

HP Operations Orchestration

Software Version: 10.20

Windows and Linux Operating Systems

Database Guide

Document Release Date: December 2014 (Rev B)
Software Release Date: November 2014



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notice

© Copyright 2005-2014 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

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

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Acknowledgements

Documentation Updates

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

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

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

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

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

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

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

You will also receive updated or new editions if you subscribe to the appropriate product support service.

Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at: **<http://www.hp.com/go/hpssoftwaresupport>**

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

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

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

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

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

To find more information about access levels, go to:

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

HP Software Solutions Now accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is

<http://h20230.www2.hp.com/sc/solutions/index.jsp>

Contents

Introduction to Preparing the Database Environment	6
Overview	6
Using Database Clusters	7
HP OO Database Sizing	8
Hardware Requirements	9
Deploying and Maintaining a Microsoft SQL Server Database	10
Workflow for Microsoft SQL Server Deployment	10
System Requirements for Microsoft SQL Server	11
Hardware Requirements	11
Software Requirements	11
Examples of Tested Deployments	12
Language Support	12
Configuring SQL Server	13
Manually Creating an HP OO Database on Microsoft SQL Server	14
Manually Creating Database Objects	17
Microsoft SQL Server Database Maintenance	18
Backing up the HP OO Database	18
Creating a Maintenance Plan	19
Always On Support	22
Deploying and Maintaining an Oracle Database	23
Workflow for Oracle Deployment	23
System Requirements for Oracle	24
Hardware Requirements	24
Software Requirements	24
Examples of Tested Deployments	24
Language Support	25
Configuring an Oracle Database	25
Manually Creating an HP OO Database on an Oracle Instance	27
Manually Creating Database Objects	28
Connecting to Oracle Using SID or Service Name	29
Oracle Database Maintenance	29
Backing up the HP OO Database	29
Creating a Maintenance Plan	30
Deploying and Maintaining a MySQL Database	33
Workflow for MySQL Deployment	33
System Requirements for MySQL	33

Hardware Requirements	34
Software Requirements	34
Examples of Tested Deployments	35
Language Support	35
Configuring MySQL	35
Manually Creating an HP OO Database on MySQL	37
Manually Creating Database Objects	38
MySQL Database Maintenance	39
Backing up the HP OO Database	39
Creating a Maintenance Plan	39
Utility for Purging Historical Data	40
Deploying and Maintaining a Postgres Database	42
Workflow for Postgres Deployment	42
System Requirements for Postgres	42
Hardware Requirements	43
Software Requirements	43
Examples of Tested Deployments	43
Language Support	44
Configuring Postgres	44
Manually Creating an HP OO Database on Postgres	45
Manually Creating Database Objects	46
Postgres Database Maintenance	46
Backing up the HP OO Database	47
Creating a Maintenance Plan	47
Utility for Purging Historical Data	48
Database Purging Flows	50
Appendix A: Additional Guidelines for Microsoft SQL Server	51
Using Windows Authentication to Access Microsoft SQL Server Databases	51
Configuring HP OO to Work with Windows Authentication	51
Appendix B: Additional Guidelines for Oracle	53
Oracle Real Application Cluster (RAC)	53
Single Client Access Name	54
Configuring HP OO to Work with Oracle RAC	54

Introduction to Preparing the Database Environment

This chapter contains information about the types of databases used with HP Operations Orchestration (HP OO).

This chapter includes:

- [Overview](#) 6
- [Using Database Clusters](#) 7
- [HP OO Database Sizing](#) 8
- [Hardware Requirements](#) 9

Overview

The term “database” may be interpreted in several ways, depending on the database vendor/technology used. In Oracle, the term “database” relates to a collection of files containing data and metadata. A single Oracle database may contain one or more schemas (and users). A Microsoft SQL Server “database” is closer in definition to Oracle’s “schema” than to Oracle’s “database”.

In order to avoid confusion, this document will use the following terms:

- Instance/server – the software and memory structures providing RDBMS services
- Database – the entity containing tables, views, indexes, and so on

HP OO requires a single database to be created. This database may co-exist with other databases contained in a database server.

You can set up an HP OO database on one of the following database server types:

- Microsoft SQL Server Standard/Enterprise (2008 R2/2012)
- Oracle 11gR2 Standard/Enterprise Server, including RAC environment
- Oracle 12cR1 Standard/Enterprise Server - regular instance (non-CDB)
- Postgres (9.1/9.2/9.3)
- MySQL Community/Standard/Enterprise Server (5.5/5.6)

See the appropriate deployment chapter below for details:

- ["Deploying and Maintaining a Microsoft SQL Server Database" on page 10](#)
- ["Deploying and Maintaining an Oracle Database" on page 23](#)
- ["Deploying and Maintaining a MySQL Database" on page 33](#)
- ["Deploying and Maintaining a Postgres Database" on page 42](#)

Appendices contain additional information that is pertinent to all database types.

Language Support

HP OO can be installed and used in any supported language (English, French, German, Spanish, Japanese, and Simplified Chinese). Databases and database servers should be properly configured in order to support the desired language.

If you are using HP OO in a multilingual environment, it is preferable that you configure your database to use the Unicode character set. See the relevant deployment chapter for detailed instructions.

If you have user input in two languages apart from English (for example, German and Chinese) then MS SQL should not be used. You should use an alternative database such as Oracle, MySQL, or Postgres with the recommended Unicode configuration for HP OO.

Important Notes

- This document is intended for trained database administrators. If you are not familiar with the type of database you wish to use, or you feel that you lack the relevant knowledge required in order to create and configure an HP OO database, see the database vendor's documentation and make sure you fully understand each action you perform as described in this guide.
- HP OO database connectivity relies on Java JDBC. If your environment requires specific adaptations or security measures, see the JDBC documentation (or database vendor documentation) to find out exactly how the JDBC connection URL should be formatted.
- This document describes the required database settings for HP OO. Settings that are not specified in this document can be left with default values or configured by your organization's DBA.

Using Database Clusters

Database clusters help to make the HP OO system more robust, by protecting the database from several types of failures.

HP OO does not provide database cluster-related features of its own, such as database connection failover. It relies on the functionality of the JDBC package being used, and the database environment capabilities.

You can install HP OO in conjunction with any type of database cluster environment that satisfies the following conditions:

- Connection pooling is supported
- A single, valid database URL can be provided
- It provides reliable transaction handling during failover (a single, complete transaction must either completely fail or succeed)

The HP OO installer does not provide database cluster-specific installation options. During installation, you need to provide the installer a simplistic / "regular" form of database connection. Database URL adaptations usually have to be made after the system is already installed in the **database.properties** configuration file.

See "[Appendix B: Additional Guidelines for Oracle](#)" on page 53 for an Oracle RAC example of a database URL adaptation.

HP OO Database Sizing

HP OO 10.20 keeps all flow and step execution data in the database by default, enabling comprehensive debugging of previous flow runs. As a result, the database size grows in accordance to the system throughput and flow complexity.

This section will help you prepare for installing HP OO. By estimating your system scale (small/standard/enterprise), you will be able to derive the amount of disk space required, derive the amount of memory (RAM) used by the database, and determine additional database installation parameters.

Step 1: Estimate the system scale according to complexity:

System Criteria\Scale	Small	Standard	Enterprise
Average number of steps per flow	50 or less	100 or more	1000 or more
Average flow duration	Seconds to minutes	1 hour or more	Up to 24 hours
Average payload size per flow ¹	1 KB or less	1 MB or more	4 MB or more

[1] – Ignore if you are not sure about this. The term "payload" refers to flow/step data of considerable size. For example, a CSV or XML file used as flow input.

Step 2: Estimate the system scale according to concurrency/frequency:

System Criteria\Scale	Small	Standard	Enterprise
Average number of flows per day	Less than 100	300 or more	1000 or more

The following table provides disk space and memory requirements for different HP OO deployment scales:

System Scale\Parameter	HP OO Database Disk Size Requirements	Memory
Small	50 GB	4 GB
Standard	500 GB	8 GB
Enterprise	Between 500 GB and 2 TB	12 GB

Notes:

- Disk space and memory values are estimates. Actual disk space and memory consumption vary, depending on the database vendor and database server configuration.

- Memory (RAM) reflects recommended database memory, not the overall amount of memory available on the database machine (virtual or physical server).
- Disk space reflects the amount of disk space required for day-to-day operation of the HP OO system and reasonable historical data retention—not including long-term database backups.

It is highly recommended to regularly purge historical data using the supplied stored procedure and database purge flows, in order to keep the database size within your environment limitations.

- The amount of additional disk space required for keeping HP OO database backups depends on the backup policy (frequency and retention period).

Hardware Requirements

The following table describes the hardware (CPU and memory) requirements recommended for the each of the database servers.

Note: The memory values reflect database memory consumption as part of the entire machine’s memory.

