
| Microsoft SQL Server 2008 R2 Enterprise Edition | 64 BIT | Service Pack 1, Service Pack 2 | Windows 2008 R2 Enterprise Edition Service Pack 1 (64-bit) |
| Microsoft SQL Server 2008 Enterprise Edition | 32 BIT | Service Pack 3 | Windows 2008 Enterprise Edition Service Pack 2 |

# About Microsoft SQL Server Deployment

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

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

  For details on installing and configuring Microsoft SQL Server, see "Installing and Configuring Microsoft SQL Server" on page 18.

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

  You can create APM databases using the Server and Database Configuration utility.

  For details on creating Microsoft SQL database, see "Creating and Configuring Microsoft SQL Server Databases" on page 26.

# Chapter 3: Manually Creating the Microsoft SQL Server Database

This chapter describes the procedure for manually creating APM databases on a Microsoft SQL Server.

This chapter includes:

- "Microsoft SQL Server Databases Overview" below
- "Manually Creating the HPE Application Performance Management Microsoft SQL Server Databases" below

## Microsoft SQL Server Databases Overview

APM uses the following databases for its persistency:

- **Management.** The APM management database.
- **Profile.** The APM profile database.
- **RTSM.** The universal configuration management database.

During the APM 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 APM.

> **Note:** You cannot create an RTSM database manually.

For details about installing an APM server, see the APM Installation Guide.

## Manually Creating the HPE Application Performance Management Microsoft SQL Server Databases

After you have installed and configured Microsoft SQL Server, you create APM databases for the storage of data collected by APM.

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

If you created the databases in advance, you can connect APM to the existing databases from the Server and Database Configuration utility, and to the existing profile databases from the **Database Management** page in **Admin > Platform > Setup and Maintenance**.

For information on the Server and Database Configuration utility, see "Setting Database Parameters" in the APM Installation Guide.

For information on creating profile databases in APM, see "Database Management Overview" in the APM Platform Administration Guide.

This section includes the following topics:

- "Creating Databases" below
- "Creating the Objects" below

# Creating Databases

The database administrator should create one of each of the following Microsoft SQL Server databases:

- Management
- RTSM
- Profile

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 "Creating and Configuring Microsoft SQL Server Databases" on page 26.

> **Note:** When the APM 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 APM database, you create objects by running SQL scripts. The object creation scripts are located in the **<Gateway Server root directory>\AppServer\webapps\site.war\DataBases\SQL_Svr_DB_Utils** directory. For the Management and Profile databases, the SQL script is dynamically created by the **generate_create_script** utility, located in the **<Gateway Server root directory>\dbverify\bin** directory.

> **Note:** The SQL script files generated by the **generate_create_script** utility contain references to the [PRIMARY] files group. In order to use a different file group, manually open the .sql file in a text edtor and replace all instances of [PRIMARY] with the desired file group.

## Creating the Management Objects

To create the APM Management objects, you must generate and run the Management database SQL script.

**To generate the Management database SQL script:**

Run one of the following scripts file located in **<Gateway Server root directory>\dbverify\bin** directory:

*For Windows:*

```
generate_create_script.bat [Generated Script Name] [dbType] [sessionType]
```

*For UNIX:*

```
./generate_create_script.sh [Generated Script Name] [dbType] [sessionType]
```

The following parameters should be used for the Management schema:

- **Generate Script Name.** The name of the SQL file to generate, for example, `Management.sql`.
- **dbType.** `SqlServer`
- **sessionType.** `Management`

**To run the Management database SQL script:**

1. Create the Management database. For details, see "Creating Databases" on the previous page.

2. Connect to the Management database and run the script generated in the previous section to deploy the Management objects.

   **Note:** If your MSSQL database has Turkish collation, make sure all content of the generated SQL script is in uppercase.

3. Run the following SQL script from the objects scripts directory:

   `management_sql_inserts.sql`

## Creating the Profile Objects

To create the APM profile objects, you must generate and run the profile database SQL script.

**To generate the Profile database SQL script:**

Run one of the following scripts file located in **<Gateway Server root directory>\dbverify\bin** directory:

*For Windows:*

`generate_create_script.bat [Generated Script Name] [dbType] [sessionType]`

*For UNIX:*

`./generate_create_script.sh [Generated Script Name] [dbType] [sessionType]`

The following parameters should be used for the Profile schema:

- **Generate Script Name.** The name of the SQL file to generate, for example, `Profile.sql`.
- **dbType.** `SqlServer`
- **sessionType.** `Profile`

To run the Profile database SQL script:

1. Create the Profile database. For details, see "Creating Databases" on the previous page.

2. Connect to the Profile database and run the script generated in the previous section to deploy the Profile objects.

   **Note:** If your MSSQL database has Turkish collation, make sure all content of the generated SQL script is in uppercase.

3. Run the following SQL script from the objects scripts directory:

   `profile_sql_inserts.sql`

# Chapter 4: Installing and Configuring Microsoft SQL Server

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

This chapter includes:

- "Workflow for Microsoft SQL Server Deployment" below
- "Installation Prerequisites" on the next page
- "Notes and Limitations" on the next page
- "Checklist for Support and Certification" on page 20
- "Installing Microsoft SQL Server" on page 20
- "Configuring Microsoft SQL Server" on page 22
- "Verifying and Modifying Server and Database Settings" on page 22

## Workflow for Microsoft SQL Server Deployment

To deploy Microsoft SQL Server for use with APM, perform the following steps:

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

   For details, see "Database File Layout" on page 27.

2. **Review installation prerequisites**

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

3. **Review server and database checklists**

   These checklists summarize the server and database configuration options supported and recommended for APM.

   For details, see "Checklist for Support and Certification" on page 20.

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

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

5. **Configure Microsoft SQL Server**

   For details, see "Configuring Microsoft SQL Server" on page 22.

6. **Create APM databases on Microsoft SQL Server**

   For details, see "Creating Databases" on page 16.

7. **Configure APM databases**

   For details, see "Configuring Databases" on page 29.

8. **Verify Microsoft SQL Server and databases**

   For details, see "Verifying and Modifying Server and Database Settings" on page 22.

9. **Set up Windows authentication**

For details, see "Using Windows Authentication to Access Microsoft SQL Server Databases" on page 41. This step is only relevant if you have selected Windows authentication instead of SQL Server authentication.

# Installation Prerequisites

- **System Requirements**

  For a list of supported operating systems, see "System Requirements" on page 12.

- **Software**

  | Software | Version Needed |
  | --- | --- |
  | Microsoft Windows Installer | 4.5 or later |

- **Database file placement**

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

- **Memory**

  There must be at least 2 GB of RAM.

- **User Accounts**

  - If you plan to perform activities outside the local machine (such as file copying to or from different servers, backing up to a shared remote repository, and so forth), you must have a domain account available for Microsoft SQL Server services.

  - If you want to install APM using the installation wizard, you must provide a user account that has database creator privileges.

- **Miscellaneous**

  - Verify that the disk where Microsoft SQL Server is to be installed is not compressed.

  - Disable antivirus software and any other applications that may interfere while installing Microsoft SQL Server.

# Notes and Limitations

- Microsoft SQL Server must be installed on a dedicated machine to host the APM databases.

- Do not change the default memory settings. Allow the Microsoft SQL Server to manage memory dynamically, except when you configure **awe enabled** support. The AWE feature was removed from SQL Server 2012, and is not recommended for use starting SQL Server 2008 R2.

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

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

- When working with Microsoft SQL Server, the CMDB collation and the SQL Server collation must be the same in the old server and new server in order for an upgrade to run properly. For a specific collation that is

different from the current server collation, install a separate Microsoft SQL Server instance where you need to specify the specific collation. For more information, see https://msdn.microsoft.com/en-us/library/ms143219(v=sql.110).aspx.

- APM does not support Microsoft SQL Server Page/Row compression.

# Checklist for Support and Certification

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

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

| Subject | Microsoft SQL Server | |
|---|---|---|
| | **Supported** | **Recommended** |
| Instances | Default, Single | |
| Authentication Mode | Mixed, Windows | |
| Collation | Case-Insensitive. APM does not support binary sort order and case sensitivity. Only case-insensitive order with a combination of accent, kana, or width settings is supported. | Use the Collation Settings dialog box to select the collation. Do not select the binary check box. Accent, kana, and width sensitivity should be selected according to the relevant data language requirements. The selected language must be the same as the OS Windows regional settings language. |
| Network Libraries | **Server:** TCP/IP and Named Pipes <br><br> **Client:** TCP/IP and Named Pipes | **Server:** TCP/IP <br><br> **Client:** TCP/IP |
| Server Configuration Options | Defaults, unless instructed otherwise | |
| Data File Properties | Manual file growth, or FILEGROWTH less than or equal to 100 MB | FILEGROWTH: ~30-100 MB |
| Collation Database Property | Server default | |
| Database Options | Defaults, unless instructed otherwise | |
| Recovery Model | Any | Full |

# Installing Microsoft SQL Server

Although the installation process is not difficult, it is important that you familiarize yourself with all of the installation details so that you select the appropriate options. Selecting the default options may, in some

cases, negatively affect the Microsoft SQL Server's performance.

Select the following options in the installation dialog boxes:

- **Feature Selection dialog box configuration.**

  - Remove Full Text Search from the list since APM does not use this indexing search feature

  - Under Destination Folder, make sure that the Data Files directory is stored on a fault-tolerant disk system, for example, RAID 1. Even though these system databases are fairly small, they are essential for the operation of Microsoft SQL Server.

- **Instance Name dialog box configuration.** For details on named instances, see "Port Allocation Options for Named Instances" on page 24. You access a default instance by specifying the server name or IP address.

- **Service Account dialog box configuration.**

  - If all Microsoft SQL Server activities are outside the local machine (for example, file copying to or from a different server, backing up to shared remote repository, replication with other servers, ActiveX script job steps, CmdExec job steps, and so forth), choose **Use a Domain User account** and specify the user name, password, and domain of a user that is a member of the local machine's administrator group, and that has the appropriate permissions for network resources.

  - If all Microsoft SQL Server activities are limited to the local machine, choose **Use the built-in System account** and select **Local system**. This selection provides SQL Server administrative privileges on the local machine only.

- **Authentication Mode dialog box.** APM works with both Windows authentication and with Microsoft SQL Server authentication (recommended).

