



Operation

Informatica MDM - Product 360

Version: 8.1.1

1 Table of Contents

1	Table of Contents	2
2	Database Operation	8
2.1	Database update	8
2.1.1	Prerequisites on Oracle	8
2.2	Maintenance tasks.....	8
2.2.1	Indices and Statistics	8
	Create Maintenance Plan for SQL Server	9
	Create Maintenance Plan for Oracle.....	14
2.2.2	Backup and Recovery	15
2.3	Tuning.....	15
2.3.1	MS-SQL Server	15
	Activity Monitor	15
	Data Collection.....	16
	Memory Settings.....	16
	Transaction Logs and the Recovery Mode.....	18
	TempDB handling.....	18
2.4	Trouble shooting	18
2.4.1	Reporting	18
	Usage statistics	18
	Cleanup all temporary reports.....	21
	Separate storage for reporting tables	21
2.4.2	Create missing entries Item/Product/Variant/Structure-attributes in repository key language	22
	Solution.....	22
	Consequences	22
2.4.3	MS-SQL Server	22
	"The query processor could not start the necessary thread resources for parallel query execution".....	22
	"Transaction (Process ID XXX) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction".....	23
2.4.4	Oracle.....	23
	Import/Export of schemas	23
	Connection Issues	23
	Oracle JDBC driver OutOfMemoryError / huge memory consumption	24
	Oracle JDBC driver OutOfMemoryError (Cache).....	24

Oracle ReportStoreTemp performance	24
Automatic expiration of passwords for Oracle DB users	25
2.5 Diagnosing and tuning poorly performing Oracle queries.....	25
2.5.1 Fixing frequency histogram issues.....	29
2.6 Oracle Import/Export for MAIN, MASTER and SUPPLIER Schemas	30
2.6.1 Preparation.....	30
Directory Object	30
2.6.2 Export.....	30
User Roles	30
Execute export datapump	31
2.6.3 Import	31
User Roles	31
Tablespaces	31
Execute Import Datapump	32
3 Server Operation	33
3.1 Maintenance Tasks	33
3.2 Monitoring	33
3.3 Tuning.....	33
3.3.1 Memory Settings.....	33
3.3.2 Garbage Collection	33
Enable GC activity log	34
3.3.3 Communication Framework.....	34
3.3.4 Mass-Data Parallel Processing	34
3.3.5 Media Asset Parallel Management.....	34
CreateZipManagement.....	34
MediaAssetProvider special MBean.....	36
HeilerClassic special MBeans.....	36
3.3.6 Database Connection Pool.....	37
3.3.7 Logging.....	38
3.3.8 Import	39
Preferences regarding CPU usage.....	39
Preferences regarding memory usage	40
3.4 Trouble Shooting.....	41
3.4.1 Anti-Virus Software affecting performance	41

3.4.2	Media Asset Provider (Classic Provider).....	41
	No preview are visible.....	41
3.4.3	Obtain a Memory Dump (Heap Dump) from the Java Virtual Machine	41
3.4.4	Enhanced Logging for Assortment Content Calculation.....	42
3.4.5	Server and client in different time zones.....	42
3.5	SNMP Monitoring.....	42
3.5.1	Overview	43
3.5.2	Default Configurations	43
	Agent.....	43
	Network Management Station	44
	Managed Objects	44
	OID Configuration	44
	Generation of OID configuration	44
	Server Remote Command	44
	Trap.....	44
	Enable SNMP	45
	Run application	45
3.5.3	Modifying configurations	45
	Change the agent's port	45
	Network Management Station	45
	Change OID Configuration.....	46
	Advanced configuration	48
3.6	DB Connection pool	49
3.6.1	High performance connection pool configuration example	49
	Hibernate	49
	UDA	51
3.7	Server Job Maintenance.....	52
3.7.1	Configure the life time of a job.....	52
	Syntax	52
	Example	52
3.7.2	Standard Job List.....	52
3.8	Soft Delete Cleanup Job.....	54
3.8.1	General information	54
3.8.2	Configuration.....	54
3.8.3	Schedule	55
3.9	Data Quality Operation.....	55

3.9.1	Logging	55
	Rule engine execution.....	55
	Data provisioning	55
	Performance analyzing.....	56
3.9.2	Performance tuning	56
3.9.3	Issue reporting guide	57
	Configuration setup	57
	Information about the execution context.....	57
	Logging data	57
	Performance.....	57
3.9.4	Configure rules that are affected by Dictionary Synchronization in order to increase performance of IDQ rule execution	57
	Situation/ motivation	57
	Solution.....	58
3.10	Check integrity of attribute names in key language	59
3.10.1	MSSQL	60
3.10.2	Oracle.....	60
3.11	Network Recovery	60
4	Desktop Operation	62
4.1	Maintenance Tasks	62
4.2	Tuning.....	62
4.2.1	Performance optimization of the "Documents" view.....	62
	Filter	63
4.2.2	Import	64
4.3	Trouble Shooting.....	65
4.3.1	User Interface is not responding any more (client freeze).....	65
	Find the problem on the client	65
	Find the problem on the server	65
4.3.2	Rich Text Editor	66
	Unable to use the clipboard actions "Cut", "Copy" and "Paste"	66
4.3.3	Workaround to prevent the "Invalid Certificate" error (Mozilla)	66
4.4	Product Paradim (EGD) limitations.....	67
5	Control Center Manual.....	67
5.1	Prerequisite	67

5.2	Function.....	67
5.3	Login	68
5.4	Dashboard	68
5.4.1	Widget creation using templates	68
5.4.2	Customize widgets	69
5.5	Monitoring	70
5.5.1	Application Server Node.....	70
	Server tab	71
5.5.2	Server Nodes	71
5.5.3	Server Node Details	71
	Clients	72
	Telemetry	72
	Log	73
	Thread Dump.....	73
	SNMP.....	74
5.6	Deployment	74
5.6.1	Installation.....	74
5.6.2	License	75
5.6.3	Configuration.....	75
5.7	Support	75
5.7.1	CSM	75
5.8	Additional Features.....	76
5.8.1	Opening multiple tabs.....	76
5.8.2	Telemetry Actions.....	76
6	Audit Trail Operation.....	76
6.1	Maintenance.....	77
6.1.1	Audit Trail access security	77
6.1.2	Audit Trail database operations.....	77
	dbInstall	77
	dbDelete.....	77
	dbInfo.....	77
	dbTruncate	77
	dbBackup.....	78
6.1.3	Available application arguments	78

6.1.4	Purge Audit Trail database strategy	79
6.1.5	Running scheduled task	79
6.1.6	Audit Trail JMS consumer.....	79
6.2	Tuning	80
6.2.1	PIM Audit Trail MXBean	80
6.2.2	Audit Trail server MXBean.....	81
6.2.3	Audit Trail Server - second instance	82
7	Log File Overview	83
7.1	Additional Logging of Dataquality	84
7.2	Additional Logging of Change Summary and Datagraph.....	84
7.3	Additional Logging of Media Asset Processing.....	85
8	How to create a thread dump	85
8.1	Control Center	85
8.2	Console Window	86
8.3	VisualVM (JMX Interface)	86
8.4	Analyzing Thread Dumps	87
9	How to enable Java Management Extensions (JMX)	89
9.1	Production environment	90
9.2	Download and Installation of JMX Tooling	90
9.2.1	Connect on productive system	90

The operation manual documents actions which need to be performed in order to keep Informatica MDM - Product 360 up and running in a stable way. It contains the backup strategy as well as tuning and troubleshooting hints.

In order to meet enterprise level requirements of our current and future customers we try to steadily improve the performance of all parts of the Product 360 application. Optimal performance comes not without a cost, you will need adjust the platform configuration to the actual use case and load scenarios you experience at the customer.

2 Database Operation

Combines maintenance, tuning and troubleshooting for the Product 360 Core Database server. Additionally to the recommendations of the operation guide, a well educated database administrator should be available for maintenance and tuning tasks.

2.1 Database update

2.1.1 Prerequisites on Oracle

In case the Product 360 database is running in archive log mode (most likely), make sure to turn archive log mode off before executing the database update setup.

Otherwise you might risk exceeding available archive log disk space, in which case any database activity will be suspended immediately, and it will stay suspended until there is free available disk space again.

2.2 Maintenance tasks

2.2.1 Indices and Statistics

The database uses indices to increase the performance for read access on columns. In Product 360 we have indices on the columns which are typically used for selecting data, for example, all logical keys are at least part of an index. Additionally to that, the database uses statistics about the data in the tables in order to choose the most effective execution plan. Indices as well as statistics need to be maintained by the database administrator on a regular basis. Usually this is being done by creating a Maintenance Task which executes every day.

However, the typical maintenance task is more something of a bulldozer in maintaining. This means, it just rebuilds or reorganizes all indices no matter if needed or not and it also kills all statistics and rebuilds them no matter if needed or good for the statistic. Sometimes statistics which are very accurate get blindly rebuild and the result is a statistic which is not that accurate anymore, leading to a lesser performance than what you could have.

So, the general recommendation (from Microsoft Consulting) is like this (pseudo code):

```
if (number of pages in index < 1000)
    do nothing
else
    if (fragmentation < 5%)
        do nothing
    else if (fragmentation between 5% and < 30%)
        reorganize index + update its statistics
    else
        rebuild index
```


Additionally to that you should update statistics for all non-indexed columns and monitor index fragmentation to adjust the fill factor.

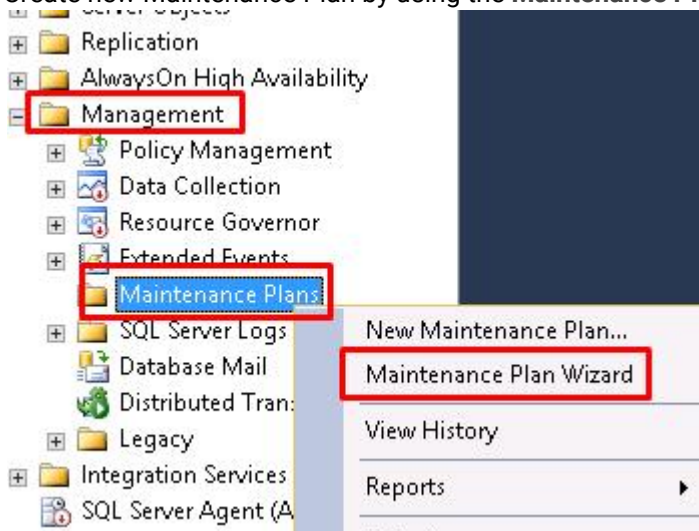
Create Maintenance Plan for SQL Server

All these steps can be combined in a single, easy to execute stored procedure which has generously been provided by Ola Hallengren. For a detailed description about the index and statistic maintenance script please refer to the online description which is available here:

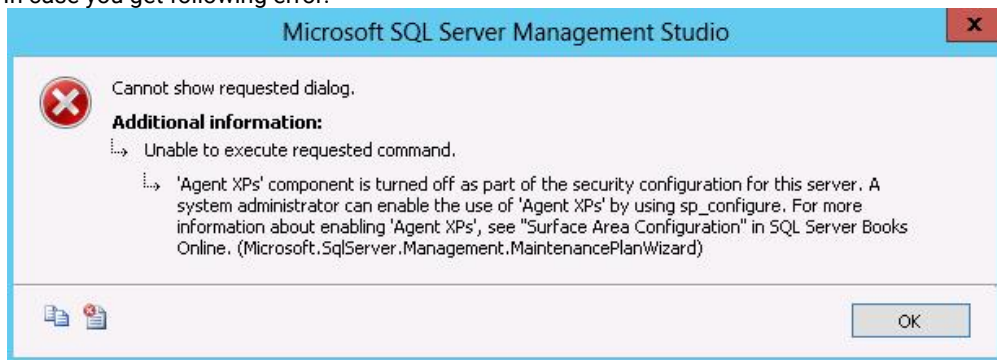
<http://ola.hallengren.com/sql-server-index-and-statistics-maintenance.html>. It will create all necessary objects for you. (For you convenience we attached the script it also to this wiki page)

Automatic execution (MSSQL2012)

1. **Execute** the MaintenanceSolution.sql script on your database. Be sure that your **SQL Server Agent** (service) is running.
2. Create new Maintenance Plan by using the **Maintenance Plans Wizard** as shown below



In case you get following error:

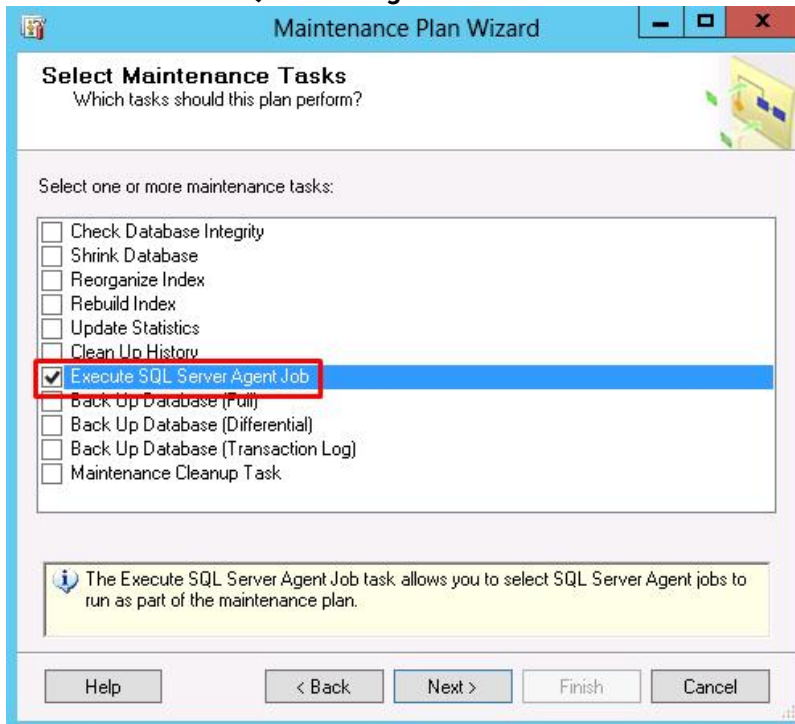


execute this statement:

```
sp_configure 'show advanced options', 1;  
GO  
RECONFIGURE;  
GO  
sp_configure 'Agent XPs', 1;
```

GO
RECONFIGURE
GO

3. Type the name of your new maintenance plan and configure your job schedule, then click on **Next**
4. Select the **Execute SQL Server Agent Job** and click on **Next twice**



5. Choose **IndexOptimize** and click on **Next**

The screenshot shows the 'Maintenance Plan Wizard' window, specifically the 'Define Execute SQL Server Agent Job Task' step. The title bar reads 'Maintenance Plan Wizard'. Below the title, the text says 'Define Execute SQL Server Agent Job Task' and 'Configure the maintenance task.'.

Under the heading 'Available SQL Server Agent jobs:', there is a table with three columns: 'Select', 'Job name', and 'Description'. The 'IndexOptimize - USER_DATABASES' row is selected, indicated by a red rectangle around the checked checkbox and the row itself.

Select	Job name	Description
<input type="checkbox"/>	CommandLog Cleanup	Source: http://ola.hallengren.com
<input type="checkbox"/>	DatabaseBackup - USER_DATABASES - LOG	Source: http://ola.hallengren.com
<input type="checkbox"/>	DatabaseIntegrityCheck - USER_DATABASES	Source: http://ola.hallengren.com
<input checked="" type="checkbox"/>	IndexOptimize - USER_DATABASES	Source: http://ola.hallengren.com
<input type="checkbox"/>	Output File Cleanup	Source: http://ola.hallengren.com
<input type="checkbox"/>	DatabaseBackup - SYSTEM_DATABASES - FU...	Source: http://ola.hallengren.com

Below the table, the 'Schedule:' section shows 'Not scheduled (On Demand)' with a 'Change...' button.

At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Finish >>', and 'Cancel'.

6. (Optional) Configure the report location and click on **Next**
7. After clicking **Finish** the task will check for any warnings or errors:

The screenshot shows the 'Maintenance Plan Wizard Progress' window. The title bar reads 'Maintenance Plan Wizard'. Below the title, the text says 'Maintenance Plan Wizard Progress' and 'Click Stop to interrupt the operation.'.

At the top, there is a green checkmark icon and the word 'Success'. Below this, a summary table shows the results:

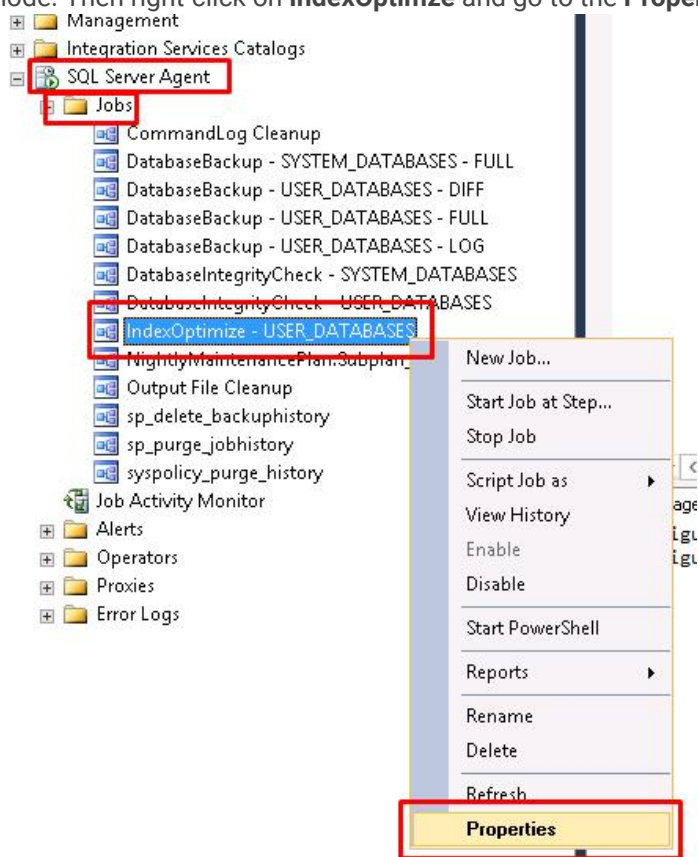
	5 Total	0 Error
5 Success		0 Warning

Below the summary, there is a 'Details:' section with a table showing the actions performed:

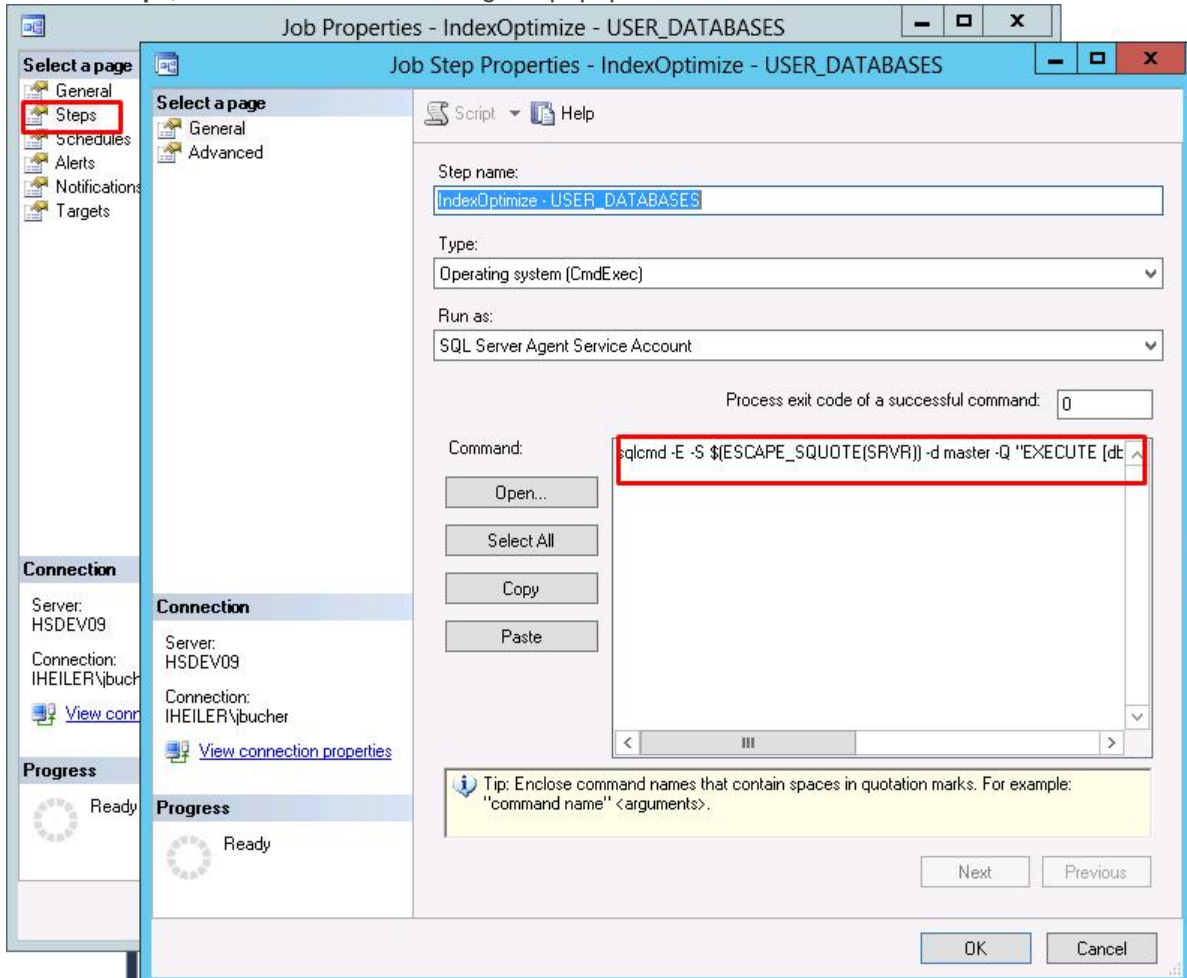
Action	Status	Message
Creating maintenance plan "NightlyMainten..."	Success	
Adding tasks to the maintenance plan	Success	
Adding scheduling options	Success	
Adding reporting options	Success	
Saving maintenance plan "NightlyMaintena..."	Success	

At the bottom, there are buttons for 'Stop', 'Report' (with a dropdown arrow), and 'Close'.

8. Click on the **SQL Server Agent** node from the left-hand side to expand the menu and open the **Jobs** node. Then right-click on **IndexOptimize** and go to the **Properties** view as shown below:



- Click on **Steps**, then **Edit** and a new Dialog will popup.



- Modify** the command statement (in this example the job will be executed on all user databases)

```
sqlcmd -E -S $(ESCAPE_QUOTE(SRVR)) -d master -Q "EXECUTE [dbo].[IndexOptimize] @Databases = 'USER_DATABASES',
@FragmentationLow = NULL, @FragmentationMedium = 'INDEX_REORGANIZE,INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE',
@FragmentationHigh = 'INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE', @FragmentationLevel1 = 5, @FragmentationLevel2 =
30, @UpdateStatistics = 'COLUMNS', @OnlyModifiedStatistics = 'Y', @LogToTable = 'Y' -b
```

Make sure to reformat above statement to a valid format.

