



Hewlett Packard
Enterprise

HPE Operations Orchestration

Software Version: 10.60
Windows and Linux Operating Systems

Database Guide

Document Release Date: May 2016
Software Release Date: May 2016

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 2016 Hewlett Packard Enterprise Development LP

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.

Documentation Updates

To download the most recent edition of a document, go to <https://softwaresupport.hp.com>.

Contents

Introduction to Preparing the Database Environment	6
Overview	6
Using Database Clusters	7
Database Security	8
HPE OO Database Sizing	8
Hardware Requirements	9
Deploying and Maintaining a Microsoft SQL Server Database	11
Workflow for Microsoft SQL Server Deployment	11
System Requirements for Microsoft SQL Server	12
Hardware Requirements	12
Software Requirements	12
Examples of Tested Deployments	13
Language Support	13
Configuring SQL Server	14
Manually Creating an HPE OO Database on Microsoft SQL Server	16
Manually Creating Database Objects	18
Microsoft SQL Server Database Maintenance	19
Backing up the HPE OO Database	19
Creating a Maintenance Plan	20
Upgrade/Rollback Instructions	23
Database Server Upgrade from 2012 or 2008R2 to Server 2014	23
Always On Support	24
Deploying and Maintaining an Oracle Database	26
Workflow for Oracle Deployment	26
System Requirements for Oracle	27
Hardware Requirements	27
Software Requirements	27
Examples of Tested Deployments	27
Language Support	28
Configuring an Oracle Database	28
Manually Creating an HPE OO Database on an Oracle Instance	29
Manually Creating Database Objects	31
Connecting to Oracle Using SID or Service Name	31
Oracle Database Maintenance	32

Backing up the HPE OO Database	32
Creating a Maintenance Plan	32
Reclaiming LOB Space	35
Upgrade/Rollback Instructions	35
Deploying and Maintaining a MySQL Database	37
Workflow for MySQL Deployment	37
System Requirements for MySQL	37
Hardware Requirements	38
Software Requirements	38
MySQL Connector	39
Examples of Tested Deployments	39
Language Support	39
Configuring MySQL	39
Manually Creating an HPE OO Database on MySQL	41
Manually Creating Database Objects	43
MySQL Database Maintenance	43
Backing up the HPE OO Database	43
Creating a Maintenance Plan	44
Utility for Purging Historical Data	44
Upgrade/Rollback Instructions	45
Deploying and Maintaining a Postgres Database	46
Workflow for Postgres Deployment	46
System Requirements for Postgres	46
Hardware Requirements	47
Software Requirements	47
Examples of Tested Deployments	47
Language Support	48
Configuring Postgres	48
Manually Creating an HPE OO Database on Postgres	49
Manually Creating Database Objects	50
Postgres Database Maintenance	50
Backing up the HPE OO Database	51
Creating a Maintenance Plan	51
Utility for Purging Historical Data	52
Upgrade/Rollback Instructions	53
Database Purging Flows	54
Appendix A: Additional Guidelines for Microsoft SQL Server	55
Using Windows Authentication to Access Microsoft SQL Server Databases	55

Configuring HPE OO to Work with Windows Authentication	55
Configuring OO to Work with Always On	56
Appendix B: Additional Guidelines for Oracle	58
Oracle Real Application Cluster (RAC)	58
Single Client Access Name	59
Configuring HPE OO to Work with Oracle RAC	59
Appendix C: Installation Wizard 'Other database' Option	61
Microsoft SQL Server Named Instance Example	61
Microsoft SQL Server Windows Authentication Example	63
Oracle RAC Example	64

Introduction to Preparing the Database Environment

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

This chapter includes:

- [Overview](#) 6
- [Using Database Clusters](#) 7
- [Database Security](#) 8
- [HPE 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

HPE 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 HPE OO database on one of the following database server types:

- Microsoft SQL Server Standard/Enterprise (2008 R2/2012/2014)
- Oracle 11gR2 Standard/Enterprise Server, including RAC environment
- Oracle 12cR1 Standard/Enterprise Server - regular instance (non-CDB), including RAC environment
- Postgres (9.1/9.2/9.3/9.4/9.5)
- MySQL Community/Standard/Enterprise Server (5.5/5.6/5.7)

See the appropriate deployment chapter below for details:

- ["Deploying and Maintaining a Microsoft SQL Server Database" on page 11](#)
- ["Deploying and Maintaining an Oracle Database" on page 26](#)
- ["Deploying and Maintaining a MySQL Database" on page 37](#)
- ["Deploying and Maintaining a Postgres Database" on page 46](#)

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

Language Support

HPE 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 HPE 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 HPE 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 HPE OO database, see the database vendor's documentation and make sure you fully understand each action you perform as described in this guide.
- HPE 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 HPE 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 HPE OO system more robust, by protecting the database from several types of failures.

HPE 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 cluster environment capabilities, such as SCAN/AG listeners.

You can install HPE 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 HPE 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 A: Additional Guidelines for Microsoft SQL Server](#)" on page 55 for an Always On configuration.

See "[Appendix B: Additional Guidelines for Oracle](#)" on page 58 for an Oracle RAC example of a database URL configuration.

Database Security

The HPE OO database is at the heart of the HPE OO system, containing the HPE OO system configuration and possibly corporate-sensitive data. It is highly recommended that you enforce strict security settings on your database by following these guidelines.

Harden your database server by following the recommendations from the database vendor and the operating system vendor. Server hardening may include, among other items:

- Restrict SSH access on Linux servers to a well-controlled set of OS users
- Enforce a strict OS user password policy – password length, complexity, lockout-policy, and so on
- Set up an audit system to detect and report break-in attempts

Harden your database user/login accounts:

- Enforce the password policy – password length, complexity, lockout-policy, and so on
- Restrict access to administration-level accounts
- Set up an audit system to detect and report break-in attempts

Database encryption at file-system level is supported as long as it does not compromise OO's performance, and completely transparent to OO. For example, Oracle TDE may be used as long as it does not cause any performance degradation or prevent LOB free-space reclaim.

HPE OO Database Sizing

HPE OO 10.x 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.

In version 10.22 and later, the amount of data kept in the database can also be controlled by setting the system persistency level. See the *HPE OO Tuning Guide* for more details.

This section will help you prepare for installing HPE 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] – The term "payload" refers to flow/step data of considerable size. For example, a CSV or XML file used as flow input, a large JSON object / REST API call data.

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 HPE OO deployment scales:

System Scale\Parameter	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 HPE OO system and reasonable historical data retention—not including long-term database backups.

It is highly recommended to regularly purge historical data using the stored procedure published on HPE Live Network and the database purge flows provided in the HPE Solutions content pack, in order to keep the database size within your environment limitations.