  To enable Microsoft SQL Server authentication:

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

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

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

  This section is important to correctly configure the database for non-Latin character sets. Check with your database administrator to select the correct collation setting.

  To correctly configure the database for non-Latin character sets:

  a. Click **Customize**.
  b. Select **Windows collation designator and sort order** and in the Collation Designator field, select the collation from the drop down list.

  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, but this is not recommended for dedicated APM databases. Because of the flexibility in collation management in Microsoft SQL Server, 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.

- **Data Directories dialog box.** Specify the destination folders for the data and log files of the user databases and temporary databases, as well as the backup directory. It is recommended to store data and log files on high-performance storage systems.
- Install the latest service pack for Microsoft SQL Server available at the Microsoft Download Center site at http://www.microsoft.com/downloads/details.aspx?FamilyID=cb6c71ea-d649-47ff-9176-e7cac58fd4bc&amp;DisplayLang=en. For the latest service pack supported to work with APM, see "Software Requirements" on page 12.

# Configuring Microsoft SQL Server

This section describes the service and server options you can configure after you have installed Microsoft SQL Server and includes the following topics:

- "Service Configuration Options" below
- "Service Configuration Options" below

## Service Configuration Options

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

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

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

## Service Configuration Options

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

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

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

# Verifying and Modifying Server and Database Settings

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

| Server/Database Setting | How to Verify/Modify the Setting |
|---|---|
| Default Instance | In the operating system's Services applet, a default Microsoft SQL Server instance and a named instance appear as SQL Server (Instance_Name). |
| Authentication Mode | In the Microsoft SQL Server Management Studio, right-click the server, choose **Properties**, and click the **Security** tab. Select **SQL Server and Windows Authentication mode (Mixed Mode)**. |
| Collation Settings | Run the following command: `sp_helpsort` |
| Network Libraries | On the Server, select **Start > Programs > Microsoft SQL Server <release number> > Configuration Tools > Configuration Manager**.<br><br>At **SQL Native Client..** choose **client protocols** and ensure the selected protocol is in the enabled state.<br><br>• Supported<br>  Shared memory, TCP/IP ,and Named Pipes for both the server and client<br>• Recommended<br>  Only TCP/IP for both the server and client. |
| View or change server configuration options | • To allow the viewing of all options, run: `EXEC sp_configure 'show advanced options', 1 reconfigure with override`<br>• To view the current values, run: `EXEC sp_configure`<br>• To change a setting, run: `EXEC sp_configure '<option>', <value>`<br><br>Some options take effect only after you run **reconfigure** with **override**, while others require restarting the Microsoft SQL Server service. For more details, see Microsoft SQL Server documentation. |
| Check whether the user who is going to create APM databases has CREATE DATABASE permissions | Log on to the Microsoft SQL Server Management Studio with the login you want to check, and run the following:<br><br>```\nUSE master\nIF PERMISSIONS() & 1 = 1\nPRINT 'User has CREATE DATABASE permissions'\nELSE\nPRINT 'User does not have CREATE DATABASE permissions'\n``` |

| Server/Database Setting | How to Verify/Modify the Setting |
|---|---|
| Check whether the APM database user has enough permissions in the database | • Log on to the Microsoft SQL Server Management Studio with the user name you want to check.<br>• Change the context of the database to the required database.<br>• Open a new query and do the following in each database:<br><pre>select case when IS_MEMBER ('db_owner')=1<br> or IS_SRVROLEMEMBER ('sysadmin')=1<br> or (IS_MEMBER ('db_ddladmin') = 1 and<br> IS_MEMBER ('db_datareader')=1 and<br> IS_MEMBER ('db_datawriter')=1 and<br> IS_MEMBER ('db_denydatareader')=0 and<br> IS_MEMBER ('db_denydatawriter')=0 )<br>then 'User has enough permissions'<br>else 'User does not have enough permissions'<br>end</pre> |
| Data and log file destination directory is not compressed (only in NTFS) | Right click the directory, choose **Properties**, and then **Advanced**. Verify that the **Compression** check box is cleared. |
| Database and database file properties (including recovery model and collation properties) | • To view the database and database file properties, run:<br>`EXEC sp_helpdb <database name>`<br>• To change the database properties, run: `ALTER DATABASE <database name> SET <option> <value>`<br>• To change the database file properties, run: `ALTER DATABASE <database> MODIFY FILE (name = <filename>, <property> = <value>)`<br>You can also view or change these properties from the Database Properties dialog box in the Enterprise Manager. |
| Microsoft SQL Server service pack version and edition | Log on to Management Studio and make the following query: `select @@version` |

## 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:**

1. Select **Start > Programs > Microsoft SQL Server > Configuration Tools > SQL Server Configuration Manager**.

2. Expand Protocols for <instance name>, and double-click **TCP/IP**.

3. In the TCP/IP Properties dialog box, on the IP Addresses tab, several IP addresses appear, in the format IP1, IP2, up to IPAll. For each address:

   a. If the TCP Dynamic Ports dialog is set to 0, indicating that the database engine is listening on dynamic ports, delete the 0.

   b. In the TCP Port box, enter the port number on which you want this IP address to listen, and click **OK**.

4. In the console pane, click SQL Server Services.

5. In the details pane, right-click SQL Server (<instance name>), and then click **Restart** to stop and restart SQL Server.

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

# Chapter 5: Creating and Configuring Microsoft SQL Server Databases

This chapter describes the creation and configuration of APM databases on a Microsoft SQL Server.

This chapter includes:

- "Creating Databases" below
- "Configuring Databases" on page 29

## Creating Databases

This section includes the following topics:

- "Database Permissions" below
- "Database File Layout" on the next page

## 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:** APM login accounts should be mapped to **dbo** in the database. Members of the sysadmin server role automatically have CREATE DATABASE permissions, and are also mapped to **dbo** in all databases. A database owner is automatically mapped to **dbo** in the database.

To check whether a user has CREATE DATABASE permissions, log on 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 on 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 can 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. Therefore, it is advisable that there be as little interference as possible with the log activity. It is usually sufficient to place the log on a RAID 1 system because of the serial writes to the log. If you have processes reading from the log (for example, triggers accessing the inserted and deleted views which are formed from the log records or transactional replication), or several log files for different databases, consider placing the log file(s) on a RAID 0+1 (striped mirror) system.
- **Data files.** Data files should be placed on a RAID 0+1 system for optimal performance.

## File and Database Properties

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

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

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

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

automatic expansion to finish. For large databases, a percentage growth increment can lead to exponential growth of the database and should be avoided.

## 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. The growth increment of the tempdb should be large enough to avoid fragmentation, yet not too large to expand in a reasonable amount of time. Create the tempdb with a minimum, initial size of 1 GB and with a growth increment of 50 MB. The tempdb database should be striped across several disks, ideally on a RAID 0+1 controller. It is recommended to move the tempdb database to its own set of disks.

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

## File Groups

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

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

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

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

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

- **Partial Restoring.** Microsoft SQL Server does not support the restoration of a single table. Even if you place a single table in a file group, you cannot restore a file group to a point in time earlier than the rest of the data. Instead, you must apply all log file backups in order to synchronize the file group with the rest of the data. Microsoft SQL Server supports partial restoration to a database with a different name. A partial restoration allows you to restore a single file group, and supports point-in-time restoration. However, you must restore the **PRIMARY** file group because it contains the SYSTEM tables.

    To 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.

For additional data on creating file groups, see "Microsoft SQL Server Data Storage Recommendation" on page 82.

## 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 about setting database sizes, see "The tempdb Database Settings" on the previous page.

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

- **master, msdb, model.** These databases, although crucial for the operation of Microsoft SQL Server, are smaller than tempdb because they store only meta data.

  It is strongly recommended to use a fault tolerant disk—ideally, RAID 1—for these databases.

  > **Note:** For APM 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 command:**

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

# Configuring Databases

After 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" below
- "Database Configuration Options" on the next page

> **Important:** If you enabled APM to create APM 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 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. After 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 26.

# 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 APM certification:

| Configuration Option | Description | Default | APM Certification in Microsoft SQL Server |
|---|---|---|---|
| ANSI NULL default (see note below) | Specifies whether the database columns are defined as NULL or NOT NULL, by default | Not set | Not set |
| ANSI PADDING | Controls the way the column stores values shorter than the defined size of the column and the way the column stores values that have trailing blanks in char, varchar, binary, and varbinary data. | OFF | ON |

| Configuration Option | Description | Default | APM Certification in Microsoft SQL Server |
|---|---|---|---|
| Auto close | Specifies whether the database shuts down after its resources are freed and all users exit. | Not set | Not set<br><br>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<br><br>**Note:** If set, constant growth/ shrinkage may cause file system fragmentation. |
| Auto update statistics | Specifies whether out-of-date statistics required by a query for optimization are built automatically during optimization | Set | Set |
| Compatibility level | The version of Microsoft SQL Server that the database appears to be (for the application) | The same version as the release installed | The same version as the release installed |
| Read only | Database is read only | Not set (READ_ WRITE) | READ_WRITE |
| Recovery | The database recovery model determines the recovery capabilities by controlling the amount of bulk operation logging (such as Select into, Bulk, Insert, Create index, LOB manipu-lation). The higher the recovery model, the higher the recovery capabilities. However, the amount of logging also increases, which may affect performance. | Full | Full (unless you are certain that the lower recovery capabilities are sufficient for your system) |
| 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 |

| Configuration Option | Description | Default | APM Certification in Microsoft SQL Server |
|---|---|---|---|
| Page verify | Specifies whether incomplete pages can be detected | Set | Set |
| Quoted identifiers enabled | 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 following command:

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

For information on SQL databases, see Microsoft SQL Server Books Online.

# Chapter 6: Maintaining Microsoft SQL Server Databases

This chapter describes the various maintenance tasks that are recommended for APM 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" below
- "Database Integrity and Fragmentation" on page 35

## Backing Up Databases

Microsoft SQL Server supports three main types of database backup — full, differential, and log. It also supports file/file group backup, which is discussed in a separate section below. 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 the next page
- "Differential Backup" on the next page
- "Log Backup" on the next page
- "File/File Group Backup" on page 35
- "Maintenance Plan" on page 35
- "Transaction Log Issues" on page 35

# 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 profile database), 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 eight contiguous 8K pages) that were changed since the last full backup. When restoring a database, you need only restore the last differential backup performed after the full backup.