- Click **OK**
- Schedule the job in the **Properties** view

Manual execution

- Download and **execute** the MaintenanceSolution.sql script.
- Execute** following SQL statement (in this example the index will get optimized on PIM_MAIN, PIM_MASTER and PIM_SUPPLIER)

SQL Statement to execute

```
EXECUTE dbo.IndexOptimize @Databases = 'PIM_MAIN, PIM_MASTER, PIM_SUPPLIER', @FragmentationLow = NULL,
@FragmentationMedium = 'INDEX_REORGANIZE,INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE', @FragmentationHigh =
'INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE', @FragmentationLevel1 = 5, @FragmentationLevel2 = 30, @UpdateStatistics
= 'COLUMNS', @OnlyModifiedStatistics = 'Y'
```

Make sure to reformat above statement to a valid format.

Create Maintenance Plan for Oracle

If the Oracle database instance is fully dedicated to PIM, it is recommended to disable the system job which automatically updates statistics on a daily basis:

```
BEGIN
DBMS_AUTO_TASK_ADMIN.DISABLE
(
  CLIENT_NAME => 'auto optimizer stats collection',
  OPERATION => NULL,
  WINDOW_NAME => NULL
);
END;
/
```

Instead, the Oracle DBA should create its own job for statistics maintenance.

It should be executed on a periodic base, and also immediately when the data situation has significantly changed (e.g. 25% of all items were touched).

Here is an example script which covers default Product 360 installations, the Oracle DBA is encouraged to adjust it accordingly to the specific Product 360 environment:

```
--*****
-- Statistics collection with SKEWONLY
-- * Automatically creates histograms on any column that shows a skew in data distribution
-- * Despite time consumption we highly recommend ESTIMATE_PERCENT = 100
-- * We recommend to delete height balanced histograms on NVARCHAR2 columns
-- * Check if the number of buckets in frequency histograms matches NDV
--*****
BEGIN
  DBMS_STATS.GATHER_SCHEMA_STATS(
    OWNNAME      => 'PIM_MAIN',
    OPTIONS      => 'gather',
    ESTIMATE_PERCENT => 100,
    METHOD_OPT    => 'for all columns size skewonly',
    CASCADE      => true,
    DEGREE       => 8
  );
  DBMS_STATS.GATHER_SCHEMA_STATS(
    OWNNAME      => 'PIM_MASTER',
    OPTIONS      => 'gather',
    ESTIMATE_PERCENT => 100,
    METHOD_OPT    => 'for all columns size skewonly',
    CASCADE      => true,
    DEGREE       => 8
  );
END;
```

```

DBMS_STATS.GATHER_SCHEMA_STATS(
  OWNNAME      => 'PIM_SUPPLIER',
  OPTIONS       => 'gather',
  ESTIMATE_PERCENT => 100,
  METHOD_OPT     => 'for all columns size skewonly',
  CASCADE      => true,
  DEGREE        => 8
);
END;
/

```

In case you still get bad performance / execution plans even after you updated the statistics, you can try to flush the shared pool to enforce the creation of new execution plans.

However, we usually only recommend this in case you are installing Product 360 updates which come with an updated version of the Product 360 database setup.

```
ALTER system flush shared_pool;
```

Index maintenance is usually a DBA task, so we don't give special recommendations about this task.

2.2.2 Backup and Recovery

Usually the backup and recovery strategy is defined by the customers IT department and it's dba. Product 360 core works fine with nightly backups using the "simple" recovery mode of the database, as well as transaction log backups during the day using the "full" recovery mode. Choose whatever your customer required in order to feel safe.



If you switch the schemas to recovery mode "full" you have to set up the transaction log backups. If you don't your transaction logs will keep on growing eventually without limits!

The backup plan should include the MASTER, SUPPLIER and MAIN schemas as well as all the runtime folders like import, export where the import and export files are stored.

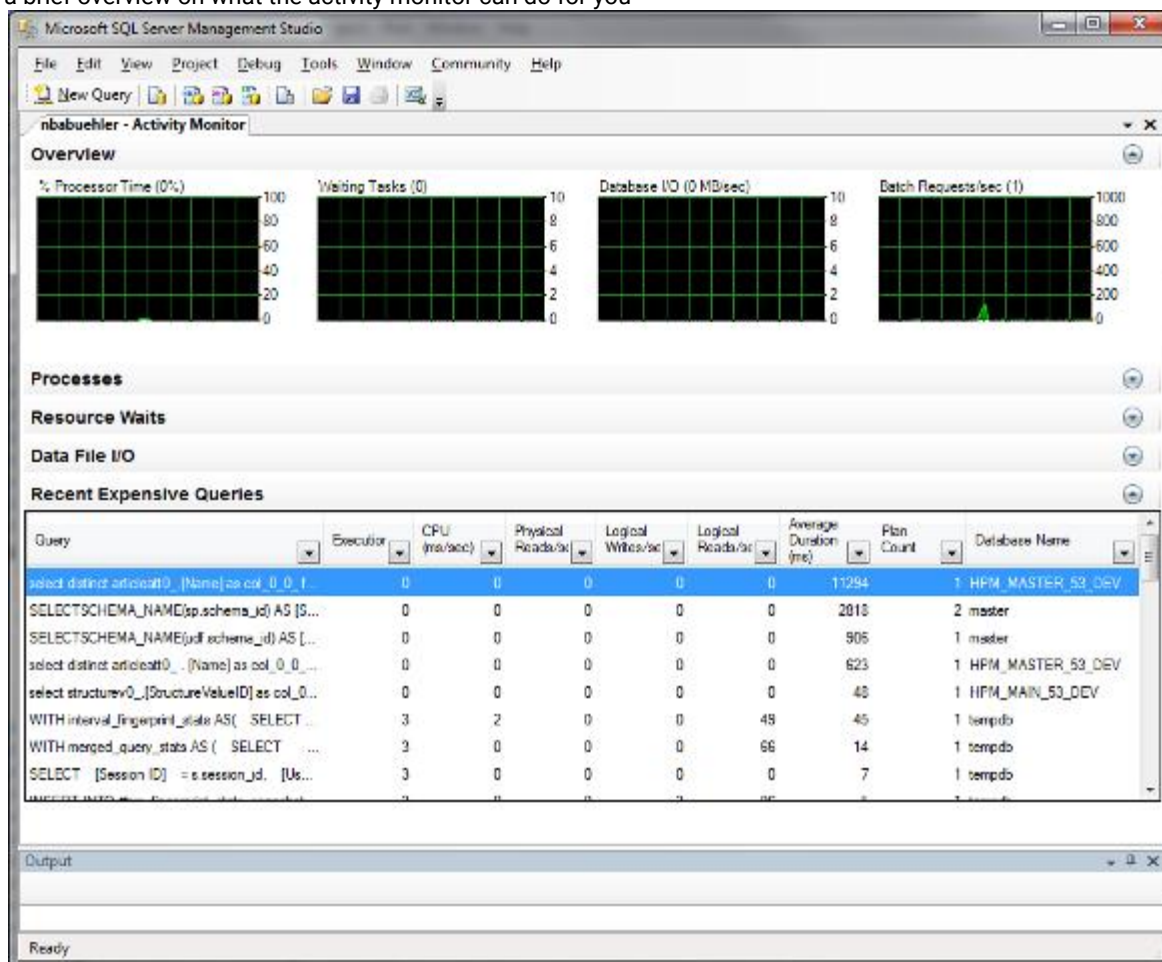
2.3 Tuning

2.3.1 MS-SQL Server

Activity Monitor

Use the Activity Monitor which is available in SQL Server to quickly identify slow statements and connection problems. Please see this knowledge base article which explains how to open the activity monitor, as well as this blog entry which gives

a brief overview on what the activity monitor can do for you



Data Collection

The activity monitor is a quite effective tool to get an overview of the current load on the database. However, sometime it's not that easy to produce the load on the application right at the time you're looking at the activity monitor. Sql Server 2008 has a new feature called "Data Collection". It might be a good idea to setup this feature and collect performance data for some days or weeks in order to see trends.

For a detailed description as how to setup this feature please refer to the MS-SQL Server Books Online: <http://msdn.microsoft.com/en-us/library/bb677356.aspx>



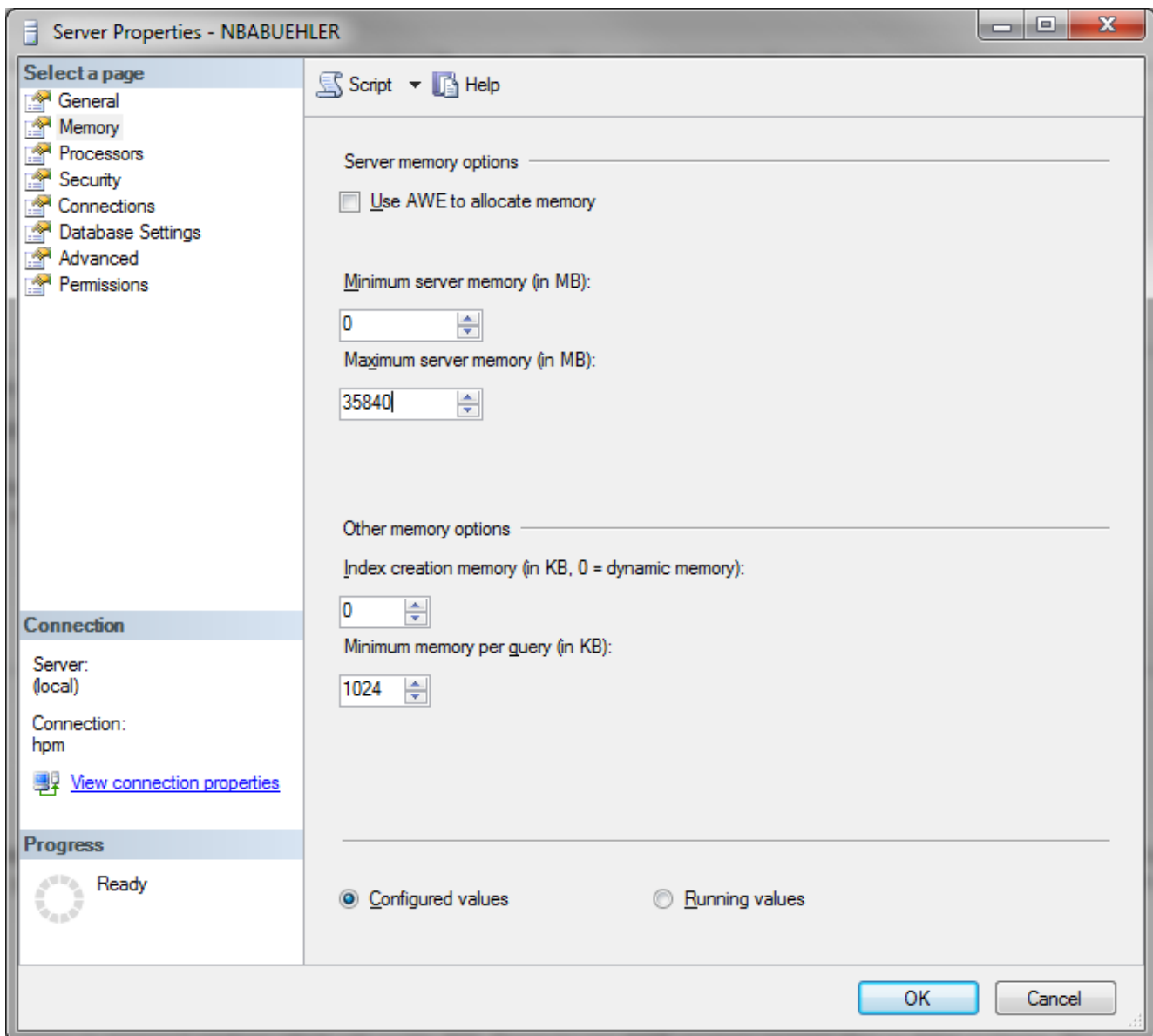
Please note that the data collection feature and the jobs it creates can not be deleted again, they can only be "disabled" so they will no longer collect any data.


Memory Settings

You should always limit the maximum memory usage of the SQL Server. In case you have a single instance on the machine, you can limit the memory to the maximum physical memory minus one GB for the operating system.

For example: The machine has 36 GB of memory, you would configure the SQL Server to use $35 \times 1024 = 35840$ MB memory, leaving the operating system 1024 MB - which should be enough, usually.

You can configure this using the Microsoft Sql Server Management Studio. Open the properties dialog of your database and adjust the memory settings like shown below.



 The "Use AWE to allocate memory" will have no effect in 64bit installations, SQL Server uses AWE access automatically when the user has "Lock pages in memory" privilege.

Lock pages in memory

Enable the SQL Server Service user with the "Lock pages in memory" privilege which will enable SQL Server to use AWE memory access also on 64bit systems. This results in a better performance for the memory access since SQL Server can directly access the memory and must not ask for it using the Virtual Memory Manager (VMM).

See also "Configuring the Server for Optimal Performance" (SQL Server 2008 Administration from Wrox Press, Chapter 11, Page 434)

Transaction Logs and the Recovery Mode

In a productive environment you should define the initial size of the database log files to the expected maximum. For both data and log files we recommend setting a proper growth value because the default of 64MB is too small.



A data base growth action always "stops the world" of the database until the files are enlarged. When this will occur often it will be a serious performance problem.

In case you are operating the Product 360 databases in **FULL** recovery mode please make sure that you take backups of the transaction logs on a regular basis. Otherwise the transaction logs will grow endlessly because SQL Server will not re-use the existing space until a backup was made.

TempDB handling

The TempDB will be used a lot since Product 360 uses transaction mode **READ_COMMITTED_SNAPSHOT** starting with PIM 7.1.

We recommend to tune the storage and the amount of data files according to the database load.

2.4 Trouble shooting

2.4.1 Reporting

A key functionality of the Product 360 - Server is the so called "reporting". A report in our sense is more or less only a list of internal database ID's. The report therefore reflects an arbitrary amount of "objects" in Product 360 - Server, identified by their ID. Reporting is a crucial part of Product 360 - Server since many functions rely on reports. For example the table based views of root entities like items or products, the export and the delete operations as well.

Usage statistics

The Informatica Software Support might ask you to provide usage statistics for the reporting in case we need to trouble shoot an issue of yours, for these you need to execute the following script on each data source (main, master and supplier):

Usage Statistic MSSQL

```
CREATE TABLE [#ReportInstances]
(
    [ReportInstance] NVARCHAR(30)
)
INSERT INTO [#ReportInstances] ( [ReportInstance] ) VALUES (N'ReportStore'), (N'ReportStoreTempA'), (N'ReportStoreTempB')

DECLARE @REPORT_TABLE_NAME NVARCHAR(30)
DECLARE @ReportInstance NVARCHAR(30)
DECLARE @SQL_STRING NVARCHAR(MAX)
DECLARE @TMP_SQL NVARCHAR(MAX)

DECLARE C0 CURSOR FOR
SELECT [ReportInstance]
FROM [#ReportInstances]
ORDER BY [ReportInstance]
OPEN C0
FETCH NEXT FROM C0 INTO @ReportInstance
WHILE @@FETCH_STATUS = 0
BEGIN
```

```

SET @SQL_STRING = ''
DECLARE C1 CURSOR FOR
SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME LIKE @ReportInstance + '[0-9]%'
ORDER BY CAST(REPLACE(TABLE_NAME, @ReportInstance, '' ) AS INT)
OPEN C1
FETCH NEXT FROM C1 INTO @REPORT_TABLE_NAME
WHILE @@FETCH_STATUS = 0
BEGIN
    IF LEN(@SQL_STRING) > 0
    BEGIN
        SET @SQL_STRING = CONCAT(@SQL_STRING, ' UNION ALL ')
    END
    SET @SQL_STRING = CONCAT(@SQL_STRING, 'SELECT * FROM dbo.[' + @REPORT_TABLE_NAME + ']')
    FETCH NEXT FROM C1 INTO @REPORT_TABLE_NAME
END
CLOSE C1
DEALLOCATE C1

SET @TMP_SQL = CONCAT('SELECT COUNT(*) AS [Number of Report IDs from ', @ReportInstance, '] FROM (', @SQL_STRING, '
X')

PRINT @TMP_SQL
EXECUTE sp_executesql @TMP_SQL

SET @TMP_SQL = CONCAT('SELECT COUNT(*) AS [Number of Temporary Report IDs from ', @ReportInstance, '] FROM (',
@SQL_STRING, ') X WHERE ReportID IN (SELECT ID FROM Report WHERE Purpose = 1)')
PRINT @TMP_SQL
EXECUTE sp_executesql @TMP_SQL

SET @TMP_SQL = CONCAT('SELECT COUNT(*) AS [Number of Permanent Report IDs from ', @ReportInstance, '] FROM (',
@SQL_STRING, ') X WHERE ReportID IN (SELECT ID FROM Report WHERE Purpose = 3)')
PRINT @TMP_SQL
EXECUTE sp_executesql @TMP_SQL

SET @TMP_SQL = CONCAT('SELECT COUNT(*) AS [Number of Orphan Report IDs from ', @ReportInstance, '] FROM (',
@SQL_STRING, ') X WHERE ReportID NOT IN (SELECT ID FROM Report)')
PRINT @TMP_SQL
EXECUTE sp_executesql @TMP_SQL

FETCH NEXT FROM C0 INTO @ReportInstance
END
CLOSE C0
DEALLOCATE C0

DROP TABLE [#ReportInstances]

SELECT COUNT([ID]) AS repTotalCount, Purpose, SUM([Count]) totalIdsCount FROM [dbo].[Report] GROUP BY Purpose

SELECT count([ID]) AS numOfReports, [Count] AS repSize, COUNT([ID])*[Count] AS totalSize FROM [dbo].[Report] WHERE Purpose=1
GROUP BY [Count] ORDER BY totalSize DESC
SELECT count([ID]) AS numOfReports, [Count] AS repSize, COUNT([ID])*[Count] AS totalSize FROM [dbo].[Report] WHERE Purpose=1
GROUP BY [Count] ORDER BY numOfReports DESC
SELECT count([ID]) AS numOfReports, [Count] AS repSize, COUNT([ID])*[Count] AS totalSize FROM [dbo].[Report] WHERE Purpose=1
GROUP BY [Count] ORDER BY repSize DESC

SELECT [Type], SUM([Count]) AS totalSize FROM [dbo].[Report] WHERE Purpose=1 AND [Count] < 10 GROUP BY [Type]
SELECT [Type], SUM([Count]) AS totalSize FROM [dbo].[Report] WHERE Purpose=1 AND [Count] < 100 GROUP BY [Type]

```

```
SELECT [Type], SUM([Count]) AS totalSize FROM [dbo].[Report] WHERE Purpose=1 AND [Count] < 200 GROUP BY [Type]
```

```
SELECT TOP 100 [ID] ,[Type] ,[Purpose] ,[Count] ,[CreationDate] FROM [dbo].[Report] WHERE Purpose=1 ORDER BY CreationDate DESC
```

Usage Statistic Oracle

```
SET SERVEROUTPUT ON
```

```
SET LINESIZE 2000
```

DECLARE

```
REPORT_TABLE_NAME VARCHAR2(30);
```

```
SQL_STRING VARCHAR2(2000);
```

```
TMP_SQL VARCHAR2(2000);
```

```
TMP_RESULT NUMBER;
```

```
TEMP_NAME NVARCHAR2(30);
```

```
TYPE TEMP_ARRAY IS VARRAY(3) OF VARCHAR2(30);
```

```
ARRAY TEMP_ARRAY := TEMP_ARRAY('ReportStore', 'ReportStoreTempA', 'ReportStoreTempB');
```

```
CURSOR C1(REPORT_INSTANCE VARCHAR2)
```

```
IS SELECT TABLE_NAME
```

```
FROM USER_TABLES
```

```
WHERE REGEXP_LIKE(TABLE_NAME, '^' || REPORT_INSTANCE || '\d')
```

```
ORDER BY CAST(REPLACE(TABLE_NAME, REPORT_INSTANCE) AS NUMBER);
```

BEGIN

```
FOR i IN 1..ARRAY.COUNT LOOP
```

```
TEMP_NAME := ARRAY(i);
```

```
SQL_STRING := '';
```

```
OPEN C1(TEMP_NAME);
```

```
LOOP
```

```
FETCH C1 INTO REPORT_TABLE_NAME;
```

```
EXIT WHEN C1%NOTFOUND;
```

```
IF LENGTH(SQL_STRING) > 0 THEN
```

```
BEGIN
```

```
SQL_STRING := SQL_STRING || ' UNION ALL ';
```

```
END;
```

```
END IF;
```

```
SQL_STRING := SQL_STRING || ' SELECT * FROM "' || REPORT_TABLE_NAME || '"';
```

```
END LOOP;
```

```
CLOSE C1;
```

```
TMP_SQL := 'SELECT COUNT(*) AS "numIds' || TEMP_NAME || '" FROM (' || SQL_STRING || ') X';
```

```
DBMS_OUTPUT.PUT_LINE(TMP_SQL);
```

```
EXECUTE IMMEDIATE(TMP_SQL) INTO TMP_RESULT;
```

```
DBMS_OUTPUT.PUT_LINE(TMP_RESULT || CHR(10));
```

```
TMP_SQL := 'SELECT COUNT(*) AS "numTempIds' || TEMP_NAME || '" FROM (' || SQL_STRING || ') X WHERE "ReportID" IN (SELECT "ID" FROM "Report" WHERE "Purpose" = 1)';
```

```
DBMS_OUTPUT.PUT_LINE(TMP_SQL);
```

```
EXECUTE IMMEDIATE(TMP_SQL) INTO TMP_RESULT;
```

```
DBMS_OUTPUT.PUT_LINE(TMP_RESULT || CHR(10));
```

```
TMP_SQL := 'SELECT COUNT(*) AS "numPermIds' || TEMP_NAME || '" FROM (' || SQL_STRING || ') X WHERE "ReportID" IN (SELECT "ID" FROM "Report" WHERE "Purpose" = 3)';
```

```

DBMS_OUTPUT.PUT_LINE(TMP_SQL);
EXECUTE IMMEDIATE(TMP_SQL) INTO TMP_RESULT;
DBMS_OUTPUT.PUT_LINE(TMP_RESULT || CHR(10));

TMP_SQL := 'SELECT COUNT(*) AS "numOrphanIds' || TEMP_NAME || '" FROM (' || SQL_STRING || ') X WHERE "ReportID" NOT IN
(SELECT "ID" FROM "Report")';
DBMS_OUTPUT.PUT_LINE(TMP_SQL);
EXECUTE IMMEDIATE(TMP_SQL) INTO TMP_RESULT;
DBMS_OUTPUT.PUT_LINE(TMP_RESULT || CHR(10));

END LOOP;
END;
/

SELECT COUNT ("ID") AS "repTotalCount", "Purpose", SUM ("Count") AS "totalIdsCount" FROM "Report" GROUP BY "Purpose";
SELECT COUNT ("ID") AS "numOfReports", "Count" AS "repSize", COUNT ("ID")*"Count" AS "totalSize" FROM "Report" WHERE
"Purpose"=1 GROUP BY "Count" ORDER BY "totalSize" DESC;
SELECT COUNT ("ID") AS "numOfReports", "Count" AS "repSize", COUNT ("ID")*"Count" AS "totalSize" FROM "Report" WHERE
"Purpose"=1 GROUP BY "Count" ORDER BY "repSize" DESC;

SELECT "Type", SUM ("Count") AS "totalSize" FROM "Report" WHERE "Purpose"=1 AND "Count" < 10 GROUP BY "Type";
SELECT "Type", SUM ("Count") AS "totalSize" FROM "Report" WHERE "Purpose"=1 AND "Count" < 100 GROUP BY "Type";

SELECT "Type", SUM ("Count") AS "totalSize" FROM "Report" WHERE "Purpose"=1 AND "Count" < 200 GROUP BY "Type";
SELECT "ID", "Type", "Purpose", "Count", "CreationDate" FROM "Report" WHERE "Purpose" = 1 ORDER BY "CreationDate" DESC;

```

Cleanup all temporary reports

The Product Manager uses an automatically scheduled system job for to delete temporary reports which are older than one day. This job is scheduled every hour (by default), therefore the Report table should not have any entry with Purpose=1 (temporary) which are older than 24 hours. However, it occurred in the past that this job could not fulfill its task in a reasonable amount of time or due some misconfiguration.