- The amount of additional disk space required for keeping HPE 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 HPE 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](#) 11
- [System Requirements for Microsoft SQL Server](#) 12
 - [Hardware Requirements](#) 12
 - [Software Requirements](#) 12
 - [Examples of Tested Deployments](#) 13
 - [Language Support](#) 13
- [Configuring SQL Server](#) 14
- [Manually Creating an HPE OO Database on Microsoft SQL Server](#) 16
 - [Manually Creating Database Objects](#) 18
- [Microsoft SQL Server Database Maintenance](#) 19
 - [Backing up the HPE OO Database](#) 19
 - [Creating a Maintenance Plan](#) 20
 - [Upgrade/Rollback Instructions](#) 23
 - [Database Server Upgrade from 2012 or 2008R2 to Server 2014](#) 23
- [Always On Support](#) 24

Workflow for Microsoft SQL Server Deployment

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

1. **Review sizing guidelines.** For details, see "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 14](#).

4. **Create an OO database on Microsoft SQL Server.** For details, see ["Manually Creating an HPE OO Database on Microsoft SQL Server"](#) on page 16.
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 55. 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 HPE OO.

Hardware Requirements

For HPE OO database sizing guidelines and hardware requirements, see ["HPE 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 HPE OO:

Microsoft SQL Server Database Releases			
Version	Type	32/64Bit	Service Pack
2014	Standard	64 Bit	1
	Enterprise	64 Bit	1
2012	Standard	64 Bit	2
	Enterprise	64 Bit	2
2008 R2	Standard	x64	3
		x86	3
	Enterprise	x64	3
		x86	3

Only the listed service packs should be installed. Newer service packs are also supported unless stated otherwise in the *HPE 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 HPE quality assurance personnel.

Database Release			Operating System
Version	32/64Bit	Service Pack	
Microsoft SQL Server 2014 Enterprise Edition	64 Bit	1	Windows 2012 Standard Edition 64 Bit
Microsoft SQL Server 2012 Enterprise Edition	64 Bit	2	Windows 2012 Standard Edition 64 Bit
Microsoft SQL Server 2008 R2 Enterprise Edition	64 Bit	3	Windows 2012 Standard Edition 64 Bit

Language Support

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

Use one of the following collations depending on your HPE 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 HPE OO database collation to one of the collations above, in order to support future HPE 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 HPE 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 HPE 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, 2012, and 2014	
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 HPE OO)

[1] At the moment, the HPE OO 10.x installer supports only SQL authentication. Windows authentication can be configured afterward. See "[Appendix A: Additional Guidelines for Microsoft SQL Server](#)" on page 55.

Microsoft SQL Server 2008R2, 2012, and 2014			
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 4-12 GB depending on system scale according to the sizing guide</i>	

Microsoft SQL Server 2008R2, 2012, and 2014			
Database Options			
	Mandatory	Recommended	Supported
Collation	Any collation listed in the “Language Support” section in “System Requirements for Microsoft SQL Server” on page 12.		
Recovery Model		Full	Simple, Full
Allow Snapshot Isolation	True		
Is Read Committed Snapshot On	True		
Auto Shrink	False		
Auto Create Statistics	True		

Note: SQL Server does not use READ-COMMITTED transaction isolation by default. It is mandatory to set the **Allow Snapshot Isolation** and **Is Read Committed Snapshot On** flags. OO fails to operate properly using any other type of transaction isolation.

Manually Creating an HPE OO Database on Microsoft SQL Server

During HPE OO setup, a new database can be created automatically by the HPE 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 HPE 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 HPE 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 HPE OO installation. In such a case, you (or your organization’s DBA) should create the database, login, and user first, and then let the HPE 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 HPE 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, login, and user were created successfully. Adapt the highlighted values to match your environment.

Note: SQL scripts are also available in the Central server installation under the **central/bin/sql**

folder.

If you are copying SQL scripts from this document, note that you may need to remove redundant line breaks from the copied version.

```
USE [master]
GO

DECLARE
    @DBNAME          NVARCHAR(255),
    @SERVERLOGIN     NVARCHAR(255),
    @LOGINPASSWD    NVARCHAR(255),
    @DBUSER         NVARCHAR(255),
    @DBCOLLATION    NVARCHAR(255),
    @qry            NVARCHAR(4000);

SET @DBNAME =          N'OODB';
SET @SERVERLOGIN =     N'OODBLOGIN';
SET @LOGINPASSWD =    N'OODBComplexPa55WorD$';
SET @DBUSER =         N'OODBUSER';
SET @DBCOLLATION =    N'SQL_Latin1_General_CP1_CS_AS';

BEGIN

    SET @qry = 'USE [master];'
              + 'CREATE DATABASE [' + @DBNAME + '] COLLATE ' + @DBCOLLATION + ';'
              + 'ALTER DATABASE [' + @DBNAME + '] SET ALLOW_SNAPSHOT_ISOLATION ON;'
              + 'ALTER DATABASE [' + @DBNAME + '] SET READ_COMMITTED_SNAPSHOT ON;'
              + 'ALTER DATABASE [' + @DBNAME + '] SET AUTO_CREATE_STATISTICS ON;'
              + 'ALTER DATABASE [' + @DBNAME + '] SET AUTO_SHRINK OFF;';
    EXEC (@qry);

    SET @qry = 'USE [master];'
              + 'CREATE LOGIN [' + @SERVERLOGIN + '] WITH PASSWORD=N'''
              + @LOGINPASSWD + ''', DEFAULT_DATABASE=[' + @DBNAME + ']';
    EXEC (@qry);

    SET @qry = 'USE ' + @DBNAME + '; EXEC sp_executesql N'''
              + 'CREATE USER ' + @DBUSER + ' FOR LOGIN [' + @SERVERLOGIN + ']'''';
    EXEC (@qry);

    SET @qry = 'USE ' + @DBNAME + '; EXEC sp_executesql N'''
              + 'EXEC sp_addrolemember [db_owner], [' + @DBUSER + ']'''';
    EXEC (@qry);

END;
GO
```

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

```
USE [OODB]
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

4. (Optional) To verify that your newly created user has sufficient privileges to create the HPE OO database, connect to the database server using **OODBLOGIN** and perform the following:

```
USE [OODB]
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 HPE OO service starts and connects to the database for the first time.

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

It is highly recommended to grant the HPE OO database user DDL-related privileges, in order to allow 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 HPE OO installation 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 HPE 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 HPE 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 HPE OO Database](#) 19
- [Creating a Maintenance Plan](#) 20
- [Upgrade/Rollback Instructions](#) 23
- [Database Server Upgrade from 2012 or 2008R2 to Server 2014](#) 23

Backing up the HPE OO Database

Microsoft SQL Server databases are either configured for the **Full** recovery model, or the **Simple** recovery model. You can back up an HPE OO database using either one of these models. As HPE 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 HPE 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 HPE OO on a daily basis.