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

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

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

# Log Backup

A log backup—unlike full and differential backups which are mainly based on backing up an image of extents—backs up transactions from the transaction log and replays them upon restoration. In order to perform a log backup, 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 following 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 the next page
- "Understanding File System Fragmentation" on the next page
- "Understanding Internal Fragmentation" on the next page
- "Understanding External Fragmentation" on page 37
- "Detecting and Handling Index Fragmentation" on page 37
- "Supplied Utilities for Monitoring and Rebuilding Indexes" on page 38
- "Utility to Rebuild All Indexes in Database" on page 38
- "Utility to Rebuild Indexes Based on the Fragmentation Level of Each Index" on page 39

# Database Integrity

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

```
DBCC CHECKDB ('database name')
```

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

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

# Understanding File System Fragmentation

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

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

If you cannot anticipate the future size of a database file, to avoid small, fragmented parts, use a large value as the file growth increment. Do not use too large a value, however, because this leads to client request timeouts when the file autogrows (for more details, see "Database Permissions" on page 26). 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 APM 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 `ALTER INDEX REBUILD` command (online or offline) and `FILLFACTOR` option. The `FILLFACTOR` option specifies the fullness of the leaf level index pages.

# Understanding External Fragmentation

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

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

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

# Detecting and Handling Index Fragmentation

## Using sys.dm_db_index_physical_stats to Detect Fragmentation

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

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

The table syntax of `sys.dm_db_index_physical_stats` is:

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

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

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

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

For more information about **sys.dm_db_index_physical_stats**, see the Microsoft SQL Server documentation.

## Handling Fragmentation

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

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

- **Between 5% and 30%.** Use the `ALTER INDEX REORGANIZE` command to reorganize the index. Index reorganization is always executed online.

- **Greater than 30%.** Use the `ALTER INDEX REBUILD` command to rebuild the index. Index rebuilding can be executed online or offline. Rebuild the indexes online to achieve the availability similar to that of the Reorganize option.

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

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

> **Note:** It is strongly recommended that you create an automatic index rebuild task for each of the APM databases. In the following databases, run this task at least once a day: the Profile database and the RTSM (CMDB) database.

## Supplied Utilities for Monitoring and Rebuilding Indexes

APM provides two utilities that can be used to detect and rebuild fragmented indexes. The **rebuild_fragmented_indexes.bat** utility uses the Logical scan fragmentation and Scan Density criteria to detect, and if instructed to rebuild, fragmented indexes. The operation of listing the fragmented tables has a very small impact on system performance and can be executed on line. The operation of rebuilding the indexes usually hinders performance as tables are partially locked during the process, and CPU and I/O are heavily utilized. It is recommended to rebuild the indexes in a maintenance window. The utilities should be run by a database administrator.

The utilities for Microsoft SQL Server are located in the **<Gateway Server root directory>\AppServer\webapps\site.war\DataBases\SQL_Svr_DB_Utils** directory on the APM Gateway Server.

## Utility to Rebuild All Indexes in Database

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

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

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

- SQL Server name
- Database name
- SA password

**Example:**

```
rebuild_indexes.bat SQL_SRVR_3 BSM_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 **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.

**Example:**

```
rebuild_fragmented_indexes.bat SQL_SRVR_3 BSM_DB_3 ad%min52 1
```

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

# Distribution Statistics

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

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

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

# Collecting Statistics for RTSM

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

An example of a scenario that warrants a manual refresh of the CMDB statistics is automated Data Flow Management (DFM) jobs. DFM is the process responsible for automatically detecting configuration items and inserting them into the CMDB.

# Utility to Refresh Statistics

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

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

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

From **<Gateway Server root directory>\AppServer\webapps\site.war\DataBases\SQL_Svr_DB_Utils**, execute **update_statistics.bat** with the following parameters:

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

**Example:**

```
update_statistics.bat SQL_SRVR_3 RTSM_DB_3 ad%min52 1
```

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

# Chapter 7: Using Windows Authentication to Access Microsoft SQL Server Databases

Unless configured otherwise, APM uses Microsoft SQL Server authentication to access Microsoft SQL Server databases. However, Windows authentication can also be used.

This chapter describes how to enable APM to use Windows authentication to access Microsoft SQL Server databases.

## Enabling APM to Work with Windows Authentication

You can enable APM to use Windows authentication instead of Microsoft SQL Server authentication to access any of the APM databases.

To enable APM to use Windows authentication to access a Microsoft SQL database, you must:

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

This section includes the following topics:

- "Configuring Microsoft SQL Server to Use Windows Authentication" below
- "Launching the APM Service with a Windows User" below
- "Creating or Connecting to a Microsoft SQL Database Using Windows Authentication" on the next page

## 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 then click **OK**.

## Launching the APM Service with a Windows User

By default, the APM 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 APM 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 HPE Application Peformance Management Service User" in the APM Installation Guide.

**To change the APM service user:**

1. Disable APM (**Start > Programs > HPE Application Performance Management > Administration > Disable HPE Application Performance Management**).

2. In Microsoft's Services window, double-click **HP Business Service Management**. The HP Business Service Management Web Server Properties (Local Computer) dialog box opens.

3. Click the **Log On** tab.

4. Select **This account** and browse to choose the user you previously defined for your Microsoft SQL Server.

5. Enter the selected user's Windows password and confirm this password.

6. Click **Apply** to save your settings and **OK** to close the dialog box.

7. Enable APM (**Start > Programs > HPE Application Performance Management > Administration > Enable HPE Application Performance Management**).

# 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 "Server Deployment and Setting Database Parameters" in the APM Installation Guide.

# Part III: Deploying and Maintaining the Oracle Server Database

# Chapter 8: Oracle Server Deployment Overview

You can set up APM schemas on an Oracle Server.

This chapter describes the following topics related to deploying Oracle Servers for use with APM.

This chapter includes:

- "About Oracle Server Deployment" below
- "System Requirements" below

## About Oracle Server Deployment

To deploy Oracle Server for use with APM, perform the following procedures:

- **Install Oracle Server.**

  For details on Oracle software installation, see the installation guide in the documentation for your specific Oracle platform. For software installation options, see "Oracle Server Configuration and Sizing Guidelines" on page 51 and "Oracle Summary Checklist" on page 71.

- **Build a database on Oracle Server to store APM data.**

  For instance configuration and sizing guidelines, see "Oracle Server Configuration and Sizing Guidelines" on page 51. For details on database instance installation, see the installation guide in the documentation for your specific Oracle platform.

- **Create one or more Oracle tablespaces to store APM data.**

  For details, see "Oracle Tablespaces" on page 54.

- **Create an Oracle user schema for APM schemas.**

  You create APM user schemas using the Server and Database Configuration utility.

## System Requirements

This section describes the system requirements for working with Oracle Server in conjunction with APM.

This section includes the following topics:

- "Hardware Requirements" on the next page
- "Software Requirements" on the next page
- "Examples of Tested Deployments" on page 46
- "Oracle Instances" on page 46

# Hardware Requirements

For APM hardware sizing guidelines, see "Hardware Requirements" on page 10.

For Oracle hardware requirements, see 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, see: http://otn.oracle.com/documentation/index.html.

# Software Requirements

The following table lists the Oracle Servers supported for working with APM. A supported option means that HPE quality assurance personnel have successfully performed basic tests on that option.

| Database Release | |
| --- | --- |
| **Version** | **System Type** |
| Oracle 11.2 (11g R2) Enterprise Edition | 64 BIT |
| Oracle 11.2 (11g R2) RAC Enterprise Edition | 64 BIT |
| Oracle 12c Enterprise Edition | 64 BIT |
| Oracle 12c RAC Enterprise Edition | 64 BIT |

**Note:**

- It is strongly recommended to apply the latest critical Oracle patches. For details, consult the Oracle documentation.

- The Oracle Partitioning option must be enabled.

- If you are using an Oracle version prior to 10.2.0.5, you must apply the Oracle patch that fixes Oracle defect # 5866410. For more details, see the Oracle website for information related to this defect number.

- Consult the Oracle documentation for supported platforms.

- If you install Oracle 12c:

  a. On the Oracle database machine, go to **oracle/product/<oracle version>/db_ 001/network/admin**.

     where **<oracle version>** is the Oracle version you are using, for example 12.1.0.2

  b. Locate **sqlnet.ora** and add the following line:

     **SQLNET.ALLOWED_LOGON_VERSION_SERVER=8**

  c. If a pluggable database is used (multitenant database feature), locate **listener.ora** and add the line **USE_SID_AS_SERVICE_LISTENER=on**.

  d. Save the file and restart **listener**.

# Examples of Tested Deployments

The following table details the deployment environments that have been rigorously tested by HPE quality assurance personnel:

| Database Release | | Operating System |
|---|---|---|
| **Version** | **System Type** | |
| Oracle 12c RAC Enterprise Edition | 64 BIT | Linux Enterprise Edition RHEL 6.5 |
| Oracle 12c Enterprise Edition | 64 BIT | Windows 2008 R2 Enterprise Edition |
| Oracle 11.2 (11g R2) RAC Enterprise Edition | 64 BIT | Linux Enterprise Edition RHEL 5 |
| Oracle 11.2 (11g R2) Enterprise Edition | 64 BIT | Linux Enterprise Edition RHEL 6.5/5 |

# Oracle Instances

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

For APM certification, do not use more than one Oracle instance. If you use more than one instance for the APM 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 APM.

# Chapter 9: Manually Creating the Oracle Schemas

This chapter describes how to manually create and populate APM user schemas. It also contains information on Oracle permissions for APM user schemas.

This chapter includes:

- "Oracle Server Databases Overview" below
- "Creating APM Schemas Manually" below
- "APM Oracle Schema Privileges" on page 50

> **Note:** When manually creating the APM databases, it is strongly recommended that you run the database schema verify program after connecting APM to these databases via the Server and Database Configuration Utility. For information on the verification process, see "Database Schema Verification" on page 91.

## Oracle Server Databases Overview

APM uses the following databases for its persistency:

- **Management.** The APM management database.
- **Profile.** The APM profile database.
- **RTSM.** The universal configuration management database.

During the APM 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 APM.

For details about installing APM server, see the APM Installation Guide.

## Creating APM Schemas Manually

The APM user schemas can be created manually or automatically using the Server and Database Configuration utility. For more information about the Server and Database Configuration utility, see "Setting Database Parameters" in the APM Installation Guide.

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. APM automatically displays the default port, **1521**.
- **SID.** The Oracle instance name that uniquely identifies the Oracle database instance being used by APM.
- **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, `APM_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 APM tablespace, see "Oracle Tablespaces" on page 54.
- **Temporary tablespace.** The name of the temporary tablespace you assigned to the user schema. The default Oracle temporary tablespace is **temp**.

> **Note:** If you are using the SQL*Plus Oracle client, make sure you are aware of its limitations. For example, a database creation command may exceed the maximum command-line length supported by SQL*Plus. For information about the limitations of SQL*Plus, see:
>
> **Release 10.2** - SQL*Plus® User's Guide and Reference - A SQL*Plus Limits, Release 10.2
>
> **Release 11.2** - SQL*Plus® User's Guide and Reference - A SQL*Plus Limits, Release 11.2
>
> If you run into problems due to an SQL*Plus limitation, see the documentation provided with your Oracle application.

This section includes the following topics:

- "Creating a Management User Schema Manually" below
- "Creating a Profile User Schema Manually" on the next page

> **Note:** You cannot create an RTSM User Schema manually.

# 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 about the Server and Database Configuration utility, see "Setting Database Parameters" in the APM Installation Guide.

> **Note:** If you cannot submit database administrator connection parameters over a non-secure connection, you may need to configure a management user schema manually .

**To generate and run the Management schema SQL script:**

1. Create the user schema:

   Create a user schema with the permissions described in "APM Oracle Schema Privileges" on page 50.

2. Generate the Management schema SQL script:

   Run the following script file located in the **<Gateway Server root directory>\dbverify\bin** directory:

   *For Windows:*

   ```
   generate_create_script.bat [Generated Script Name]  [dbType]  [sessionType]
   ```

   *For UNIX:*

   ```
   ./generate_create_script.sh [Generated Script Name]  [dbType]  [sessionType]
   ```

   The following parameters should be used for the Management schema:

   - **Generate Script Name.** The name of the SQL file to generate, for example, `management.sql`.

   - **dbType.** `Oracle`

   - **sessionType.** `Management`

3. Run the Management schema SQL script:

   a. Connect to the Management schema and run the script generated in step 2 to deploy the Management objects.

   b. Run the following SQL script located in **<Gateway Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utils**:

      ```
      management_ora_inserts.sql
      ```

   c. When the script is finished running, commit the change.

# Creating a Profile User Schema Manually

Profile user schemas contain performance data collected by APM, and are generally large databases. For information on the size of the profile databases, see "Oracle Server Configuration and Sizing Guidelines" on page 51.

Profile user schemas can be created manually or automatically using the **Manage Profile Databases** page in **Admin > Platform > Setup and Maintenance**. For information about creating profile databases automatically, see "Database Management Overview" in the Platform Administration Guide.

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 **Manage Profile Databases** page in **Admin > Platform > Setup and Maintenance**. For more information about connecting to profile user schemas from Platform Administration, see "Database Management Overview" in the Platform Administration Guide.

**Note:** If you cannot submit database administrator connection parameters over a non-secure connection, you may need to configure Profile user schemas manually.

**To generate and run the Profile schema SQL script:**

1. Create the user schema:

   Create a user schema with the permissions described in "APM Oracle Schema Privileges" on the next page.

2. Generate the Profile schema SQL script:

   Run the following script file located in the **<Gateway Server root directory>\dbverify\bin** directory:

   *For Windows:*

   ```
   generate_create_script.bat [Generated Script Name]  [dbType]  [sessionType]
   ```

   *For UNIX:*

   ```
   ./generate_create_script.sh [Generated Script Name]  [dbType]  [sessionType]
   ```

   The following parameters should be used for the Profile schema:

   - **Generate Script Name.** The name of the SQL file to generate, for example, `profile.sql`.

   - **dbType.** `Oracle`

   - **sessionType.** `Profile`

3. Run the Profile schema SQL script:

    a. Connect to the Profile schema and run the script generated in step 2 to deploy the Profile objects.

    b. Run the following SQL script located in **<Gateway Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utils**:

