

# HP Operations Orchestration

Software Version: 10.20

Windows and Linux Operating Systems

## Database Maintenance Procedures

Document Release Date: November 2014

Software Release Date: November 2014



## Legal Notices

### Warranty

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

The information contained herein is subject to change without notice.

### Restricted Rights Legend

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

### Copyright Notice

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

### Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

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

UNIX® is a registered trademark of The Open Group.

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

## Documentation Updates

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

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

To check for recent updates or to verify that you are using the most recent edition of a document, go to: <http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: <http://h20229.www2.hp.com/passport-registration.html>

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

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

## Support

Visit the HP Software Support Online web site at: <https://softwaresupport.hp.com/>

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

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

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

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

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

To find more information about access levels, go to:

[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)

**HP Software Solutions Now** accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is <http://h20230.www2.hp.com/sc/solutions/index.jsp>

# Contents

- HP OO Database Maintenance Procedures ..... 5
- Microsoft SQL Server Database Maintenance ..... 6
  - Procedure for Maintaining Indexes and Statistics ..... 6
  - Procedure for Purging Historical Data ..... 8
- Oracle Maintenance Procedures .....10
  - Procedure for Maintaining Indexes and Statistics ..... 10
  - Procedure for Purging Historical Data ..... 11
- MySQL Maintenance Procedures .....13
  - Procedure for Purging Historical Data ..... 13
- Postgres Maintenance Procedures .....15
  - Procedure for Purging Historical Data ..... 15

# HP OO Database Maintenance Procedures

HP OO database maintenance tasks include:

- Updating statistical information about indexes and tables
- Rebuilding indexes
- Purging historical data
- Reclaiming free space

Use the procedures and other tools described in the following chapters in order to keep your HP OO database in good shape.

To update statistics and maintain indexes, you can also use other tools in accordance with your company policy. For data purging, it is highly recommended to use only HP provided procedures in order not to compromise database structure or data.

## Database Size

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

## Database Guide Document

This document only specifies instructions for installing database maintenance procedures. See the *HP OO 10.20 Database Guide* for detailed instructions on how to configure your HP OO database.

# Microsoft SQL Server Database Maintenance

This chapter provides information on how to install and use HP OO 10.20 Microsoft SQL Server database maintenance procedures. These procedures are applicable for Microsoft SQL Server 2008R2 and 2012.

## Procedure for Maintaining Indexes and Statistics

Download the latest **MSSQL.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.20** and unpack it.

To install and use the HP OO maintenance stored procedures:

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

```
USE [master]
GO

GRANT VIEW SERVER STATE TO oouser
GO
```

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

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

- **OO\_DB\_MAINTENANCE\_LOG.sql** (optional)
- **OOCmdExec.sql**
- **OOIndexMaintenance.sql**

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

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

- **OO\_DB\_MAINTENANCE\_LOG.sql** (optional)
- **OOCmdExec.sql**
- **OOIndexMaintenance.sql**

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

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

```
USE [OO]
GO

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

Explanation of the above code:

- Replace “OO” with the actual name of your database. Note that there are three occurrences.
- The **@FragmentationXXX** parameters set the script’s fragmentation level sensitivity and course of action in each case. These threshold levels and subsequent actions are recommended by Microsoft’s documentation. Tune these values with caution.
- **@SortInTempdb** (once set to ‘Y’) lets you perform sorting operations during index reorganization/rebuild in **tempdb** rather than in memory, for better performance. If you choose to use this option, make sure you have sufficient free space in **tempdb**.
- **@Indexes** is a filter for including/excluding indexes in the maintenance operation. It is recommended to keep this filter as is, to analyze all indexes.
- **@TimeLimit** is the timeout in seconds for the maintenance operation to complete. Set it in accordance with your maintenance window boundaries, if applicable.

- **@LockTimeout** is the timeout in seconds to wait for object lock. Once expired, the specific operation fails and the procedure continues to the next object.
- **@LogToTable** determines whether maintenance operation results should be logged to a table. This lets you keep track of the maintenance operations and helps procedure debugging.
- **@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.

## Procedure for Purging Historical Data

Download the latest **MSSQL.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.20** and unpack it.

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

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

OOGetErrorInfo.sql

OOPurgeHistory.sql

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

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

OOGetErrorInfo.sql

OOPurgeHistory.sql

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

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

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



```
,@AdvancedMode = 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 record.
- **@PurgeExecutionsInBatchesOf** determines how many flows are handled together. Smaller values imply smaller, more frequent transactions, and higher values imply less frequent, larger transactions. 10,000 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 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.
- **@AdvancedMode** 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.