Prior to Version 6.0 this job deleted all reports (and their corresponding entries in the ReportStoreTemp table) in a **single transaction**. This may lead to an "explosion" of the transaction log since the ReportStoreTemp table might contain millions of items. In case you find old reports in the Report table (in any Product 360 - Server schema) you can execute the following SQL statements to completely delete all temporary reports in one small step, keeping the transaction log usage low.

Please note that you should shutdown the Product 360 - Server during this - since currently running operations might rely on a temporary report.

Delete ALL Temporary Reports

```

DELETE FROM [Report] WHERE [Purpose] = 1
TRUNCATE TABLE [ReportStoreTemp]

```

With PIM - Server versions >= 6.0 the delete job is more aware of this problem. Big reports will be deleted one by one, small reports in a bulk delete operation. You can configure the boundaries of "big reports" in the plugin_customization.ini file. Additionally to that, the delete job can also be configured to delete reports which are younger than a day - for example a few hours.

Separate storage for reporting tables

On heavy load databases we recommend to put the corresponding reporting tables for each schema:

- Report
- ReportStore
- ReportStoreTempA

- ReportStoreTempB

on a high-performance storage separated from the rest of the Product 360 tables.

2.4.2 Create missing entries Item/Product/Variant/Structure-attributes in repository key language

This can happen if the language of the repository got changed during maintenance, if there is any custom code in the system which adds entries by not honoring the key language or an import which is ignoring the fact the the key language should be filled always. The repository language is configured in the server.properties and should get never touched in a running system.

Since PIM 7.0.01 the Product 360 Core will check this behavior during server startup. The server won't start if there are not all attributes maintained in the repository key language.


The occurred message in the server log file will look like this:

The database 'HPM_MAIN' contains invalid structure feature entries. '3' entries have been found which aren't maintained in the repository language German.

Solution

Create the entries for the missing language. The provided script is checking the item attributes and structure attributes. It will lookup of the missing data entries and copy an existing entry with the lowest language ID for the missing repository language.

Because the name of the created attribute has to be unique, the name will be '<name of the attribute to copy><defined delimiter in the cmd file><id of the attribute to copy>'. This name will be cut at the cmd defined length (by default DB 'Name' column length).

 It is possible that this script creates invalid data if the attribute name of the the attribute to copy is as long as the defined max length in the cmd file and this name is already used by another attribute in the same context.

createMissingAttributes_mssql_pim7.rar

createMissingAttributes_oracle_pim7.rar

createMissingAttributes_mssql_pim8.rar

createMissingAttributes_oracle_pim8.rar

Consequences

The repository language **MUST NOT** be changed as soon as entity data such as items/products/variants or structures/structure groups have been created and exist in the database. In such a situation, the stability of the system can no longer be guaranteed since logical key fields most likely will contain null values.

2.4.3 MS-SQL Server

"The query processor could not start the necessary thread resources for parallel query execution"

During high-end benchmark tests the following error message has been thrown by the SQL Server "The query processor could not start the necessary thread resources for parallel query execution". After some research we came to the conclusion that the only currently known solution to this problem is to reduce the load on the database server. Several articles from Microsoft or Microsoft close resources explained this to be the only way to avoid this problem. See also this article from Microsoft: <http://msdn.microsoft.com/en-us/library/aa337448.aspx>

So, how to reduce the load on the database server? First approach is to reduce the number of concurrent connections using the [DB Connection pool](#) settings as described in the Tuning Advisory. Additionally to that please always inform the

Informatica Software Support. It will be very helpful if you can describe which action lead to the error, steps to reproduce would be great too since we're constantly investigating this issue and try to find a better solution.

In case this problem occurs often on your system you might want to observe the processor queue, this can give us hints on the processor utilisation at the time the error occurred.

"Transaction (Process ID XXX) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction"

In case you encounter database deadlocks you can use SQL Server Profiler to find the cause for the particular deadlock.

You need to run a server-side trace to capture deadlock information, you can use the script available here: <http://www.sqlservercentral.com/scripts/deadlock/66808/>

Once the deadlock has occurred, you can read the trace file with the script available here: <http://www.sqlservercentral.com/scripts/deadlock/66809/>

The content of the trace file will show which transactions resp. which SQL statements are responsible for the deadlock so further actions can be taken.

2.4.4 Oracle

Import/Export of schemas

To obtain some prepared notes on how to import export schemas from/to an Oracle instance, please raise a request with Informatica.

Connection Issues

If your project is having sporadic connection issues with Oracle database maybe these topics are helpful:

1. Maximum amount of processes. Because there were more than one application on the Oracle database the number of 300 was too small. We had good experiences with 1000 (minimum 300 x count of applications)
2. open_cursors, also too small values. We had good experiences with 1000 (minimum 300 x count of applications)
3. Log-Files are full. Mostly the Listener log file.



Switch off listener logging

Carry out following steps to configure the listener logging:

- a. Open [listener.ora](#) in `..\product\11.2.0\dbhome_1\NETWORK\ADMIN`
- b. Add **LOGGING_<Listernname> = OFF**
- c. Replace `<Listernname>` through your `listernname` e.q. 11GR2
- d. Save the changes
- e. Restart db and listener service.

Possible Values:

[ON| OFF]

4. tempdb full. Index rebuild after installation of standard classification systems!
5. Audit-Trail: waste logs also and makes them full. Towards description of Oracle the parameter „audit_trail“ is NOT deactivated per default! We deactivated this parameter on all of our instances.



Attention: when applying changes via enterprise manager console please ensure that the changes are applied in spfile, too.

Oracle JDBC driver OutOfMemoryError / huge memory consumption

Oracle JDBC driver may course OutOfMemoryError. The reason is that Oracle driver tries to allocate huge buffer for the incoming data. Sizes of these buffers depend on a single result set row size and 'prefetchRowSize' parameter of the prepared statement. By default this parameter is set to 10, but to increase list model performance it has been set to 10000 for all fragment SPs calls.

There are following different ways to avoid OutOfMemoryError in the oracle jdbc driver:

- Use jdbc driver parameter to limit buffer size in the UDA configuration file:

```
<property name="connectionProperties">oracle.jdbc.maxCachedBufferSize=<sizeInBytes>;</property>
```

- Set 'rowPrefetchSize' in the com.heiler.ppm.fragment.server.fragment extension point to override the global value for a certain fragment
- Set 'defaultRowPrefetchSize' in the com.heiler.ppm.fragment.server/preferences.ini or in the plugin_customization.ini

```
# Global value to control number of rows that will be fetch from the database in one round-trip
# using fragment stored procedure. It helps jdbc driver to find trade-off between memory consumption
# and db fetch performance. Default value set by many jdbc drivers is 10, but to improve list model data access
# we need to set this value to a much higher value.
#
# Some JDBC driver implementations use this value to estimate incoming raw data buffer size as
# defaultRowPrefetchSize*rowSize. Such buffers may have very large size (several hunderd megabyte).
# Set this parameter to a low value if application server has small amount
# of memory. You can also experiment with this value if garbage collection pessure is too high.
#
# This global value can be overridden per fragment using optional 'rowPrefetchSize' attribute of the
# com.heiler.ppm.article.core.server.Fragment extension point. It is recommended to use this attribute if
# the average number of rows returned by the fragment can be estimated in development time.
# defaultRowPrefetchSize = 10000
```

See Oracle JDBC MemoryManagement for more info

Oracle JDBC driver OutOfMemoryError (Cache)

The listModel (Product 360 - Server Table) access with NCLOBs may result in an OutOfMemoryError.

The reason of this dump is oracle driver cache for CLOB fields. JDBC statements hold references to huge byte arrays.

This strategy usually reduces number of DB round-trips and improves performance but it is takes a lot of memory.

Try setting **oracle.jdbc.FreeMemoryOnEnterImplicitCache** connection property to true. This will instruct Oracle driver to do cache clean-up more frequently.

see oracle driver do for more detail: http://download.oracle.com/docs/cd/E14072_01/appdev.112/e13995/oracle/jdbc/pool/OracleDataSource.html

Oracle ReportStoreTemp performance

Product 360 - Server reporting framework uses ReportStore and ReportStoreTemp tables to save report ids.

Under heavy load report save rate can be very high and ReportStoreTemp tables may grow up to dozens of million records.

If possible we recommend putting these tables on a high performance storage.

Automatic expiration of passwords for Oracle DB users

When setting up the Oracle DB and the users needed for Product 360 it is important to not set the password of each of them to automatically expire. Once the expiration date is met the Product 360 Server and components will not be able to start up anymore.

2.5 Diagnosing and tuning poorly performing Oracle queries

This is a big topic, but we'll focus in on what a bad query plan looks like, what a good query plan looks like, and one example of how to improve a poor performing query. There is no generic way to tune all poorly performing queries, but hopefully this page can educate you on the tools you need to identify and diagnose a poor performing query and an example of how this particular query was tuned. The first part of this how-to is PIM specific, but the important parts are only specific to Oracle.

The first step is to find the poor performing query. In our example, a PIM Desktop user issued a query and noticed poor performance:



The first step is to go through the logs to try to find the query. As PIM uses hibernate, the following categories (org.hibernate.SQL and org.hibernate.type.descriptor.sql.BasicBinder) had to be updated to TRACE in the log4j.xml file to enable logging of SQLs as well as bind parameters:

```
<!-- Use "TRACE" to see the debug output from UDA and activate tracing in stored procedures.
      Check also server.properties property "db.default.debug.show_sql" -->
<category name="UDA" additivity="false">
  <priority value="TRACE"/>
  <appender-ref ref="UdaAppender"/>
</category>

<!-- Use "TRACE" to see the debug output from Hibernate
      org.hibernate.cache           Log all second-level cache activity
      org.hibernate.hql.ast.AST     Log HQL and SQL ASTs during query parsing
      org.hibernate.jdbc            Log all JDBC resource acquisition
      org.hibernate.pretty          Log the state of all entities (max 20 entities) associated with the session at flush time
      org.hibernate.SQL             Log all SQL DML statements as they are executed (include org.hibernate.type to see binds
and values)
      org.hibernate.secure          Log all JAAS authorization requests
      org.hibernate.tool.hbm2ddl    Log all SQL DDL statements as they are executed
      org.hibernate.transaction     Log transaction related activity
      org.hibernate.type            Log all JDBC parameters
      org.hibernate                Log everything -->
<category name="org.hibernate.SQL" additivity="false">
  <level value="TRACE" />
  <appender-ref ref="UdaAppender"/>
</category>
<category name="org.hibernate.type.descriptor.sql.BasicBinder" additivity="false">
  <level value="TRACE" />
  <appender-ref ref="UdaAppender"/>
</category>
```

Changes in this file do not require a server restart, and after the above query is executed again, you will need to find the query in the logs.

Here's the query after we've replaced the necessary bind parameters:

```
set linesize 1000
```

```

explain plan for
select articlerev0_.ID, articlerev0_.Identifier
from
  "ArticleRevision" partition(P_PRODUCT) articlerev0_
left outer join
  "ArticleDetail" partition(P_PRODUCT) articledet1_
    on articlerev0_.ID=articledet1_.ArticleRevisionID
    and articledet1_.DeletionTimestamp = TO_TIMESTAMP('9999-12-31','yyyy-mm-dd')
left outer join
  "ArticleLang" partition(P_PRODUCT) articlelan2_
    on articlerev0_.ID=articlelan2_.ArticleRevisionID
    and (
      articlelan2_.LanguageID=9
      and articlelan2_.Res_LK_Text100_01='LT'
      and articlelan2_.Res_LK_Int_01=0
      and articlelan2_.ChannelID=1
      and articlelan2_.EntityID=21007
    )
    and articlelan2_.DeletionTimestamp = TO_TIMESTAMP('9999-12-31','yyyy-mm-dd')
left outer join
  "ArticleAttribute" partition(P_PRODUCT) articleatt3_
    on articlerev0_.ID=articleatt3_.ArticleRevisionID
    and (
      articleatt3_.NameInKeyLanguage='Heel Type'
    )
    and articleatt3_.DeletionTimestamp = TO_TIMESTAMP('9999-12-31','yyyy-mm-dd')
left outer join
  "ArticleAttributeValue" partition(P_PRODUCT) articleatt4_
    on articleatt3_.ID=articleatt4_.ArticleAttributeID
    and (
      articleatt4_.Identifier='ALL'
      and articleatt4_.LanguageID=9
    )
    and articleatt4_.DeletionTimestamp = TO_TIMESTAMP('9999-12-31','yyyy-mm-dd')
where
  articlerev0_.RevisionID = 1
  and articlerev0_.DeletionTimestamp = TO_TIMESTAMP('9999-12-31','yyyy-mm-dd')
  and articledet1_.Res_Bit_02=1
  and articlelan2_.Res_Text250_05='41'
  and (
    articleatt4_.Value is not null
  )
  and articlerev0_.EntityID=1100;

```

Explaining the plan query for this resulted in the following. Notice the difference between the estimated row count and the actual row count? This results in Oracle choosing an incorrect plan with an incorrect join order and many nested loops.



Also notice that an INDEX RANGE SCAN was performed on index XAK2_ArticleAttribute in table ArticleAttribute, which resulted in an incorrect estimated row count.

To investigate further, a 10053 Optimizer trace is needed. In the same session from where you issued the query and explain plan, do the following. Pay attention to the commented lines below. You'll need to turn on tracing, execute the query, then turn off tracing. Then you'll need to view the trace files, preferably with a utility:

```

-- activate generating trace files
ALTER SESSION SET EVENTS='10053 trace name context forever, level 1';
-- deactivate generating trace files
ALTER SESSION SET EVENTS '10053 trace name context off';
-- get filenames for trace session
select u_dump.value || '/' || instance.value || '_ora_' || v$process.spid
|| nvl2(v$process.traceid, '_' || v$process.traceid, null ) || '.trc'"Trace File"
from V$PARAMETER u_dump
cross join V$PARAMETER instance
cross join V$PROCESS
join V$SESSION on v$process.addr = V$SESSION.paddr
where u_dump.name = 'user_dump_dest'
and instance.name = 'instance_name'
and V$SESSION.audsid=sys_context('userenv','sessionid');

```

Navigate to the Explain Plan section of the trace file to confirm you are looking at the proper query in the trace file:

```

----- Explain Plan Dump -----
----- Plan Table -----

=====
Plan Table
=====
-----+-----
+-----+
| Id | Operation                                | Name                                | Rows | Bytes | Cost | Time      | Pstart |
Pstop |
+-----+
| 0  | SELECT STATEMENT                        |                                     |      |      | 10   |           |        |
|
| 1  | NESTED LOOPS                            |                                     | 1    | 116   | 10   | 00:00:01 |        |
|
| 2  | NESTED LOOPS                            |                                     | 1    | 116   | 10   | 00:00:01 |        |
|
| 3  | NESTED LOOPS                            |                                     | 1    | 84    | 7    | 00:00:01 |        |
|
| 4  | PARTITION REFERENCE SINGLE              |                                     | 1    | 55    | 5    | 00:00:01 | KEY(AP)|
KEY(AP)|
| 5  | TABLE ACCESS BY LOCAL INDEX ROWID     | ArticleAttribute                   | 1    | 55    | 5    | 00:00:01 | 2      | 2
|
| 6  | INDEX RANGE SCAN                        | XAK2_ArticleAttribute              | 1    |       | 4    | 00:00:01 | 2      | 2
|
| 7  | TABLE ACCESS BY GLOBAL INDEX ROWID     | ArticleRevision                    | 1    | 29    | 2    | 00:00:01 | 2      | 2
|
| 8  | INDEX UNIQUE SCAN                       | PK_ArticleRevision                | 1    |       | 1    | 00:00:01 |        |
|
| 9  | PARTITION REFERENCE SINGLE              |                                     | 1    |       | 2    | 00:00:01 | KEY(AP)|
KEY(AP)|
| 10 | INDEX UNIQUE SCAN                       | XAK1_ArticleAttributeValue         | 1    |       | 2    | 00:00:01 | KEY(AP)|
KEY(AP)|
| 11 | TABLE ACCESS BY LOCAL INDEX ROWID     | ArticleAttributeValue              | 1    | 32    | 3    | 00:00:01 | 1      | 1
|
+-----+

```

In my trace file the explain plan dump for the query starts at line 6613.

Using a trace file viewer is recommended - general steps to find the issue are:

1. Find the proper 'optimizing query block' (OPTIMIZING QB) section for your sql (If using v10053.exe look for the OPTIMIZING QB section on the left hand side).
2. Look for the Access path analysis for the table containing the wrong statistic (ArticleAttribute in our case).
3. Look for the column containing the index giving the wrong estimated row count. In our case, the XAK2_ArticleAttribute index is on the NameInKeyLanguage column.

At this point you'll want to familiarize yourself with how Oracle uses histograms to manage statistics.



If you look closely at section 3, you'll notice that the NDV (Number of Distinct Values) is 383, but the #Bkts (Buckets) is 255 from the HtBal (Height Balanced) histogram. The NameInKeyLanguage value of 'Heel Height' was unfortunately not in one of the buckets, which causes the "Using prorated density: 0.000000 of col #54 as selectivity of out-of-range/non-existent value pred" message you see. This causes the optimizer to incorrectly guess that this value doesn't exist in this table and that joining to this table would also return no rows for the query (prorated density of 0).

To fix this, we need to delete the histogram. Here's a script which deletes all histograms for all varchars with > 254 distinct values, as the number of buckets isn't sufficient to cover all values so a height balanced histogram isn't desired for this scenario (in general, height balanced histograms are almost useless for varchar2 columns).

Script which deletes all height balanced histograms for nvarchar2 columns with more than 254 distinct values

```
-- script which deletes all height balanced histograms for nvarchar2 columns with more than 254 distinct values
BEGIN
  FOR COL_ITEM IN (
    SELECT stats.OWNER, stats.TABLE_NAME, stats.COLUMN_NAME
    FROM DBA_TAB_COL_STATISTICS stats
    INNER JOIN DBA_TAB_COLUMNS cols
    ON stats.TABLE_NAME = cols.TABLE_NAME
    AND stats.COLUMN_NAME = cols.COLUMN_NAME
    WHERE cols.DATA_TYPE = 'NVARCHAR2'
    AND stats.HISTOGRAM <> 'NONE'
    AND stats.NUM_DISTINCT > 254
    AND stats.OWNER = cols.OWNER
    AND stats.OWNER IN ('HBC_HPM_MAIN', 'HBC_HPM_MASTER', 'HBC_HPM_SUPPLIER')
    ORDER BY stats.OWNER, stats.TABLE_NAME, stats.COLUMN_NAME
  )
  LOOP
    DBMS_STATS.DELETE_COLUMN_STATS (OWNNAME=>COL_ITEM.OWNER, TABNAME=>'"' || COL_ITEM.TABLE_NAME || '"', COLNAME=>'"' ||
COL_ITEM.COLUMN_NAME || '"', COL_STAT_TYPE=>'HISTOGRAM');
  END LOOP;
END;
/
```

Finally, to prevent Oracle's automatic statistics job from regenerating the histogram you need to call the set_table_prefs function.

Pass size 1 as the METHOD_OPT to prevent histogram creation:

```
--This does not work as column and table name are lowercase and must be passed with double quote.
--TODO is to fix this call to work with mixed case table and column names
BEGIN
  DBMS_STATS.set_table_prefs('HBC_HPM_MASTER', '"ArticleAttribute"', 'METHOD_OPT',
    'for all columns size skewonly for columns size 1 "NameInKeyLanguage"');
```

```
END;  
/
```

After fixing this and a similar issue in the Res_Text250_05 column for table ArticleLang, we get the following plan. While the actual and estimated row counts are off, the join order is now more efficient (Good: ArticleLang, ArticleDetail, ArticleAttribute, ArticleAttributeValue vs. Bad: ArticleAttribute, ArticleDetail, ArticleLang, ArticleAttributeValue previously). For outer joins in general, the most limiting tables should be joined first, which for PIM means ArticleAttribute and ArticleAttributeValue should usually be joined last as they are larger than the Article table by some factor.



2.5.1 Fixing frequency histogram issues

The second issue look for the Res_Text250_05 column for table ArticleLang:

```
Column (#13):  
NewDensity:0.000053, OldDensity:0.000000 BktCnt:9392, PopBktCnt:9375, PopValCnt:114, NDV:208  
Column (#13):  
NewDensity:0.000122, OldDensity:0.000000 BktCnt:4113, PopBktCnt:4091, PopValCnt:103, NDV:208  
Column (#13): Res_Text250_05( Part#: 1  
AvgLen: 4 NDV: 208 Nulls: 10118996 Density: 0.000122  
Histogram: Freq #Bkts: 125 UncompBkts: 4113 EndPtVals: 125  
Column (#13): Res_Text250_05(  
AvgLen: 4 NDV: 208 Nulls: 10118996 Density: 0.000122  
Histogram: Freq #Bkts: 125 UncompBkts: 4113 EndPtVals: 125  
...  
...  
Using density: 0.000122 of col #13 as selectivity of unpopular value pred
```

Issue: The value selected did not fall into any of the histogram buckets.