```
profile_ora_inserts.sql
```

    c. When the script is finished running, commit the change.

# APM 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 **<Gateway Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_Utils** directory) describes the required database permissions that must be granted to the APM user. These permissions are also used by the APM platform components to create a new Oracle user:

- `GRANT "CONNECT" TO <HPE Application Performance Management Oracle user schema>`

- `GRANT CREATE SEQUENCE TO <HPE Application Performance Management Oracle user schema>`

- `GRANT CREATE TABLE TO <HPE Application Performance Management Oracle user schema>`

- `GRANT CREATE TRIGGER TO <HPE Application Performance Management Oracle user schema>`

- `GRANT UNLIMITED TABLESPACE TO <HPE Application Performance Management Oracle user schema>`

- `GRANT CREATE VIEW TO <HPE Application Performance Management Oracle user schema>`

- `GRANT CREATE PROCEDURE TO <HPE Application Performance Management Oracle user schema>`

- `ALTER USER <HPE Application Performance Management Oracle user schema> DEFAULT ROLE ALL`

If the schema is being used for the RTSM database (CMDB schema ), then the following permission is also needed:

`GRANT CREATE TYPE TO <HPE Application Performance Management Oracle user schema for RTSM>`

> **Note:** APM supports any user with higher permissions. For APM certification, use an Oracle user that has the exact Oracle permissions described above.

# Chapter 10: Oracle Server Configuration and Sizing Guidelines

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

This chapter includes:

## Oracle Parameter Settings

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

| Parameter Name | APM Deployment | | Remarks |
| --- | --- | --- | --- |
| | Small | Large | |
| DB_BLOCK_SIZE | 8K | 8K-16K | Should be a multiple of the operating system block size. |
| DB_CACHE_ADVICE | ON | ON | For gathering statistics when tuning is required |
| SGA_TARGET | 1 GB | 4 GB and higher | Oracle 10g. See remarks below the table. |
| MEMORY_TARGET | 2 GB | 5 GB and higher | Oracle 11g. See remarks below the table. |
| LOG_BUFFER | 1 MB | 5 MB | |
| DB_FILE_ MULTIBLOCK_ READ_COUNT | Oracle default value | Oracle default value | |
| PROCESSES | 200 | 400 | Add an additional 100 as a safety net |
| SESSIONS | 225 | 445 | (1.1 * PROCESSES) + 5 |
| OPTIMIZER_ INDEX_ COST_ ADJ parameter value | 100 | 100 | Affects performance |

| Parameter Name | APM Deployment | | Remarks |
| | Small | Large | |
| --- | --- | --- | --- |
| TIMED_ STATISTICS | True | True | |
| LOG_ CHECKPOINT_ INTERVAL | 0 | 0 | |
| LOG_ CHECKPOINT_ TIMEOUT | 0; or greater than or equal to 1800 | 0; or greater than or equal to 1800 | |
| OPTIMIZER_ MODE | ALL_ ROWS | ALL_ ROWS | |
| CURSOR_ SHARING | Exact | Exact | |
| OPEN_CURSORS | 800 | 800 | |
| COMPATIBLE | The same as the release installed | The same as the release installed | |
| SQL_TRACE | False, True | False | |
| UNDO_ MANAGEMENT | Auto | Auto | |
| UNDO_ RETENTION | Oracle default value | Oracle default value | Automatic tuning is performed in Oracle 10g and up |
| RECYCLEBIN | Off | Off | |
| NLS_LENGTH_ SEMANTICS | BYTE | BYTE | This parameter controls the length definition of varchar columns. |
| WORKAREA_SIZE_ POLICY | AUTO | AUTO | |
| PGA_AGGREGATE_ TARGET | 400 MB | 1 GB and higher | |
| STATISTICS_LEVEL | TYPICAL | TYPICAL | Enables tuning if required. |
| STAR_ TRANSFORMATION_ ENABLED | TRUE | TRUE | Determines whether star transformation can be applied to queries by the optimizer. Setting the parameter to false affects the performance of End User Management reports. |

| Parameter Name | APM Deployment | | Remarks |
| | Small | Large | |
| --- | --- | --- | --- |
| _ALWAYS STAR_ TRANSFORMATION | TRUE | TRUE | This hidden parameter tells optimizer to always favor star transformation when applicable. |
| OPTIMIZER_ CAPTURE_SQL_ PLAN_BASELINES | FALSE | FALSE | Oracle 11g. Controls Automatic Plan Capture as part of Oracle 11g SQL Management Base (SMB). |
| AUDIT_TRAIL | NONE | NONE | In Oracle 11g, the default is changed from none to db, which means that out-of-the-box database auditing is written to the SYS.AUD$ audit trail table. It is advisable to change this value to none to avoid growth of the system tablespace. |
| CURSOR_ SPACE_ FOR_ TIME | False | False | |
| USE_STORED_ OUTLINES | False | False | Oracle Default |
| BLANK_ TRIMMING | False | False | |
| FIXED_DATE | Not set | Not set | APM uses the SYSDATE function for generating system time as part of the application process. |
| _PARTITION_ LARGE_EXTENTS | FALSE | FALSE | Relevant for Oracle 11.2.0.2 only. When this hidden parameter is set to TRUE, it affects the size of partitions in native partitioned tables. The initial extent allocated for each partition is very large, thus causing unwanted growth of database data files. |

**Note:**

- **SGA_TARGET.** Setting this parameter configures Oracle to automatically determine the size of the buffer cache (**db_cache_size**), shared pool (**shared_pool_size**), large pool (**large_pool_size**), java pool (**java_pool_size**), and streams pool (**streams_pool_size**).

  The value configured for SGA_TARGET sets the total size of the SGA components.

  When SGA_TARGET is set (that is, its value is not 0), and one of the above pools is also set to a non-zero value, the pool value is used as the minimum value for that pool.

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

# Oracle Tablespaces

An Oracle tablespace is an Oracle object that is a logical container of database objects, for example, tables, indexes, and so forth. When working with APM, you must create one or more dedicated default tablespaces for your APM user schemas. You may also want to create a dedicated temporary tablespace for APM. 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 APM, but not certified for use with APM, since it can cause the system to consume all available disk space.

This section includes:

- "Locally Managed Tablespaces" below
- "Creating an Oracle Tablespace" below

## 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 APM.

For information on locally managing temporary tablespace using `TEMPFILE`, see "Temporary Tablespace Settings" on page 56.

## Creating an Oracle Tablespace

You create a APM 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 **<APM 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 APM installation. For more information, see "Data Tablespace Settings" on the next page.

**To create an APM 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. Note that the actual size needed varies depending on the amount of data APM 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" below
- "Data Tablespace Default Storage Settings" below
- "System Tablespace Settings" on the next page
- "Temporary Tablespace Settings" on the next page
- "Redo Log Settings" on the next page
- "Undo Segment Settings" on page 57

## Data Tablespace Settings

The following table specifies the recommended sizes for APM tablespaces:

| Tablespace | APM Deployment | | Remarks |
| | Small | Large | |
| --- | --- | --- | --- |
| Management | 5 GB | 10 GB | The specified size is a minimum requirement. |
| Profile | 30 GB | 600 GB | The actual size is dependent on the EPS of the Profile database metrics and on the purging policy of the Profile database data tables as defined in the Partition and Purging Manager. |
| CMDB | 1 GB | 20 GB | The specified size is a minimum requirement. |

## Data Tablespace Default Storage Settings

The storage settings for data tablespaces should be:

- Locally managed tablespace
- Automatic segment space management

- Automatic local extent management

## System Tablespace Settings

The following table specifies the recommended settings for system tablespaces:

| Tablespace | APM Deployment | |
| --- | --- | --- |
| | Small | Large |
| SYSTEM | 2 GB | 5 GB |
| SYSAUX | 2 GB | 5 GB |

The System Tablespace storage default settings should be:

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

## Temporary Tablespace Settings

The following table specifies the recommended settings for temporary tablespaces:

| Tablespace | APM 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><li>Should be locally managed (Uniform allocation).</li><li>Tablespaces should be of a temporary type (use of TEMPFILE).</li><li>Segment space management in temporary tablespaces is manual.</li></ul> |

## Redo Log Settings

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

| Setting | APM Deployment | |
| --- | --- | --- |
| | Small | Large |
| Redo log file size | 100 MB | 200 MB - 500 MB |

| Setting | APM 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 | APM 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 | | |

The Undo Tablespace storage default settings should be:

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

# Using RAID Configuration

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

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

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

Note the following RAID configuration recommendations:

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

performance of distributed parity (RAID 5) is not as good as that of simple mirroring (RAID 1).

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

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

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

**Note:**

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

# ASM Storage

As of APM 9.00, Oracle 11g Automatic Storage Management (ASM) is supported as a storage option for Oracle database files, in addition to the traditional file system option.

For details on configuring and administrating ASM, see the Oracle Database Storage Administrator's Guide 11g.

For an ASM Overview, best practices, and technical papers, see http://www.oracle.com/technology/products/database/asm/index.html.

Pay attention to the following tasks:

- Preparing and mapping of the ASM disks, configuring the ASM disk groups, and setting the correct redundancy.
- Usage of the Oracle ASMLIB driver for improved discovery and management of ASM disks and for enhanced I/O processing (Linux only).
- Proper setting of ASM_DISKSTRING according to your disks discovery path.
- Rebalance operation manual and automatic control options.

# Oracle Advanced Security

> **Note:** To configure your Oracle database to use Oracle Advanced Security Encryption Algorithms, you may need to ask your database administrator for assistance.

The following Oracle Advanced Security Encryption Algorithms were successfully tested with APM:

- AES256 AES with a 256-bit key size
- RC4_256 RSA RC4 with a 256-bit key size
- AES192 AES with a 192-bit key size
- RC4_40 RSA RC4 with a 40-bit key size

To enable Network Data Encryption on the APM side:

1. In the **ddoracle** section of the **/opt/HP/BSM/conf/jdbc.drivers.properties** file, add the following lines:

   - ddoracle.EncryptionLevel = required

   - ddoracle.EncryptionTypes = <Encryption Algorithm>

2. Create a **/opt/HP/BSM/odb/conf/jdbc.properties** file with the following content:

   - Oracle = ddoracle

   - ddoracle.EncryptionLevel = required

   - ddoracle.EncryptionTypes = <Encryption Algorithm>

# Chapter 11: Maintaining an Oracle Server Database

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

This chapter includes:

- "Database Maintenance and Tuning" below
- "Oracle Database Backup and Recovery" on page 66

## 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 the next page
- "Database Load Behavior" on the next page
- "CPU and I/O" on the next page
- "Oracle Alert File" on the next page
- "Archive Log - File System" on the next page
- "Tablespace Storage Space" on page 62
- "Space Management" on page 62
- "Collecting Statistics for Databases" on page 62
- "Oracle 10g Automated Statistics Collection" on page 62
- "Oracle 11g Automated Statistics Collection" on page 63
- "Star Transformation in Profile Schemas" on page 63
- "Advanced Statistics Collection for Profile Schemas" on page 64
- "Collection Statistics for RTSM" on page 64
- "Index Fragmentation" on page 64

# System Global Area (SGA)

Always configure your SGA to fit physical memory and avoid using swapping. It is recommended that you not set the SGA for more than 70 percent of system physical memory, leaving enough memory for additional system and client processes. In Oracle 11g, the parameters `MEMORY_TARGET` or `MEMORY_MAX_TARGET` cannot be greater than the shared memory file system (**/dev/shm**) on your operating system.

# Database Load Behavior

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

It is also recommended that you monitor I/O load on the system to identify I/O contention. After you determine which disk is most loaded, you can use the AWR output to determine which particular Oracle data file is the cause of the contention and consider changing the I/O storage configuration.

# CPU and I/O

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

You can use **perfmon** on Windows, or **top** in UNIX and the storage system management tools, to monitor the above resources.

# Oracle Alert File

Oracle registers abnormal events in the **alert.log** file, whose location is defined by the `BACKGROUND_DUMP_DEST` parameter. As of Oracle 11g, the `BACKGROUND_DUMP_DEST` parameter is replaced by `DIAGNOSTIC_DEST` as part of Oracle Automatic Diagnostic Repository (ADR).

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

# Archive Log - File System

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

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

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

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