Database\Scale	Small/Standard				Enterprise			
	CPUs		RAM		CPUs		RAM	
	Small	Standard	Min	Rec	Min	Rec	Min	Rec
SQL Server	2	4	4 GB	8 GB	4	12	8 GB	12 GB
Oracle	2	4	4 GB	8 GB	4	12	8 GB	12 GB
MySQL	2	4	4 GB	8 GB	4	12	8 GB	12 GB
Postgres	2	4	4 GB	8 GB	4	12	8 GB	12 GB

Min = Minimal value, Rec = Recommended value

In addition to the generalized hardware requirements above, refer to the appropriate hardware requirements and software requirements sections per database.

Deploying and Maintaining a Microsoft SQL Server Database

In order to deploy HP OO using Microsoft SQL Server, you must have an existing SQL Server database service. If you need to create a new database service, see the relevant documentation provided by Microsoft, because this information is not included within this guide. However, this guide contains recommendations for the SQL Server configuration.

It is also recommended to use the SQL Server Agent service in order to comfortably schedule data purging and index maintenance jobs.

This chapter includes the following sections:

- [Workflow for Microsoft SQL Server Deployment](#) 10
- [System Requirements for Microsoft SQL Server](#) 11
 - [Hardware Requirements](#) 11
 - [Software Requirements](#) 11
 - [Examples of Tested Deployments](#) 12
 - [Language Support](#) 12
- [Configuring SQL Server](#) 13
- [Manually Creating an HP OO Database on Microsoft SQL Server](#) 14
 - [Manually Creating Database Objects](#) 17
- [Microsoft SQL Server Database Maintenance](#) 18
 - [Backing up the HP OO Database](#) 18
 - [Creating a Maintenance Plan](#) 19
- [Always On Support](#) 22

Workflow for Microsoft SQL Server Deployment

To deploy HP OO using Microsoft SQL Server, perform the following steps:

1. **Review sizing guidelines.** For details, see "HP OO Database Sizing" in "[Introduction to Preparing the Database Environment](#)" on page 6.
2. **Review Hardware and Software Requirements.** For details, see "[System Requirements for Microsoft SQL Server](#)" on the next page.
3. **Configure Microsoft SQL Server.** For details, see "[Configuring SQL Server](#)" on page 13.

4. **Create HP OO database on Microsoft SQL Server.** For details, see ["Manually Creating an HP OO Database on Microsoft SQL Server" on page 14.](#)
5. (Optional) **Set up Windows authentication.** For details, see "Using Windows Authentication to Access Microsoft SQL Server Databases" in ["Appendix A: Additional Guidelines for Microsoft SQL Server" on page 51.](#) This step is only relevant if you are using Windows authentication instead of SQL Server authentication.

System Requirements for Microsoft SQL Server

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

Hardware Requirements

For HP OO database sizing guidelines and hardware requirements, see "HP OO Database Sizing" and "Hardware Requirements" in ["Introduction to Preparing the Database Environment" on page 6.](#)

For Microsoft SQL Server hardware requirements, see the relevant installation guide for your Microsoft SQL Server release and operating system.

Software Requirements

The following table lists the Microsoft SQL Server releases supported by HP OO:

Microsoft SQL Server Database Releases			
Version	Type	32/64Bit	Service Pack
2012	Standard	64 Bit	2
	Enterprise	64 Bit	2
2008 R2	Standard	x64	2
		x86	2
	Enterprise	x64	2
		x86	2

Only the listed service packs should be installed. Newer service packs are also supported unless stated otherwise in the *HP OO Release Notes*.

See the Microsoft documentation for supported platforms.

Examples of Tested Deployments

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

Database Release			Operating System
Version	32/64Bit	Service Pack	
Microsoft SQL Server 2012 Enterprise Edition	64 Bit	2	Windows 2012 Standard Edition 64 Bit
Microsoft SQL Server 2008 R2 Enterprise Edition	64 Bit	2	Windows 2012 Standard Edition 64 Bit

Language Support

In Microsoft SQL Server, unlike other databases, HP OO database does not use Unicode collation.

Use one of the following collations depending on your HP OO installation language:

Language	Database Collation
English	SQL_Latin1_General_CP1_CS_AS
Japanese	Japanese_Unicode_CS_AS
Simplified Chinese	Chinese_Simplified_Stroke_Order_100_CS_AS
German	SQL_Latin1_General_CP1_CS_AS
French	French_100_CS_AS
Spanish	SQL_Latin1_General_CP1_CS_AS

Note: When an MS SQL database is created with a case-sensitive collation, object names such as tables, keys, and so on become case-sensitive as well.

For example, for the OO_STEP_LOG_BINDINGS table, a command such as `SELECT * FROM oo_step_log_bindings` is seen as using an invalid object name.

If you are currently using a different collation, it is highly recommended that you change your HP OO database collation to one of the collations above, in order to support future HP OO versions.

The following procedure is an example of how to change an existing database collation:

1. Connect to your database server using an administrative login (for example, "sa").
2. Disconnect all existing sessions to the specific HP OO database that you intend to change.

Important! The command will fail unless this database has exactly 0 sessions/connections.

3. Run the following code (change `my_database` to the actual name of the database):

```
USE [master]  
GO  
ALTER DATABASE [my_database] COLLATE Japanese_Unicode_CS_AS  
GO
```

Note: This change does not alter existing column collations. Any new column or table will use the new collation by default from this point on. The new collation sorting rules are applied immediately. In other words, the new collation affects sorting behavior and future data, rather than existing data.

If you have user input in two languages apart from English (for example, German and Chinese) then MS SQL should not be used. You should use an alternative database such as Oracle, MySQL, or Postgres with Unicode configuration.

Configuring SQL Server

This section contains information on Microsoft SQL Server and database configuration settings.

You can install an HP OO database in any SQL Server environment including clustered environments.

Legend:

- **Mandatory** configuration options/values appear in **bold/orange** font.
- **Recommended** configuration options/values appear in **bold/purple** font.
- Supported configuration options/values appear in normal font, and may show as a comma-separated list.
- *Comments* appear in *italic* font.

Microsoft SQL Server 2008R2 and 2012	
Server Options/Features	
Configuration Item	Supported Configuration Options
Server Configuration Options	Defaults, unless instructed otherwise
Instances	Default, Single
Authentication Mode	Mixed, Windows ¹
Full-Text Search	(Not required for HP OO)

[1] At the moment, the HP OO 10.10 installer supports only SQL authentication. Windows authentication can be configured afterward. See "[Appendix A: Additional Guidelines for Microsoft SQL Server](#)" on page 51.

Microsoft SQL Server 2008R2 and 2012			
Instance/Server Options			
	Mandatory	Recommended	Supported
Server Collation		SQL_Latin1_General_CP1_CS_AS	Any Collation
Network Libraries	Server: TCP/IP Client: TCP/IP		
Concurrent Connections	>= 800	0 (unlimited)	
Max Server Memory	> 4GB	2,147,483,647 (default, unlimited) <i>Allocate 4-12GB depending on system scale according to the sizing guide</i>	

Microsoft SQL Server 2008R2 and 2012			
Database Options			
	Mandatory	Recommended	Supported
Collation	Any collation listed in the "Language Support" section in "System Requirements for Microsoft SQL Server" on page 11.		
Recovery Model		Full	Simple, Full
Allow Snapshot Isolation	True		
Is Read Committed Snapshot On	True		
Auto Shrink	False		
Auto Create Statistics	True		

Manually Creating an HP OO Database on Microsoft SQL Server

During HP OO setup, a new database can be created automatically by the HP OO installer or an existing database can be used.

If during installation, you are authorized to connect to the database server as **sysadmin** (that is, connect as “sa”), use the installer’s “Create the database/schema” option, and you can skip this section.

This section describes the procedure for manually creating an HP OO database, login, and user on Microsoft SQL Server.

Note: Only the database, login and user are created at this point; objects such as tables and indexes are not created yet. These objects are created once HP OO starts for the first time.

This section is relevant for you if, for example, due to security restrictions, you do not wish to use login/user credentials with elevated privileges during the HP OO installation. In such a case, you (or your organization’s DBA) should create the database, login, and user first, and then let the HP OO installer connect to the pre-existing database using “lower” privileges.

To create a database, you must connect to the SQL Server instance using a login that has **CREATE DATABASE** permission.

- Members of the sysadmin server roles automatically have **CREATE DATABASE** permission, and are also mapped to dbo in all databases.
- Perform the following procedures only if you are an experienced Microsoft SQL Server database administrator.
- If you prefer to use the database creation wizard/GUI, make sure you select all options that correspond with the T-SQL code presented below. For example, make sure you set **Allow Snapshot Isolation** to TRUE under the **Options** page/**Other Options** pane/**Miscellaneous** tab in the New Database dialog box.
- Not all database creation options are specified—only those that differ from the default value. When in doubt, use default values.

Selecting a specific collation also means that an HP OO system that uses SQL Server is limited to the set of languages supported by that specific collation. For example, if the **SQL_Latin1_General_CP1_CS_AS** collation is used, English, German, and Spanish characters may be used, but Japanese characters may not. If **Japanese_Unicode_CS_AS** is used, French accent characters will not be presented properly. For the complete specification of each collation, see the Microsoft SQL Server documentation.