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

Timing:

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

Retention:

Retention depends on your business guidelines and regulations.

Creating a Maintenance Plan

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

Supplied Utilities for HPE OO Database Housekeeping

HPE 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.

Note: SQL scripts are also available in the Central server installation under the **central/bin/sql** folder.

If you are copying SQL scripts from this document, note that you may need to remove redundant line breaks from the copied version.

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 HPE 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 HPE OO is least active.

Utility for Maintaining Indexes and Statistics

Download the latest **MSSQL.zip** pack from HPE Live Network under **OO DB Maintenance Scripts and Procedures > HPE Operations Orchestration 10.x** and unpack it.

To install and use the HPE 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 HPE 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 HPE OO.

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

DECLARE
    @DBNAME NVARCHAR(255),
    @IDXFLTR NVARCHAR(270);

SET @DBNAME = N'OO';
SET @IDXFLTR = @DBNAME + '.dbo.%';

EXECUTE [dbo].[OOIndexMaintenance]
@DatabaseName = @DBNAME
,@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 = @IDXFLTR
,@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 two 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 HPE OO history purging stored procedure:

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

```
00PurgeHistory.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 HPE OO user.
3. Run the following T-SQL script and verify that the new objects were created successfully:

```
00PurgeHistory.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].[00PurgeHistory]
@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.

Upgrade/Rollback Instructions

Prior to any upgrade or rollback of HPE OO, take the following steps:

1. Purge non-essential data from the database.
Upgrade and rollback operations convert flow execution data. Less data means a shorter upgrade/rollback. If all flow execution data is essential, perform a complete database backup first and then purge.
2. Back up your database, preferably after the database was purged.
3. Schedule the operation to the database server’s quiet period.
4. Verify that no shared resources are exhausted – temporary space, transaction log, and similar resources.

Following any upgrade or rollback of HPE OO, take the following steps:

1. Install relevant versions of maintenance and purge stored procedures.
Stored procedures and purge flows are not upgraded/rolled back by the HPE OO upgrade utility.
2. Reschedule database maintenance and purge routines (stored procedures and/or purge flows).

Database Server Upgrade from 2012 or 2008R2 to Server 2014

If OO 10.60 database server is being upgraded from 2012 or 2008R2 to 2014, you must modify the OO Central server configuration in order to use a different JDBC connector jar.

SQL Server 2008R2 and 2012 are interfaced by OO using JTDS 1.3.0 connector, while 2014 is interfaced using Microsoft JDBC 4.2.

If your SQL Server database server is being upgraded to 2014:

1. Edit central server **database.properties** file located under the **central\conf** folder.
2. Modify the **db.driver** value as follows:
From:
net.sourceforge.jtds.jdbc.Driver
To:
com.microsoft.sqlserver.jdbc.SQLServerDriver
3. Modify the **db.url** value as follows:
From:
jdbc:jtds:sqlserver://**db_host:db_port/db_name**;sendStringParametersAsUnicode=true
To:

```
jdbc:sqlserver://db_host:db_port;databaseName=db_name;sendStringParametersAsUnicode=true
```

If you have edited your JDBC URL and used additional options, use Microsoft's JDBC 4.2 document in order to format the new URL correctly.

4. Repeat all above steps for each central server in the OO system.
5. Restart OO central servers.

If OO is being upgraded to 10.60 and the SQL Server database is being upgraded to 2014 simultaneously, follow these steps:

1. Complete the OO upgrade procedure using the current database (2008R2 or 2012).
Follow all the instructions in the sections above as usual – purge beforehand, backup, etc. It is highly recommended to perform a sanity test for the OO 10.60 system using the current database.
2. Make a backup of the OO 10.60 database (full, copy-only backup is preferred). Backup OO's login role, if necessary.
3. Upgrade the SQL Server database server to 2014.
4. Restore OO's database from the backup if necessary (optional).
If OO's database was upgraded along with the database server, there is no need to restore it.
Reconfigure the login role authentication and database ownership if necessary.
5. Perform the steps as described in "[Microsoft SQL Server Database Maintenance](#)" on page 19

At this point, you must configure Central servers to use Microsoft JDBC instead of JTDS.

Always On Support

Microsoft SQL Server 2008R2 and 2012 "Always On" is supported by HPE OO as a legacy client^[1] to implement a high availability and/or disaster recovery solution.

[1] HPE OO does not support the "MultiSubnetFailover" feature for SQL Server 2008R2 and 2012, and requires that **RegisterAllProvidersIP** is set to 0 in the cluster/listener configuration.

Connection to SQL Server 2014 "Always On" is supported including the "MultiSubnetFailover" feature, but have not been certified for HPE OO 10.60.

See "[Appendix A: Additional Guidelines for Microsoft SQL Server](#)" on page 55 for additional information.

High Availability Setup

As HPE OO requires a single point of contact, it is essential that an Availability Group Listener (AG listener) exist in the high availability setup. HPE OO's database connection is defined by a single URL - containing a single hostname.

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

In the case of a database failover scenario, while database connectivity is lost, HPE OO will repeatedly attempt to connect to the database until a new READ-WRITE capable instance accepts the connection. The redirection of the configured database hostname to a different IP address should be done by the AG Listener and cluster environment.

Note: HPE 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, HPE OO should be halted, reconfigured (edit the **database.properties** file), and then started again once the new database instance is available.

See the *HPE OO Installation, Upgrade, and Configuration Guide* for details on reconfiguring the database connection details.

Deploying and Maintaining an Oracle Database

In order to deploy HPE 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](#) 26
- [System Requirements for Oracle](#) 27
 - [Hardware Requirements](#) 27
 - [Software Requirements](#) 27
 - [Examples of Tested Deployments](#) 27
 - [Language Support](#) 28
- [Configuring an Oracle Database](#) 28
- [Manually Creating an HPE OO Database on an Oracle Instance](#) 29
 - [Manually Creating Database Objects](#) 31
 - [Connecting to Oracle Using SID or Service Name](#) 31
- [Oracle Database Maintenance](#) 32
 - [Backing up the HPE OO Database](#) 32
 - [Creating a Maintenance Plan](#) 32
 - [Reclaiming LOB Space](#) 35
 - [Upgrade/Rollback Instructions](#) 35

Workflow for Oracle Deployment

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

1. **Review sizing guidelines.** For details, see ["HPE 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 28.
4. **Create a Database.** For details, see ["Manually Creating an HPE OO Database on an Oracle Instance"](#) on page 29.
5. (Optional) **Connect HPE OO to an Oracle RAC environment.** For details, see ["Oracle Real Application Cluster \(RAC\)"](#) in ["Appendix B: Additional Guidelines for Oracle"](#) on page 58. This step is only relevant if you are using HPE OO in an Oracle RAC environment.