## Oracle Maintenance Procedures

This chapter provides information on how to install and use HP OO 10.20 Oracle database maintenance procedures. These procedures are applicable for Oracle 11g R2.

### Procedure for Maintaining Indexes and Statistics

Download the latest **Oracle.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.20** and unpack it.

To install and use the HP OO maintenance stored procedures:

1. Log in to Oracle as “system” or any other user with a DBA role, and run the following commands. These system privileges are required in order to 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 00;  
GRANT ANALYZE ANY TO 00;  
GRANT ALTER ANY INDEX TO 00;
```

Adapt the highlighted user name to match your environment.

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

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

```
SET serveroutput ON size 100000  
  
DECLARE x integer := 0;  
  
BEGIN  
    hp_oo_db_maint.IndexMaintenance(3, 15, 1, x);  
END;
```

Explanation regarding the above code (parameters reference values - left to right):

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

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

## Procedure for Purging Historical Data

To install and use HP OO history purging stored procedure:

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

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

```
SET serveroutput ON SIZE 100000

DECLARE x integer := 0;

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

```
END;
```

Explanation about the above code (parameters reference values - left to right):

- The **pPurgeExecutionsOlderThan** 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.

- **pPurgeExecutionsInBatchesOf** determines how many flows are handled together. Smaller values imply smaller, more frequent transactions, and higher values imply less frequent, larger transactions. 10,000 is recommended for most systems.
- **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.
- **pAdvancedMode** 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.

## MySQL Maintenance Procedures

This chapter provides information on how to install and use HP OO 10.20 MySQL database maintenance procedures. These procedures are applicable for MySQL 5.5 – 5.6.

The user created for HP OO should have “ALL PRIVILEGES” granted on the HP OO database. If this is not the case, grant “CREATE ROUTINE”, “ALTER ROUTINE” and “EXECUTE” to the HP OO user by an authorized account (such as “root”).

### Procedure for Purging Historical Data

Download the latest **MySQL.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.20** and unpack it.

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

1. Log into MySQL as “oouser” (the user created for HP OO).
2. Run the **00PurgeHistory.sql** script and verify that the new procedure was created and compiled successfully.
3. Tune the stored procedure according to the comments embedded in the script.

**Note:** MySQL does not allow setting default values for procedure parameters. Always run this procedure using explicit values!

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

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

Explanation regarding the above code (parameters reference values - left to right):

- 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.
- **pPurgeExecutionsInBatchesOf** determines how many flows are handled together. Smaller values imply smaller, more frequent transactions, and higher values imply less frequent, larger transactions. 10,000 is recommended for most systems.
- **pShouldPurgeExecutionSummary** determines if the **OO\_EXECUTION\_SUMMARY** table should be purged. It is recommended to use “0” by default (do not purge this table).

It is recommended to keep data in this table, because 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 the 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.
- **pAdvancedMode** 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.

## Postgres Maintenance Procedures

This chapter provides information on how to install and use HP OO 10.20 Postgres database maintenance procedures. These procedures are applicable for Postgres 9.2 – 9.3.

The user created for HP OO should be able to create and execute any function (procedure) within the HP OO database context. If this is not the case, grant relevant privileges to the HP OO user by an authorized account (such as “postgres”).

### Procedure for Purging Historical Data

Download the latest **PostgreSQL.zip** pack from HP Live Network under **OO DB Maintenance Scripts and Procedures > HP Operations Orchestration 10.20** and unpack it.

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

1. Log into Postgres as “oouser” (the user created for HP OO).
2. Run the **OOPurgeHistory.sql** script and verify that the new function was created and successfully.
3. Tune the stored procedure 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.

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

Explanation regarding the above code (parameters reference values - left to right):

- The **pPurgeExecutionsOlderThan** 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.
- **pPurgeExecutionsInBatchesOf** determines how many flows are handled together. Smaller values imply smaller, more frequent transactions, and higher values imply less frequent, larger transactions. 10,000 is recommended for most systems.
- **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.
- **pAdvancedMode** 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.