To create a database:

1. Log in to Microsoft SQL Server as “sa” or any other login with a **sysadmin** role or the **CREATE DATABASE** permission.
2. Run the following T-SQL script and verify that the database was created successfully:

```
USE [master]
GO

CREATE DATABASE [<Enter the DB Name>]
```

```
ON PRIMARY
( NAME = N'OO',
  FILENAME = N'D:\path\to\data\OO.ndf',
  SIZE = 4MB,
  MAXSIZE = UNLIMITED, FILEGROWTH = 1MB )
LOG ON
( NAME = N'OO_log',
  FILENAME = N'D:\path\to\log\OO_log.ldf',
  SIZE = 1MB,
  MAXSIZE = UNLIMITED,
  FILEGROWTH = 10%)
COLLATE SQL_Latin1_General_CP1_CS_AS
GO

ALTER DATABASE [OO] SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE [OO] SET READ_COMMITTED_SNAPSHOT ON
ALTER DATABASE [OO] SET AUTO_CREATE_STATISTICS ON
ALTER DATABASE [OO] SET AUTO_SHRINK OFF
GO
```

Adapt the highlighted values to match your environment.

Note: The N's in the T-SQL script are optional, and are only required in certain situations. For example, you would require the N's in the following situation:

- The database service is installed on a native-Japanese Windows server, and you are connecting to the server using a client from an native-English Windows machine.

3. Run the following T-SQL script (still as "sa") to create a login and user for the HP OO system, and test them to verify that you can log in successfully:

```
USE [master]
GO

CREATE LOGIN [oouser] WITH PASSWORD=N'???' ,
  DEFAULT_DATABASE=[OO],
  DEFAULT_LANGUAGE=[us_english];
ALTER LOGIN [oouser] ENABLE;
GO

USE [OO]
GO

CREATE USER oouser FOR LOGIN [oouser];
GO
EXEC sp_addrolemember N'db_owner', N'oouser'
```


Adapt the highlighted values to match your environment. The login may be of any type (for example, Windows/Credentials-based) as long as you make sure that **oouser** has ownership on the HP OO database.

4. (Optional) In order to verify that database objects can be created by the new login and user, connect to the server as **oouser** and perform the following:

```
USE [oo]
GO

CREATE TABLE [dbo].[TEST_TABLE](
    [TEST_COLUMN] [int] NULL
)
GO

INSERT INTO [dbo].[TEST_TABLE] ([TEST_COLUMN]) VALUES ( 1 );
INSERT INTO [dbo].[TEST_TABLE] ([TEST_COLUMN]) VALUES ( 2 );
GO
```

Verify that the table was created and that it contains two rows. You can now manually or otherwise drop the table

5. (Optional) To verify that your newly created user has sufficient privileges to create the HP OO database, log in to the database server as **oouser** and perform the following:

```
USE [oo]
GO

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
```

Manually Creating Database Objects

Once the database, login, and user are in place, the database objects (tables, indexes, and so on) are created when the HP OO service starts and connects to the database for the first time.

You could create the database objects manually if the user created for HP OO does not have permissions to create or modify database objects (limited to DML operation only).

It is highly recommended to grant the HP OO database user DDL-related privileges, in order to allow HP OO to perform upgrade operations without manual intervention, as these operations sometimes require modifications to the database structure.

To manually create the database objects:

1. Extract the **mssql.sql** files from the HP OO DVD (or zip file) under **docs\sql**.
2. Log in to Microsoft SQL Server as the relevant user, who is permitted to create and modify database objects in the HP OO database.
3. Run the **mssql.sql** file and verify that no errors occurred.

Microsoft SQL Server Database Maintenance

This section describes the various maintenance tasks that are recommended for HP OO databases created on Microsoft SQL server, such as backing up the database, checking database integrity, handling index fragmentation, and monitoring the database.

This section includes:

- [Backing up the HP OO Database](#) 18
- [Creating a Maintenance Plan](#) 19

Backing up the HP OO Database

Microsoft SQL Server databases are either configured for the **Full** recovery model, or the **Simple** recovery model. You can back up an HP OO database using either one of these models. As HP OO keeps all configuration and operational history in a single database, always backup the complete database.

Consider the following guidelines when you create your backup plan for HP OO:

Backup method:

The backup method depends mainly on business considerations—how much information “may” be lost? What is the maximum time for system recovery? If you need to be able to perform point-in-time recovery, and only “allow” a few hours-worth of data loss, use the full recovery model and perform full or differential backups daily, and transaction log backup every N hours depending on your business requirements.

If your organization is more tolerant to data loss, you can use the simple recovery model and perform a daily or weekly full backup.

Backup frequency:

Daily backup is recommended, especially if you are using/modifying HP OO on a daily basis.

You should back up once a month at the very least.

Timing:

Schedule backup for the time when HP OO is least active.

Retention:

Retention depends on your business guidelines and regulations.

Creating a Maintenance Plan

Maintaining an HP OO database includes rebuilding the index and reclaiming free space. Use the scripts and tools described in this section, in order to keep the HP OO database in good shape.

Supplied Utilities for HP OO Database Housekeeping

HP OO provides a set of scripts for index maintenance, statistics maintenance, and history purging. These scripts create stored procedures that you can tune and schedule to run periodically.

It is recommended to use these procedures, but you can also use other methods in accordance with company policy, as long as indexes and statistics are kept in good shape.

Note that index rebuilding online (without HP OO system downtime) requires an enterprise grade database. Make sure you are running an enterprise version of Microsoft SQL Server before attempting online index rebuilding.

Also, note that maintenance activity usually consumes additional resources from the database. This is why it is important to schedule maintenance activity at the times when HP OO is least active.

Utility for Maintaining Indexes and Statistics

Download the latest **MSSQL.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.10** and unpack it.

To install and use the HP OO maintenance stored procedures:

1. Log in to Microsoft SQL Server as “sa” or any member of the **sysadmin** role and run the following code in order to give the HP OO user the ability to access **dm_os_performance_counters** dynamic management view (DMV):

```
USE [master]
GO

GRANT VIEW SERVER STATE TO oouser
GO
```

Replace “oouser” with the actual user created for HP OO.

2. Edit the following T-SQL scripts and replace each “USE <your_db_name_here>” in the file headers to your actual HP OO database name. For example, if your database name is “OOPROD”, replace it with “USE OOPROD”.
 - **OO_DB_MAINTENANCE_LOG.sql** (optional)
 - **OOcmdExec.sql**
 - **OOIndexMaintenance.sql**

Do not skip this step; otherwise, the set of procedures will not be created in the correct database.

3. Log in to Microsoft SQL Server as the HP OO user.
4. Run the following T-SQL scripts in the given order and verify that the new objects were created

successfully:

- **OO_DB_MAINTENANCE_LOG.sql** (optional)
- **OOCmdExec.sql**
- **OOIndexMaintenance.sql**

5. Tune your stored procedures according to the comments embedded in the script.

The following example shows how this procedure can be used. For detailed explanations, see the guidelines provided as comments in the procedure header.

```
USE [OO]
GO

EXECUTE [dbo].[OOIndexMaintenance]
    @DatabaseName = 'OO'
    ,@FragmentationLow = NULL
    ,@FragmentationMedium = 'INDEX_REORGANIZE,INDEX_REBUILD_ONLINE,INDEX_
REBUILD_OFFLINE'
    ,@FragmentationHigh = 'INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE'
    ,@FragmentationLevel1 = 5
    ,@FragmentationLevel2 = 30
    ,@SortInTempdb = 'N'
    ,@Indexes = 'OO.dbo.%'
    ,@TimeLimit = 1800
    ,@LockTimeout = 20
    ,@LogToTable = 'Y'
    ,@Execute = 'Y'
GO
```

Explanation of the above code:

- Replace "OO" with the actual name of your database. Note that there are three occurrences.
- The **@FragmentationXXX** parameters set the script's fragmentation level sensitivity and course of action in each case. These threshold levels and subsequent actions are recommended by Microsoft's documentation. Tune these values with caution.
- **@SortInTempdb** (once set to "Y") lets you perform sorting operations during index reorganization/rebuild in tempdb rather than in memory, for better performance. If you choose to use this option, make sure you have sufficient free space in **tempdb**.
- **@Indexes** is a filter for including/excluding indexes in the maintenance operation. It is recommended to keep this filter as is, to analyze all indexes.
- **@TimeLimit** is the timeout in seconds for the maintenance operation to complete. Set it in accordance with your maintenance window boundaries, if applicable.
- **@LockTimeout** is the timeout in seconds to wait for object lock. Once expired, the specific operation fails and the procedure continues to the next object.
- **@LogToTable** determines whether maintenance operation results should be logged to a table. This lets you keep track of the maintenance operations and helps procedure debugging.

Note that it requires creating the table using the **OO_DB_MAINTENANCE_LOG.sql** script.

- **@Execute** determines whether actual operations (such as index rebuild) are performed or not. If this parameter is set to 'N', the procedure performs a “dry run” and shows an analysis of the relevant objects.