System Requirements for Oracle

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

Hardware Requirements

For HPE OO database sizing guidelines and hardware requirements, see "HPE 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 HPE OO:

Oracle Releases			
Version	Type	32/64Bit	Patch Set
11g R2	Standard	64 Bit	11.2.0.1 – 11.2.0.4 ^[1]
	Enterprise	64 Bit	11.2.0.1 – 11.2.0.4 ^[1]
12c R1 Regular instance, non-CDB	Standard	64 Bit	12.1.0.1 - 12.1.0.2
	Enterprise	64 Bit	12.1.0.1 - 12.1.0.2

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

[1] It is highly recommended to apply Oracle's "DATABASE PATCH SET UPDATE 11.2.0.4.6", which solves an issue with rollback to previous HPE OO versions.

See the Oracle documentation for supported platforms.

Note that Oracle 12c R1 is only supported in its regular, backward-compatible instance form. HPE OO 10.x does not support connecting to an Oracle 12c container database (CDB).

The Oracle 12c RAC environment is supported in backward-compatible form. New features such as multiple cluster subnets for SCAN listener are not supported.

Examples of Tested Deployments

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

Database Release			Operating System
Version	32/64Bit	Patch Set	
Oracle 11g R2 Enterprise Edition	64 Bit	11.2.0.4.6	Red Hat Enterprise Linux 6.5 64 Bit
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	12.1.0.2	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 HPE 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
UNDO_MANAGEMENT		AUTO	Automatic, Manual
Undo size	>= 4GB	6GB – 10GB	

Oracle Database 11gR2 / 12cR1			
Instance/Server Options			
Instance configuration options	Defaults, unless instructed otherwise		
	Mandatory	Recommended	Supported
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 HPE OO. If HPE 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 HPE OO. If HPE 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 HPE OO Database on an Oracle Instance

During HPE OO setup, a new database can be created automatically by the HPE 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 HPE 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 HPE 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 HPE OO installation. In such a case, you (or your organization’s DBA) should create the user (database) first, and then let the HPE 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.

Note: SQL scripts are also available in the Central server installation under the **central/bin/sql** folder.

If you are copying SQL scripts from this document, note that you may need to remove redundant line breaks from the copied version.

```
DEFINE OOUSER=OOUSER
DEFINE OOPASS=OOPA55wOrD
DEFINE DEFTBS=USERS
DEFINE TMPTBS=TEMP
CREATE USER &&OOUSER
IDENTIFIED BY &&OOPASS
DEFAULT TABLESPACE &&DEFTBS TEMPORARY TABLESPACE &&TMPTBS
QUOTA UNLIMITED ON &&DEFTBS ACCOUNT UNLOCK
;

GRANT CONNECT TO &&OOUSER;
GRANT CREATE VIEW, CREATE SEQUENCE, CREATE TABLE, CREATE PROCEDURE TO &&OOUSER;
GRANT SELECT ON ALL_USERS TO &&OOUSER;
```

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 HPE 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 HPE OO service starts and connects to the database for the first time.

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

It is highly recommended to grant the HPE OO database user DDL-related privileges, in order to allow HPE 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 HPE OO installation zip file under **docs\sql**.
2. Edit the **oracle.sql** file and prefix each object with the HPE OO user, to make sure the objects are created under the HPE OO user.
3. Connect to the HPE OO database as the relevant user, who is permitted to create and modify database objects in the HPE OO database.
4. Run the **oracle.sql** file and verify that no errors occurred, and that all objects are created and owned by the HPE 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. In version 10.20 and later, HPE 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 HPE 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 HPE OO Database](#) 32
- [Creating a Maintenance Plan](#) 32
- [Reclaiming LOB Space](#) 35
- [Upgrade/Rollback Instructions](#) 35

Backing up the HPE OO Database

Oracle databases can be backed up using several tools, such as **expdp** and **RMAN**. An HPE 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 HPE 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 HPE OO on a daily basis.

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

Timing:

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

Retention:

Retention depends on your business guidelines and regulations.

Creating a Maintenance Plan

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

Supplied Utilities for HPE OO Database Housekeeping

HPE 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.

Note: SQL scripts are also available in the Central server installation under the **central/bin/sql** folder.

If you are copying SQL scripts from this document, note that you may need to remove redundant line breaks from the copied version.

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 HPE 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 HPE OO is least active.

Utility for Maintaining Indexes and Statistics

Download the latest **Oracle.zip** pack from HPE Live Network under **OO DB Maintenance Scripts and Procedures > HPE Operations Orchestration 10.x** and unpack it.

To install and use the HPE 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 “OOUSER” (the user created for HPE OO).
3. Run the **HPE_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 SIZE 100000  
  
DECLARE x integer := 0;  
  
BEGIN  
    hpe_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 HPE OO.

Utility for Purging Historical Data

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

1. Log in to Oracle as “OOUSER” (the user created for HPE OO).
2. Run the **HPE_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
  hpe_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.

Utility for Reclaiming LOB Space

To install and use the HPE OO LOB space reclamation stored procedure:

1. Log in to Oracle as “OO” (the user created for HPE OO).
2. Run the **HPE_OO_DB_MAINT.sql** script (only if you have not run it already) and verify that the new package and procedures were created successfully.

The following example shows how this procedure may be used.

```
SET serveroutput ON SIZE 100000 DECLARE x integer := 0;  
BEGIN  
hpe_oo_db_maint.ForceLobShrink(x);  
END;
```

Notes:

Several conditions must be met in order for the shrink operation to succeed:

- TABLESPACE and database tables must not be compressed.
- ROW MOVEMENT must be enabled for all tables.
- TABLESPACE must be configured as AUTO segment space management.

See the Oracle documentation for the complete set of conditions.

Shrinking LOB segments of considerable size (tenths or hundreds of Giga-Bytes) may take some time to complete and may consume a considerable amount of database resources—CPU and I/O. Therefore, it is recommended to execute this process when the database server is least active.

Reclaiming LOB Space

Oracle databases are not quick when it comes to freeing space occupied by LOB segments. Depending on the configuration of the database server, LOB segment space may not be released immediately after purge operations. As LOB space may have a significant impact on the database size, you can add a post-purge operation to force freeing of LOB segments.