Solution: Add more buckets such that all values in the column can be assigned a bucket by forcing the optimizer to analyze 100 percent of the rows. As the number of distinct values in this column is less than 254 (Oracle max # of buckets) this is a suitable solution:

```
-- script which adds more buckets for frequency histograms  
DECLARE  
    BUCKETS_BUFFER NUMBER := 0;  
BEGIN  
    FOR COL_ITEM IN (  
        SELECT stats.OWNER, stats.TABLE_NAME,  
               LISTAGG('FOR COLUMNS "' || stats.COLUMN_NAME || '" SIZE ' || CASE WHEN stats.NUM_DISTINCT +  
BUCKETS_BUFFER > 254 THEN 254 ELSE stats.NUM_DISTINCT + BUCKETS_BUFFER END, ', ') WITHIN GROUP (ORDER BY stats.COLUMN_NAME)  
        AS METHOD_OPT  
        FROM DBA_TAB_COL_STATISTICS stats  
        INNER JOIN DBA_TAB_COLUMNS cols  
            ON stats.TABLE_NAME = cols.TABLE_NAME  
            AND stats.COLUMN_NAME = cols.COLUMN_NAME  
        WHERE cols.DATA_TYPE = 'NVARCHAR2'  
        AND stats.HISTOGRAM = 'FREQUENCY'  
        AND stats.NUM_DISTINCT > stats.NUM_BUCKETS  
        AND stats.OWNER = cols.OWNER  
        AND stats.OWNER IN ('HBC_HPM_MAIN', 'HBC_HPM_MASTER', 'HBC_HPM_SUPPLIER')  
        GROUP BY stats.OWNER, stats.TABLE_NAME  
        ORDER BY stats.OWNER, stats.TABLE_NAME
```

```

        )
    LOOP
        DBMS_STATS.GATHER_TABLE_STATUS
        (
            OWNNAME => COL_ITEM.OWNER,
            TABNAME => ' ' || COL_ITEM.TABLE_NAME || ' ',
            ESTIMATE_PERCENT => 100,
            METHOD_OPT => COL_ITEM.METHOD_OPT,
            CASCADE => FALSE
        );
    END LOOP;
END;
/

```

2.6 Oracle Import/Export for MAIN, MASTER and SUPPLIER Schemas

A "how to" guideline for exporting and importing Product Manager Schemas from and to an Oracle 11G. Informatica Software Development or Support might ask you to provide the data of a certain installation in order to reproduce bugs or performance issues. This guide shall help support, partners and consulting to provide this.



The terms backup/restore which are usually used in the SQL Server world map to export/import in Oracle. The actual backup/restore functionality of Oracle is bound to a specific oracle instance (more or less) and is only meant for error/data loss recovery, but not to transfer the data to a different database instance.

2.6.1 Preparation

The whole export/import process can not be performed by the sys user at all. Therefore we suggest to create a new user for administrative purposes. In the following sections we will refer to this user as `hlradmin`. The effective permissions a user needs to have differ between the export and import process, thus they are described in the corresponding sections.

Create the hlradmin user

```
CREATE USER "HLRADMIN" IDENTIFIED BY "heiler" PROFILE "DEFAULT" ACCOUNT UNLOCK;
```

Directory Object

The directory object in oracle maps to a physical directory path. This directory is used to store the dumpfile(s). The directory object is needed for the export as well as the import.

Create the directory object

```
CREATE DIRECTORY "HLR_DIR" AS 'c:\heiler\oracle\impexp'
```

2.6.2 Export

User Roles

The `hlradmin` user needs the following privileges to export data

- resource

- connect
- select permissions on all objects for the MAIN, MASTER and SUPPLIER users

Grant needed privileges to hlradmin user

```
GRANT CONNECT TO "HLRADMIN";
GRANT RESOURCE TO "HLRADMIN";
GRANT EXP_FULL_DATABASE TO "HLRADMIN";
```

Execute export datapump

2.6.3 Import

User Roles

The hlradmin user needs the following privileges to perform the import:

- imp_full_database
- create user
- create any table
- create any index
- create any sequence
- create any procedure
- create tablespace

Grant needed privileges to hlradmin user

```
GRANT IMP_FULL_DATABASE TO "HLRADMIN";
GRANT CREATE USER TO "HLRADMIN";
GRANT CREATE ANY TABLE TO "HLRADMIN";
GRANT CREATE TABLE WITH ADMIN OPTION TO "HLRADMIN";
GRANT CREATE ANY INDEX TO "HLRADMIN";
GRANT CREATE INDEX WITH ADMIN OPTION TO "HLRADMIN";
GRANT CREATE ANY SEQUENCE TO "HLRADMIN";
GRANT CREATE SEQUENCE WITH ADMIN OPTION TO "HLRADMIN";
GRANT CREATE ANY PROCEDURE TO "HLRADMIN";
GRANT CREATE PROCEDURE WITH ADMIN OPTION TO "HLRADMIN";
GRANT CREATE TABLESPACE TO "HLRADMIN";
```

Tablespaces

To import a dump file which contains the tablespaces of the MAIN, MASTER and SUPPLIER schemas, you need to have corresponding tablespaces in the target database. You can execute the following pl/sql snippet for each needed schema in order to create the tablespaces.

Create empty tablespaces

```
-- Temporary tablespace
CREATE TEMPORARY TABLESPACE "<SCHEMA_NAME>_TEMP"
TEMPFILE 'c:/heiler/oracle/impexp/<SCHEMA_NAME>_Temp.dbf'
SIZE 1024M REUSE
AUTOEXTEND ON NEXT 128M
MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL
```

```

UNIFORM SIZE 1M;
-- Data tablespace
CREATE TABLESPACE "<SCHEMA_NAME>_DATA"
DATAFILE 'c:/heiler/oracle/impexp/<SCHEMA_NAME>_Data.dbf'
SIZE 1024M REUSE
AUTOEXTEND ON NEXT 128M
MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
-- Separate index tablespace
CREATE TABLESPACE "<SCHEMA_NAME>_INDEX"
DATAFILE 'c:/heiler/oracle/impexp/<SCHEMA_NAME>_Index.dbf'
SIZE 1024M REUSE
AUTOEXTEND ON NEXT 128M
MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;

```

Execute Import Datapump

Parameter file

```

# the directory object
DIRECTORY=HLR_DIR
# the name of the ora dump file, located in the physical file directory mapped to DIRECTORY
DUMPFILE=MYEXPORT01.DMP
# the schemas to be imported
# attention: you have to import all PIM - Server schemas (there're dependencies between the schema objects)
SCHEMAS=CUST_MAIN,CUST_MASTER,CUST_SUPPLIER
# specify the schema name mappings, necessary if schema names to be imported are different from local schema names
# schemas are created automatically if they don't exist
# pattern:
# schema name in dump file:local schema name
REMAP_SCHEMA=CUST_MAIN:HPM_MAIN,CUST_MASTER:HPM_MASTER,CUST_SUPPLIER:HPM_SUPPLIER
# specify the tablespace name mappings, necessary if tablespace names to be imported are different from local tablespace
names
# tablespace are NOT created automatically, you have to create them before
# pattern:
# tablespace name in dump file:local tablespace name (must exist)
REMAP_TABLESPACE=
CUST_MAIN_INDEX:HPM_MAIN_INDEX,CUST_MAIN_DATA:HPM_MAIN_DATA,CUST_MAIN_TEMP:HPM_MAIN_TEMP,CUST_MASTER_INDEX:HPM_MASTER_INDEX,
CUST_MASTER_DATA:HPM_MASTER_DATA,CUST_MASTER_TEMP:HPM_MASTER_TEMP,
CUST_SUPPLIER_INDEX:HPM_SUPPLIER_INDEX,CUST_SUPPLIER_DATA:HPM_SUPPLIER_DATA,CUST_SUPPLIER_TEMP:HPM_SUPPLIER_TEMP
# use this option to only create the sql file
# no import will be executed
# SQLFILE=SQL_IMPORT.SQL

```

execute import cmd

```
impdp USERID=HLRADMIN/heiler parfile="<path to parameter file>" | pause
```



If you get errors regarding insufficient access rights of user HLRADMIN, try to execute with SYSTEM-User:

```
impdp USERID=SYSTEM/heiler parfile="<path to parameter file>" | pause
```




While importing, you might get several Oracle errors regarding the users `..HPM_BULKLOAD` and `..HPM_REPSTORE`:

ORA-39083: Objekttyp OBJECT_GRANT konnte nicht erstellt werden, Fehler:

ORA-01917: Benutzer oder Funktion KRAMP_HPM_REPSTORE ist nicht vorhanden

Fehlerhafte SQL ist:

```
GRANT FLASHBACK ON "KRAMP_HPM_MAIN"."User" TO "KRAMP_HPM_REPSTORE"
```

ORA-39083: Objekttyp OBJECT_GRANT konnte nicht erstellt werden, Fehler:

ORA-01917: Benutzer oder Funktion KRAMP_HPM_BULKLOAD ist nicht vorhanden

Fehlerhafte SQL ist:

```
GRANT ALTER ON "KRAMP_HPM_MAIN"."UnitSystemUnitMapLang" TO "KRAMP_HPM_BULKLOAD"
```

They do not have any impact on the import result, so you can ignore them.

3 Server Operation

Combines maintenance, tuning and troubleshooting for the Product 360 application server.

3.1 Maintenance Tasks

- Backup Strategy
- Background job should be configured to be automatically deleted after a certain amount of time. See [Server Job Maintenance](#) for details.

3.2 Monitoring

Monitoring tools which can communicate using the SNMP protocol can be used to monitor the Product 360 application servers. Please see [SNMP Monitoring](#) for configuration details on the SNMP functionality.

3.3 Tuning

Here is the list of settings and configuration options that may affect system performance. These settings depend on system workloads, hardware and environment.

3.3.1 Memory Settings

As expected, the memory settings are crucial for the application server of the PIM - Server. You can configure the available memory in the `<installation root>/_environnement.conf` file. Please note that you can configure the maximum amount of memory (`MEM_MAX`).



The minimum amount of available heap memory will always be set to the same value as the `MEM_MAX`, since for an application server it makes no sense otherwise.

3.3.2 Garbage Collection

Garbage collection configuration is a very complicated thing to do. Informatica PIM Software engineers spend a lot of time on site with customers to find the right garbage collection algorithm and settings. All needed parameters have already been

adjusted by us, the only configuration which is needed for every customer installation is the number of garbage collection threads which can be used. PIM - Server uses a parallel garbage collection algorithm, therefore you can decide how many threads the garbage collection can use. The GC_THREADS setting in the <installation root>/_environment.conf file should be set to about 75% of the available CPU cores. For example, in case you have 8 Cores (including hyper-threading cores), you would set this to about 6 threads.



In case you need to further adjust the garbage collection settings you can do so in the <installation root>/service/wrapper.conf file in the GC_OPTIONS section. Please note that the console_debug.cmd launch file will use different garbage collection algorithms since the debug mode is not able to handle the ones we use for production.

Enable GC activity log

In order to analyze the GC events in detail you can enable the GC activity log for each Product 360 Server. Therefor add the following parameters in wrapper.conf:

```
wrapper.java.additional.24 = -XX:+PrintGCDetails  
wrapper.java.additional.25 = -XX:+PrintGCDateStamps  
wrapper.java.additional.26 = -Xloggc:./logs/gc_%p.log
```

(ensure, that the number at the end of wrapper.java.additional.- parameters is continuously incrementing over the whole wrapper.conf file)

After that you have to restart the Product 360 Server. From then on, each Server start produces a new gc_<pid>.log file (whereas <pid> is the OS specific Process ID of the P360 Server) in the server's log folder. This gc_<pid>.log file can then be analyzed e.g. online via <http://gceasy.io/>.

3.3.3 Communication Framework

Number of worker threads in server network settings

3.3.4 Mass-Data Parallel Processing

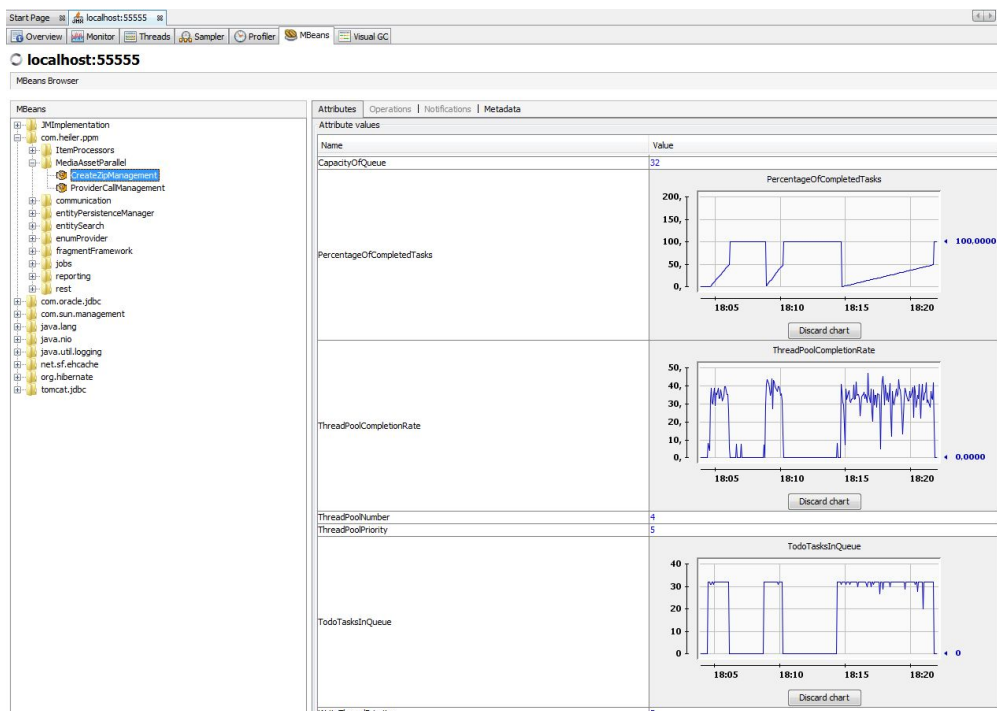
Item Processor Threadpool

3.3.5 Media Asset Parallel Management

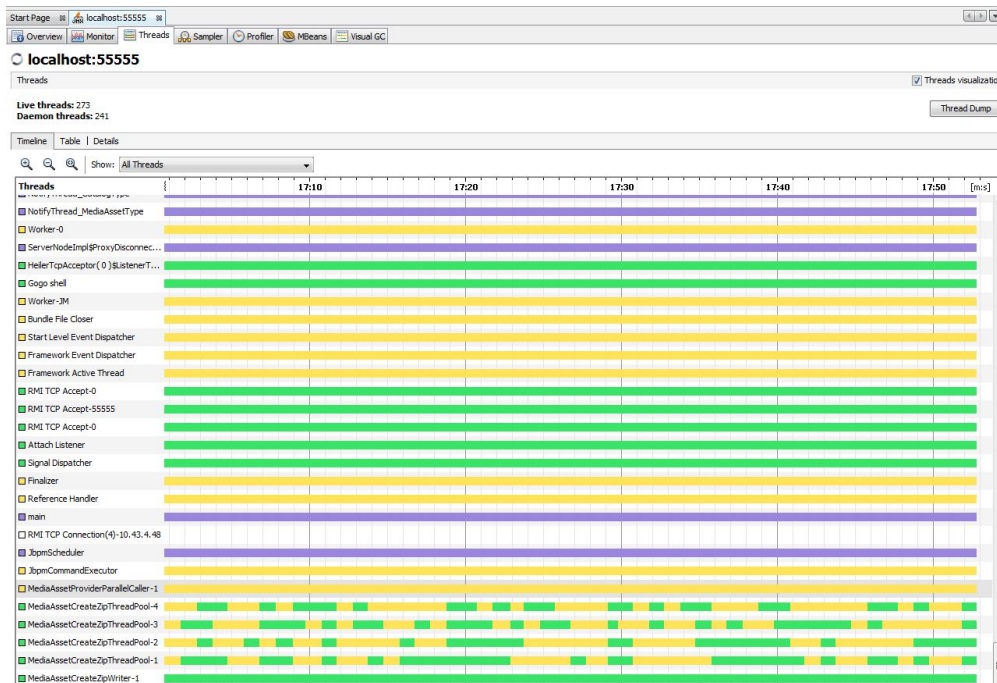
There are several registered MBeans under the category **com.heiler.ppm** → **MediaAssetParallel** with the JMX interface which are applied to observe and improve the performance in run time.

CreateZipManagement

It provides the management of the thread pool and other relevant parameters(e.g. the size of the BlockingQueue to avoid OutOfMemoryError) by zip creation phase.



The following screenshot indicates the corresponding threads in run time.



MediaAssetProvider special MBean

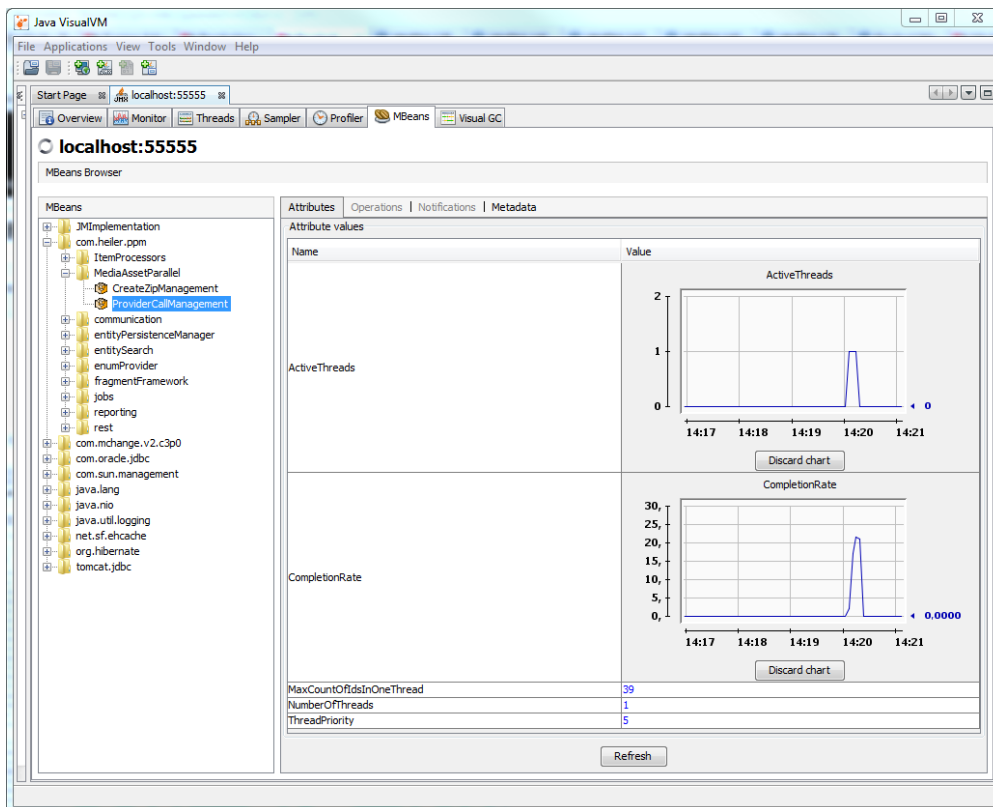
MediaAssetProviderParallelManager

i It was named as "ThreadPoolManagement", and changed as "ProviderCallManagement" and then as "MediaAssetProviderParallelManager" since 7.1.01.00.

It provides the management of the multi-threads call for the corresponding MediaAssetProvider methods.

A special thread pool is applied to improve the performance for some methods of the default IMediaAssetProvider, such methods should have any parameter composed of a long id list (e.g. getUNCpaths(String[] identifiers, ...)). Currently only the PIM - Media Manager 'HMM' MediaAssetProvider supports it. This thread pool settings can be observed and adjusted with the JMX interface during runtime. It is meanful for some scenarios like export with mass data.

The following screenshot indicates the thread pool utilization. The user can adjust the values of the attributes "MaxCountOfIdsInOneThread" and "NumberOfThreads" to improve the performance, if the "ActiveThreads" reaches always the maximum value and the "CompletionRate" is but too low.



HeilerClassic special MBeans

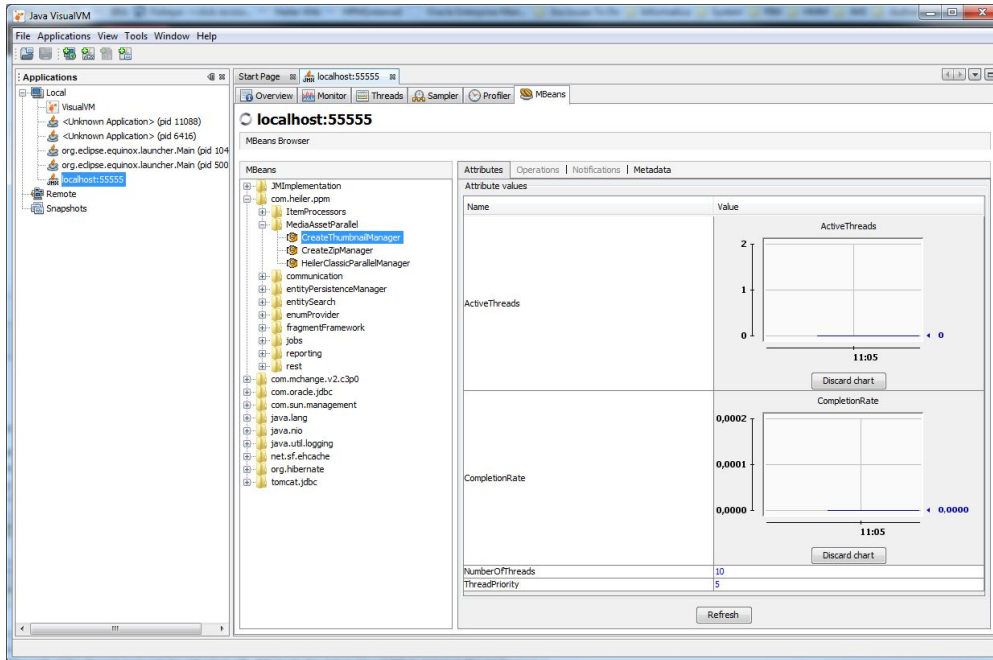
i since 7.1.01.00 HeilerClassic registers two MBeans which can be also observed and adjusted with JMX interface.

CreateThumbnailManager

It provides the management of thread pool which schedules all graphicsmagick convert jobs to create thumbnails. The default NumberOfThread is 10.