```
SQL> alter session set NLS_DATE_FORMAT = 'DD-MON-YYYY';
SQL> select FIRST_TIME as "Day",
```

```
        COUNT(*) as "Number of files"
        from V$LOG_HISTORY
        group by FIRST_TIME
        order by 1 asc;
```

# Tablespace Storage Space

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

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

# Space Management

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

Delete this text and replace it with your own content.

# Collecting Statistics for Databases

The APM 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 APM deployment, it is recommended that you collect statistics for all APM objects (tables and indexes).

When working with large APM 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 the tables and indexes created by the Partitioning and Purging Manager). After your APM system is stable, you should collect statistics once a day.

# Oracle 10g Automated Statistics Collection

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

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

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

# Oracle 11g Automated Statistics Collection

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

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

**To manually collect statistics for all APM objects on a database schema:**

1. Log on to the APM relevant schema (for example, profile schema) using SQL*Plus (a third party Oracle client).

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 on 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.
> - 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.

# Star Transformation in Profile Schemas

The APM Profile database was redesigned in version 9.00 in order to provide better performance in a complex reporting environment. This design is based on the star model, which is a classic implementation for data warehouse environments.

To provide an optimal support for this design, Oracle provides a feature named **star transformation**, which transforms ordinary queries into star queries. The main reporting system for APM, End User Management reports, is based upon this feature.

The following database parameters should be set at the system level to enable star queries:

- **star_transformation_enabled=true.** This parameter enables star transformation queries.
- **_always star_transformation=true.** This hidden parameter tells optimizer to always favor star transformation when applicable.

For details on these parameters, see "Oracle Parameter Settings" on page 51.

# Advanced Statistics Collection for Profile Schemas

The star model requires broader, fine-grained statistics for the profile schema, facilitating the optimal use of star transformation queries.

As of release 9.20, APM manages the database statistics for the profile schemas internally. When a new profile schema is created, database statistics are locked using Oracle **dbms_stats** APIs. Oracle default statistics collection jobs (such as the Oracle 11g automated statistics collection job) have no effect on profile schemas. Database administrators should refrain from collecting statistics for profile schemas using the FORCE parameter.

# Collection Statistics for RTSM

Unlike some databases, where queries are predefined and can be tuned according to the expected database size, RTSM database (CMDB schema) 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 RTSM, it is recommended that you manually refresh statistics if major changes to the RTSM schema objects have occurred, such as bulk insert transactions triggered by automated DFM jobs. DFM is the process responsible for automatically detecting configuration items (CIs) and inserting them into RTSM.

# Index Fragmentation

APM schemas consists of Oracle B-tree indexes for enhancing searches on table columns.

> **Note:** In APM profile schema, Oracle bitmap indexes are used in addition to B-tree indexes. The following sections are relevant to B-tree indexes.

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

## Index Maintenance Utility

APM'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 **\<Gateway Server root directory>\AppServer\webapps\site.war\DataBases\ORA_DB_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, 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.
- Bitmap indexes are used in the APM profile schema to support star queries. They enable better performance for reports when filtering dimension columns. They are used in some of the fact tables in the profile schema. Insert statements are performed against these tables on a regular basis. It is recommended to rebuild these bitmap indexes as part of your organization's index maintenance plan.

# 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" below
- "Oracle Recovery Manager - RMAN" on the next page

## Available Backup Methods

This section describes the various backup methods that are available.

### Cold Backup

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

After 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, see the Oracle Backup and Recovery Guide for your Oracle release.

## Hot Backup

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

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

After 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, APM may also experience some performance degradation due to disk load.

For more information, see the Oracle Backup and Recovery Guide for your Oracle release.

## Data Pump Export Import

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

The Oracle Data Pump export utility dumps schema structure and contents into one or more Oracle binary files called a dump file. This method can be used to transfer data between two schemas in the same database, or between two separate Oracle databases. To load exported data back into the database, use the Data Pump import utility. For more information about Oracle Data Pump utilities, see Oracle Database Utilities in the Oracle documentation for your release.

> **Note:** APM does not require you to use a specific backup method; however, it is recommended that your backup method accommodate APM'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 about RMAN, see the Oracle Recovery Manager User's Guide for your release.

# Chapter 12: Configuring the Oracle Client for APM

This chapter describes how to configure the Oracle Client for APM 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 APM databases.

## Oracle Client Versions and Operating System Platforms

The Oracle Client versions and operating system platforms that are supported and recommended for working with APM are identical to the Oracle Server versions and platforms. For details, see the Oracle Server requirements table in "Software Requirements" on page 45.

## Oracle Client Installation

To install the Oracle Client, see 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

To work with APM, 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, see the Oracle documentation.

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

If you are using a RAC deployment, you can configure your tnsnames entry to work with the RAC. For an example of a RAC entry, see "Support for Oracle Real Application Cluster" on page 96.

> **Note:**
>
> - The APM 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.
>
> - The Easy Connect Naming method eliminates the need to look up names in the tnsnames.ora files for TCP/IP environments. For details, see the Oracle Database Net Services Administrator's Guide in the Oracle documentation set of your release.

# Chapter 13: Oracle Summary Checklist

This chapter contains a checklist summarizing the requirements for APM support and certification.

For more information about the Oracle database configuration settings that should be used when working with Oracle Server and APM, see "Oracle Server Configuration and Sizing Guidelines" on page 51.

## Checklist for APM Support and Certification

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

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

| Option | Supported | Recommended | Remarks | For more information, see |
|---|---|---|---|---|
| Oracle edition | Enterprise | Enterprise | | |
| Dedicated APM server | Not necessary | Not necessary. It is recommended to dedicate an instance for APM | | |
| Use of multiple Oracle instances | Yes | No | The configuration of all instances must match in a certified environment. | "Oracle Instances" on page 46 |
| Use of non-default port | Yes | Yes | | |
| Undo management | Automatic; Manual | Automatic | Set UNDO_ MANAGEMENT parameter to AUTO in a certified environment | |
| Oracle Shared Servers connection method | Yes | No | APM uses a connection pool architecture. Use the dedicated server connection method in a certified environment. | |
| Oracle replication | No full support | No | | |

| Option | Supported | Recommended | Remarks | For more information, see |
|---|---|---|---|---|
| Operating System file compression | No | No | Not supported by Oracle; Causes abnormal behavior and affects performance. | |
| Database control files required | Greater than or equal to 2 | 3 | Preferable on different disks. | |
| Redo log groups | Greater than or equal to 3 | 4 | Oracle enables the software mirroring of redo log files. This is achieved by creating at least two members of redo log in each group. Members of the same group should reside on different disks. | |
| Character set | WE8ISO8859P1; UTF8,AL32UTF8 | AL32UTF8 | | |
| OPEN_CURSORS | 800 | 800 | | |
| Working in archive log mode | True; False | True | | |