Utility for Purging Historical Data

To install and use the HP OO history purging stored procedure:

1. Edit the following T-SQL scripts and replace each “USE <your_db_name_here>” in the file headers with your actual HP OO database name. For example, if your database name is “OOPROD”, replace it with “USE OOPROD”.

```
OOGetErrorInfo.sql
```

```
OOPurgeHistory.sql
```

Do not skip this step; otherwise, the procedure will not be created in the correct database.

2. Log in to Microsoft SQL Server as the HP OO user.
3. Run the following T-SQL scripts in the given order and verify that the new objects were created successfully:

```
OOGetErrorInfo.sql
```

```
OOPurgeHistory.sql
```

4. Tune your stored procedures according to the comments embedded in the script.

The following example shows how this procedure may be used. See the guidelines provided as comments in the procedure header for detailed explanations.

```
USE [OO] GO

EXECUTE [dbo].[OOPurgeHistory]
@PurgeExecutionsOlderThan = 90
,@PurgeExecutionsInBatchesOf = 1000
,@ShouldPurgeExecutionSummary = 0
,@verbose = 1
,@StopPurgingProcessAfter = 4
,@DeepClean = 0
GO
```

Explanation of the above code:

- Replace “OO” with the actual name of your database.
- The **PurgeExecutionsOlderThan** parameter determines how many days are kept, (protected) relative to the time the procedure starts running. By default, 90 days are kept. Older data is deleted, starting with the oldest records.
- **@PurgeExecutionsInBatchesOf** determines how many flows are handled together. Smaller values imply smaller, more frequent transactions, and higher values imply less frequent, larger transactions. 1000 is recommended for most systems.
- **@ShouldPurgeExecutionSummary** determines if the **OO_EXECUTION_SUMMARY** table should be purged. The default value is "0" (do not purge this table). It is recommended to keep data in

this table as it does not consume a lot of space. Use "1" only if you want to completely remove any reference to the relevant flows.

- **@verbose** determines the verbosity level. "0" corresponds to "quiet" output, "1" corresponds to normal output, and "2" prints out detailed information.
- **@StopPurgingProcessAfter** is the timeout in hours for the operation to complete. Set it in accordance with your maintenance window boundaries if applicable.
- **@DeepClean** determines whether deep cleansing is performed. For example, searching for "orphan" records that may bloat the database unnecessarily. The default is "0" (off). Note that setting this flag to "1" prolongs the procedure run time, but the timeout limit is still imposed.

Always On Support

Microsoft SQL Server "Always On" is supported for HP OO as a high availability and/or disaster recovery solution.

High Availability Setup

As HP OO requires a single point of contact, it is essential that an Availability Group Listener (AG listener) exist in the high availability setup.

Normally, HP OO interfaces with the AG listener in order to connect to the availability group primary replica (a READ-WRITE capable instance).

In case of a database failover scenario, while database connectivity is lost, HP OO will repeatedly attempt to connect to the database until a new READ-WRITE capable instance accepts the connection.

Note: HP OO has no use for a read-only database (thus has no use for read-only routing).

Disaster Recovery Solution

In a disaster recovery setup, HP OO should be halted, reconfigured (edit the **database.properties** file), and then started again once the new database instance is available.

See the *HP OO Administration Guide* for details on reconfiguring the database connection details.

Deploying and Maintaining an Oracle Database

In order to deploy HP OO using Oracle, you must have an existing Oracle database service. If you need to create a new database instance/service, see the relevant documentation provided by Oracle, because this information is not included within this guide. However, this guide contains recommendations for the Oracle instance configuration.

This chapter includes the following sections:

- [Workflow for Oracle Deployment](#) 23
- [System Requirements for Oracle](#) 24
 - [Hardware Requirements](#) 24
 - [Software Requirements](#) 24
 - [Examples of Tested Deployments](#) 24
 - [Language Support](#) 25
- [Configuring an Oracle Database](#) 25
- [Manually Creating an HP OO Database on an Oracle Instance](#) 27
 - [Manually Creating Database Objects](#) 28
 - [Connecting to Oracle Using SID or Service Name](#) 29
- [Oracle Database Maintenance](#) 29
 - [Backing up the HP OO Database](#) 29
 - [Creating a Maintenance Plan](#) 30

Workflow for Oracle Deployment

To deploy HP OO using Oracle, perform the following steps:

1. **Review sizing guidelines.** For details, see "HP OO Database Sizing" in ["Introduction to Preparing the Database Environment"](#) on page 6.
2. **Review Hardware and Software Requirements.** For details, see ["System Requirements for Oracle"](#) on the next page.
3. **Configure an Oracle Database.** For details, see ["Configuring an Oracle Database"](#) on page 25.
4. **Create a Database.** For details, see ["Manually Creating an HP OO Database on an Oracle Instance"](#) on page 27.
5. (Optional) **Connect HP OO to an Oracle RAC environment.** For details, see "Support for Oracle Real Application Cluster" in ["Appendix B: Additional Guidelines for Oracle"](#) on page 53. This step is only relevant if you are using HP OO in an Oracle RAC environment.

System Requirements for Oracle

This section describes the system requirements for working with Oracle in conjunction with HP OO.

Hardware Requirements

For HP OO database sizing guidelines and hardware requirements, see "HP OO Database Sizing" and "Hardware Requirements" in ["Introduction to Preparing the Database Environment" on page 6](#).

For Oracle hardware requirements, see the relevant installation guide for your Oracle release and operating system.

Software Requirements

The following table lists the Oracle releases supported by HP OO:

Oracle Releases			
Version	Type	32/64Bit	Patch Set
11gR2	Standard	64 Bit	11.2.0.1 – 11.2.0.4
	Enterprise	64 Bit	11.2.0.1 – 11.2.0.4
12cR1 Regular instance, non-CDB	Standard	64 Bit	12.1.0.1
	Enterprise	64 Bit	12.1.0.1

Only the listed patch sets should be installed. Newer patch sets are also supported unless stated otherwise in the *HP OO Release Notes*.

See the Oracle documentation for supported platforms.

Note that Oracle 12c R1 is only supported in its regular, backward-compatible instance form. HP OO 10.20 does not support connecting to an Oracle 12c container database (CDB).

Examples of Tested Deployments

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

Database Release			Operating System
Version	32/64Bit	Patch Set	
Oracle 11g R2 Enterprise	64 Bit	11.2.0.4.0	Red Hat Enterprise Linux 6.3 64 Bit

Database Release			Operating System
Version	32/64Bit	Patch Set	
Edition			
Oracle 11g R2 Enterprise Edition	64 Bit	11.2.0.4.0	Windows 2012 Standard Edition 64 Bit
Oracle 12c R1 Enterprise Edition	64 Bit	12.1.0.1	Red Hat Enterprise Linux 6.5 64 Bit

Language Support

The Oracle instance character set should be set to **AL32UTF8**. This enables using any Unicode character (and practically all common characters in all languages).

Configuring an Oracle Database

This section contains information on the Oracle instance and database configuration settings.

You can install an HP OO database in an Oracle clustered environment (Oracle RAC or other).

Legend:

- **Mandatory** configuration options/values appear in **bold/orange** font.
- **Recommended** configuration options/values appear in **bold/purple** font.
- Supported configuration options/values appear in normal font, and may show as a comma-separated list.
- *Comments* appear in *italic* font.

Oracle Database 11gR2 / 12cR1			
Instance/Server Options			
Instance configuration options	Defaults, unless instructed otherwise		
	Mandatory	Recommended	Supported
PROCESSES	>= 500		
SESSIONS	>= 555		
TIMED_STATISTICS		TRUE	TRUE, FALSE
OPEN_CURSORS	>= 900		
Shared/Dedicated Server		Dedicated	Dedicated, Shared

Oracle Database 11gR2 / 12cR1			
Instance/Server Options			
Instance configuration options	Defaults, unless instructed otherwise		
	Mandatory	Recommended	Supported
UNDO_MANAGEMENT		AUTO	Automatic, Manual
Undo size	>= 4GB	6GB – 10GB	
Memory Management		ASMM	AMM, ASMM
MEMORY_TARGET		0 (disabled)	>= 5G (for AMM)
SGA_TARGET		8G – 12G	>= 4G (for ASMM)
SGA_MAX_SIZE		8G – 12G	>= 4G (for ASMM)
PGA_AGGREGATE_TARGET		1G – 2G	>= 500M (for ASMM)

- Note that all values reflect resources required by HP OO. If HP OO shares the Oracle instance with other users, these values should be added to whatever is currently consumed by others.
- See the sizing guide to determine values displayed as range.

Oracle Database 11gR2 / 12cR1			
Instance/Server Options			
	Mandatory	Recommended	Supported
File system			ASM, Any
Storage options		Locally managed tablespace	
		Automatic segment space management (ASSM)	
		Automatic local extent management	
ARCHIVELOG mode		ARCHIVELOG	ARCHIVELOG, NOARCHIVELOG
Redo Log total size	>= 600MB	1GB	

- Note that all values reflect resources required by HP OO. If HP OO shares the Oracle instance with other users, these values should be added to whatever is currently consumed by others.
- See the sizing guide to determine values displayed as range.