The general structure of the command is:

```
ALTER TABLE FOO MODIFY LOB ( LOB_COLUMN_NAME ) (shrink space);
```

The list of tables and columns involved may vary from one HPE OO version to another.

Note: The ability to shrink space depends on Tablespace definitions. See the Oracle documentation about reclaiming space.

The Oracle maintenance stored procedures package contains a stored procedure specifically for reclaiming LOB space: **hpe_oo_db_maint.ForceLobShrink**. See [Utility for Reclaiming LOB Space](#).

Upgrade/Rollback Instructions

Rolling back HPE OO 10.5x and later to previous versions requires patching Oracle 11gR2 database servers to 11.2.0.4.6. Apply Oracle’s “DATABASE PATCH SET UPDATE 11.2.0.4.6” prior to rollback operation.

Prior to any upgrade or rollback of HPE OO, take the following steps:

1. Purge non-essential data from the database.

Upgrade and rollback operations convert flow execution data. Less data means a shorter upgrade/rollback. If all flow execution data is essential, perform a complete database backup first and then purge.

2. Back up your database, preferably after the database was purged.
3. Schedule the operation to the database server's quiet period.
4. Verify that no shared resources are exhausted – temporary space, UNDO/REDO, and similar resources.

Following any upgrade or rollback of HPE OO, take the following steps:

1. Install relevant versions of maintenance and purge stored procedures.
Stored procedures and purge flows are not upgraded/rolled back by the HPE OO upgrade utility.
2. Reschedule database maintenance and purge routines (stored procedures and/or purge flows).

Deploying and Maintaining a MySQL Database

In order to deploy HPE 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](#) 37
- [System Requirements for MySQL](#) 37
 - [Hardware Requirements](#) 38
 - [Software Requirements](#) 38
 - [MySQL Connector](#) 39
 - [Examples of Tested Deployments](#) 39
 - [Language Support](#) 39
- [Configuring MySQL](#) 39
- [Manually Creating an HPE OO Database on MySQL](#) 41
 - [Manually Creating Database Objects](#) 43
- [MySQL Database Maintenance](#) 43
 - [Backing up the HPE OO Database](#) 43
 - [Creating a Maintenance Plan](#) 44
 - [Utility for Purging Historical Data](#) 44
 - [Upgrade/Rollback Instructions](#) 45

Workflow for MySQL Deployment

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

1. **Review sizing guidelines.** For details, see "HPE 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 39.
4. **Create HPE OO database on MySQL.** For details, see "Manually Creating an HPE OO Database on MySQL" on page 41.

System Requirements for MySQL

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

Hardware Requirements

For HPE OO database sizing guidelines and hardware requirements, see "HPE 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 HPE 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
5.7	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.

For MySQL 5.6.1 - 19, the recommendation is 256M, but for MySQL 5.6.20 - 21, the recommendation is 2GB.

See the MySQL documentation for supported platforms.

MySQL Connector

The MySQL connector jar is not provided with the OO installation. Download and place it under each HPE OO Central `<installation>/central/lib` folder.

Verify that the connector version you are using is fully compatible with your database server version. Currently, the only supported version is 5.1.35.

Upgrade Note: Because prior to version 10.60, MySQL JDBC 5.1.21 was in use, make sure to download and use 5.1.35 during the upgrade process.

Examples of Tested Deployments

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

Database Release			Operating System
Version	32/64Bit	Patch	
MySQL Server 5.6.19 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 HPE 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 and 5.7			
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-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		
innodb_log_file_size (for MySQL 5.6.22 - 5.7)	256M		
explicit-defaults-for-timestamp	0		
sql-mode	Default or none		
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	
binlog_format		row	
innodb_flush_log_at_trx_commit		2	
innodb_flush_method		O_DIRECT (Linux Only)	

MySQL 5.5, 5.6 and 5.7			
Instance/Server Options			
innodb_doublewrite		0	
MySQL 5.5, 5.6 and 5.7			
Other Options			
Server Configuration Options	Defaults, unless instructed otherwise		
	Mandatory	Recommended	Supported
[client]			
default-character-set	utf8		
[mysql]			
default-character-set	utf8		
[mysqldump]			
max_allowed_packet	250M		

- MySQL options may be written using underscore or hyphen in different contexts. Use the correct form according to your server version and usage context.
- Using none-default sql-mode options may result in unexpected behavior.

Manually Creating an HPE OO Database on MySQL

During HPE OO setup, a new database can be created automatically by the HPE 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 HPE 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 HPE 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 HPE OO installation. In such a case, you (or your organization’s DBA) should create the database first, and then let the HPE 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.

Note: SQL scripts are also available in the Central server installation under the **central/bin/sql** folder.

If you are copying SQL scripts from this document, note that you may need to remove redundant line breaks from the copied version.

```
SET @O0DB='O0DB';
SET @O0USER='O0USER';
SET @O0PASS='O0Pa55WorD';

SET @SQL1 = CONCAT('CREATE DATABASE IF NOT EXISTS `',@O0DB,`\` COLLATE utf8_bin');
SET @SQL2 = CONCAT('CREATE USER `',@O0USER,`\`@`%' IDENTIFIED BY
`',@O0PASS,`\`');
SET @SQL3 = CONCAT('GRANT ALL PRIVILEGES ON `',@O0DB,`\`.* TO `',@O0USER,`\`');
PREPARE stmt1 FROM @SQL1;
PREPARE stmt2 FROM @SQL2;
PREPARE stmt3 FROM @SQL3;
EXECUTE stmt1;
EXECUTE stmt2;
EXECUTE stmt3;
DEALLOCATE PREPARE stmt1;
DEALLOCATE PREPARE stmt2;
DEALLOCATE PREPARE stmt3;
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 **ouser** and perform the following:

```
USE O0;

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 HPE OO service starts and connects to the database for the first time.

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

It is highly recommended to grant the HPE OO database user DDL-related privileges, in order to allow HPE 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 HPE OO installation zip file under **docs\sql**.
2. Connect to the HPE OO database as the relevant user, permitted to create and modify database objects in the HPE 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 HPE 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 HPE OO Database](#) 43
- [Creating a Maintenance Plan](#) 44
- [Utility for Purging Historical Data](#) 44
- [Upgrade/Rollback Instructions](#) 45

Backing up the HPE OO Database

You can back up MySQL database using several tools, such as **mysqldump** or **mysqlbackup**. You can back up the HPE 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 HPE 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 HPE OO on a daily basis.

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

Timing:

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

Retention:

Retention depends on your business guidelines and regulations.