| Oracle Partitioning Option | Enabled | Enabled | Used by the APM Partition Manager to partition and purge Profile database data | |
| Autoextend option in tablespace files | Yes | No | | |
| Locally managed data tablespace | Yes | Yes | | "Locally Managed Tablespaces" on page 54 |
| Tablespace extent management | Local automatic for application schemas; Local uniform for TEMP tablespace | Local automatic for application data tablespaces; Local uniform for TEMP tablespace | | "Oracle Tablespace Settings" on page 55 |

| Option | Supported | Recommended | Remarks | For more information, see |
|---|---|---|---|---|
| Automatic Segment Space Management Tablespace (ASSM) | Yes | Yes | | |

## Oracle Client Configuration

In order to work with APM, 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, see the Oracle documentation.

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

If you are using a RAC deployment, you can configure your tnsnames entry to work with the RAC. For an example of a RAC entry, see "Support for Oracle Real Application Cluster" on page 96.

> **Note:**
> - The APM 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.
> - The Easy Connect Naming method eliminates the need to look up names in the tnsnames.ora files for TCP/IP environments. For details, see the Oracle Database Net Services Administrator's Guide in the Oracle documentation set of your release.

## Oracle Server and Client Requirements

For the Oracle Server and Client versions and operating system platforms that are supported for working with APM, see "Software Requirements" on page 45.

### Setting Oracle Initialization Parameters

For a list of recommended settings for Oracle server initialization parameters, see "Oracle Parameter Settings" on page 51.

# Part IV: Appendixes

# Appendix A: Loader Management

This appendix explains how APM 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 APM is being reported to the database. HPE Software Support also uses loader information when helping to troubleshoot data problems.

This appendix includes:

## Overview of Loader Management

The APM Loader enables efficient and persistent reporting of data from data collectors to the Gateway Server and on to the database.

The APM Loader works in the following manner:

1. Data streams (through HTTP requests) from a data collector (for example, Business Process Monitor, Real User Monitor, 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.

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

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

7. 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" below.

**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 78.

## 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.

   After 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 is sent to the **Recovery Persistency** directory. When it arrives, 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 APM 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" below.

3. **Flatten Failure**

   The data that flows from Business Process Monitor and Webtrace is hierarchical 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.

1. 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 HPE Software Support, you can change the default logging level. For details, see "Changing the Default Logging Level" on the next page. For a definition of APM log levels, see "Log Severity Levels" in the APM Platform Administration Guide.

2. 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 HPE Software Support. For a description of the loader directories, see "Location of Persistency Directories" above.

3. 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 the APM Platform Administration Guide.

# 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.

> **Note:** 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" below.

The loader log files are located at:

**<Gateway Server root directory>\log\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 HPE 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:

   **<Gateway Server root directory>\conf\core\Tools\log4j\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:

   **<Gateway Server root directory>\conf\core\Tools\log4j\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:

## 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:

**<GatewayServer Root Directory>\log\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.

## Loader Statistics

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

**<GatewayServer Root Directory>\log\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 APM, such as SiteScope 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:

**http://<Gateway Server Name>:8080/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

# Appendix B: Microsoft SQL Server Data Storage Recommendation

This appendix contains recommendations for creating Microsoft SQL Server file groups in which to store APM data. This information applies to advanced users only.

## Creating Microsoft SQL Server File Groups

When APM databases are created automatically by the Server and Database Configuration utility, APM stores database objects in two file groups: the **PRIMARY** file group for system tables and the **userdata001** file group for user tables. If you are setting up a large deployment of APM, it is recommended that you create additional file groups in which to store APM 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 APM 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 APM 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).

# Appendix C: Microsoft SQL Server Configuration Options

Most server configuration options are dynamically configured by Microsoft SQL Server. For APM certification, you should not change the default options unless you are instructed to do so by HPE 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).

The default values should be used except for the following items:

- Agent XPs (A) = 1
- awe enabled (A) = 1. This should only be set when more than 4GB is required on a 32Bit machine. Please note that this feature is deprecated in Microsoft SQL Server 2012.

| Configuration Option | Default Value for SQL Server 2012 | Default Value for SQL Server 2008 and 2008 R2 | APM Certification |
|---|---|---|---|
| access check cache bucket count (A) | 0 | 0 | Default |
| access check cache quota(A) | 0 | 0 | Default |
| ad hoc distributed queries(A) | 0 | 0 | Default |
| affinity I/O mask (A, RR) | 0 | 0 | Default |
| affinity64 I/O mask (A, only available on 64-bit version of SQL Server) | 0 | 0 | Default |
| affinity mask (A) | 0 | 0 | Default |
| affinity64 mask (A, RR), only available on 64-bit version of SQL Server | 0 | 0 | Default |
| Agent XPs (A) | 0 (Changes to 1 when SQL Server Agent is started. Default value is 0 if SQL Server Agent is set to automatic start during Setup) | 0 (Changes to 1 when SQL Server Agent is started. Default value is 0 if SQL Server Agent is set to automatic start during Setup) | 1 |
| allow updates (Obsolete. Do not use. Will cause an error during reconfigure.) | 0 | 0 | Default |

| Configuration Option | Default Value for SQL Server 2012 | Default Value for SQL Server 2008 and 2008 R2 | APM Certification |
|---|---|---|---|
| awe enabled (A, RR) | | 0 | Default, Unless more than 4GB memory is required on a 32Bit machine. This feature was deprecated in Microsoft SQL Server 2012. |
| backup compression default | 0 | 0 | Default |
| blocked process threshold(A) | 0 | 0 | Default |
| c2 audit mode (A, RR) | 0 | 0 | Default |
| clr enabled | 0 | 0 | Default |
| common criteria compliance enabled (A, RR) | 0 | 0 | Default |
| contained database authentication | 0 | N/A | Default |
| cost threshold for parallelism (A) | 5 | 5 | Default |
| cross db ownership chaining | 0 | 0 | Default |
| cursor threshold (A) | -1 | -1 | Default |
| Database Mail XPs (A) | 0 | 0 | Default |
| default full-text language(A) | 1033 | 1033 | Default |
| default language | 0 | 0 | Default |
| default trace enabled (A) | 1 | 1 | Default |
| disallow results from triggers (A) | 0 | 0 | Default |
| EKM provider enabled | 0 | 0 | Default |
| filestream_access_level | 0 | 0 | Default |
| fill factor (A, RR) | 0 | 0 | Default |
| ft crawl bandwidth (max), see ft crawl bandwidth(A) | 100 | 100 | Default |
| ft crawl bandwidth (min), see ft crawl bandwidth(A) | 0 | 0 | Default |
| ft notify bandwidth (max), see ft notify bandwidth(A) | 100 | 100 | Default |

| Configuration Option | Default Value for SQL Server 2012 | Default Value for SQL Server 2008 and 2008 R2 | APM Certification |
|---|---|---|---|
| ft notify bandwidth (min), see ft notify bandwidth(A) | 0 | 0 | Default |
| index create memory (A, SC) | 0 | 0 | Default |
| in-doubt xact resolution (A) | 0 | 0 | Default |
| lightweight pooling (A, RR) | 0 | 0 | Default |
| locks (A, RR, SC) | 0 | 0 | Default |
| max degree of parallelism(A) | 0 | 0 | Default |
| max full-text crawl range(A) | 4 | 4 | Default |
| max server memory (A, SC) | 2147483647 | 2147483647 | Default |
| max text repl size | 65536 | 65536 | Default |
| max worker threads (A) | 0<br><br>Zero auto-configures the number of max worker threads depending on the number of processors, using the formula (256+ (<processors>-4) * 8) for 32-bit SQL Server and twice that for 64-bit SQL Server. | 0<br><br>Zero auto-configures the number of max worker threads depending on the number of processors, using the formula (256+ (<processors>-4) * 8) for 32-bit SQL Server and twice that for 64-bit SQL Server. | Default |
| media retention (A, RR) | 0 | 0 | Default |