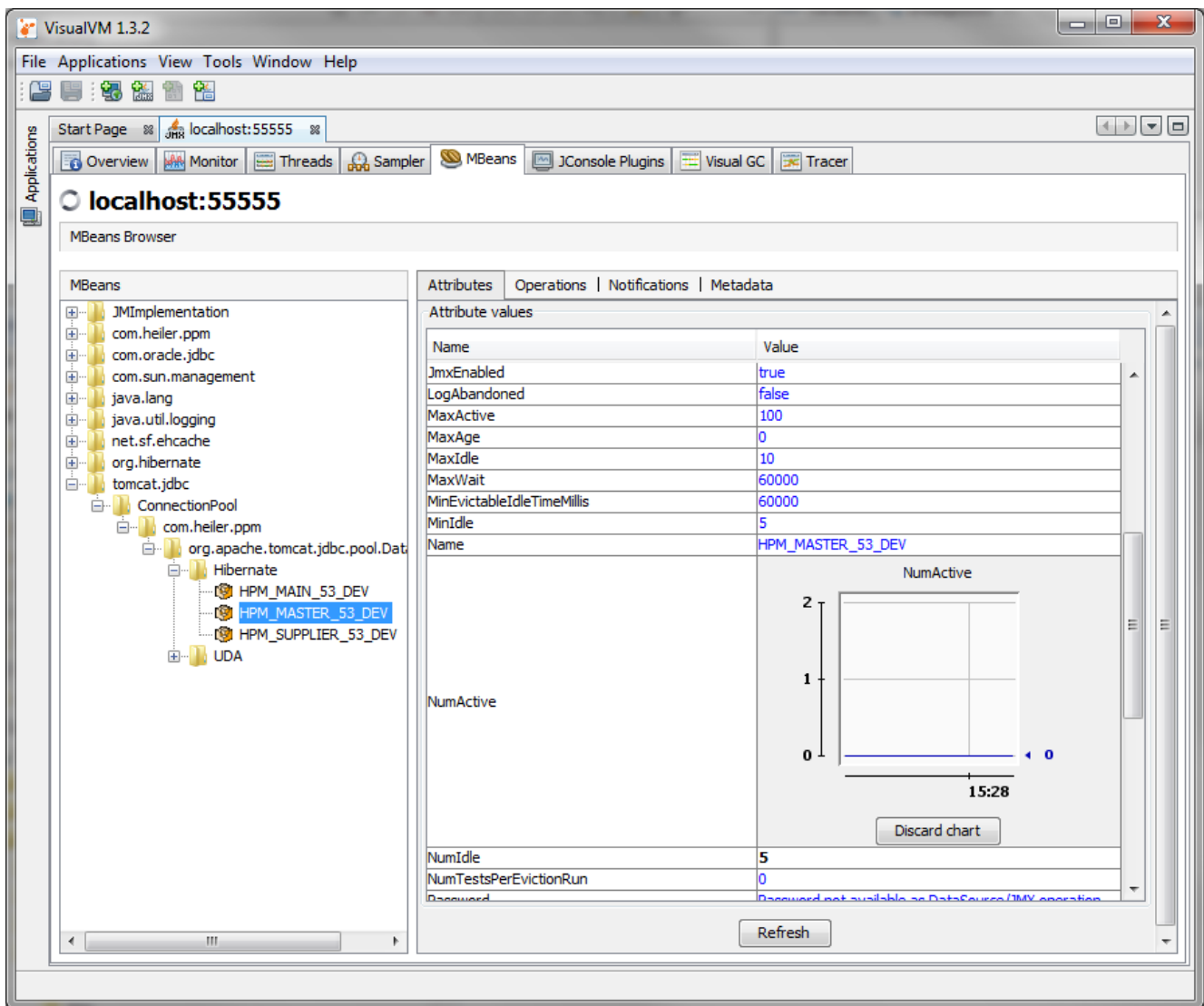
HeilerClassicParallelManager

It provides the management of thread pool which schedules tasks which can be performed parallelly for the corresponding MediaAssetProvider methods(e.g. GetSubCategoriesTask).



3.3.6 Database Connection Pool

The database connection pool settings can be adjusted in the corresponding configuration files. More detailed information on the configuration of the connection pool can be found here [DB Connection pool](#). During runtime the connection pool utilization can be observed with the JMX interface (see Screenshot). Most important here is the maximum used attribute which indicates if the maximum number of connections has already been used. You might need to increase the maximum connections in case the limit is reached often, or continuously.



This screenshot shows the `NumActive` attribute of the `ConnectionPool` for the `HPM_MASTER` connection pool. Currently there are zero connections in use. When this number reaches the `MaxActive` limit, the application load might be too high for the number of connections, adjusting the connection pool is necessary then. Additionally to `NumActive` and `MaxActive` you might want to check the `WaitCount`, which indicates how long a thread needs to wait until it acquires a connection from the pool. In case there are enough connections left in the pool, this number should be about zero.



The PIM - Server has two connection pool configurations which are of interest here, the `Hibernate` and the `UDA` one. The `Hibernate` pool is used for detail model persistence, therefore it's utilized in detail model modifications, import and merge. The `UDA` pool is used mostly for read-only mass-data retrieval like list model based functionality such as table views and export.

3.3.7 Logging

PIM - Server has a central configuration file for all logging activities in the server, as well as in the client. By default the logging trace levels are set to `INFO`, thus logging messages with severity `INFO`, `WARNING` and `ERROR` will be logged in the

console log, errors additionally in the error.log file. Please make sure that TRACE or DEBUG severity levels are disabled in production environments, since they dramatically slow down the performance. Logging is configured in the <installation root>/configuration/hpm/log4j.xml file.

3.3.8 Import

Preferences regarding CPU usage

Especially in Single Server scenarios in which the Import Jobs are executed throughout the day, it might be necessary to limit the CPU resource consumption of a single import process.


For this there are several preferences available which can be set in the plugin_customization.ini file.


```
# Specifies the maximum number of threads used for importing objects.
# Default is twice the number of cores.
# Allowed value is an integer greater than 0.
# com.heiler.ppm.importer5.core/importer.maxThreads=2

# Factor for the number of parallel threads.
# The number of parallel threads is computed by multiplying the number of cores and this factor.
# The result is rounded.
# Value is a double; default is 2.0
# This value is only considered if importer.maxThreads is not set.
com.heiler.ppm.importer5.core/importer.parallelThreadsFactor = 2.0

# Allow to limit number of the import jobs running in parallel
# Only the actual database import (phase 2) is limited.
# -1 means unlimited (default)
com.heiler.ppm.importer5.core/importer.maxNumberOfDatabaseImportsAtSameTime = -1
```


Additionally to these global settings, a user can activate the so called "debug mode" for the Import by pressing **Shift + F12** when **inside of the Import Perspective** in the Desktop Client. With activated debug mode, additional settings are available when scheduling the import.

 Import



Schedule import

Specify when the import is to be executed.



As soon as possible ▾

Time

Structure

Heiler Standard ▾

Import mode

Import or update all objects ▾

Error mode

Tolerant (object is imported as far as possible) ▾

☐ Perform test run before import, cancel import on errors

Test Settings

☐ Generate gold standard file

☐ Test against gold standard file

Number of threads:

☐ Use always bulkload

☐ Use never bulkload

Bulkload size:

☐ Clear catalog

Raw performance test (number of objects):

Storage mode for intermediate memory (0-2):

OK

Cancel

⚠ Please do not change any of those additional settings except the Number of Threads!

Preferences regarding memory usage

When importing CSV or Excel files, per default all rows are read into heap memory in order to presort them. This speeds up processing if the data for one item is stored in rows that are spread. So when importing large CSV or Excel data files, consider to increase the job servers' heap size accordingly.

If presorting is not needed (because the data files are already presorted) or desired, it can be disabled by adding the following preference in the server's `plugin_customization.ini`: **`com.heiler.ppm.mapper.core/presortRowsDefault = false`**



Importing large data files might become slower, when presorting is deactivated (see preference above) and the import data file is not already presorted.

3.4 Trouble Shooting

- Log files
- What's needed for a support call

3.4.1 Anti-Virus Software affecting performance

In the past we had some customer performance issues which were bound to the anti-virus scanning on the application server. Although we can not say that this is always the case for all deployments or all anti-virus products, we do recommend to exclude the PIM relevant directories from the scanning if applicable by the customers local security policy. If scanning can't be avoided in your situation, you should consider to reduce the lifetime of import or export jobs. The shorter they are, the less files will be in the import/export directories - which might increase scanning performance again.

3.4.2 Media Asset Provider (Classic Provider)

No preview are visible

Ensure following things:

1. Is the file extension unsupported?(E.g. pdf, ps, eps files because the ghostscript is not supported in PIM any more)
2. Is the graphicsMagick(official verified version GraphicsMagick-1.3.14-Q16) installed?
3. Check that the `blackListExtensions` parameter of the `C:\heiler\server\configuration\HPM\plugin_customization.ini` file does not contain the file extension
4. Has the user who has started the application server write access to the media folder defined in the "Media Asset Server (MAS) Settings" section of the `server.properties` file?
Is it possible to open the image files from another computer by the explorer and another program?
5. Important: An unsuccessful rendering of a thumbnail leads to an entry in the database table `MediaAssetFile` of the "[db_prefix]_MAIN_[db_suffix]" database.
Therefore, after a "subsequent" adjustment of these points this table must be modified. Set the bit-field "SupportThumbnail" to NULL.

Check 3rd party tools:

The technical background is that PIM - Server runs on a 64-bit machine as a 64-bit Java process.

PIM - Server starts `gm.exe` (Graphicsmagick) in a several process space (Graphicsmagick is a 32-bit process).

3.4.3 Obtain a Memory Dump (Heap Dump) from the Java Virtual Machine

There are several ways to obtain a heap dump during runtime.

- Heap dump on out of memory
By default all Product Manager launch configurations (development and runtime) contain the Java parameter `-XX:+HeapDumpOnOutOfMemoryError` which performs a heap dump when an out of memory error occurs.
- Heap dump on Ctrl+Break
Beginning with Java Version 1.5.0_16 the Java VM supports the `-XX:+HeapDumpOnCtrlBreak` parameter which will create a heap dump when you press Ctrl + Break in the Java console. For this the PIM - Server must be started with the `debug_console.cmd` file.
- Heap dump via `jcmd` utility
This console utility is included in JDK. Use `"jcmd <processId> GC.heap_dump <filename>"` for creating a heap dump, e.g. `"jcmd 14904 GC.heap_dump D:\heapdump.hprof"`
For details see <https://docs.oracle.com/javase/8/docs/technotes/guides/troubleshoot/tooldescr006.html>.

We suggest to use the Eclipse Memory Analyzer to analyze the Heap Dump, especially in case you have large heap sizes. It's the only tool we know so far which is able to open large heap dumps at all (you will need a lot of memory during the indexing of the heap dump in the memory analyzer - but this must not be physical memory 😊)

3.4.4 Enhanced Logging for Assortment Content Calculation

In case there are issues in terms of an assortment's content and there are items included/excluded which do not fit to the defined assortment rules, please add the following lines to the server's log4j.xml and execute an assortment update:

Assortment Log

```
<appender name="AssortmentFileAppender" class="org.apache.log4j.RollingFileAppender">
  <param name="File" value="logs/.assortment.log"/><!-- = Runtime directory -->
  <param name="MaxFileSize" value="10240KB"/> <!-- = 10MB -->
  <param name="MaxBackupIndex" value="10"/>
  <param name="Threshold" value="TRACE"/>
  <layout class="org.apache.log4j.PatternLayout">
    <!-- The default pattern: DateTime Priority [Thread] [Category] Message\n -->
    <param name="ConversionPattern" value="%d{ISO8601} %-5p [%t] [%c{1}] %m%n"/>
    <!-- Pattern to output the caller's file name and line number -->
    <!-- param name="ConversionPattern" value="%d{ISO8601} %-5p [%t] (%F:%L) - %m%n"/>
  </layout>
</appender>

<!-- Default logging level for assortment modifying and updating.
This category uses its own appender and does not route log events to other appenders.
Use "TRACE" to activate assortment loggers -->
<category name="assortment" additivity="false">
  <priority value="TRACE"/>
  <appender-ref ref="AssortmentFileAppender"/>
</category>
```

The enhanced logging will write detailed messages regarding assortment content calculation to *.assortment.log* file located in the server's logging folder. Please provide this log file to Informatica support.

3.4.5 Server and client in different time zones

At the moment it is only possible to use P360 server and client in different time zones with a max time offset of 23 hours.

Theoretically it is possible to have a server client constellation with a 26 hour offset. This is not supported. More than 23 hours offset ends up in exceptions like

```
java.lang.IllegalArgumentException: Timestamp format must be yyyy-mm-dd hh:mm:ss[.fffffffff]
    at java.sql.Timestamp.valueOf(Timestamp.java:237)
    ...
```

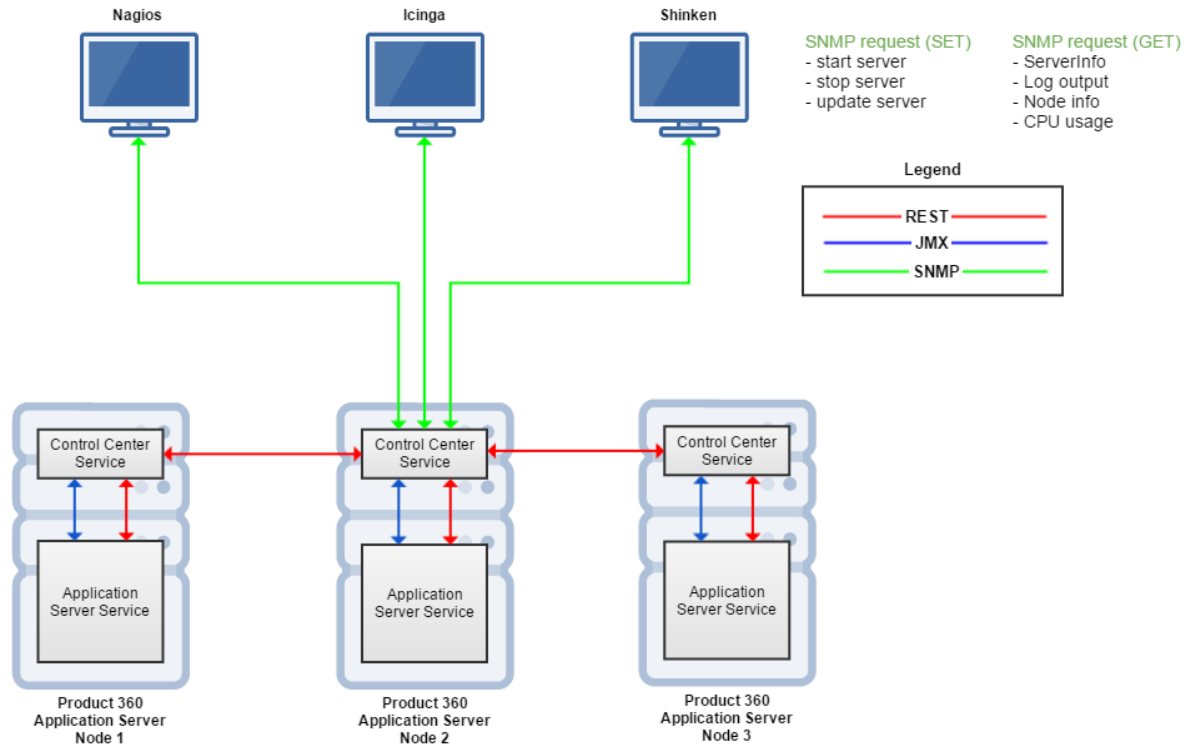
3.5 SNMP Monitoring



SNMP Monitoring feature will be available with the version **8.0.03.00**

SNMP (Simple Network Monitoring Protocol) is an open standard for monitoring applications inside a bigger IT infrastructure. This page provides details as how to set up this monitoring feature for Informatica MDM - Product 360.

3.5.1 Overview



The SNMP capable tools like Icinga, Nagios or others connect to a single Control Center service on any of the application server nodes. All control center services are connected with each other and therefore are able to provide the monitoring data for all the nodes in the Product 360 application server cluster. Communication between the Control Center services is based on REST, the communication between the Control Center service and its Application Server Service is done using REST and JMX.

3.5.2 Default Configurations

The system controller is delivered with standard configurations. Therefore these configurations can be modified in the configuration files.

Agent

- By default an SNMP agent is accessible through the port 161. This port number is also used by the system controller as default configuration.
- The system controller supports only communication using SNMP Version 3.
- The default protocol used by the system controller for communication through SNMP is UDP.
- Every notification is disabled by default except notification about the server status (STOPPED, STOPPING, and STARTING, RUNNING)

Network Management Station

- The control center provides by default configurations for a single Network Management Station (single monitoring). This station includes an SNMP manager in order to send SNMP request and receives traps. The Network Management Station is located by default on the same host where the control center is running and traps are sent to the port 162.
- The default configuration of the Network Management Station offers 2 default users: one with a lower security level for sending GET request and receiving traps and the second user with a higher security level for sending both SET (start, stop and update server) and GET request and receiving traps. One of these users has the default username `controller`, the password `controller` and uses the authentication protocol SHA. The second user has the default username `Administrator`, the password `Administrator` and a private password `Administrator` and uses the authentication protocol SHA and the private protocol AES. To perform server remote control, users with this security level should be used.

Managed Objects

Every managed object is enabled by default to the managed server and a trap is sent once by events like Server STOPPED, STOPPING, STARTING, RUNNING. If you want traps to be sent by other events you should configure it (see section managed objects configuration).

OID Configuration

The generated OID configuration of each managed object is the result of the concatenation of the enterprises OID, the server OID and the location of the objects value in the MIB. By default of the enterprise OID is `.1.3.6.1.4.1.200`. The part `iso.org.dod.internet.private.enterprise(.1.3.6.1.4.1)` is the prefix. The number 200 represents the default enterprise identifier. Please change the 200 to your standard enterprise Id. If you don't have an enterprise identifier, please visit the link <http://pen.iana.org/pen/PenApplication.page> to request a unique identifier for your enterprise. Each managed server has an OID. If your environment has 2 servers, the first will have the OID 1.1 and the second 1.2. If you already have an SNMP Agent in your system that uses those OID, you should provide another to prevent MIB conflicts. (See section Modifying configuration)

Generation of OID configuration

At the program start a file is generated within the configuration directory with the name `"oid-data-config.properties"`. At the first successful connection of the system controller with each managed server a section is added to this file with the generated managed object OIDs. This operation is done for all managed servers. The generated OID are also reachable through SNMP by requesting the server description using the description OID. Each server has its description OID. It is found in the `oid-data-config.properties`.

Server Remote Command

All servers can be started, stopped or updated remotely using the remote command OID. To perform this operation you need a user that has the security level 2. The default remote command OID is `.1.3.6.1.4.1.200.3`. This OID is generated and is found in the servers description. The value expected for this object is a string that uses the following syntax: `command (start/stop/update) + "," + SERVER IDENTIFIER (e.g. pim-server1) + "," + MODE (kill, grace)`. The argument MODE is needed just to stop the operation. All given arguments should be separated by a semicolon. Errors in system controller during remote command operations are notified to Network Management Stations via traps. The trap OID of this error is a concatenation of the generated remote command OID and the sub string `.1`. If the generated remote command OID is `.1.3.6.1.4.1.200.3` the trap error Id will be `.1.3.6.1.4.1.200.3.1`. This information is useful for reading traps.

Trap

The trap is generated to announce an event. The generated traps by the Control Center uses a structure. It is a list of information about the event. The first element of this list represents the affected server and the second describes the event. The third element is the event level (Critical, Warning, and Ok) and the last the system time. The event level is used for management application to represent services with different colors (red, yellow and green) depending on the event level. For Monitoring Applications like Icinga or Nagios this level is represented by digits from 0 to 2 with following meanings:

- 0 = OK
- 1 = Warning

- 2 = Critical

Let us take an example. If at 7:55 the system detects that the server identified by PIM-SERVER1 is stopped. It will send the Trap with following information [PIM-SERVER1, STOPPED, 2, 7:55]. If 5 minutes after the stop the system detects that this is running (maybe the system administrator has start it after the event), a trap will be sent with following information [PIM-SERVER1, RUNNING, 0, 8:00]. Every trap has an OID. But all trap for the same event has the same OID. For each managed object the generated OID is used both for querying objects value via SNMP and for sending traps.

E.g. if the generated OID for the managed object `serverStatus` of the server identified by PIM-SERVER1 is . 1.3.6.1.4.1.200.1.1.2.1 and the notification is enabled, all Trap notifications about this object will be sent using the same OID. The information about the structure of the trap sent and its OID are useful because you will need these for translation the trap and submit the result to the Network Management Station via passive check.

Enable SNMP

To enable SNMP, go to the configuration directory and find the file `NetworkConfig.xml`.

Edit this file as follow:

- Go to the xml element "`snmpdataconfigfile`"
- Remove the line which says : "`<!-- settings for snmp`"
- Also remove the line below the xml element "`snmp`", which says "`-->`"
- Save the file

Now find the `ClusterixConfig.xml` in the same directory and edit that file as follow:

- Again remove the line that says: "`<!-- settings for snmp`"
- Also remove the line at the bottom below the xml element "`NetworkManagementSystems`" that says " `-->`"
- Save the file and restart the application

Run application

After successful start of the Application, you should see the message Agent access `x.x.x.x:yyy` where `x.x.x.x` is the IP address of the host on which the agent is running and `yyy` the port number on which it is reachable. If this message appears, the system controller is ready for monitoring.

3.5.3 Modifying configurations

Change the agent's port

Use the VM argument `-Dclusterix.snmp.port` to change the agent's port. E.g. to set the port to 165 set the argument as follows `-Dclusterix.snmp.port=165`.

Network Management Station

Change the Trap receiver's port

1. Go to the configuration directory
2. Find the file `ClusterixConfig.xml`
3. Open it and go to the XML- root tag `NetworkManagementSystems` ->Inside this tag you will see a XML-tag with the name `NetworkManagementSystem`
4. Find his XML- element `<port>162</port>`
5. Change 162 to the port number you wish to assign.
6. Save the file and restart the application

Change the host address

1. Go to the configuration directory
2. Find the file `ClusterixConfig.xml`
3. Open it and go to the XML- root tag `NetworkManagementSystems`.

4. Inside this tag you will see a XML- tag with the name NetworkManagementSystem
5. Find his XML- element <host>127.0.0.1</host>
6. Change 127.0.0.1 to the IP address of your management station.
7. Save the file and restart the application.

Change user credentials

1. Go to the configuration directory
2. Find the file ClusterixConfig.xml
3. Open it and go to the your Network Management System definition
4. Go to the root element Users
5. Inside this tag you will see a XML- tag with the name User
6. Go to a user definition and edit his properties:
7. you want to change the username, find the XML-Element <name>admin</name>
8. Change admin to a preferred username.

If you want to change the password, find the XML-Element In the XML-Element <password>Informatica</password> Change **Informatica** to a preferred password. If you want to change the private password, and the XML element In the XML element <privatePassword>Informatica</ privatePassword> Change **Informatica** to your preferred private password.

At the moment, the private password is encrypted using the Advanced Encryption Standard with a 128 bit key.

Add a user to the Network Management System configuration

1. Go to the configuration directory
2. Find the file ClusterixConfig.xml
3. Open it and go to the your Network Management System definition Go to the XML- root Users
4. Inside this tag you should find a XML- tag with the name User
5. Duplicate a user definition and edit the properties as explained above. Leave the XML element privatePassword, and privateProtocol blank (<privatePassword> </ privatePassword>, <privateProtocol></ privateProtocol>)
- if the user you want to add belongs to the security level 1.
6. Save the file and restart the application.

Remove a user from the Network Management System configuration

1. Go to the configuration directory
2. Find the file ClusterixConfig.xml
3. Open it and go to the your Network Management System definition
4. Go to the XML - root Users
5. Inside this tags you will see a XML- tag with the name User
6. Remove the tag of the user you want to delete to your system.
7. Save the file and restart the application. Please note that the SNMP Agent needs at least one user with security level 2 to start.

Change OID Configuration

Server OID

1. 1. Go to the configuration directory
2. 2. Find the file NetworkConfig.xml
3. 3. This file contains the Network configuration of all managed server. Each managed server is represented by an XML-tag node.
4. 4. Find the node of the server which the configuration should be edited.
5. 5. Find the XML-Element <snmp oid=1.1>
6. 6. Replace 1.1 with the Id you want. Please note that the Id consists of separated numbers.
7. 7. Save the file and restart the application.

