











# Introduction to Preparing the Database Environment

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

This chapter includes:

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

## Overview

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

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

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

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

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

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

See the appropriate deployment chapter below for details:

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

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





































# Deploying and Maintaining an Oracle Database

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

This chapter includes the following sections:

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

## Workflow for Oracle Deployment

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

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





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

## Language Support

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

## Configuring an Oracle Database

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

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

Legend:

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

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



# Manually Creating an HP OO Database on an Oracle Instance

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

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

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

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

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

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

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

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

To create a database:

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

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













# Deploying and Maintaining a MySQL Database

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

This chapter includes the following sections:

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

## Workflow for MySQL Deployment

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

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

## System Requirements for MySQL

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

## Hardware Requirements

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

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

## Software Requirements

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

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

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

See the MySQL documentation for supported platforms.

## Examples of Tested Deployments

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

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

## Language Support

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

## Configuring MySQL

This section contains information on MySQL and database configuration settings.

Legend:

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

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



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

## Manually Creating an HP OO Database on MySQL

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

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

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

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

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

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

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

To create a database:

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

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

Adapt the highlighted values to match your environment.

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

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

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

## Manually Creating Database Objects

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

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

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

To manually create the database objects:

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

# MySQL Database Maintenance

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

This section includes:

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

## Backing up the HP OO Database

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

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

### **Backup method:**

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

### **Backup frequency:**

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

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

### **Timing:**

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

### **Retention:**

Retention depends on your business guidelines and regulations.

## Creating a Maintenance Plan

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

### **Recommended Utility for Database Maintenance**

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

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

Here is an example of how to run this utility:

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

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

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

## Utility for Purging Historical Data

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

1. Download the latest **MySQL.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.21** and unpack it.
2. Log in to MySQL database as "ouser" (the user created for HP OO).
3. Run the **00PurgeHistory.sql** script and verify that the new procedure was created successfully.
4. Tune your stored procedures according to the comments embedded in the script.

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

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

Explanation about the above code:

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







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

## Hardware Requirements

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

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

## Software Requirements

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

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

Only supported versions should be used.

See the Postgres documentation for supported platforms.

## Examples of Tested Deployments

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

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

## Language Support

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

## Configuring Postgres

This section contains information on Postgres configuration settings.

Legend:

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

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

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

# Manually Creating an HP OO Database on Postgres

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

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

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

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

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

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

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

To create a database:

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

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

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

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

Adapt the highlighted values to match your environment.

**Note:** In Windows environments, instead of 'en\_US.UTF-8', use:

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

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

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

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

## Manually Creating Database Objects

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

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

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

## Postgres Database Maintenance

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

This section includes:

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

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

## Backing up the HP OO Database

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

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

### Backup method:

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

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

### Backup frequency:

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

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

### Timing:

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

### Retention:

Retention depends on your business guidelines and regulations.

## Creating a Maintenance Plan

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

### Recommended Utility for Database Maintenance

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

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

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

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

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

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

## Utility for Purging Historical Data

HP OO 10.21 keeps all flow and step execution data in the database by default, enabling comprehensive debugging of previous flow runs. As a result, the database size grows in accordance to the system throughput and flow complexity. It is highly recommended to track your database size and make sure that old, irrelevant information is periodically purged.

Reclaiming free space in Postgres database requires two phases:

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

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

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

To install and use HP OO history purging stored procedure:

1. Download the latest **PostgreSQL.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.21** and unpack it.
2. Log in to Postgres database as “ouser” (the user created for HP OO).
3. Run the **OOPurgeHistory.sql** script and verify that the new function was created successfully.
4. Tune your stored function according to the comments embedded in the script.

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

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

Explanation about the above code:

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



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

## Database Purging Flows

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

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

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

# Appendix A: Additional Guidelines for Microsoft SQL Server

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

## Using Windows Authentication to Access Microsoft SQL Server Databases

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

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

## Configuring HP OO to Work with Windows Authentication

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

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

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

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

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

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

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

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

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

# Appendix B: Additional Guidelines for Oracle

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

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

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

This appendix includes:

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

## Oracle Real Application Cluster (RAC)

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

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

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

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

- Oracle RAC database name: OORAC
- Machine names: Server1, Server2
- On each machine, there is an Oracle instance of OORAC:
  - SID on Server1: OORAC1
  - SID on Server2: OORAC2
- On each machine, there is a virtual IP (Server1-Vip and Server2-Vip):

- Server1-Vip is assigned to Server1
- Server2-Vip is assigned to Server2

The virtual IP is in addition to the static IP assigned to the machine.

- The listeners on both servers are listening on the default port 1521 and support the database service OORAC.

## Single Client Access Name

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

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

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

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

## Configuring HP OO to Work with Oracle RAC

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

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

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

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

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

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

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