Creating a Maintenance Plan

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

Recommended Utility for Database Maintenance

In order to keep the HPE 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 HPE 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 HPE 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 HPE OO history purging stored procedure:

1. Download the latest **MySQL.zip** pack from HPE Live Network under **OO DB Maintenance Scripts and Procedures > HPE Operations Orchestration 10.x** and unpack it.
2. Log in to MySQL database as “ouser” (the user created for HPE OO).
3. Run the **OOPurgeHistory.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 OOPurgeHistory(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. 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.
- **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.

Upgrade/Rollback Instructions

Prior to any upgrade or rollback of HPE OO, take the following steps:

1. Purge non-essential data from the database.
Upgrade and rollback operations convert flow execution data. Less data means a shorter upgrade/rollback. If all flow execution data is essential, perform a complete database backup first and then purge.
2. Back up your database, preferably after the database was purged.
3. Schedule the operation to the database server's quiet period.
4. Verify that no shared resources are exhausted – temporary space, transaction log, and similar resources.
5. Make a backup copy of your MySQL connector jar file. Consider using a newer one for upgrade scenarios.

Following any upgrade or rollback of HPE OO, take the following steps:

1. Install relevant versions of maintenance and purge stored procedures.
Stored procedures and purge flows are not upgraded/rolled back by the HPE OO upgrade utility.
2. Reschedule database maintenance and purge routines (stored procedures and/or purge flows).

Deploying and Maintaining a Postgres Database

In order to deploy HPE 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	46
• System Requirements for Postgres	46
• Hardware Requirements	47
• Software Requirements	47
• Examples of Tested Deployments	47
• Language Support	48
• Configuring Postgres	48
• Manually Creating an HPE OO Database on Postgres	49
• Manually Creating Database Objects	50
• Postgres Database Maintenance	50
• Backing up the HPE OO Database	51
• Creating a Maintenance Plan	51
• Utility for Purging Historical Data	52
• Upgrade/Rollback Instructions	53

Workflow for Postgres Deployment

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

1. **Review sizing guidelines.** For details, see "[HPE 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 48.
4. **Create HPE OO database on Postgres.** For details, see "[Manually Creating an HPE OO Database on Postgres](#)" on page 49.

System Requirements for Postgres

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

Hardware Requirements

For HPE OO database sizing guidelines and hardware requirements, see "HPE 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 HPE 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
9.4	x86 32-bit
	x86 64-bit
9.5	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 HPE 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

Database Release		Operating System
Version	32/64Bit	
Postgres 9.3.2	64 Bit	Red Hat Enterprise Linux 6.3 64 Bit
Postgres 9.4.6	64 Bit	Red Hat Enterprise Linux 7 64 Bit
Postgres 9.5.1	64 Bit	Red Hat Enterprise Linux 7 64 Bit

Language Support

Postgres determines character set and collation at the database level. The HPE 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

Postgres 9.1 - 9.3			
Instance/Server Options			
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 HPE OO Database on Postgres

During HPE OO setup, a new database can be created automatically by the HPE 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 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 HPE 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 HPE OO installation. In such a case, you (or your organization’s DBA) should create the database, login, and user first, and then let the HPE 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.

Note: SQL scripts are also available in the Central server installation under the **central/bin/sql** folder.

If you are copying SQL scripts from this document, note that you may need to remove redundant line breaks from the copied version.

```
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 HPE OO service starts and connects to the database for the first time.

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

1. Extract the **postgres.sql** file from the HPE OO installation zip file under **docs\sql**.
2. Connect to the HPE OO database as the HPE 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 HPE 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 HPE OO Database](#)51
- [Creating a Maintenance Plan](#) 51
- [Utility for Purging Historical Data](#)52
- [Upgrade/Rollback Instructions](#) 53

Backing up the HPE 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 HPE 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 HPE 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 HPE OO on a daily basis.

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

Timing:

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

Retention:

Retention depends on your business guidelines and regulations.

Creating a Maintenance Plan

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

Recommended Utility for Database Maintenance

In order to keep the HPE 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 HPE 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 HPE 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

HPE OO 10.x 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.

“Vacuum Full” Operation

PostgreSQL database is only able to reclaim Large Objects (LOBs) space when the **Vacuum Full** operation is run. A considerable amount of HPE OO's database space is stored as LOBs. Therefore, it is important to schedule a routine **Vacuum Full** operation - at times in which OO is least active. The **Vacuum Full** operation requires *exclusive table locks*, and therefore may interfere with HPE OO's activity.

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

1. Download the latest **PostgreSQL.zip** pack from HPE Live Network under **OO DB Maintenance Scripts and Procedures > HPE Operations Orchestration 10.x** and unpack it.
2. Log in to Postgres database as “oouser” (the user created for HP
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. 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.
- **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.

Upgrade/Rollback Instructions

Prior to any upgrade or rollback of HPE OO, take the following steps:

1. Purge non-essential data from the database.
Upgrade and rollback operations convert flow execution data. Less data means a shorter upgrade/rollback. If all flow execution data is essential, perform a complete database backup first and then purge.
2. Verify that you have sufficient disk space to support a "vacuum full" operation, and initiate a "vacuum full" operation.
3. Back up your database, preferably after the database was purged.
4. Schedule the operation to the database server's quiet period.
5. Verify that sufficient free disk space is available for transaction log files.

Following any upgrade or rollback of HPE OO, take the following steps:

1. Install the relevant version of the purging stored procedure.
Stored procedures and purge flows are not upgraded/rolled back by the HPE OO upgrade utility.
2. Reschedule the database purge routines (stored procedure and/or purge flows).
3. Reschedule the routine database "vacuum full" operations.

Database Purging Flows

In HPE OO 10.20 and later, database purging flows are available in the **HPE 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 HPE OO deployment on Microsoft SQL Server.

Using Windows Authentication to Access Microsoft SQL Server Databases

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

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

Configuring HPE OO to Work with Windows Authentication

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

To enable HPE 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.

Note: When copying scripts, note that you may need to remove redundant line breaks from the copied version.

For SQL Server 2008R2 and 2012:

```
db.username=[USERNAME]  
db.password=[the string generated by encrypt-password.bat]
```

```
jdbc.url=jdbc\:jtds\:sqlserver\://[DB_HOSTNAME]\:[PORT]/[DB_NAME]  
;sendStringParametersAsUnicode=true  
;domain\=[DOMAIN_NAME]
```

For SQL Server 2014:

```
db.username=[USERNAME]  
db.password=[the string generated by encrypt-password.bat]
```