Enterprise OID

The enterprise OID entry is found in the Object configuration file of each server.

1. Go to the configuration directory
2. Find the file NetworkConfig.xml .This file contains the Network configuration of all managed server. Each managed server is represented by the XML-tag node.
3. Find the node-definition of the server which the configuration should be edited.
4. Find the XML-Element <snmpdataconfigfile name="file name">
5. Check the assigned file name and remember it
6. Look this file in the configuration directory and open it
7. Go to the XML-tag general
8. Edit the XML-element enterprise(change the last number of the property id (200) to your enterprise ID)
9. Save the file and restart the application.

Modifying Managed Object configuration

1. Go to the configuration directory and find the file NetworkConfig.xml
2. This file contains the Network configuration of all managed server. Each managed server is represented by the xml-tag node.
3. Find the node-definition of the server for which configurations are to be edited.
4. Find the xml-Element <snmpdataconfigfile name=file name>
5. Check the assigned file name and remember it
6. Look for the file in the configurations directory and open it
7. Go to the XML-tag rest for rest managed objects and jmx for jmx managed objects:

Enable/disable a managed object

1. Check the property append
2. Set the value of this property to true to enable the object or false to disable the object.

Attention: Enabling/disabling objects is a critical operation because it affects all your monitoring system configurations (your management application and the system controller) if you already had a running monitoring solution. After this operation all or some managed objects become a new OID depending on their position in the oid-config-file. Please compare the new generated OID of each managed object with the OID you previously configured in your monitoring application to be sure that it is the same. If it is not the same use the new generated OID of the object. Otherwise you can have the following problems/errors in your monitoring application:

- Some OID are not found (If you disabled managed objects)
- Some OID are found but their value belong to another objects.

Enable/disable notification for a managed object

This works if the property append is set to true

1. Check the property notification_enabled
2. Set the value of this property to true to enable notification or false to disable the notification.

Customizing notification for a managed object

This works if the properties append and notification are set to true.

- If the property type equals to String, assign a value to the property pattern. This represents the value you expected for this managed object. If another value is found over the time, a Trap-notification will be send to all register Network Management Station (see section Network Management Station above) with the actual value of the object.
- If the value of the property type different to String (e.g. int or long).Please assign a range to the property range This represents the possible values you expect for the managed object over the time.

Note: if a value is found over the time, that do no belong to the range, a notification will be send register Network Management Stations with the actual value of the object. To give the range follow the syntax [+min value+.. +max value +]. The bracket [] means that the minimal and the maximal value belong to the range. Attention: if you have a syntax error during range configuration or the value you assigned to min is greater than the value of max the system controller will start but the configured notification will be ignored. Please check the log file for error description.

Add notification to managed object

You can add notifications to every managed object you want. But be sure that append is set to true.

- To add a notification, add the following property to the managed object: `notification_enabled=true`.
- If this object has the type String adds the property `pattern=`The expected value of this object otherwise adds the property `range= [min...max]` (see section customizing notification for a managed object above).

Advanced configuration

Multiple Monitoring

The standard configuration allows the system controller to send trap just to one Network Management Station. To enable multiple monitoring do the following:

1. Go to the configuration directory
2. Find the file `ClusterixConfig.xml`
3. Open it and go to the XML- root `NetworkManagementSystems`. In this tag you will see a XML- tag with the name `NetworkManagementSystem`
4. Duplicate this tag and edit like explained above (see section Network Management Station).
5. Save the file and restart the application. Every Trap will be sent to all Network Management Stations listed in this configuration file. If a trap is sent to a Network Management Station and an acknowledge is not received, the trap will be sent twice. If the second trap is lost, an error will be logged.

Add a new Managed objects

This works just for the JMX communication.

1. Go to the configuration directory and find the file `NetworkConfig.xml`. This file contains the Network configuration of all managed server. Each managed server is represented by the xml-tag node.
2. Find the node-definition of the server which the configuration should be edited.
3. Find the xml element `<snmpdataconfigfile name=file name>`
4. Look the assigned file name and remember it
5. Look this file in the configuration directory and open it
6. Go to the XML-tag JMX
7. Duplicate a JMX-Object definition
8. Edit as follow
 - Object name: set the MBean Object Name. You can use the JConsole Windows (see <http://docs.oracle.com/javase/tutorial/jmx/remote/jconsole.html>) to have the right name of the MBean Object.
 - Element name: set the attribute name of the MBean Object to query.
 - Set the type of this attribute as showed in the JConsole Windows. Note: If the type is String you cannot have the graphic visualization of this object in your Network Management Station. It is done just for performance data.
 - Set access to `read_only` or `read_write` depending on your choice. Note that if you set the access type to `read_write` the value of this object can be change via SNMP-Set request. And you should need special rights for this.
 - Set `append` to true to enable the managed object.
 - Add notification as explained in the section "Add notification" above if you need any
 - Save the file and restart the application.

Note: If the given MBean object name of the managed object that you try to add or the MBean attribute you query does not exist or is not correctly spelled, the managed object will be ignored and an error will be logged.

Add/remove managed Server

1. Go to the configuration directory and find the file `NetworkConfig.xml`. This file contains the Network configuration of all managed server. Each managed server is represented by a xml-tag node.
2. Duplicate a node-definition and edit it as follow:
 - Go to his xml element JMX and give the port number of this server
 - Go to his XML-element OID and give an SNMP OID to this server (e.g 1.3)

- Go to the XML-element `snmpdataconfigfile` and replace the file name with `snmp-data3.xml`
- Duplicate a `snmp-data-denition` file and rename it (e.g `snmp-data3.xml`)
- Configure managed objects as explained above
- Save the file and restart the application.

A new managed server is added to your system controller and you can query the value of the managed object and receiving traps using the generated OIDs.

3.6 DB Connection pool

PIM Core database workload consists of a large amount of parallel requests. To handle such workload PIM Core is using DB connection pool. Connection pool holds established jdbc connections and returns them to the pool when higher level data access layer such as Hibernate or UDA close them. Such pool helps to save time required to establish and prepare jdbc connection. However connection pool increase memory consumption. PIM Core maintains one connection pool per data source (MAIN,MASTER,SUPPLIER) and per data access module (UDA,Hibernate), total 6 connection pools. Quartz and JBPM maintain their own connection pools. Due to performance reasons PIM Core is using `tomcatjdbc` connection pool which is optimized for concurrent access. Popular implementations DBCP and c3p0 are not suitable for PIM Core workloads.

It is important to configure connection pool appropriately. See <https://tomcat.apache.org/tomcat-8.0-doc/jdbc-pool.html> for more detail on tomcat jdbcpool configuration options.

To improve jdbc performance the connection pool is capable of configuring prepared statement caches. This means prepared statements are cached per connection, and please note that this will significantly increase memory consumption! Also the connections need to be configured in a way that they do not remove idling connections too fast otherwise you will render the cache useless.



JMX

Use hibernate and connection pool JMX managed beans to analyse workload and adjust values on production systems, see also [Tuning Advisory](#)

3.6.1 High performance connection pool configuration example

The default configuration should be sufficient for regular workloads. However in context of having many concurrent users and/or system jobs (import, export, data quality, etc.) the connection pool might become a bottle neck. The connection pool will throw an error in case the connections are exhausted, but this will not necessarily happen because there is a wait time until the error will be thrown. To adjust connection pool settings, the following data access template files must be adjusted.



Oracle

When running PIM on Oracle do not enable caching of prepared statements unless you make sure that initialization parameter **OPEN_CURSOR** is set high enough. The limit is counted globally for all connections that belong to the same pool.

Hibernate

The template files when running PIM on Oracle for the Hibernate connection pools are located in the **server\configuration\HPM\database** folder:

- `main.properties.template.ORA11g`
- `master.properties.template.ORA11g`
- `supplier.properties.template.ORA11g`

In case you are running PIM on Microsoft SQL Server, the corresponding template files have the MSSQL2008 suffix. Make sure to modify the template files since PIM will always generate the real property files during server startup.



Configuration for 250 maximum active connections

```
# increase validation interval (value in milliseconds) to avoid permanent connection
check which is unnecessary in today's networks
hibernate.tomcatjdbc.pool.validationInterval = 120000

# increase eviction interval (value in milliseconds) when the pool should check which
idling connections could be removed from the pool
hibernate.tomcatjdbc.pool.timeBetweenEvictionRunsMillis = 30000

# increase idle time (value in milliseconds) before the connection may be removed
from the pool
hibernate.tomcatjdbc.pool.minEvictableIdleTimeMillis = 1800000

# increase the amount of initial connections since creating new connections is a very
time intensive task
hibernate.tomcatjdbc.pool.initialSize = 50

# increase the amount if minimum idling connections which will be kept open at any
time to fully utilize statement cache and avoid hard parsing repeating queries
hibernate.tomcatjdbc.pool.minIdle = 125

# increase the amount of maximum idling connections to avoid closing idling
connections during peak times
hibernate.tomcatjdbc.pool.maxIdle = 250

# increase the amount of maximum idling connections to avoid the situation that PIM
processes are starving from available connections
hibernate.tomcatjdbc.pool.maxActive = 250

# lower the maximum wait time (value in milliseconds) until the connection pool will
throw an exception that the pool is exhausted
hibernate.tomcatjdbc.pool.maxWait = 10000

# enable jdbc interceptor for statement cache, increase max value if required by
application load pattern
hibernate.tomcatjdbc.pool.jdbcInterceptors =
ConnectionState;StatementFinalizer;StatementCache(prepared=true,callable=false,max=25
0)

# Oracle only: this will force the jdbc driver to enable its builtin statement cache
to avoid hard parsing repeating queries
hibernate.tomcatjdbc.pool.connectionProperties =
oracle.jdbc.implicitStatementCacheSize=250

# Oracle only: prepared statements (INSERT or UPDATE) can be batched if batch_size is
> 1
hibernate.jdbc.batch_size = 50
```

All other settings should be left untouched. Increasing the amount of connections and/or statement cache size will result in higher memory consumption, so make sure that application server and database memory target is sized properly.

UDA

The template file when running PIM on Oracle for the UDA connection pools is located in the **server\configuration\HPM\database\uda** folder:

- uda-jdbcpool.xml.template.ORA11g

In case you are running PIM on Microsoft SQL Server, the corresponding template file has the MSSQL2008 suffix. Make sure to modify the template file since PIM will always generate the real property files during server startup.

This example will adjust the same values like the Hibernate connection pool (please note that this is a xml file):



Configuration for 250 maximum active connections

```
<!-- increase validation interval (value in milliseconds) to avoid permanent
connection check which is unnecessary in today's networks -->
<property name="validationInterval">120000</property>

<!-- increase eviction interval (value in milliseconds) when the pool should check
which idling connections could be removed from the pool -->
<property name="timeBetweenEvictionRunsMillis">30000</property>

<!-- increase idle time (value in milliseconds) before the connection may be removed
from the pool -->
<property name="minEvictableIdleTimeMillis">1800000</property>

<!-- increase the amount of initial connections since creating new connections is a
very time intensive task -->
<property name="initialSize">50</property>

<!-- increase the amount if minimum idling connections which will be kept open at any
time to fully utilize statement cache and avoid hard parsing repeating queries -->
<property name="minIdle">125</property>

<!-- increase the amount of maximum idling connections to avoid closing idling
connections during peak times -->
<property name="maxIdle">250</property>

<!-- increase the amount of maximum idling connections to avoid the situation that
PIM processes are starving from available connections -->
<property name="maxActive">250</property>

<!-- lower the maximum wait time (value in milliseconds) until the connection pool
will throw an exception that the pool is exhausted -->
<property name="maxWait">10000</property>

<!-- enable jdbc interceptor for statement cache, increase max value if required by
application load pattern -->
<property
name="jdbcInterceptors">ConnectionState;StatementFinalizer;StatementCache(prepared=true,callable=false,max=250)</property>
```

```
<!-- Oracle only: this will force the jdbc driver to enable its builtin statement
cache to avoid hard parsing repeating queries -->
<property name="connectionProperties">oracle.jdbc.implicitStatementCacheSize=250</
property>
```

Again, all other settings should be left untouched. Increasing the amount of connections and/or statement cache size will result in higher memory consumption, so make sure that application server and database memory target is sized properly.

3.7 Server Job Maintenance

Server jobs store their current state as well as all protocol entries in the database. A special job is executed once a day and is able to delete other, finished, jobs.

3.7.1 Configure the life time of a job

A job's life time is either defined as default life time by the job contribution itself (`plugin.xml`), or with the `plugin_customization.ini` file.

The values defined in the `plugin_customization.ini` will always have precedence over the default of the contribution.

To define a life time you need the unique job identifier (as listed in the table below), e.g. for the merge it's "Merge".

Syntax

```
com.heiler.ppm.jobhistory.cleanup.server/{JobTypeIdentifier}.lifeTime = {amount in days}
```

Example

```
com.heiler.ppm.jobhistory.cleanup.server/Merge.lifeTime = 60
```

Means that already executed merge jobs are deleted after 60 days of their execution.

3.7.2 Standard Job List

List of standard jobs with their identifiers in order to configure their lifetime.

Job identifier	Job name	Job group	Default life time	Multiple Job entries
AssignDocumentJob	Media manager document assignment	System processes	3	TRUE
CheckTaskEscalatedJob	Check for escalated tasks	System processes	0	TRUE
CleanUpJobHistory	Remove obsolete processes	System processes	0	TRUE
CleanUpReports	Remove obsolete query results	System processes	0	FALSE
CleanUpSoftdelete	Remove deleted objects	System processes	0	FALSE
com.heiler.ppm.revision.jobType	Versioning	Data maintenance	10	TRUE

Job identifier	Job name	Job group	Default life time	Multiple Job entries
com.heiler.ppm.workflow.server.CleanUpFinished	Remove obsolete workflows	System processes	0	FALSE
CopyValueIntoStructureGroupsJob	Update structure group features	System processes	0	TRUE
create.thumbnails	Thumbnail generation	System processes	3	TRUE
create.thumbnails.rec	Recursive thumbnail generation	System processes	0	TRUE
DataQualitySeries	Data quality check (scheduled)	Data quality	0	TRUE
DictionarySynchronization	Dictionary synchronization (manual)	Data quality	0	TRUE
ExportDryRun	Quality check	Data maintenance	0	FALSE
ExportSeries	Export operations (repeated)	Publications	0	TRUE
FindReplace	Find and Replace	Data maintenance	2	TRUE
FindReplaceSetValue	Set	Data maintenance	2	TRUE
FulltextSearchJobSeries	Index creation (repeated)	Search index	30	TRUE
FulltextSearchSingleJob	Index creation (one-off)	Search index	30	TRUE
Import	Import operations	Data transfer	0	TRUE
Inbox	Hotfolder data processing	Data transfer	0	TRUE
InboxImport	Import operations by hotfolder	Data transfer	0	TRUE
ManualCleanUpReports	Remove obsolete query results (manual)	System processes	0	FALSE
MapArticlesToProductGroupJob	Product item feature transfer	System processes	0	TRUE
MapArticlesToStructureGroupJob	Update item-structure group assignments	System processes	0	TRUE
MapEGDsToStructureGroupJob	Update EGD-structure group assignments	System processes	0	TRUE

Job identifier	Job name	Job group	Default life time	Multiple Job entries
Merge	Merge operations	Data transfer	0	TRUE
MigrateStructures	Migrate Structures	Data maintenance	0	TRUE
NewMediaAssetDerivativeSchema	New derivative schema	System processes	0	TRUE
ReferenceDataDeployment	Deployment of reference data (scheduled)	Data quality	0	TRUE
SingleDataQuality	Data quality check (manual)	Data quality	0	FALSE
SingleExport	Export operations (one-off)	Publications	0	FALSE
SupplierExchangeExport	Export operations (Supplier Portal)	Publications	5	TRUE
TestExport	Test export operations	Publications	0	FALSE
update.version	Updating image versions	System processes	0	TRUE
UpdateTasksJob	Update tasks	System processes	0	TRUE

3.8 Soft Delete Cleanup Job

3.8.1 General information

The soft delete cleanup job is a maintenance job which performs a hard delete (physical delete) of entities which have been already soft deleted. The hard delete cleanup is only triggered for expired entities and only if the lifetime is not unlimited. With version higher 8.0.01, the lifetime is set to one year per default for all entities. The expiration date is calculated by means of the current date as basis. So if today is February the 5th and you have a lifetime of 1 month, then the calculated expire date will be January the 5th, which means that all entities deleted prior to this date will be removed permanently. The default life time property defines life time for all entity types. This can be overwritten per root and sub entity type.

3.8.2 Configuration

You can define the cleanup life time in the *plugin_customization.ini*. Search for `cleanup.lifeTime.default`.

Example default life times for all entities:

```
cleanup.lifeTime.default=1d
```

```
cleanup.lifeTime.default=1y
```

If you want to define a different life time for different root and sub entities then you need to uncomment the property `cleanup.lifeTime.custom.file` and refer to a custom file defining the soft delete properties. The custom file has to be located in the folder `~/server/configuration/HPM`. The `cleanup.lifeTime.custom.file` property referring to the custom file could look like this, for example:

```
cleanup.lifeTime.custom.file = softDeleteLifeTime.properties
```

In your customized soft delete life time properties file you can define the different life times for the entities. The most concrete definition will be used. So if you define a life time for a root entity, then you can overwrite it for the sub entity. But this is only valid if the life time for the sub entity is shorter than the one of the parent.

A valid configuration:

```
cleanup.lifeTime.default = unlimited
ArticleType.lifeTime = 1y
ArticleLangType.lifeTime = 1m
StructureGroupLangType = 6d
```

3.8.3 Schedule

To schedule the job repetitively, uncomment the property `cleanup.job.repeatPattern` and define the pattern. From the version 8.0.01.00 this job is scheduled to run every day at 24:00. To disable the scheduling, you have to uncomment the property and leave it empty, then the job won't be executed automatically but may be started manually in the process overview (Select "System processes > Remove deleted objects", select the process in the "Current and past processes" view and call "Execute immediately" from the context menu).

Example for running the job daily at 24:00:

```
cleanup.job.repeatPattern = 0 00 00 * * ? *
```

To disable scheduling:

```
cleanup.job.repeatPattern =
```

For more information refer to the Quartz Enterprise Job Scheduler documentation for Cron Expressions

3.9 Data Quality Operation

3.9.1 Logging

Additional logging information for data quality execution can be increased on several levels: rule engine execution, data provisioning and performance analysis

Rule engine execution

In order to see logging information of the execution of a data quality rule by the rule engine set the category 'IDQ' to DEBUG in server log4j.xml

```
<category name="IDQ">
  <priority value="DEBUG"/>
</category>
```

As opposed to normal logging information a server restart is required for changes to take effect.

The logs can be found in separate files in the server logs folder. Each of them has a pattern like <Execution Thread name>-<Executed rule name>-<Execution start time>.log

Data provisioning

In order to see logging information of the data provisioning (data input into and data output from rule) set the category 'DQ_EXECUTION' to DEBUG in server log4j.xml.

```
<category name="DQ_EXECUTION">
  <priority value="DEBUG"/>
```

```
</category>
```

The output is available in the usual error.log respectively console.log files. Example output:

```
13:01:25,166 DEBUG [exe-IDQ-SDK-Check_PresetValues-2] [DQ_EXECUTION] DQ IN line: 0 (rule: preset violation)
  inObjectID: '33762@1' (field: Object_ID)
  inAttributeName: 'Color' (field: ArticleAttribute:name)
  inLanguage: 'English' (field: ArticleAttributeLang:language)
  inStructure: '10058' (field: ArticleAttributeStructureGroupAttributeMap:structure)
  inStructureGroup: '232@10058' (field: ArticleAttributeStructureGroupAttributeMap:structureGroup)
  inStructureFeature: '74@10058' (field: ArticleAttributeStructureGroupAttributeMap.StructureAttribute)
  inAttributeValue: 'green2' (field: ArticleAttributeValue:Value)
  inValueIdentifier: 'DEFAULT' (field: ArticleAttributeValue:identifier)
  inStructurePresetValue: '' (field: ArticleAttributeStructureGroupAttributeMap.DomainValue)
  inDataType: 'Character string' (field: ArticleAttribute.Datatype)

13:01:25,317 DEBUG [exe-IDQ-SDK-Check_PresetValues-1] [DQ_EXECUTION] DQ OUT line: 0 (rule: preset violation)
  status_code: '0' (field: QualityStatusEntry.Status)
  status_message: 'Primary, Pflichtgruppe, Feature 'Color': Value 'green2' (English, DEFAULT) of attribute 'Color' does not conform to the preset values' (field: QualityStatusEntry.Message)
  object_id: '33762@1' (field: Object_ID)
```

Performance analyzing

Enable the following category to DEBUG for additional information about DQ initialization parameters and DQ Executor Pool usage.

```
<category name="com.heiler.ppm.dataquality.server">
  <priority value="DEBUG"/>
</category>
```

3.9.2 Performance tuning

- Increase number of maximum jobs executed in parallel. Set org.quartz.threadPool.threadCount in quartz.properties accordingly.
- since 7.1.06: use synchronous execution of dq via service api instead of asynchronous jobs and waiting for job finished status.
F.g. when executing rules inside a bpm workflow. This decreases the general load of the job framework and gets rid of the overhead of the job scheduling; See Service Api: Rest Management API -> Rest Data Quality API -> Rule Execution.

The following properties in the preferences.ini may be increased to improve performance for many concurrent executions:

```
# Maximum of executors per IDQ rule in executor object pool. Use higher numbers for better concurrent rule execution.
dataquality.executionPool.maxTotalPerRule=3
# Total maximum of executors across all IDQ rules. Roughly, this is maxTotalPerRule multiplied with number of frequently used IDQ rules.
dataquality.executionPool.maxTotal=200
# If true, one executor for each DQ rule is created on server start. If false, rule executors are created on demand. Set to false to speed up
# server startup time. First rule execution takes longer in this case.
dataquality.executionPool.populateOnStart=false
# Enables an alternative implementation for the DQ Rule Check_IsEmpty that doesn't use the IDQ SDK. Set to
# true in case you need the ultimate performance and have huge numbers of is empty checks.
```



```
dataquality.alternativeCheckIsEmpty=false
# Specifies the page size when loading the list model of data to be checked for DataQuality rule execution.
dataquality.executor.pageSize=10000
```

An additional note to the property `dataquality.executionPool.maxTotalPerRule`: This value should not be set too high, so that the overhead of creating objects does not decrease the possible performance gain that comes with more parallel executors.

3.9.3 Issue reporting guide

This guide explains required information and steps to retrieve them in order to ensure a certain overall quality level for issues concerning Data Quality relevant topics.

Please provide as much information as described as follows in order to have a quick success in solving the issues.

Configuration setup

All DQ relevant configuration and rule files contained in the server configuration subfolder 'dataquality'. This can simply be done by zipping the whole folder and attaching it to the issue.

Screenshots of specific rule configurations are also helpful, especially when they are custom ones. The same is true for the configured execution triggers.

It may also be important to provide information about the setup workflows if executions are called from there.