Manually Creating an HP OO Database on an Oracle Instance

During HP OO setup, a new database can be created automatically by the HP OO installer or a pre-existing database can be used.

If during installation, you are authorized to connect to the database server as dba (connect as "SYSTEM"), use the "create the database/schema" option, and you can skip this section.

Note: In some cases, the term "database" is used but in the case of Oracle, it should be interpreted as "user".

This section describes the procedure for manually creating an HP OO database in an Oracle instance.

Note: Only the database is created at this point; objects such as tables and indexes are not created yet. These objects are created once HP OO starts for the first time.

This section is relevant for you if, for example, due to security restrictions, you do not wish to use user credentials with elevated privileges during the HP OO installation. In such a case, you (or your organization's DBA) should create the user (database) first, and then let the HP OO installer connect to the pre-existing database using basic privileges.

To create a database, you must connect to the Oracle instance using a login that has **CREATE USER** system privilege—for example, system user.

- Any user with the DBA role has sufficient privileges to create the new user.
- Perform the following procedures only if you are an experienced Oracle database administrator.
- If you prefer to use the database creation wizard/GUI, make sure you select all options that correspond with the SQL code presented below.
- Not all database creation options are specified—only those that differ from the default value. When in doubt, use default values.

To create a database:

1. Log in to the oracle as "system" or any other user with a DBA role.
2. Run the following SQL script and verify that the database was created successfully:

```
CREATE USER OO
  IDENTIFIED BY ???????
  DEFAULT TABLESPACE <default tablespace for OO>
  TEMPORARY TABLESPACE <temporary tablespace for OO>
  QUOTA UNLIMITED ON <default tablespace for OO>
  ACCOUNT UNLOCK
;
```

```
GRANT CONNECT TO OO;  
GRANT CREATE VIEW, CREATE SEQUENCE, CREATE TABLE, CREATE PROCEDURE TO OO;
```

Adapt the highlighted values to match your environment.

Note that as of Oracle 12c, “UNLIMITED TABLESPACE” is no longer part of the “RESOURCE” role. Make sure you specify “QUOTA UNLIMITED” in the user creation command.

3. (Optional) In order to verify that database objects can be created by the new user, connect to the Oracle instance as HP OO and perform the following:

```
CREATE TABLE TEST_TABLE(  
    TEST_COLUMN int NULL  
);  
  
INSERT INTO TEST_TABLE (TEST_COLUMN) VALUES ( 1 );  
INSERT INTO TEST_TABLE (TEST_COLUMN) VALUES ( 2 );  
COMMIT;
```

Verify that the table was created and that it contains two rows. You can now manually or otherwise drop the table.

Manually Creating Database Objects

Once the database is in place, the database objects (tables, indexes, and so on) are created when the HP OO service starts and connects to the database for the first time.

You can create the database objects manually if the user created for HP OO does not have permissions to create or modify database objects (limited to DML operation only).

It is highly recommended to grant the HP OO database user DDL-related privileges, in order to allow HP OO to perform upgrade operations without manual intervention, as these operations sometimes require modifications to database structure.

To manually create the database objects:

1. Extract the **oracle.sql** file from the HP OO 10.20 DVD (or zip file) under **docs\sql**.
2. Edit the **oracle.sql** file and prefix each object with the HP OO user, to make sure the objects are created under the HP OO user.
3. Connect to the HP OO database as the relevant user, who is permitted to create and modify database objects in the HP OO database.
4. Run the **oracle.sql** file and verify that no errors occurred, and that all objects are created and owned by the HP OO user.

Connecting to Oracle Using SID or Service Name

Connecting to an Oracle database server requires you to specify either the SID (system ID) or Service Name. As of version 10.20, HP OO supports specifying the service name during installation (using either the Installation Wizard or a silent installation).

The following examples show how to format JDBC URL for each option, as it should appear in the Central **database.properties** file under the **central\conf** folder.

Connecting to an Oracle instance using the SID:

```
jdbc.url=jdbc\:oracle\:thin\:@DB_HOSTNAME_OR_IP\:PORT\:SID
```

Connecting to an Oracle instance using the Service Name:

```
jdbc.url=jdbc\:oracle\:thin\:@//DB_HOSTNAME_OR_IP\:PORT/SERVICE_NAME
```

Oracle Database Maintenance

This section describes the various maintenance tasks that are recommended for HP OO databases created on Oracle, such as backing up the database, checking database integrity, handling index fragmentation, and monitoring the database.

This section includes:

- [Backing up the HP OO Database](#) 29
- [Creating a Maintenance Plan](#) 30

Backing up the HP OO Database

Oracle databases can be backed up using several tools, such as **expdp** and **RMAN**. An HP OO database can be backed up using any type of method/tool as long as the complete database is backed up.

Consider the following guidelines when you create your backup plan for HP OO:

Backup method:

The backup method depends mainly on business considerations—how much information “may” be lost? What is the maximum time for system recovery? If you need to be able to perform point-in-time recovery, and only “allow” a few hours-worth of data loss, use the full recovery model and perform full or differential backups daily, and transaction log backup every N hours depending on your business requirements.

Backup frequency:

Daily backup is recommended, especially if you are using/modifying HP OO on a daily basis.

You should back up once a month at the very least.

Timing:

Schedule backup for the time when HP OO is least active.

Retention:

Retention depends on your business guidelines and regulations.

Creating a Maintenance Plan

Maintaining an HP OO database includes rebuilding the index and reclaiming free space. Use the scripts and tools described in this section, in order to keep the HP OO database in good shape.

Supplied Utilities for HP OO Database Housekeeping

HP OO provides a set of scripts for index maintenance, statistics maintenance, and history purging. These scripts create a package containing stored procedures that you can tune and schedule to run periodically.

It is recommended to use these procedures, but you can also use other methods in accordance with company policy, as long as indexes and statistics are kept in good shape.

Note that index rebuilding online (without HP OO system downtime) requires an enterprise grade database. Make sure you are running an enterprise version of Oracle before attempting online index rebuilding.

Also, note that maintenance activity usually consumes additional resources from the database. This is why it is important to schedule maintenance activity at the times when HP OO is least active.

Utility for Maintaining Indexes and Statistics

Download the latest **Oracle.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.10** and unpack it.

To install and use the HP OO maintenance stored procedures:

1. Log in to Oracle as “system” or any other user with a DBA role, and run the following commands. These system privileges are required in order to that verify the stored procedure created in the following steps has the explicit (not role-based) privileges to execute the index analysis and rebuild:

```
GRANT CREATE TABLE TO OO;  
GRANT ANALYZE ANY TO OO;  
GRANT ALTER ANY INDEX TO OO;
```

Adapt the highlighted user name to match your environment.

2. Log in to Oracle as “OO” (the user created for HP OO).
3. Run the **HP_OO_DB_MAINT.sql** script and verify that the new package and procedures were created successfully.
4. Tune your stored procedures according to the comments embedded in the script.

The following example shows how this procedure can be used. For detailed explanations, see the guidelines provided as comments in the procedure header.

```
SET serveroutput ON 100000
```

```
DECLARE x integer := 0;

BEGIN
  hp_oo_db_maint.IndexMaintenance(3, 15, 1, x);
END;
```

Stored procedure parameters:

- **pMaxHeight (IN)** - The minimal index height threshold for index rebuilding. The Oracle documentation recommends 3. Smaller values may result in unnecessary rebuilding operations.
- **pMaxLeafsDeleted (IN)** - The minimal deleted leaves threshold for index rebuilding. The Oracle documentation recommends 15. Smaller values may result in unnecessary rebuilding operations.
- **pRebuild (IN)** - Should indexes be rebuilt (1) or only perform a dry-run (0). A dry-run will show only recommendations for index rebuilding.
- **pReturnValue (OUT)** - The number of rebuilt indexes

Note: ONLINE index rebuilding should only be performed when the enterprise edition is used. Otherwise, the index rebuilding operation may lock tables and indexes and may interfere with the operation of HP OO..

Utility for Purging Historical Data

To install and use the HP OO history purging stored procedure:

1. Log in to Oracle as “OO” (the user created for HP OO).
2. Run the **HP_OO_DB_MAINT.sql** script (only if you have not run it already) and verify that the new package and procedures were created successfully:
3. Tune your stored procedures according to the comments embedded in the script.

The following example shows how this procedure may be used. See the guidelines provided as comment in the procedure header for detailed explanations.

```
SET serveroutput ON SIZE 100000

DECLARE x integer := 0;

BEGIN
  hp_oo_db_maint.PurgeHistory(90,10000,0,1,4,0,x);
  DBMS_OUTPUT.put_line('A total of ' || TO_CHAR(x) || ' flows were handled.');
```

Explanation about the above code:

- The **pPurgeExecutionsOlderThan** parameter determines how many days are kept (protected) relative to the time the procedure starts running. Older data is deleted, starting with the oldest records. This parameter has no default value and must be specified.
- **pPurgeExecutionsInBatchesOf** determines the maximum number of flows to handle in each batch. The default value is 10000.