```
jdbc.url=jdbc\:sqlserver\://[DB_HOSTNAME]\:[PORT];databaseName\=[DB_  
NAME];sendStringParametersAsUnicode=true;integratedSecurity=true;
```

For SQL server 2014, there might be a need to copy sqljdbc_auth.dll to the system path of the central server machine. For more details please refer to Microsoft JDBC documentation.

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

Note that the **jdbc.url** parameter is broken in this document into several lines for readability. When copying this example onto your **database.properties** file, remove all line breaks so the **jdbc.url** parameter is composed of a single line containing no white spaces.

Configuring OO to Work with Always On

For SQL Server 2008R2 and 2012

HPE OO connects to SQL Server 2008R2 and 2012 databases using the jtds connector. Thus, it is only able to register a single IP address for the Availability Group Listener. In addition, HPE OO does not support Multi-Subnet Failover.

Cluster Configuration

Configure your cluster environment as follows:

1. Set the cluster option **RegisterAllProvidersIP** to 0 for the network name associated with the Availability Group Listener.
2. Optionally, decrease the **HostRecordTTL** value for the same cluster resource to reduce database reconnection time.

See the Microsoft SQL Server Documentation (“Create or Configure an Availability Group Listener”) for a complete description of configuration options and examples.

Installing HPE OO using Availability Group Listener

Prior to installing HPE OO, verify that the Availability Group Listener network name resolves to an accessible IP address from all Central machines.

1. In the HPE OO installation wizard **Database Connection Configuration** page, enter the Availability Group Listener network name in the **Hostname or IP address** field.
2. Enter the rest of the details and click the **Test Connection** button.

Providing the Availability Group Listener network name guarantees that HPE OO's Central server(s) will be able to reconnect to the database in the case of a failure.

For silent installation, use:

```
db.url=jdbc:jtds:sqlserver://[AG-NET-NAME]:[PORT]/[DB-NAME];sendStringParametersAsUnicode=true
```

For SQL Server 2014

HPE OO connects to SQL Server 2014 databases using the Microsoft JDBC 4.2 connector.

Here is an example on how to format JDBC URL in order to connect to an Always On cluster:

```
db.url=jdbc:sqlserver://[AG-NET-NAME];instanceName=[NAMED-INST-NAME];  
databaseName=[DB-NAME];multiSubnetFailover=true;applicationIntent=ReadWrite;  
sendStringParametersAsUnicode=true
```

Note that **multiSubnetFailover** and **applicationIntent** are added in order to allow more complex behaviors.

Note that Multi-Subnet Failover scenarios in conjunction with SQL Server 2014 Always On cluster were not certified with HPE OO 10.60.

For additional information and connection options, refer to "JDBC Driver Support for High Availability, Disaster Recovery" in the Microsoft documentation and "Setting the Connection Properties" in the Microsoft JDBC documentation.

Appendix B: Additional Guidelines for Oracle

This appendix contains the configuration that needs to be done for HPE OO to work with Oracle 11gR2 and 12cR1 Real Application Cluster (RAC) environments. This information is for advanced users only.

This appendix includes:

- [Oracle Real Application Cluster \(RAC\)](#) 58
- [Single Client Access Name](#) 59
- [Configuring HPE OO to Work with Oracle RAC](#) 59

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 Cluster name: OORAC
- Service Name: ORCL.MY.DOMAIN
- 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.

- SCAN listener uses a virtual IP, usually published using DNS/GNS:
SCAN-Vip
- The local listeners on both servers listen on the default port 1521 and support the database service

OORAC. SCAN listener(s) reside on one of the cluster nodes, and may failover along with their virtual IP addresses in the case of a failure.

Note: Although 12c SCAN allows configuring multiple SCAN listeners (per subnet), HPE OO only supports connecting to a single 12c SCAN listener.

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. 11g SCAN addresses, virtual IP addresses, and public IP addresses must all be on the same subnet. 12c SCAN allows configuring multiple SCAN listeners – one per subnet. HPE OO only supports connecting to a single 12c SCAN listener.

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

Configuring HPE OO to Work with Oracle RAC

Connecting to a SCAN Listener Virtual IP

1. In the HPE OO installation wizard **Database Connection Configuration** page, enter the SCAN listener virtual IP address/network name in the **Hostname or IP address** field.
2. Select the **Service Name** radio button instead of the **SID** radio button and enter the Oracle RAC service name.
3. Enter the rest of the details and click the **Test Connection** button.

Providing the SCAN listener virtual IP address guarantees that HPE OO's Central server(s) will be able to reconnect to the database cluster in the case of a failure.

This connection method is the preferred one as it enables the HPE OO installation wizard to create the database as well as populate it.

For silent installation use (replace the highlighted text with the actual values):

```
db.url=jdbc:oracle:thin:@//[SCAN-vip]:[PORT]/[ORCL.MY.DOMAIN]
```

Load-Balancing Using Explicit Connection String

Although using Oracle's SCAN listener virtual IP is the preferred method, you can also provide an explicit connection string by using the **Other database** installation option. For complete details, see "Oracle RAC Example" in ["Appendix C: Installation Wizard 'Other database' Option" on page 61](#).

Here is an example of how a load balancing connection string would appear in Central's **database.properties** file.

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 as follows, while replacing the highlighted items with the values that match your environment.

Note: When copying scripts, note that you may need to remove redundant line breaks from the copied version.