Information about the execution context

Data Quality executions have following main informations

- The rule configurations containing information about the data provisioning
- The actual data quality rules that are used in the rule configuration
- The execution may contain more than one rule configuration, mostly if the execution is done for specific channels containing several configurations
 - If the rule configurations are executed as jobs as f.g. by manual executions, see the job log details in the processes overview for the used rule configurations.
 - For triggered executions this information may also be retrieved by looking at the trigger configuration
 - Execution may be also triggered from calls inside workflows
- Can the problem be pinned down to a specific rule configuration or rule? If so, Please specify them. Is the rule a custom rule and if so, what is it supposed to do?

Logging data

- General logging data as always (error.log, console.log)
- DQ specific logging data as described in the previous 'Logging' section. This may require to change the logging level to DEBUG as described there and then execute the rules again in order to see the additional logging information.

Performance

- Provide information about the settings as described in the Performance tuning section
- Is the execution manually or triggered e.g. after import or every night or for each item change?
- How many items are affected? Execution for single item or many items of a catalog or other data set f.g.?
- Are there many executions in parallel?

3.9.4 Configure rules that are affected by Dictionary Synchronization in order to increase performance of IDQ rule execution

Situation/ motivation

- For each IDQ rule, PIM creates executor objects and stores them in a pool, because creation is expensive. Executors are used to execute IDQ rules in parallel. These executors are created lazily on demand. Also IDQ SDK internally synchronizes executor creation, which means, creation is queued.

- Changes to dictionaries aka reference tables are only picked up by the rule if the executor is disposed and re-created
- Currently, on the PIM side, it is not known which reference tables are used by which IDQ rule. Hence, whenever, a dictionary is synchronized, all executors are disposed, and created lazily on demand.
- Because of that the performance drastically degrades whenever a dictionary is synchronized AND many new executors are requested for IDQ rule execution.

Solution

- It is now possible, to configure, which IDQ rules are affected by which dictionary (aka Reference Table). Therefore the **DictionaryRuleConfiguration.xml** is used, which is stored in the PIM server's folder /configuration/HPM/dataquality, and looks like this:

DictionaryRuleConfiguration.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<dictionaryToRuleConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="DictionaryRuleConfiguration.xsd">
  <isActive>true</isActive>
  <dictionary>
    <dictionaryIdentifier>Informatica_DQ_Content\Dictionaries\General\profanity_infa.dic</dictionaryIdentifier>
    <affectedRules>
      <ruleIdentifier>Informatica_PIM_Content/PreDefined_Rules/Check_Profanity</ruleIdentifier>
      <ruleIdentifier>Informatica_PIM_Content/Custom_Rules/MyCustomRule1</ruleIdentifier>
    </affectedRules>
  </dictionary>
</dictionaryToRuleConfiguration>
```

- This configuration is only active, if this XML file is placed in the mentioned folder AND if **<isActive>** is set to true. Otherwise, the behavior is the same as before, means with every dictionary synchronization ALL rule executor objects are disposed and created lazily on demand.
- The **<dictionary>** element defines, which rules (defined by the **<ruleIdentifier>**s within the **<affectedRule>** element) are affected by a specific dictionary (specified by the **<dictionaryIdentifier>**). You can define as many **<dictionary>** elements as you wish.
- The **<dictionaryIdentifier>** has to be copied from the dictionary view (part of the Dictionary perspective) in the PIM Desktop client.
- The **<ruleIdentifier>** is composed of the folders and subfolders of the IDQ rule and the name of the IDQ rule itself. This information can be found through the Data quality perspective in the PIM Desktop Client:
- When a dictionary, which is configured here, is synchronized, ONLY the executor objects of the configured affected rules are destroyed and recreated lazily on demand - all other rule executor objects are not hit.
- When a dictionary, which is NOT configured here, is synchronized, the behavior is the same as before, means ALL rule executor objects are disposed and created lazily on demand.



Be aware, that as soon as a specific dictionary is configured in the **DictionaryRuleConfiguration.xml** file, synchronization of that dictionary NEVER leads to a destruction of not affected rule executor objects. That means that if those "not affected" rules executor objects might anyhow have dependencies to the configured dictionary, they might work on a very old state of this dictionary, which might lead to an inconsistent behavior when executing IDQ rules.



The **DictionaryRuleConfiguration.xml** configuration is only loaded once during server startup. Means, after changing that file, the PIM server has to be restarted, in order to activate the changes.

If this something goes wrong, during loading of this configuration file (e.g. the corresponding **DictionaryRuleConfiguration.xsd** is not existing in the /configuration/HPM/dataquality folder, or there is an syntax error in the **DictionaryRuleConfiguration.xml** file), an ERROR is logged in the server's log file. But still the server starts up and the behavior is the same as before, means with every dictionary synchronization ALL rule executor objects are disposed and created lazily on demand.

Situation/ motivation For each IDQ rule, PIM creates executor objects and stores them in a pool, because creation is expensive. Executors are used to execute IDQ rules in parallel. These executors are created lazily on demand. Also IDQ SDK internally synchronizes executor creation, which means, creation is queued. Changes to dictionaries aka reference tables are only picked up by the rule if the executor is disposed and re-created. Currently, on the PIM side, it is not known which reference tables are used by which IDQ rule. Hence, whenever, a dictionary is synchronized, all executors are disposed, and created lazily on demand. Because of that the performance drastically degrades whenever a dictionary is synchronized AND many new executors are requested for IDQ rule execution. Solution It is now possible, to configure, which IDQ rules are affected by which dictionary (aka Reference Table). Therefor the DictionaryToRuleConfiguration.xml is used, which is stored in the PIM server's folder /configuration/HPM/dataquality, and looks like this: <?xml version="1.0" encoding="UTF-8"?>

```
<dictionaryToRuleConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="DictionaryRuleConfiguration.xsd"> <isActive>true</isActive> <dictionary>
<dictionaryIdentifier>Informatica_DQ_Content\Dictionary\General\profanity_infa.dic</dictionaryIdentifier>
<affectedRules> <ruleIdentifier>Informatica_PIM_Content/PreDefined_Rules/Check_Profanity</ruleIdentifier>
<ruleIdentifier>Informatica_PIM_Content/Custom_Rules/MyCustomRule1</ruleIdentifier> </affectedRules> </dictionary></
dictionaryToRuleConfiguration>
```

This configuration is only active, if this XML file is placed in the mentioned folder AND if <isActive> is set to true. Otherwise, the behavior is the same as before, means with every dictionary synchronization ALL rule executor objects are disposed and created lazily on demand. The <dictionary> element defines, which rules (defined by the <ruleIdentifier>s within the <affectedRule> element) are affected by a specific dictionary (specified by the <dictionaryIdentifier>). You can define as many <dictionary> elements as you wish. The <dictionaryIdentifier> has to be copied from the dictionary view (part of the Dictionary perspective) in the PIM Desktop client. Informatica MDM - Product 360 > .Configure specific rules that are affected by Dictionary Synchronization in order to increase performance of IDQ rule execution v7.0 > image2016-9-22

13:56:50.png The <ruleIdentifier> is composed of the folders and subfolders of the IDQ rule and the name of the IDQ rule itself. This information can be found through the Data quality perspective in the PIM Desktop Client: Informatica MDM - Product 360 > .Configure specific rules that are affected by Dictionary Synchronization in order to increase performance of IDQ rule execution v7.0 > image2016-9-22 14:2:48.png When a dictionary, which is configured here, is synchronized, ONLY the executor objects of the configured affected rules are destroyed and recreated lazily on demand - all other rule executor objects are not hit. When a dictionary, which is NOT configured here, is synchronized, the behavior is the same as before, means ALL rule executor objects are disposed and created lazily on demand.

Be aware, that as soon as a specific dictionary is configured in the DictionaryToRuleConfiguration.xml file, synchronization of that dictionary NEVER leads to a destruction of not affected rule executor objects. That means that if those "not affected" rules executor objects might anyhow have dependencies to the configured dictionary, they might work on a very old state of this dictionary, which might lead to an inconsistent behavior when executing IDQ rules.

The DictionaryToRuleConfiguration.xml configuration is only loaded once during server startup. Means, after changing that file, the PIM server has to be restarted, in order to activate the changes. If this something goes wrong, during loading of this configuration file (e.g. the corresponding DictionaryToRuleConfiguration.xsd is not existing in the /configuration/HPM/dataquality folder, or there is an syntax error in the DictionaryToRuleConfiguration.xml file), an ERROR is logged in the server's log file. But still the server starts up and the behavior is the same as before, means with every dictionary synchronization ALL rule executor objects are disposed and created lazily on demand.

3.10 Check integrity of attribute names in key language

The name of an attribute must exist in the key language otherwise errors may occur during operation. The key language can be specified in the server.properties.

If the key language is changed, it is possible that attributes exist that do not have a name in the key language. The following database script can be used to check if all attributes have a name in the key language.

3.10.1 MSSQL

The script for MASTER and SUPPLIER

```
SELECT COUNT(aa1.ID)
FROM [ArticleAttribute] aa1
INNER JOIN [ArticleAttributeLang] aa1 ON aa1.ID = aa1.[ArticleAttributeID]
WHERE aa1.ID NOT IN (
    SELECT aa2.ID
    FROM [ArticleAttribute] aa2
    INNER JOIN [ArticleAttributeLang] aa2 ON aa2.ID = aa2.[ArticleAttributeID]
    WHERE aa2.[LanguageID] = :keyLanguageID )
```

The script for MAIN

```
SELECT COUNT(sga1.ID)
FROM [StructureGroupAttribute] sga1
INNER JOIN [StructureGroupAttributeLang] sga1 ON sga1.ID = sga1.[StructureGroupAttributeID]
WHERE sga1.ID NOT IN (
    SELECT sga2.ID
    FROM [StructureAttribute] sga2
    INNER JOIN [StructureGroupAttributeLang] sga2 ON sga2.ID = sga2.[StructureGroupAttributeID]
    WHERE sga2.[LanguageID] = 9 )
```

3.10.2 Oracle

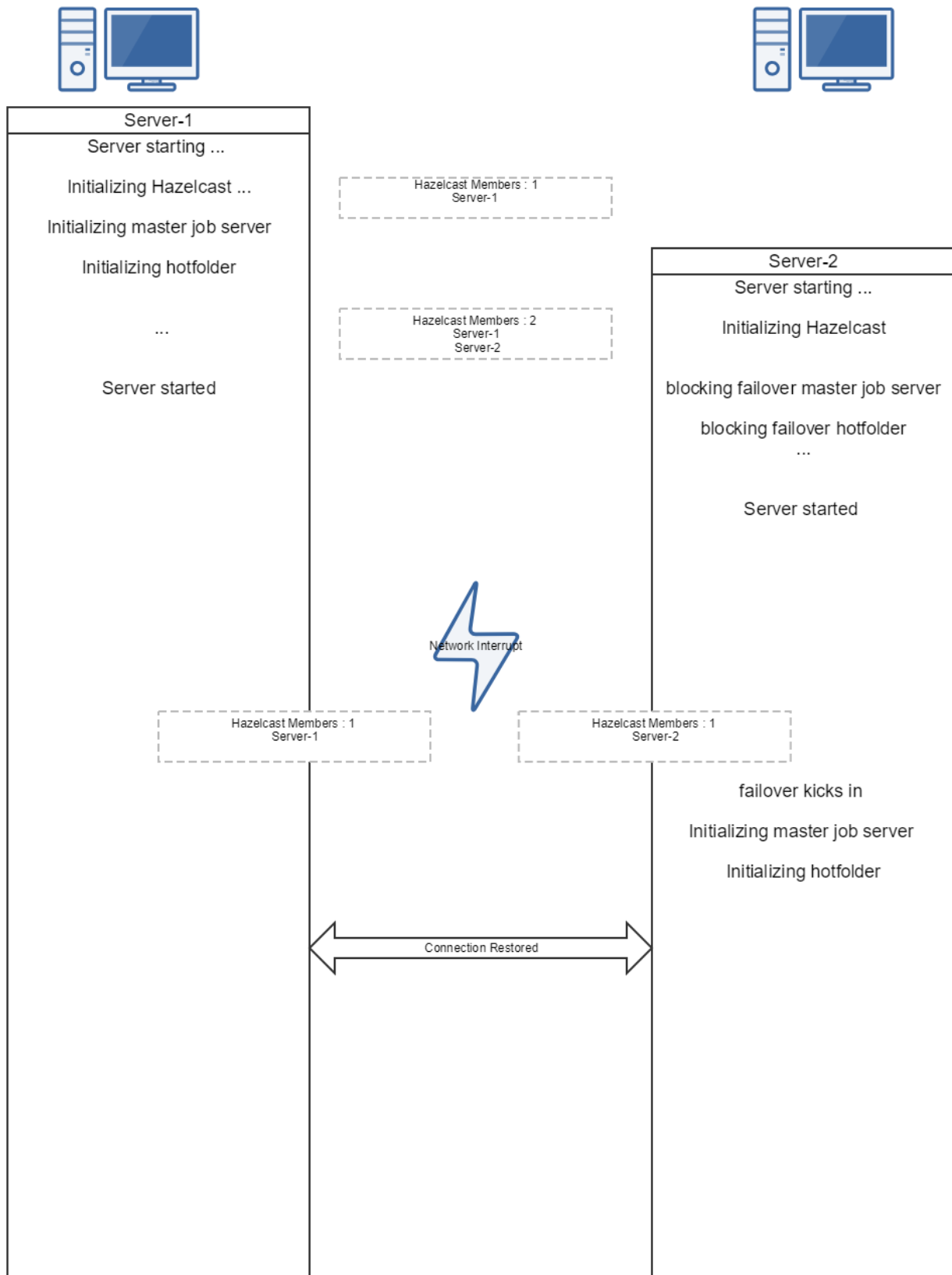
The script for MASTER and SUPPLIER

```
SELECT COUNT(aa1.ID)
FROM "ArticleAttribute" aa1
INNER JOIN "ArticleAttributeLang" aa1 ON aa1.ID = aa1."ArticleAttributeID"
WHERE aa1.ID NOT IN (
    SELECT aa2.ID
    FROM "ArticleAttribute" aa2
    INNER JOIN "ArticleAttributeLang" aa2 ON aa2.ID = aa2."ArticleAttributeID"
    WHERE aa2."LanguageID" = :keyLanguageID )
```

The script for MAIN

```
SELECT COUNT(sga1.ID)
FROM "StructureGroupAttribute" sga1
INNER JOIN "StructureGroupAttributeLang" sga1 ON sga1.ID = sga1."StructureGroupAttributeID"
WHERE sga1.ID NOT IN (
    SELECT sga2.ID
    FROM "StructureAttribute" sga2
    INNER JOIN "StructureGroupAttributeLang" sga2 ON sga2.ID = sga2."StructureGroupAttributeID"
    WHERE sga2."LanguageID" = 9 )
```

3.11 Network Recovery





4 Desktop Operation

Combines maintenance, tuning and troubleshooting for the PIM - Desktop

4.1 Maintenance Tasks

- Backup strategy
- Workspace configuration

4.2 Tuning

4.2.1 Performance optimization of the "Documents" view

PIM - Server shows in its Documents view the content of a media asset category delivered by the integrated media asset provider.

However, the respective media asset providers vary in the data they deliver for a media asset and in the reaction time/speed. For instance, it might be a problem to access a preview image at provider "A", or the calculation of an image dimension at provider "B" is a problem.

Therefore, the following requirement to media asset providers has been introduced:

- The media asset provider must deliver for each media asset an (unique) identifier, the name, and size in high speed.

If this requirement is not fulfilled, it makes no sense to integrate such a media asset provider.

Furthermore, the load of this view has been optimized so that it is based on the properties returned by the respective media asset provider:

When you are opening a media asset category, a list of all media assets contained by this category is requested from the media asset provider.

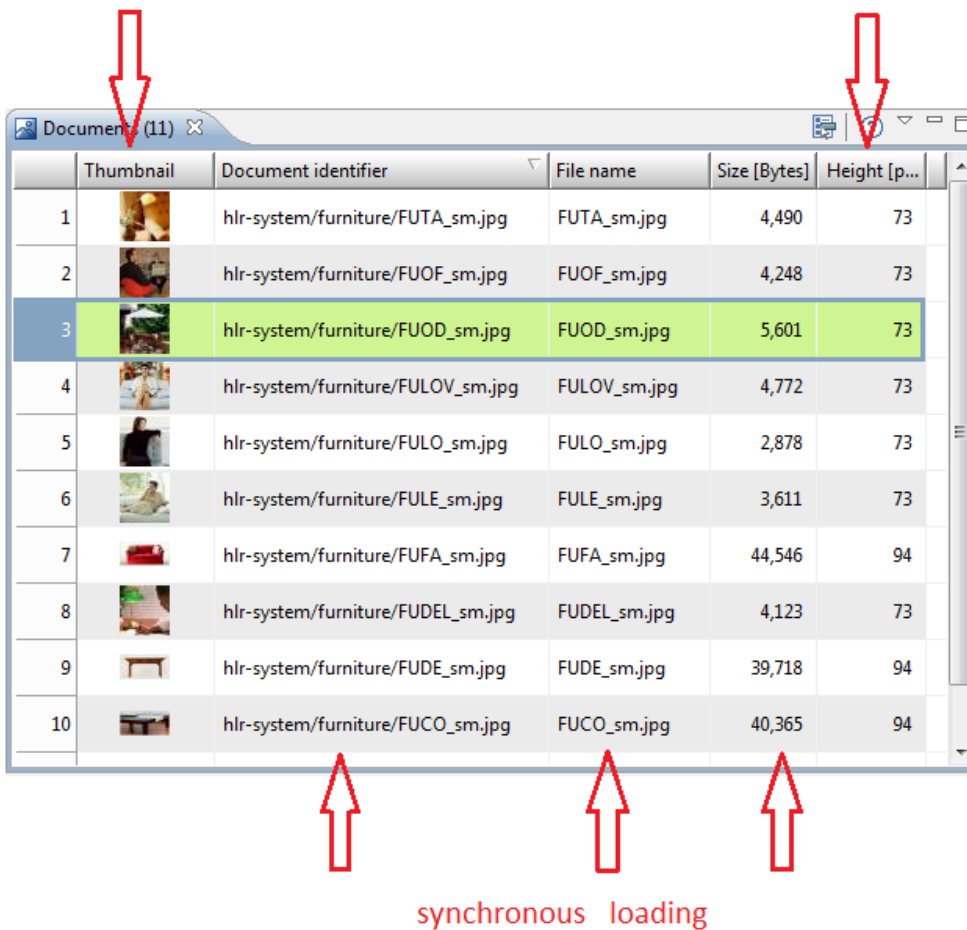
The media asset provider has to return the identifiers, the names, and the file sizes immediately.











The values for all other columns the media asset provider did not immediately return can be returned with a "delayed" status. In this case, the view requests these values for each row separately, when the view "scrolls" in the visible field area. The performance of the view can be improved by hiding unimportant columns.


It is also possible to hide the preview image column. This increases the performance even more.

asynchronous loading

asynchronous loading, if delayed
by the media asset system



	Thumbnail	Document identifier	File name	Size [Bytes]	Height [p...
1		hlr-system/furniture/FUTA_sm.jpg	FUTA_sm.jpg	4,490	73
2		hlr-system/furniture/FUOF_sm.jpg	FUOF_sm.jpg	4,248	73
3		hlr-system/furniture/FUOD_sm.jpg	FUOD_sm.jpg	5,601	73
4		hlr-system/furniture/FULOV_sm.jpg	FULOV_sm.jpg	4,772	73
5		hlr-system/furniture/FULO_sm.jpg	FULO_sm.jpg	2,878	73
6		hlr-system/furniture/FULE_sm.jpg	FULE_sm.jpg	3,611	73
7		hlr-system/furniture/FUFA_sm.jpg	FUFA_sm.jpg	44,546	94
8		hlr-system/furniture/FUDEL_sm.jpg	FUDEL_sm.jpg	4,123	73
9		hlr-system/furniture/FUDE_sm.jpg	FUDE_sm.jpg	39,718	94
10		hlr-system/furniture/FUCO_sm.jpg	FUCO_sm.jpg	40,365	94

 Note: Thumbnails of pdf documents are not displayed in the documents view!

Filter

The Documents view provides an interface enabling to filter the content on e.g. JPEG documents. To do this, you have to register an implementation of the Interface

```
com.heiler.ppm.mediaasset.ui.api.IFileDataFilter
```

by implementing the method

```
com.heiler.ppm.mediaasset.ui.views.DocumentsOfCategoryTableView.registerFileDataFilter(IFileDataFilter)
```

of the view. This filter will then be called every time the view is populated with data. From a technical point of view this filter will be called inside the respective IContentProvider implementation.

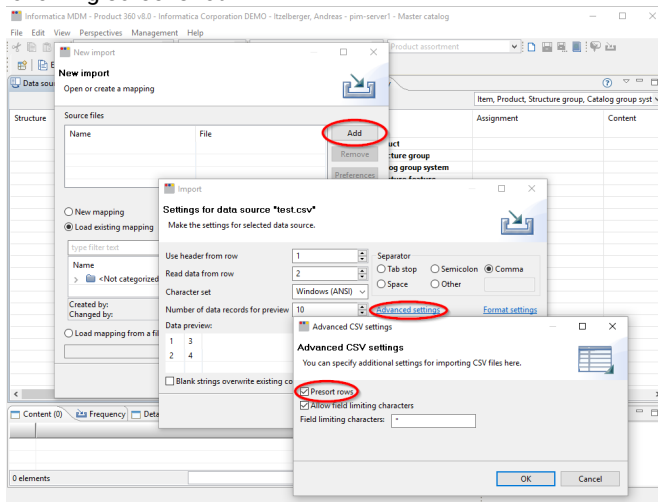
```
public interface IFileDataFilter
```