- **pShouldPurgeExecutionSummary** determines if the **OO_EXECUTION_SUMMARY** table should be purged. The default value is "0" (do not purge this table). It is recommended to keep data in this table as it does not consume a lot of space. Use "1" only if you want to completely remove any reference to the relevant flows.
- **pVerbose** determines verbosity level. "0" corresponds to "quiet" output, "1" corresponds to normal output, and "2" prints out detailed information. Set it in accordance with your maintenance window boundaries if applicable.
- **pStopPurgingProcessAfter** is the timeout in hours for the operation to complete.
- **pDeepClean** determines whether deep cleansing is performed. For example, searching for "orphan" records that may bloat the database unnecessarily. The default is "0" (off). Note that setting this flag to "1" prolongs the procedure run time, but the timeout limit is still imposed.

Deploying and Maintaining a MySQL Database

In order to deploy HP OO using MySQL, you must have an existing MySQL database. If you need to create a new database service, see the relevant documentation provided by MySQL, because this information is not included within this guide. However, this guide contains recommendations for the MySQL configuration.

This chapter includes the following sections:

- [Workflow for MySQL Deployment](#) 33
- [System Requirements for MySQL](#) 33
 - [Hardware Requirements](#) 34
 - [Software Requirements](#) 34
 - [Examples of Tested Deployments](#) 35
 - [Language Support](#) 35
- [Configuring MySQL](#) 35
- [Manually Creating an HP OO Database on MySQL](#) 37
 - [Manually Creating Database Objects](#) 38
- [MySQL Database Maintenance](#) 39
 - [Backing up the HP OO Database](#) 39
 - [Creating a Maintenance Plan](#) 39
 - [Utility for Purging Historical Data](#) 40

Workflow for MySQL Deployment

To deploy HP OO using MySQL, perform the following steps:

1. **Review sizing guidelines.** For details, see "HP OO Database Sizing" in "[Introduction to Preparing the Database Environment](#)" on page 6.
2. **Review Hardware and Software Requirements.** For details, see "[System Requirements for MySQL](#)" below.
3. **Configure MySQL** . For details, see "[Configuring MySQL](#)" on page 35.
4. **Create HP OO database on MySQL** . For details, see "[Manually Creating an HP OO Database on MySQL](#)" on page 37.

System Requirements for MySQL

This section describes the system requirements for working with MySQL in conjunction with HP OO.

Hardware Requirements

For HP OO database sizing guidelines and hardware requirements, see "HP OO Database Sizing" and "Hardware Requirements" in ["Introduction to Preparing the Database Environment" on page 6](#).

For MySQL hardware requirements, see the relevant installation guide for your MySQL release and operating system.

Software Requirements

The following table lists the MySQL releases supported by HP OO:

MySQL Database Releases		
Version	Type	32/64Bit
5.5	Community	x86 32-bit
		x86 64-bit
	Standard	x86 32-bit
		x86 64-bit
	Enterprise	x86 32-bit
		x86 64-bit
5.6	Community	x86 32-bit
		x86 64-bit
	Standard	x86 32-bit
		x86 64-bit
	Enterprise	x86 32-bit
		x86 64-bit

Note: For MySQL 5.6.20 and 5.6.21, the requirements for the `innodb_log_file_size` have increased significantly.

See the MySQL documentation for supported platforms.

Examples of Tested Deployments

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

Database Release			Operating System
Version	32/64Bit	Patch	
MySQL Server 5.6.13 Community Edition	64 Bit		Windows 2012 Standard Edition 64 Bit
MySQL Server 5.6.12 Community Edition	64 Bit		Red Hat Enterprise Linux 6.3 64 Bit

Language Support

MySQL Server character set should be set to utf8. This lets you use any Unicode character (and practically all common characters in all languages). Note that the HP OO database uses the utf8_bin collation.

Configuring MySQL

This section contains information on MySQL and database configuration settings.

Legend:

- **Mandatory** configuration options/values appear in **bold/orange** font.
- **Recommended** configuration options/values appear in **bold/purple** font.
- Supported configuration options/values appear in normal font, and may show as a comma-separated list.
- *Comments* appear in *italic* font.

MySQL 5.5 – 5.6			
Instance/Server Options			
Server Configuration Options	Defaults, unless instructed otherwise		
[mysqld]	Mandatory	Recommended	Supported
character-set-server	utf8		
collation-server	utf8_bin		
transaction-isolation	READ-		

MySQL 5.5 – 5.6			
Instance/Server Options			
	COMMITTED		
max_allowed_packet	250M		
max_connections	>= 1000		
default-storage-engine	INNODB		
innodb_log_file_size (for MySQL 5.6.1 - 19)	256M		
innodb_log_file_size (for MySQL 5.6.20 - 21)	2GB		
max_connect_errors		100000000	
innodb_file_per_table		1	
innodb_thread_concurrency		0	
table_open_cache		1000	
sort_buffer_size		2M	
read_buffer_size		2M	
tmp_table_size		400M	
max_heap_table_size		400M	
innodb_buffer_pool_size		4096M	
innodb_additional_mem_pool_size		20M	
innodb_locks_unsafe_for_binlog		1	
binlog_format		row	
innodb_flush_log_at_trx_commit		2	
innodb_flush_method		O_DIRECT	
innodb_doublewrite		0	

MySQL 5.5 – 5.6			
Other Options			
Server Configuration Options	Defaults, unless instructed otherwise		
	Mandatory	Recommended	Supported

MySQL 5.5 – 5.6			
Other Options			
[client]			
default-character-set	utf8		
[mysql]			
default-character-set	utf8		
[mysqldump]			
max_allowed_packet	250M		

Manually Creating an HP OO Database on MySQL

During HP OO setup, a new database can be created automatically by the HP OO installer or a pre-existing database can be used.

If during installation, you are authorized to connect to the database server using the DBA role (connect as “root”), use the “create the database/schema” option, and you can skip this section.

This section describes the procedure for manually creating an HP OO database on MySQL.

Note: Only the database and user are created at this point; objects such as tables and indexes are not created yet. These objects are created once HP OO starts for the first time.

This section is relevant for you if, for example, due to security restrictions, you do not wish to use login credentials with elevated privileges during the HP OO installation. In such a case, you (or your organization’s DBA) should create the database first, and then let the HP OO installer connect to the pre-existing database using basic privileges.

To create a database, you must connect to the SQL Server instance using a user that has **CREATE** permission (at the very least).

- **root** has all privileges. Any member of the DBA role will also be able to create the user and database.
- Perform the following procedures only if you are an experienced MySQL database administrator.
- If you prefer to use the MySQL Workbench GUI, make sure you select all options that correspond with the SQL code presented below.
- Not all database creation options are specified—only those that differ from the default value. When in doubt, use default values.

To create a database:

1. Log in to the MySQL as “root” or any other member of the DBA role.
2. Run the following SQL script and verify that the database was created successfully:

```
CREATE DATABASE IF NOT EXISTS `00` COLLATE utf8_bin;  
CREATE USER 'oouser'@'%' IDENTIFIED BY '??????';  
GRANT ALL PRIVILEGES ON `00`.* to 'oouser';  
FLUSH PRIVILEGES;
```

Adapt the highlighted values to match your environment.

3. Test your newly created connection to the database and verify that you are able to log in successfully.
4. (Optional) In order to verify that database objects can be created by the new login and user, connect to the server as **oouser** and perform the following:

```
USE 00;  
  
CREATE TABLE TEST_TABLE(  
    TEST_COLUMN int NULL  
);  
  
INSERT INTO TEST_TABLE (TEST_COLUMN) VALUES ( 1 );  
INSERT INTO TEST_TABLE (TEST_COLUMN) VALUES ( 2 );
```

Verify that the table was created and that it contains two rows. You can now manually or otherwise drop the table.

Manually Creating Database Objects

Once the database and user are in place, the database objects (tables, indexes, and so on) are created when the HP OO service starts and connects to the database for the first time.

You can create the database objects manually if the user created for HP OO does not have permissions to create or modify database objects (limited to DML operation only).

It is highly recommended to grant the HP OO database user DDL-related privileges, in order to allow HP OO to perform upgrade operations without manual intervention, as these operations sometimes require modifications to database structure.

To manually create the database objects:

1. Extract the **mysql.sql** file from the HP OO DVD (or zip file) under **docs\sql**.
2. Connect to the HP OO database as the relevant user, permitted to create and modify database objects in the HP OO database.
3. Run the **mysql.sql** file and verify that no errors occurred.

MySQL Database Maintenance

This section describes the various maintenance tasks that are recommended for HP OO databases created on MySQL, such as backing up the database, checking database integrity, handling index fragmentation, and monitoring the database.

This section includes:

- [Backing up the HP OO Database](#) 39
- [Creating a Maintenance Plan](#) 39
- [Utility for Purging Historical Data](#) 40

Backing up the HP OO Database

You can back up MySQL database using several tools, such as **mysqldump** or **mysqlbackup**. You can back up the HP OO database using any type of method/tool as long as the complete database is backed up.