```
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=ORCL.MY.DOMAIN)))
```

Note that the **jdbc.url** above is broken into several lines for readability, while it should appear as a single line in the actual configuration file.

When Load Balancing is set to "on", failover between listeners is active by default.

Appendix C: Installation Wizard 'Other database' Option

This appendix contains information about the **Other database** option in the HPE OO installation wizard.

This option enables using specific JDBC driver and connection options. Use this option if:

- You want to use a different version of the JDBC driver, other than the ones provided in the HPE OO installation (see the note below).
- You want to provide a JDBC connection URL yourself, to include an option or options which are not currently provided by the standard database connection options.

NOTES:

- Connection is restricted to the types and versions of database described in the "Overview" section in ["Introduction to Preparing the Database Environment" on page 6](#).
- Use a non-provided JDBC driver at your own risk. The following JDBC drivers are supported:
 - jtids-1.3.0.jar – for SQL Server 2008R2 and 2012
 - sqljdbc4-4.2.jar – for SQL Server 2014
 - ojdbc7-12.1.0.2.jar
 - postgresql-9.4.1207.jar
 - mysql-connector-java-5.1.35 (not provided in the HPE OO installation)

When you install HPE OO using the **Other database** option, the installation wizard does not create the database and/or related user or roles. You must create these beforehand, using the installation wizard. Detailed instructions on how to create the database/user/role can be found in this document, under the "Manually Creating an HPE OO Database" section in each database chapter.

Microsoft SQL Server Named Instance Example

SQL Server 2008R2 and 2012

The following example show a connection to a Microsoft SQL Server Named Instance using jtids JDBC Connector. The database, login role, and user are pre-created by the DBA, and the login role is the owner of the database (has full DML and DDL privileges).

Use the following details in the installation wizard, and revise the highlighted values to match your environment.

Use JDBC URL Option #1 when a unique TCP port is assigned to each instance.

Use JDBC URL Option #2 when **Database Browser :service** is active and is able to route connections based on the instance name. Note that TCP port number does not appear in this option.

Field	Value	Comments
JDBC Driver jar	C:\my\path\jtds-1.3.0.jar	
JDBC Driver class name	net.sourceforge.jtds.jdbc.Driver	
JDBC URL Option #1	jdbc:jtds:sqlserver://<DB_IP_OR_HOSTNAME>:<DB_PORT>/<DB_NAME>;instance=<INSTANCE_NAME>;sendStringParametersAsUnicode=true	
JDBC URL Option #2	jdbc:sqlserver://<DB_IP_OR_HOSTNAME>:<INSTANCE_PORT>;databaseName=<DB_NAME>;sendStringParametersAsUnicode=true	
Username	<LOGIN_ROLE>	
Password	<LOGIN_ROLE_PASSWORD>	

SQL Server 2014

The following example show a connection to a Microsoft SQL Server Named Instance using Microsoft JDBC Connector. The database, login role, and user are pre-created by the DBA, and the login role is the owner of the database (has full DML and DDL privileges).

Use the following details in the installation wizard, and revise the highlighted values to match your environment.

Use JDBC URL Option #1 when a unique TCP port is assigned to each instance.

Use JDBC URL Option #2 when Database Browser :service is active and is able to route connections based on the instance name. Note that TCP port number does not appear in this option.

Field	Value	Comments
JDBC Driver jar	C:\my\path\sqljdbc4-4.2.jar	
JDBC Driver class name	net.sourceforge.jtds.jdbc.Driver	
JDBC URL Option #1	jdbc:sqlserver://<DB_IP_OR_HOSTNAME>:<INSTANCE_PORT>;databaseName=<DB_NAME>;sendStringParametersAsUnicode=true	INSTANCE_PORT represents the separate instance
JDBC URL Option #2	jdbc:sqlserver://<DB_IP_OR_HOSTNAME>;instanceName=<INSTANCE_NAME>;databaseName=<DB_NAME>;sendStringParametersAsUnicode=true	If Database Browser service is active
Username	<LOGIN_ROLE>	
Password	<LOGIN_ROLE_PASSWORD>	

Microsoft SQL Server Windows Authentication Example

SQL Server 2008R2 and 2012

The following example shows a connection to a Microsoft SQL Server using Windows Authentication and jtds JDBC connector. The database is pre-created by the DBA, and the Windows login account is the owner of the database (has full DML and DDL privileges).

Use the following details in the installation wizard, and revise the highlighted values to match your environment:

Field	Value	Comments
JDBC Driver jar	C:\my\path\jtds-1.3.0.jar	Use jtds JDBC driver only
JDBC Driver class name	net.sourceforge.jtds.jdbc.Driver	
JDBC URL	jdbc:jtds:sqlserver://<DB_IP_OR_HOSTNAME>:<DB_PORT>/<DB_NAME>;domain=<DOMAIN_NAME>;sendStringParametersAsUnicode=true	
Username	<WINDOWS_USERNAME>	
Password	<WINDOWS_USERNAME_PASSWORD>	

Note: Ignore the error regarding incompatible collation, but make sure to set your collation to one of the supported collations described in the "Language Support" section in ["Introduction to Preparing the Database Environment" on page 6](#).

It is recommended but not required, that HPE OO Central service(s) is run as the same user that is used for database authentication. Note that using a domain account for authentication may enforce security policies that may change from time to time by the domain administrator.

Whenever the domain account password changes, the new password must be encrypted and the new encrypted password must be placed in the **database.properties** file located under **<OO installation>/central/conf**. If an HPE OO cluster is used, this operation must be repeated for all HPE OO central servers.

SQL Server 2014

The following example show connection to a Microsoft SQL Server 2014 using Windows Authentication and Microsoft JDBC Connector.

Prerequisites:

1. The database is pre-created by the DBA
2. The relevant Windows account is the owner of the database (has full DML and DDL privileges)
3. HPE OO's installation wizard must be unzipped and extracted to a folder form

4. **sqljdbc_auth.dll** must be downloaded and copied to OO's installation wizard under **<installer_folder>\java\bin**. This dll file is not provided along with HPE OO's installation and is available for download from Microsoft web site. Download Microsoft JDBC 4.2 and extract 64bit version of **sqljdbc_auth.dll**.

Installation steps:

1. Run HPE OO's installer by executing **<installer_folder>\installer.bat** as the same Windows account that owns the OO database.
2. In the **Connectivity** page check the **Do not start central server after installation...** checkbox.
3. In the **Database Connection** page select the **Other database** option and fill in the details according to the table below. Edit and replace highlighted parts with actual values.
4. After the installation:
 - a. Verify **sqljdbc_auth.dll** is located in **<central installation>\java\bin**.
 - b. Configure the service to run under the same Windows account that owns the OO database.
 - c. Start HPE OO's service and verify successful connection to the database.

Repeat the steps above for all central server installations

Use the following details in the installation wizard, and revise the highlighted values to match your environment:

Field	Value	Comments
JDBC Driver jar	C:\my\path\sqljdbc4-4.2.jar	
JDBC Driver class name	com.microsoft.sqlserver.jdbc.SQLServerDriver	
JDBC URL	jdbc:sqlserver://<DB_IP_OR_HOSTNAME>:<DB_PORT>;databaseName=<DB_NAME>;integratedSecurity=true;sendStringParametersAsUnicode=true	
Username	Fully qualified Windows account: DOMAIN\USERNAME	
Password	---- keep empty ----	

Note: Ignore the error about incompatible collation, but make sure to set your collation to one of the supported collations described in the "Language Support" section in ["Introduction to Preparing the Database Environment" on page 6](#).

Note that using a domain account for authentication may enforce security policies that may change from time to time by the domain administrator.

Oracle RAC Example

The following example shows a connection to an Oracle RAC cluster using explicit JDBC URL. The database (user) must be pre-created by the DBA, and must be granted DML and DDL privileges.

Use the following details in the installation wizard, and revise the highlighted values to match your environment:

Field	Value	Comments
JDBC Driver jar	C:\my\path\ojdbc7-12.1.0.2.jar	
JDBC Driver class name	oracle.jdbc.OracleDriver	
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=RAC1.MY.DOMAIN)))	
Username	<PRE-CREATED-DATABASE>	
Password	< PRE-CREATED-DATABASE-PASSWORD>	