| min memory per query (A) | 1024 | 1024 | Default |
| min server memory (A, SC) | 0 | 0 | Default |
| nested triggers | 1 | 1 | Default |
| network packet size (A) | 4096 | 4096 | Default |
| Ole Automation Procedures (A) | 0 | 0 | Default |
| open objects (A, RR, obsolete) | 0 | 0 | Default |

| Configuration Option | Default Value for SQL Server 2012 | Default Value for SQL Server 2008 and 2008 R2 | APM Certification |
|---|---|---|---|
| optimize for ad hoc workloads (A) | 0 | 0 | Default |
| PH_timeout (A) | 60 | 60 | Default |
| precompute rank (A) | 0 | 0 | Default |
| priority boost (A, RR) | 0 | 0 | Default |
| query governor cost limit(A) | 0 | 0 | Default |
| query wait (A) | -1 | -1 | Default |
| recovery interval (A, SC) | 0 | 0 | Default |
| remote access (RR) | 1 | 1 | Default |
| remote admin connections | 0 | 0 | Default |
| remote login timeout | 10 | 20 | Default |
| remote proc trans | 0 | 0 | Default |
| remote query timeout | 600 | 600 | Default |
| Replication XPs Option (A) | 0 | 0 | Default |
| scan for startup procs (A, RR) | 0 | 0 | Default |
| server trigger recursion | 1 | 1 | Default |
| set working set size (A, RR, obsolete) | 0 | 0 | Default |
| show advanced options | 0 | 0 | Default |
| SMO and DMO XPs (A) | 1 | 1 | Default |
| SQL Mail XPs (A) | N/A | 0 | Default |
| transform noise words (A) | 0 | 0 | Default |
| two digit year cutoff (A) | 2049 | 2049 | Default |
| user connections (A, RR, SC) | 0 | 0 | Default |
| User Instance Timeout (A, only appears in SQL Server 2008 Express.) | N/A | 60 | Default |
| user instances enabled (A, only appears in SQL Server 2008 Express.) | N/A | 0 | Default |

| Configuration Option | Default Value for SQL Server 2012 | Default Value for SQL Server 2008 and 2008 R2 | APM Certification |
|---|---|---|---|
| user options | 0 | 0 | Default |
| xp_cmdshell (A) | 0 | 0 | Default |

# Appendix D: Changing the Database Host

This section describes the procedure to migrate your APM database schema to a different server.

1. Stop all APM servers. This is required to prevent data loss and schema structure mismatch in the target environment.

2. Duplicate the APM schemas you wish to migrate.

   This can be performed by an offline database backup or a database level export. For details, see one of the following two sections depending on your database type:

   Microsoft SQL: "Backing Up Databases" on page 33.

   Oracle: "Oracle Database Backup and Recovery" on page 66.

3. Modify sessions table in your management database according to your new configuration.

   For details, see the **APM Installation Guide > Disaster Recovery for APM > Cleanup Procedure** and locate the step about switching references in the session table on the management database to the backup databases.

4. Run the Server and Database Configuration utility.

   Run the Server and Database Configuration utility, specifying the location of the new database host. For details, see the **APM Installation Guide > Server Deployment and Setting Database Parameters**.

5. Start all APM servers.

# Appendix E: Database Schema Verification

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

This appendix includes:

## About the Verification Process

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

During the course of the verification process, you may need to specify the user name and password of an administrative user, 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 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 95.

> **Note:** APM 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 APM schemas as the login. By assigning an administrator schema to perform datapump operations, you do not have to grant additional permissions to APM 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.hpe.com**), you may be prompted for administrative credentials separately for each host name. This may cause a degradation in the verification tool's performance. It is therefore recommended to use the same notation every time you reference the same server.

## Running the Verification Process

You run the database schema verify program from an APM 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 **<APM root directory>\dbverify\bin** directory and run the **run_db_verify.bat** file.

   - If you are running the verify program from a Linux platform, make sure that the **$DISPLAY** environment variable is set. Open an X-terminal window, move to the **<APM 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.
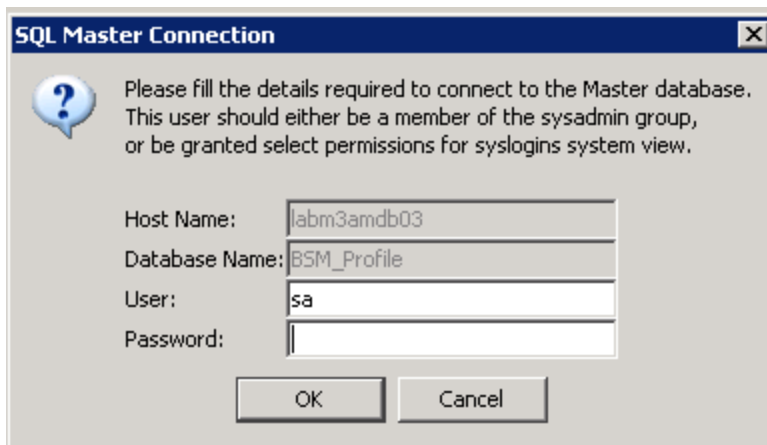
     

     In the **User** and **Password** boxes, type the user name and password of a user with permissions for the database. (The User box displays the default Microsoft SQL Server administrator user name, **sa**; by default, there is no password.) Click **OK**.

   - In the Oracle Dictionary Connection dialog box, specify the details required to connect to the Oracle database.
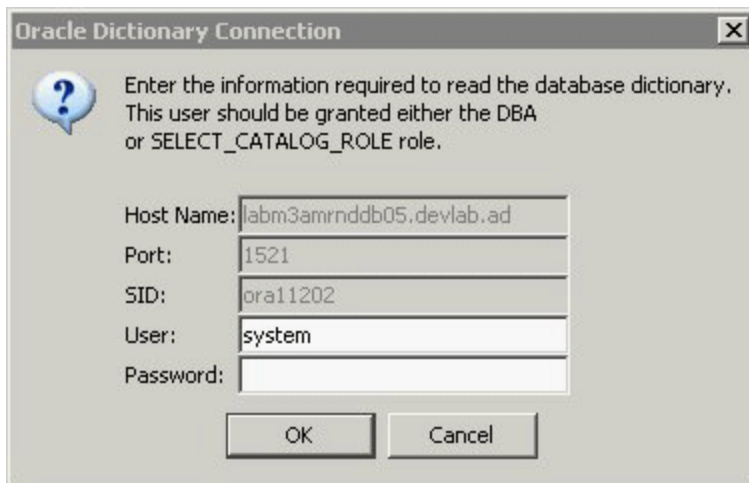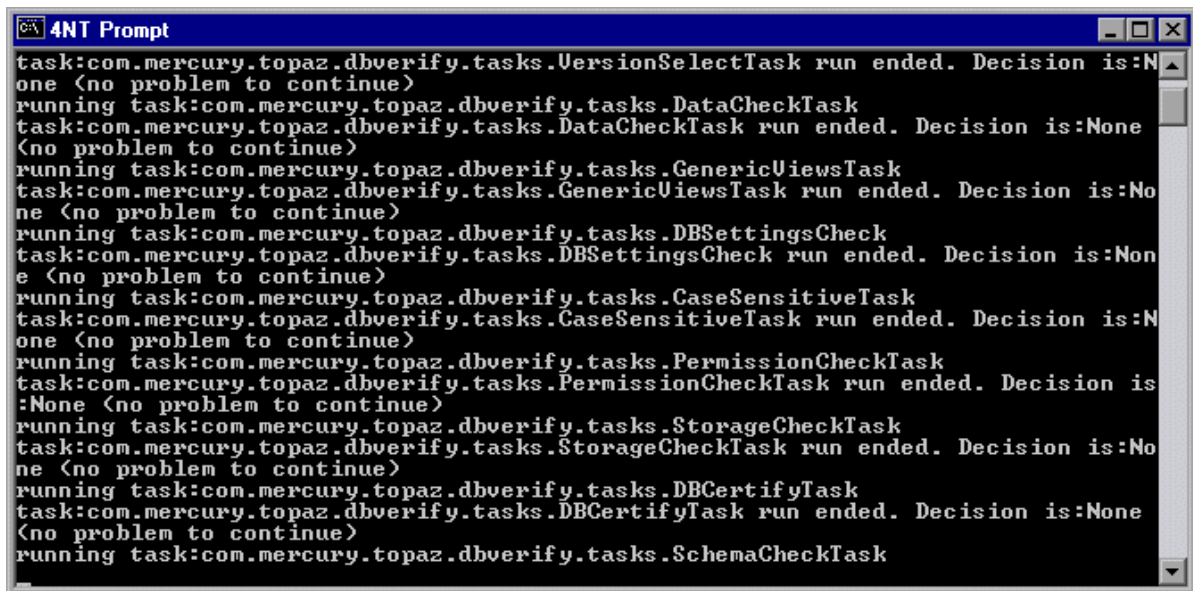
In the **User** and **Password** boxes, type the user name and password of a user with permissions for the database, and click **OK**.

> **Note:**
>
> - You are prompted for connection data for each different server on which your APM databases reside.
>
> - In order to connect using Windows authentication, leave the user and password fields blank. The login user must be defined explicitly in Microsoft SQL Server, not via a group.

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



4. If problems occur during the database verification, a dialog box is displayed listing the errors.
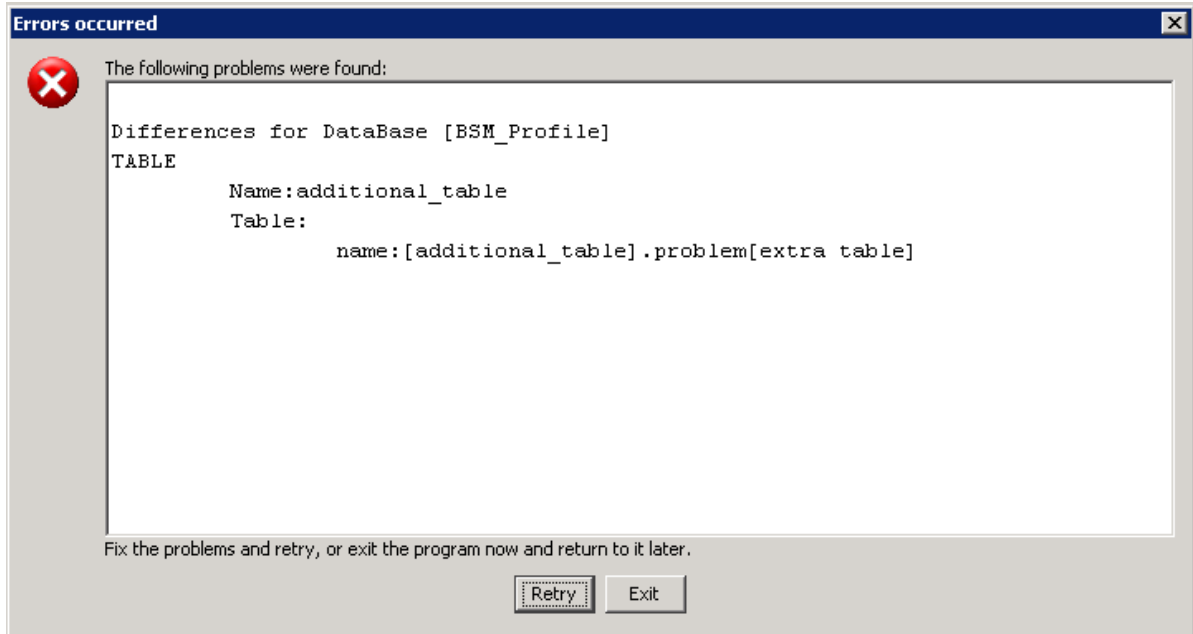
Either fix the problems found and click **Retry**, or click **Exit** and rerun the database schema verify program at a later date. A series of scripts, one for each problematic database, are created under **<APM root directory>\dbverify\tmp**. If you are unable to fix the problems, contact HPE Software Support for assistance.

You can view a log file of the errors located in the **<APM root directory>\dbverify\log** directory.

5. If the verification process determines that fixing the database may include some lengthy operations, a dialog box is displayed.
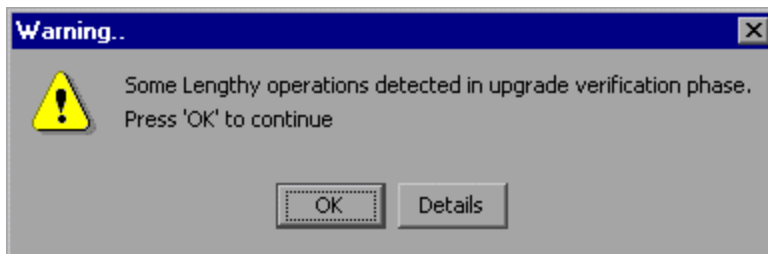


Click **OK** to continue, or **Details** to display the estimated times that the lengthy operations detected could take during an upgrade.

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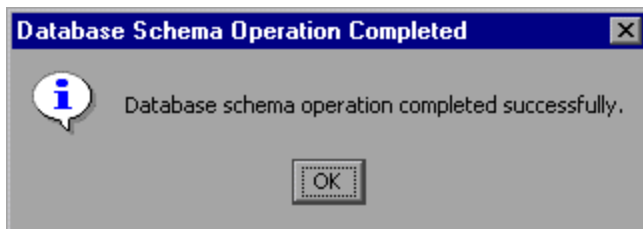


Click **OK**.

**Note:** For details on database verification troubleshooting issues, see the HPE Software Self-solve knowledge base (http://softwaresupport.hpe.com). To enter the knowledge base, you must log on with your HPE 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 with certain administrative (read-only) permissions to the database. You can create users with minimum privileges by running one of the appropriate scripts from **<HPE APM root directory>\dbverify\pre\create_admin_user_mssql.sql** for Microsoft SQL Server (this must be run as an sa user) or **<HPE APM root directory>\dbverify\pre\create_admin_user_oracle.sql** for Oracle (this must be run as a system user).

# Appendix F: Support for Oracle Real Application Cluster

This appendix contains the configuration that need to be done for HPE Application Performance Management to work with Oracle Real Application Cluster. This information applies to advanced users only.

This appendix includes:

- "About Oracle Real Application Cluster (RAC)" below
- "Single Client Access Name" on the next page
- "Client Side Configuration for Oracle RAC" on the next page
- "Server Side Configuration" on page 100
- "Create/Connect to One of the APM Databases" on page 101

## About Oracle Real Application Cluster (RAC)

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

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

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

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

- Oracle RAC database name: BSMRAC
- Machine names: Server1, Server2
- On each machine there is an Oracle instance of BSMRAC:
    - SID on Server1: BSMRAC1
    - SID on Server2: BSMRAC2
- 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 BSMRAC.
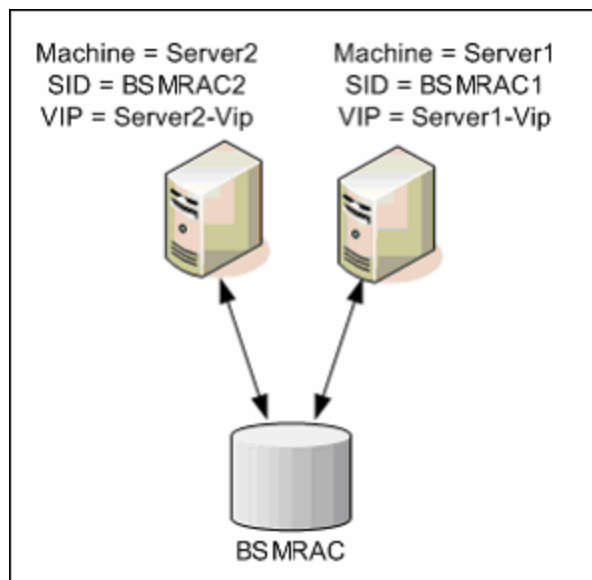
## Single Client Access Name

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

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

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

In APM 9.x when using Oracle 11g RAC, it is recommended to use the SCAN method.

## Client Side Configuration for Oracle RAC

APM uses DataDirect's JDBC driver to connect to regular Oracle databases and to Oracle RAC databases.

> **Note:** For information on how to edit the profile database to point to the new Oracle RAC database, see *Switch references in the Sessions table on the management database of the DR environment to the backup databases* in the APM Installation Guide.

Make the following changes in APM'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 < APM root directory>\conf\bsm-tnsnames.ora.**

   The format of **bsm-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. In the case of Oracle RAC 11g R2 and later, it contains only the SCAN. The SCAN contains all the details needed for connecting to the node:

  ○ **HOST** contains the SCAN value. If SCAN is not defined, then HOST contains the node virtual IP address. It is important to use the virtual IP address and not the static IP address 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, **bsm-tnsnames.ora** would be configured as:

```
BSMRAC =
(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 = BSMRAC)
 )
 )
```

**Note:** If APM schemas are deployed in more than one Oracle database or service, bsm-tnsnames.ora should contain one entry for each database or database service. When all schemas are under the same RAC service, only one entry is required.

2. **On all servers, edit the file <APM 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=<APM root directory>\\conf\\bsm-tnsnames.ora;TNSServerName=${sid}
      ```

      where **${sid}** correlates to the service name entered in the Service and Database Configuration step. This parameter is saved per APM schema; therefore, if there is more than one service name in the system, make sure that **bsm-tnsnames.ora** includes an entry for each service name.

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

      ○ **UNIX.**

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

3. **On all servers, open the directory <APM root directory>\odb\conf. Locate the jdbc.properties file.**

   a. Find the line starting with **cmdb.url**.

   b. Replace this with the following line:

   ```
   cmdb.url = jdbc:mercury:oracle:TNSNamesFile=<APM_HOME>\\conf\\bsm-tnsnames.ora;TNSServerName=<SERVICE NAME>
   ```

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

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

   ```
   Oracle = ddoracle
   cmdb.url =
   ```

```
jdbc:mercury:oracle:TNSNamesFile=<APM_HOME>\\conf\\bsm-
tnsnames.ora;TNSServerName=<SERVICE NAME>
```

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

d. If your server is running UNIX/LINUX OS, replace all the double back slashes with single slashes.

4. Switch references in the Sessions table on the management database of the DR environment to the backup databases.

a. Run the following query to retrieve all database names:

**SELECT * FROM SESSIONS**

**where SESSION_NAME like '%Unassigned%'**

b. Update the following columns in each received row with the following values:

○ **SESSION_NAME:** Replace with the new restored database name (only where SESSION_ NAME is like '%Unassigned%'). Use the following script:

UPDATE SESSIONS set SESSION_NAME='Unassigned<NEW_DB_Server_name><NEW_ schema_name><DB_User_name>'

WHERE SESSION_NAME='Unassigned<OLD_DB_Server_name><OLD_schema_ name><old_DB_User_name>'

○ **SESSION_DB_NAME:** Replace with the new restored schema name. Use the following script:

UPDATE SESSIONS set SESSION_DB_NAME='<<NEW_schema_name>'

WHERE SESSION_DB_NAME='<OLD_schema_name>'

○ **SESSION_DB_HOST:** Replace with the new restored database host name. Use the following script:

UPDATE SESSIONS set SESSION_DB_HOST='<<NEW_host_name>'

WHERE SESSION_DB_HOST='<OLD_host_name>'

○ **SESSION_DB_PORT:** Replace with the new restored port name. Use the following script:

UPDATE SESSIONS set SESSION_DB_PORT='<NEW_port_name>'

WHERE SESSION_DB_PORT='<OLD_port_name>'

○ **SESSION_DB_SID:** Replace with the new restored session ID name. Use the following script:

UPDATE SESSIONS set SESSION_DB_SID='<<<NEW_SID_name>>>'

WHERE SESSION_DB_SID='<<<OLD_SID_name>>>'

○ **SESSION_DB_UID:** Replace with the new restored name. Use the following script:

UPDATE SESSIONS set SESSION_DB_UID='<NEW_UID_name>'

WHERE SESSION_DB_UID='<OLD_UID_name>'

○ **SESSION_DB_SERVER:** Replace with the new restored server name. Use the following script:

UPDATE SESSIONS set SESSION_DB_SERVER='<NEW_server_name>'

WHERE SESSION_DB_SERVER='<OLD_server_name>'

# 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 after it is connected.

## Recommendation for APM Database

It is recommended to use the session based algorithm since APM 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 APM 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. In Oracle 11g, the SCAN virtual IP can be used. |
| Port | The local listener port on the Oracle machine or the port of the SCAN listener. |
| SID | The service name of the database.<br>**Note:** This must be the same as `<SERVICE_NAME>` entered in **bsm-tnsnames.ora**. |

In the example, the parameters would be:

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

# Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on Database Guide (Application Performance Management 9.40)**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to docteam@hpe.com.

We appreciate your feedback!