Consider the following guidelines when you create your backup plan for HP OO:

Backup method:

The backup method depends mainly on business considerations—how much information “may” be lost? What is the maximum time for system recovery? If you need to be able to perform point-in-time recovery, and only “allow” a few hours-worth of data loss, use the full recovery model and perform full or differential backups daily, and transaction log backup every N hours depending on your business requirements.

Backup frequency:

Daily backup is recommended, especially if you are using/modifying HP OO on a daily basis.

You should back up once a month at the very least.

Timing:

Schedule backup for the time when HP OO is least active.

Retention:

Retention depends on your business guidelines and regulations.

Creating a Maintenance Plan

Maintaining an HP OO database includes rebuilding the index and reclaiming free space. Use the scripts and tools described in this section, in order to keep the HP OO database in good shape.

Recommended Utility for Database Maintenance

In order to keep OO database in good shape, it is recommended to schedule **mysqlcheck** utility to run during a system maintenance window.

Important! Note that this operation locks tables! Only perform this action during a maintenance window when the HP OO system is not operating!

Here is an example of how to run this utility:

```
mysqlcheck -uouser -p???? -os --auto-repair OO
```

Replace "ouser" and "OO" with the actual HP OO user name and database name, respectively.

It is recommended not to provide the password explicitly. See the MySQL documentation for recommendations on how to secure database passwords.

Utility for Purging Historical Data

To install and use the HP OO history purging stored procedure:

1. Download the latest **MySQL.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.20** and unpack it.
2. Log in to MySQL database as "ouser" (the user created for HP OO).
3. Run the **00PurgeHistory.sql** script and verify that the new procedure was created successfully.
4. Tune your stored procedures according to the comments embedded in the script.

The following example shows how this procedure may be used. See the guidelines provided as comments in the procedure header for detailed explanations.

```
CALL 00PurgeHistory(90,10000,0,1,4,0,@res);
```

Explanation about the above code:

- The **pPurgeExecutionsOlderThan** parameter determines how many days are kept (protected) relative to the time the procedure starts running. It is recommended to keep at least 90 days. Older data is deleted, starting with the oldest records.
- **pPurgeExecutionsInBatchesOf** determines the maximum number of flows to handle in each batch. The default value is 10000.
- **pShouldPurgeExecutionSummary** determines if the OO_EXECUTION_SUMMARY table should be purged. The default value is "0" (do not purge this table). It is recommended to keep data in this table as it does not consume a lot of space. Use "1" only if you want to completely remove any reference to the relevant flows.
- **pVerbose** determines verbosity level. "0" corresponds to "quiet" output, "1" corresponds to normal output, and "2" prints out detailed information.
- **pStopPurgingProcessAfter** is the timeout in hours for the operation to complete. Set it in accordance with your maintenance window boundaries if applicable.

- **pDeepClean** determines whether deep cleansing is performed. For example, searching for “orphan” records that may bloat the database unnecessarily. The default is “0” (off). Note that setting this flag to “1” prolongs the procedure run time, but the timeout limit is still imposed.

Deploying and Maintaining a Postgres Database

In order to deploy HP OO using Postgres, you must have an existing Postgres database service. If you need to create a new database service, see the relevant documentation provided by Postgres, because this information is not included within this guide. However, this guide contains recommendations for the Postgres configuration.

This chapter includes the following sections:

- [Workflow for Postgres Deployment](#) 42
- [System Requirements for Postgres](#) 42
 - [Hardware Requirements](#) 43
 - [Software Requirements](#) 43
 - [Examples of Tested Deployments](#) 43
 - [Language Support](#) 44
- [Configuring Postgres](#) 44
- [Manually Creating an HP OO Database on Postgres](#) 45
 - [Manually Creating Database Objects](#) 46
- [Postgres Database Maintenance](#) 46
 - [Backing up the HP OO Database](#) 47
 - [Creating a Maintenance Plan](#) 47
 - [Utility for Purging Historical Data](#) 48

Workflow for Postgres Deployment

To deploy HP OO using Postgres, perform the following steps:

1. **Review sizing guidelines.** For details, see "HP OO Database Sizing" in ["Introduction to Preparing the Database Environment"](#) on page 6.
2. **Review Hardware and Software Requirements.** For details, see ["System Requirements for Postgres"](#) below.
3. **Configure Postgres.** For details, see ["Configuring Postgres"](#) on page 44.
4. **Create HP OO database on Postgres.** For details, see ["Manually Creating an HP OO Database on Postgres"](#) on page 45.

System Requirements for Postgres

This section describes the system requirements for working with Postgres in conjunction with HP OO.

Hardware Requirements

For HP OO database sizing guidelines and hardware requirements, see "HP OO Database Sizing" and "Hardware Requirements" in ["Introduction to Preparing the Database Environment" on page 6](#).

For Postgres hardware requirements, see the relevant installation guide for your Postgres release and operating system.

Software Requirements

The following table lists the Postgres releases supported by HP OO:

Postgres Database Releases	
Version	Type
9.1	x86 32-bit
	x86 64-bit
9.2	x86 32-bit
	x86 64-bit
9.3	x86 32-bit
	x86 64-bit

Only supported versions should be used.

See the Postgres documentation for supported platforms.

Examples of Tested Deployments

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

Database Release		Operating System
Version	32/64Bit	
Postgres 9.2.3	64 Bit	Windows 2012 Standard Edition 64 Bit
Postgres 9.1.9	64 Bit	Red Hat Enterprise Linux 6.3 64 Bit
Postgres 9.3.2	64 Bit	Red Hat Enterprise Linux 6.3 64 Bit

Language Support

Postgres determines character set and collation at the database level. The HP OO database uses Unicode (utf8) encoding and collation. This lets you use any Unicode character (and practically all common characters in all languages).

Configuring Postgres

This section contains information on Postgres configuration settings.

Legend:

- **Mandatory** configuration options/values appear in **bold/orange** font.
- **Recommended** configuration options/values appear in **bold/purple** font.
- Supported configuration options/values appear in normal font, and may show as a comma-separated list.
- *Comments* appear in *italic* font.

Postgres 9.1 - 9.3			
Instance/Server Options			
Instance Configuration Options	Defaults, unless instructed otherwise		
	Mandatory	Recommended	Supported
max_connections	>= 1000		
default_transaction_isolation	'read committed'		
autovacuum	on		
track_counts	on		
shared_buffers	>=512MB ¹		
effective_cache_size	>=2048MB ¹		
work_mem	>=1MB ¹		
maintenance_work_mem	>=32MB ¹		
lc_messages		'en_US.UTF-8'	Any
lc_monetary		'en_US.UTF-8'	Any

[1] - Minimal values. See the Postgres documentation on how to tune these values in accordance with your environment.

Manually Creating an HP OO Database on Postgres

During HP OO setup, a new database can be created automatically by the HP OO installer or a pre-existing database can be used.

If during installation, you are authorized to connect to the database server as a privileged user (connect as “postgres”), use the “create the database/schema” option, and you can skip this section.

This section describes the procedure for manually creating an HP OO database on Postgres.

Note: Only the database and role are created at this point; objects such as tables and indexes are not created yet. These objects are created once HP OO starts for the first time.

This section is relevant for you if, for example, due to security restrictions, you do not wish to use login/user credentials with elevated privileges during the HP OO installation. In such a case, you (or your organization’s DBA) should create the database, login, and user first, and then let the HP OO installer connect to the pre-existing database using basic privileges.

To create a database, you must connect to the Postgres instance using a login that has **CREATEUSER** and **CREATEDB** privileges at the very least.

- The **postgres** built-in user has all the required privileges.
- Perform the following procedures only if you are an experienced Postgres database administrator.
- If you prefer to use the PgAdmin GUI, make sure you select all options that correspond with the SQL code presented below.
- Not all database creation options are specified—only those that differ from the default value. When in doubt, use default values.

To create a database:

1. Log in to Postgres as “postgres” or any other login role with **CREATEUSER** and **CREATEDB** privileges.
2. Run the following SQL script and verify that the database was created successfully:

```
CREATE ROLE oouser LOGIN
UNENCRYPTED PASSWORD '???????'
NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE NOREPLICATION;

CREATE DATABASE "OO"
WITH OWNER = oouser
ENCODING = 'UTF8'
TABLESPACE = pg_default
```

```
LC_COLLATE = 'en_US.UTF-8'  
LC_CTYPE = 'en_US.UTF-8'  
CONNECTION LIMIT = 1000;
```

Adapt the highlighted values to match your environment.

Note: In Windows environments, instead of 'en_US.UTF-8', use:

```
LC_COLLATE = 'English_United States.1252'  
LC_CTYPE = 'English_United States.1252'.
```

3. (Optional) In order to verify that database objects can be created by the new login and user, connect to the server as **oouser** and perform the following:

```
CREATE TABLE TEST_TABLE(  
    TEST_COLUMN int NULL  
);  
  
INSERT INTO TEST_TABLE (TEST_COLUMN) VALUES ( 1 );  
INSERT INTO TEST_TABLE (TEST_COLUMN) VALUES ( 2 );
```

Verify that the table was created and that it contains two rows. You can now manually or otherwise drop the table.

Manually Creating Database Objects

Once the database and role are in place, the database objects (tables, indexes, and so on) are created when the HP OO service starts and connects to the database for the first time.

To manually create the database objects (instead of the HP OO service):

1. Extract the **postgres.sql** file from the HP OO DVD (or zip file) under **docs\sql**.
2. Connect to the HP OO database as the HP OO database user.
3. Run the **postgres.sql** file and verify that no errors occurred.

Postgres Database Maintenance

This section describes the various maintenance tasks that are recommended for HP OO databases created on Postgres, such as backing up the database, checking database integrity, handling index fragmentation, and monitoring the database.

This section includes:

- [Backing up the HP OO Database](#) 47
- [Creating a Maintenance Plan](#) 47

- [Utility for Purging Historical Data](#)48

Backing up the HP OO Database

You can back up a Postgres database using several tools, such as the `pg_dump` or `pg_backup` script. You can back up the HP OO database using any type of method/tool as long as the complete database is backed up.

Consider the following guidelines when you create your backup plan for HP OO:

Backup method:

The backup method depends mainly on business considerations—how much information “may” be lost? What is the maximum time for system recovery? If you need to be able to perform point-in-time recovery, and only “allow” a few hours-worth of data loss, use the full recovery model and perform full or differential backups daily, and transaction log backup every N hours depending on your business requirements.

If your organization is more tolerant to data loss, you can use the simple recovery model and perform a daily or weekly full backup.

Backup frequency:

Daily backup is recommended, especially if you are using/modifying HP OO on a daily basis.

You should back up once a month at the very least.

Timing:

Schedule backup for the time when HP OO is least active.

Retention:

Retention depends on your business guidelines and regulations.

Creating a Maintenance Plan

HP OO Postgres database maintenance mainly includes table REINDEX, as `autovacuum` needs to be activated. Use the example below, in order to keep the HP OO database in good shape.

Recommended Utility for Database Maintenance

In order to keep the HP OO database in good shape, it is recommended to run the REINDEX action during a system maintenance window.

Important! Note that this operation locks tables! Only perform this action during a maintenance window when the HP OO system is not operating!

Here is an example of how to REINDEX a complete database using the `reindexdb` utility:

```
reindexdb -d OO -U oouser -w ?????
```

Replace “OO” and “oouser” with actual HP OO database and user names.

It is recommended not to provide the password explicitly. See the Postgres documentation for recommendations on how to secure database passwords.

Utility for Purging Historical Data

HP OO 10.20 keeps all flow and step execution data in the database by default, enabling comprehensive debugging of previous flow runs. As a result, the database size grows in accordance to the system throughput and flow complexity. It is highly recommended to track your database size and make sure that old, irrelevant information is periodically purged.

Reclaiming free space in Postgres database requires two phases:

1. Space is marked as deleted – following a DELETE command.
2. A background “vacuum” process runs to allow free space to be reused.

Note that after the execution of the purge procedure described below, the vacuum process will usually spring into action and start “vacuuming” tables and indices.

It is important to allow the vacuum process to complete successfully before starting another purge. If the purge and vacuum overlap, they stall each other as they compete for object locks.

To install and use HP OO history purging stored procedure:

1. Download the latest **PostgreSQL.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.20** and unpack it.
2. Log in to Postgres database as “ouser” (the user created for HP OO).
3. Run the **OOPurgeHistory.sql** script and verify that the new function was created successfully.
4. Tune your stored function according to the comments embedded in the script.

The following example shows how this procedure may be used. See the guidelines provided as comments in the procedure header for detailed explanations.

```
SELECT OOPurgeHistory(90,10000,0,1,4,0);
```

Explanation about the above code:

- The **pPurgeExecutionsOlderThan** parameter determines how many days are kept (protected) relative to the time the procedure starts running. It is recommended to keep at least 90 days. Older data is deleted, starting with the oldest records.
- **pPurgeExecutionsInBatchesOf** determines the maximum number of flows to handle in each batch. The default value is 10000.
- **pShouldPurgeExecutionSummary** determines if the OO_EXECUTION_SUMMARY table should be purged. The default value is “0” (do not purge this table). It is recommended to keep data in this table as it does not consume a lot of space. Use “1” only if you want to completely remove any reference to the relevant flows.
- **pVerbose** determines verbosity level. “0” corresponds to “quiet” output, “1” corresponds to normal output, and “2” prints out detailed information.
- **pStopPurgingProcessAfter** is the timeout in hours for the operation to complete. Set it in accordance with your maintenance window boundaries if applicable.

- **pDeepClean** determines whether deep cleansing is performed, for example searching for “orphan” records that may bloat the database unnecessarily. The default is “0” (off). Note that setting this flag to “1” prolongs the procedure run time, but the timeout limit is still imposed.

Database Purging Flows

As of HP OO 10.20, database purging flows are available in the **HP Solutions** content pack under **Integrations/Hewlett-Packard/Operations Orchestration/10.x/Database**.

These flows enable you to clear old data from the database, based on time. For example, you can use the purging flows to clear data such as run inputs, outputs, stepLog, Studio debugger events, and auditing (if auditing is enabled).

It is recommended to deploy and schedule these flows to run periodically, in order to keep the database size under control and maintain optimal system performance.

Appendix A: Additional Guidelines for Microsoft SQL Server

This appendix contains additional guidelines relevant for HP OO deployment on Microsoft SQL Server.

Using Windows Authentication to Access Microsoft SQL Server Databases

Unless configured otherwise, HP OO uses Microsoft SQL Server authentication to access Microsoft SQL Server databases. Note that the HP OO installer currently does not support using Windows authentication during HP OO installation. However, Windows authentication can be used once HP OO is installed.

This appendix describes how to enable HP OO to use Windows authentication to access Microsoft SQL Server databases.

Configuring HP OO to Work with Windows Authentication

You can enable HP OO to use Windows authentication instead of Microsoft SQL Server authentication to access HP OO databases.

To enable HP OO to use Windows authentication to access a Microsoft SQL database:

1. Encrypt the Windows user password using the **encrypt-password.bat** utility located under **<OO installation>/central/bin** by running:

```
encrypt-password.bat --encrypt --password <password>
```

Save the generated string in order to use it in the next step.
2. Back up your current **database.properties** file located under **<OO installation>/central/conf** if you have an existing (usable) database connection.
3. Edit the **database.properties** file located under **<OO installation>/central/conf**, and change only the relevant parameter syntax to match the following example:

```
db.username=<USERNAME>  
jdbc.url=jdbc\:jtds\:\  
sqlserver\://<hostname>\:<port>/<db_name>;\
```

```
sendStringParametersAsUnicode\=true;\  
domain\<=<DOMAIN NAME>  
db.password=<the string generated by encrypt-password.bat>
```

Replace all the highlighted items with the correct values that match your environment.

Note that the **jdbc.url** parameter is broken into several lines using trailing backslash characters.

Appendix B: Additional Guidelines for Oracle

This appendix contains additional guidelines relevant for HP OO deployment on Oracle.

This appendix contains the configuration that needs to be done for HP OO to work with Oracle Real Application Cluster. This information is for advanced users only.

Note: The HP OO installer currently does not support using an Oracle RAC connection during the HP OO installation. During installation, regular connection properties must be provided. However, you can use the Oracle RAC connection once HP OO is installed.

This appendix includes:

- [Oracle Real Application Cluster \(RAC\)](#) 53
- [Single Client Access Name](#) 54
- [Configuring HP OO to Work with Oracle RAC](#) 54

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: OORAC
- Machine names: Server1, Server2
- On each machine, there is an Oracle instance of OORAC:
 - SID on Server1: OORAC1
 - SID on Server2: OORAC2
- 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 OORAC.

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.

The SCAN method is recommended when using HP OO in an Oracle 11g RAC environment.

Configuring HP OO to Work with Oracle RAC

To enable HP OO to connect to an Oracle RAC environment, complete the following steps:

1. Back up your current `database.properties` file located under `<OO installation>/central/conf` if you have an existing (usable) database connection.
2. Edit the `database.properties` file located under `<OO installation>/central/conf`, and change only the relevant parameter syntax to match the following example:

```
jdbc.url=jdbc:oracle:thin:@\  
(DESCRIPTION=\  
(LOAD_BALANCE=on)\  
(ADDRESS_LIST=\  
(ADDRESS=(PROTOCOL=TCP)(HOST= Server1-Vip)(PORT=1521))\  
(ADDRESS=(PROTOCOL=TCP)(HOST= Server2-Vip)(PORT=1521))\  
(CONNECT_DATA=(SERVICE_NAME= OORAC)))
```

Replace the highlighted items with the values that match your environment.

Note that the `jdbc.url` parameter is broken into several lines using trailing backslash characters.

Set the **Load Balancing** and **Failover** parameter values in accordance with your preferences.

When **Load Balancing** is on, **Failover** is on by default.