```
{
    public FileData[] filter( FileData[] fieldToFilter );
}
```

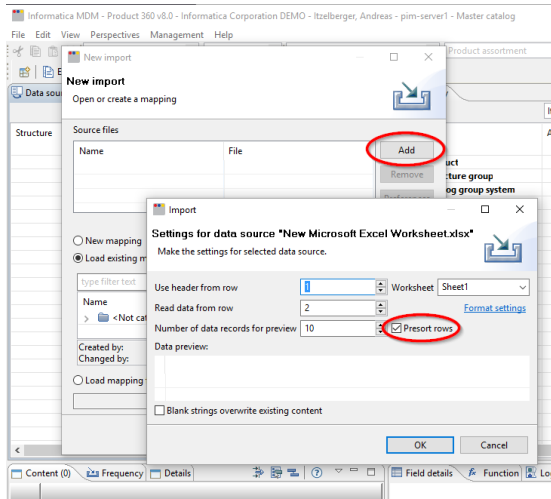
4.2.2 Import

When adding CSV or Excel files to an import project, per default all rows are read into heap memory in order to presort them. This speeds up processing if the data for one item is stored in rows that are spread. So when adding large CSV or Excel data files, consider to increase the client's heap size accordingly.

If presorting is not needed (because the data files are already presorted) or desired, it can be disabled by opening the "Advanced settings" dialog after selecting the CSV file, and here unchecking the checkbox "Presort rows". See also the following screenshot:



For Excel data files find the "Presort rows" checkbox directly in the Settings dialog, after selecting the Excel file:



This "Presort rows" settings only affects the adding of data files to the import project in P360 Desktop Client, but not the import itself. For disabling presorting rows in the import itself, please see the chapter ["Server Operation"](#) (of the Installation and Operation guide) and here the sub chapter "Import", "Preferences regarding memory usage".



Adding large data files to an import project might become slower, when presorting is deactivated (see setting above) and the import data file is not already presorted.

4.3 Trouble Shooting

- Log Files
- Workspace log
- Server connection lost/reconnect

4.3.1 User Interface is not responding any more (client freeze)

The currently supported operating systems of the Heiler Product Manager do only support a single GUI thread. It is responsible for reading and dispatching operating system events like mouse clicks and keyboard strokes. Usually the main thread reads an event from the operating system, and dispatches this event further to whoever listens on it (in our case SWT widgets). It then waits for all those event listeners to complete and then read again the next events from the operating system (the read-and-dispatch loop is synchronously). It performs this in an endless loop as long as the application is running.

In the PIM - Desktop this thread is also the `main` thread, which makes it quite easy to identify. Therefore, whenever the client stops responding it must be related to the client's `main` thread being busy with other things then reading and dispatching OS events. For example it's waiting on some operation to complete.

Find the problem on the client

To pinpoint the reason for the freeze you will need to know what the client is doing (or better waiting for) at the moment of the freeze. For this you will need a [How to create a thread dump](#) of the client.

1. Perform a thread dump of the client at the moment of the freeze
2. Find the `main` thread in the dump and check the stack trace - see where it ends.
3. In case the `main` thread waits on the completion of a server request you will need to move your investigation to the server.
In this case the stack trace of the client's `main` thread usually stops somewhere in the communication framework, e.g. `ClientNodeImpl(AbstractNode).sendRequest(NodeIdentifier, Message, long)`
4. In case the stack trace makes absolutely no sense for you, don't panic, that's not unusual 😊. With the information you collected we should be able to help you identify and fix the problem.
5. In case the client's main thread is waiting for a server request to complete, you can move on step further and check the server's threads.

Find the problem on the server

Since the server has no user interface, the main thread of the server is more or less irrelevant for this task. All requests from the communication framework are handled by the communication worker threads. The only problem is to find the right one. Currently you can't determine this by just looking at the thread names, you really have to check all communication framework threads. Communication worker threads are named `CommunicationWorker-<Number>`. Most of them should be in parking position, waiting for a request of a client (e.g. `Unsafe.park(boolean, long)` line: not available [native method]), those can be ignored, of course.



If you take multiple thread dumps of the server, you can check which thread "didn't move" between those dumps, making it easier to find the responsible one.

When you finally found the communication worker thread which is responsible, it might happen that this thread is itself waiting. Waiting for some other thread or the database. From there on it gets complicated and leads to debugging multi-threading issues and database locking problems. It would go beyond the scope of this wiki, so we would encourage you to contact Heiler Software Support if you need any further assistance on this.

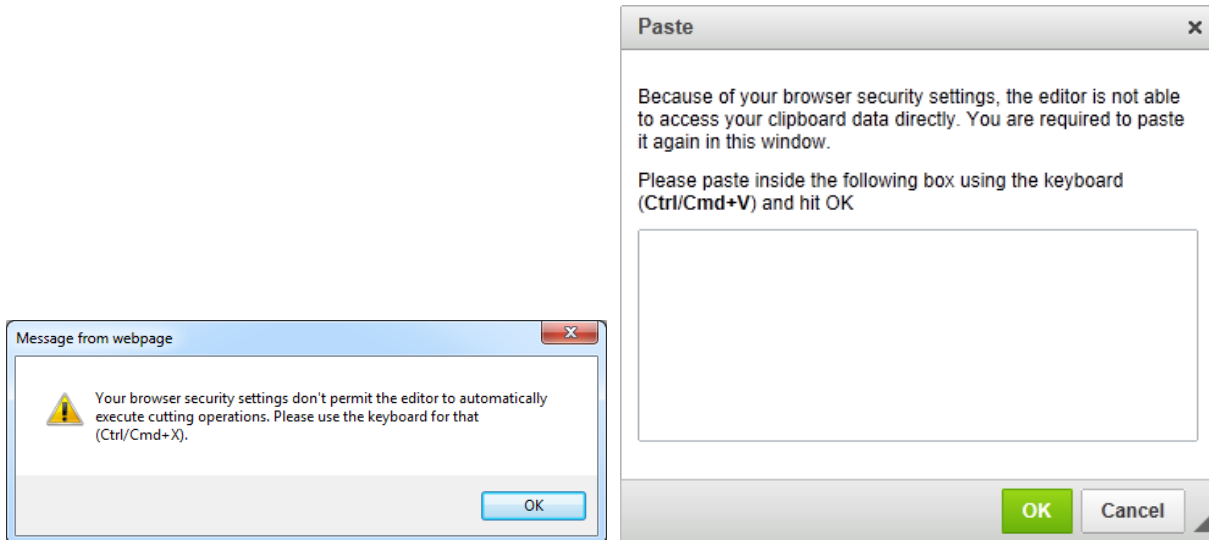
4.3.2 Rich Text Editor

Unable to use the clipboard actions "Cut", "Copy" and "Paste"

Due to the browser setting it may be impossible to use the following RichText-Editor buttons



In this case one of the following dialogs may be shown:

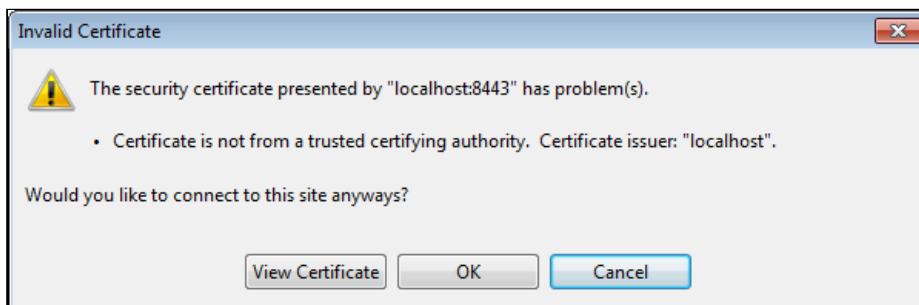


To change this behavior open the "Internet Explorer" and do following steps:

1. Go to Tools -> Internet Options.
2. Go to "Security" tab.
3. Select "Internet" zone, then click on "Custom level..." button.
4. Scroll down to "Scripting" section (at the bottom few).
5. Under "Allow Programmatic clipboard access" option, check or select (tick) to enable

4.3.3 Workaround to prevent the "Invalid Certificate" error (Mozilla)

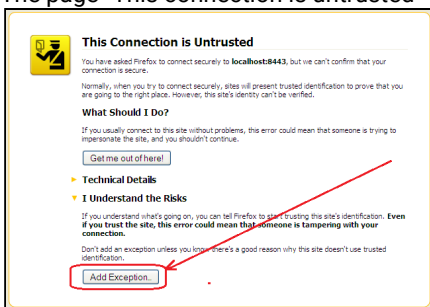
If the browser control loads any URL which uses an **unsigned** (or self-signed) SSL certificate, an error dialog appears:



Since there is no way to add an "exception" to the XULRunner, the following workaround can be used:

1. Open Firefox and load the domain which uses the invalid certificate (i.e. "https://localhost:8443/")

2. The page "This connection is untrusted" will be shown (see below):



3. Click on the button "Add exception..." and confirm the dialog which appears
4. Go to the "Firefox" profile folder (typically "C:\Users\[USER_NAME]\AppData\Roaming\Mozilla\Firefox\Profiles\[PROFILE_NAME]\")
5. Copy the file "cert_override.txt"
6. Go to the XULRunner profile folder (typically "C:\Users\[USER_NAME]\AppData\Roaming\Mozilla\eclipse\")
7. Paste the file "cert_override.txt"

Alternatively it is possible to use another workaround:

1. Open the file "prefs.js" in the folder "C:\Users\[USER_NAME]\AppData\Roaming\Mozilla\eclipse\".
2. Add the following line to the file and save the file:
`user_pref("network.automatic-ntlm-auth.trusted-uris", [host url of IdP server]);`

Next time you start PIM Desktop the browser control will load the URL without any errors.

Alternatively just use a "trusted" certificate (see https://en.wikipedia.org/wiki/Certificate_authority) or switch the setting `http.client.proxy` in "server.properties" to an HTTP URL (instead of HTTPS).

Startig with 8.0.03 it is possible to login to the desktop client via SAML. In this case the same rules apply to the certificate of the SAML IDP server. Generally it is recommended to use a trusted certificate, however If necessary the workaround with the `cert_override.txt` can be applied here as well.

4.4 Product Paradim (EGD) limitations

Please note that the EGD (Extended Generic Data) based product paradigm, which was used in the past for the first PIM Product paradigm, ist now with PIM 7 no longer in Support. The EGD itself will remain as basis for custom entities, but without the `EGDAttributeMapping` entity.

This information is also about some sideeffects in the PIM 7 user interface, yet. For example you are having in the organization perspective double menu entries for product assortments. One is for the old, egd based, product assortment, one for the new product assortments. Unfortunately you cannot see the difference in one glance. Workaround until this is removed with PIM 8 is to try out default object rights on your product assortment. If it does not have the adjusted effect you have to choose the other selection in the default object rights combo box.

5 Control Center Manual

5.1 Prerequisite

In order to get the Control Center running you need to follow the installation manual for the Server Installation

5.2 Function

The Control Center is a central application to help to administrate the systems of your cluster. It provides basic functionality like:

- Install and update your Product 360 system in the cluster
- Start server(s)

- Stop servers(s)
- See logs
- See running jobs
- See information of servers
- Gather configuration settings of the system
- Shutdown clients
- Send message to clients
- create thread dumps
- .
- .

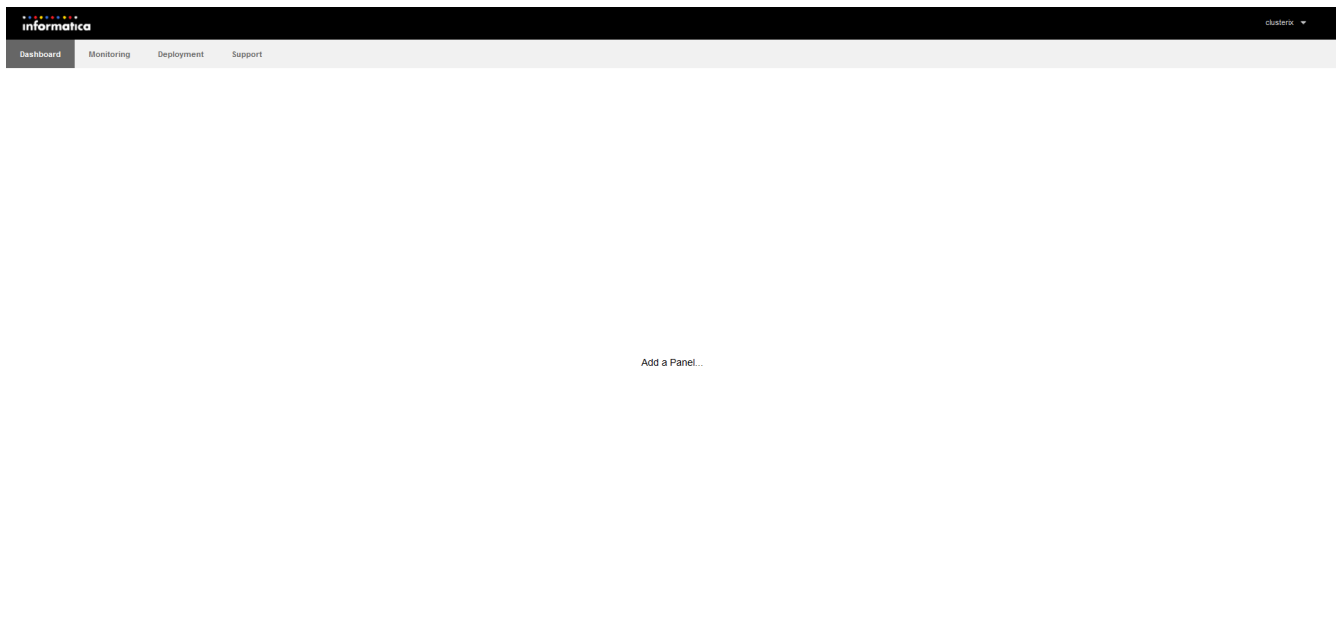
5.3 Login

To open the web interface of the Product 360 Control Center you have to know where the service is running. The URL consists of the host name and the port which is defined in the `ClusterixConfig.xml`. By default the port is 9000. So when the service is running on the instance called `myComputer` then you can open the web interface by calling `myComputer:9000`

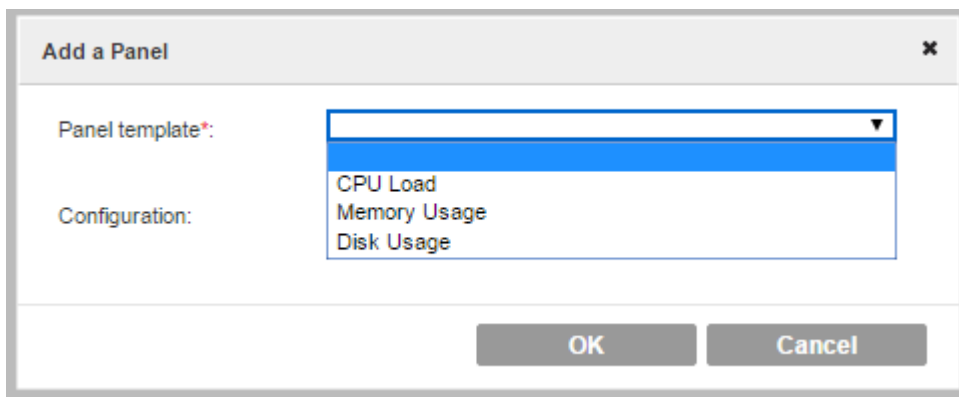
The login credentials are also defined in the file `ClusterixConfig.xml`. By default you can login with user:`clusterix` password:`clusterix`.

5.4 Dashboard

5.4.1 Widget creation using templates



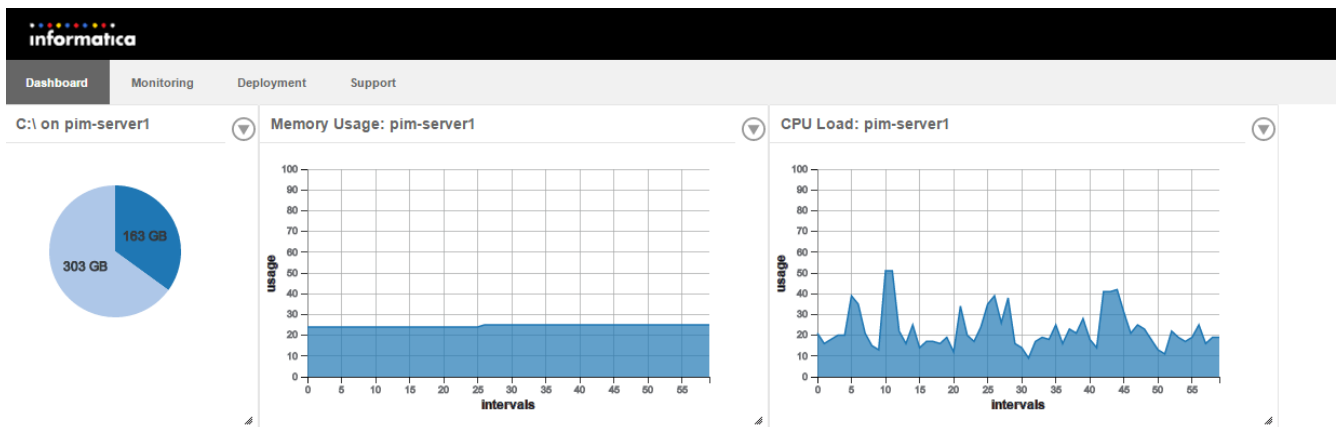
The dashboard is your personal area, where you can pint and save graphical widgets. It allows you to monitore data of a selected server. There are a number of widgets which may help you doing so. If you enter your dashboard the first time, you will be greeted with an empty panel. To add widgets to your dashboard, you can either select the label in the middel of the screen or left click on the **dashboard** tab in the top left corner.



Doing so opens up a window, giving you the opportunity to add your desired panel. Selecting your wanted panel template and left clicking on the 'ok' button, places the selected widget on your dashboard.

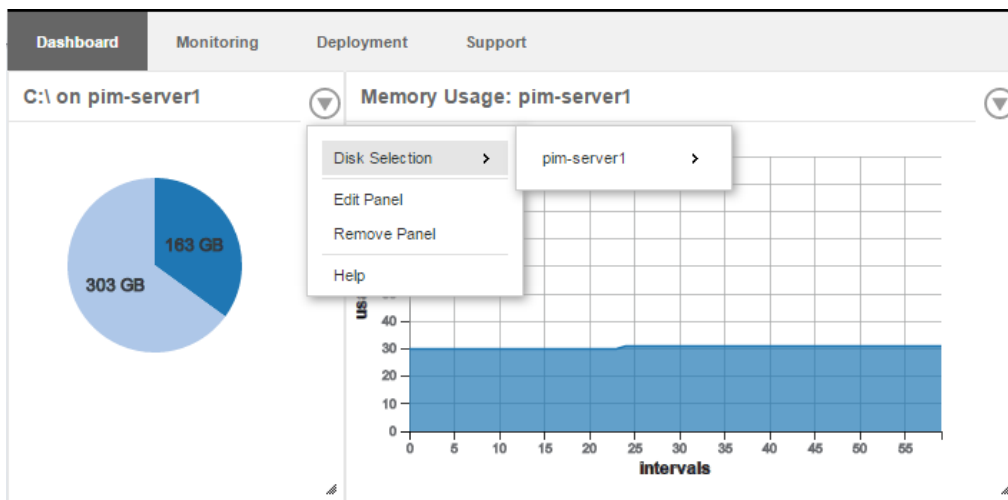
5.4.2 Customize widgets

The dashboard saves your current state and allows you to view your data at any given moment.



To customize your personal dashboard, you can select the arrow symbol in the top right corner of every single widget. Doing so allows you to:

- **select a server** that you want to monitor
- **edit the panel** itself
- **remove** the wanted widget.



Every change will be saved once you leave your dashboard and is loaded up when you enter it, making it your own personal dashboard.

5.5 Monitoring

The Monitoring tab allows you to monitor all systems inside your network. From here on out you are able to watch different categories of all application servers that are currently available in your network system. The left side of the monitoring tab contains a navigation tree, which is displaying the network itself and all available features for each server as nodes. The right side contains all associated data of the selected node inside the navigation tree.

5.5.1 Application Server Node

The screenshot shows the 'Monitoring' tab with the 'Application Server' node selected. The left sidebar shows a navigation tree with 'Network' and 'Application Server' nodes. The main content area displays the 'Application Server' view. It includes a table with columns: Identifier, Server Types, State, Host, CPU, Cores, MemoryMax (MB), Disk SpaceMax (GB), and Additional Information. The table shows one entry for 'pim-server1' with a state of 'RUNNING'. Below the table, there are sections for 'Clients', 'Jobs', and 'Server Actions'. The 'Selected Servers' section has a text input field and 'Start' and 'Stop' buttons. The 'Client Message' section has a 'Selected Clients' text input field, a 'Message' text input field, and a 'Send' button. The 'Client Shutdown' section is also visible at the bottom.

Clicking on the **Application Server** tab, opens the application server view. This tab contains information like:

- All running application servers and their status
- All connected clients
- All running Jobs that are currently running on the servers

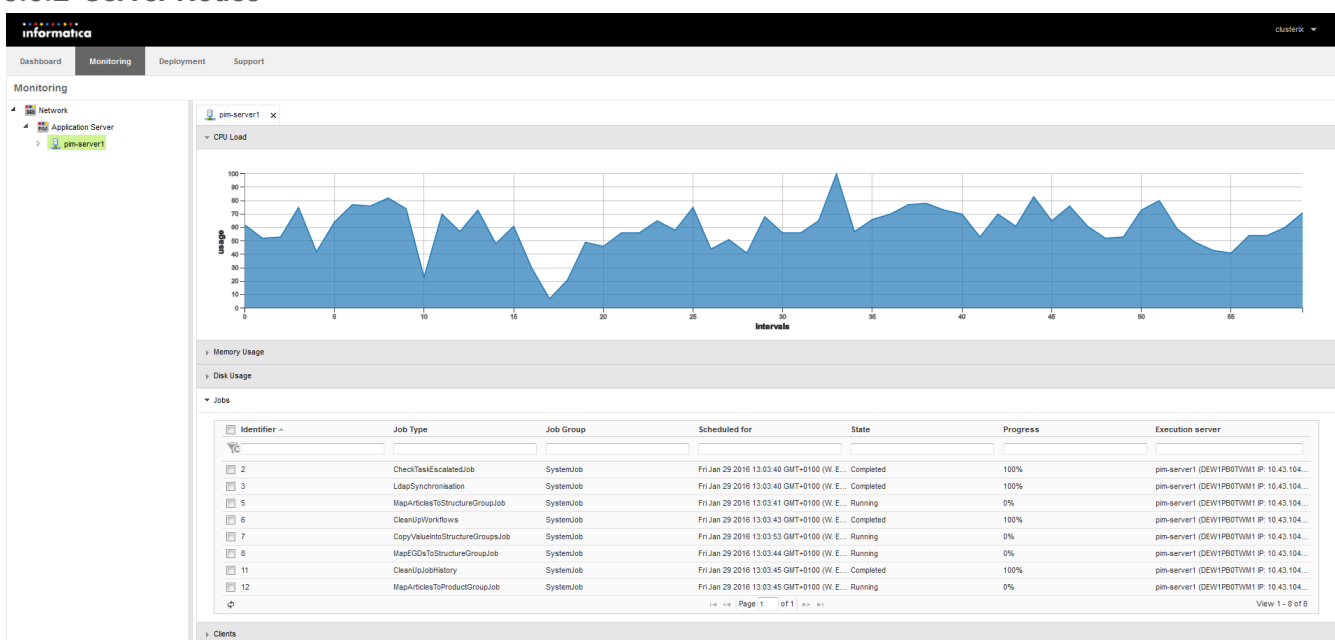
You are also able to select a running server and **start** or **stop** it. If there are clients connected to a server, you can select a client from the **Client Message** tab and send them a message. In the **Client Shutdown** tab you are able to select a client and shut it down.

Server tab

The Server tab presents server data about every configured server inside the current network. Together with the general data, which can be seen in the graphic above, you can view the current memory consumption of each server. The shown memory consumption describes how much memory has been allocated to the PIM server **application** running on the system and how much that application is currently using. The allocated memory on each server application can increase automatically if the application finds it necessary. This depends on how much free physical memory is available on the system and if the application is in need of more memory.

The example above has a system 'pim-server1' with a certain amount of physical memory. Currently about 3.5 GB memory has been allocated to the application. The application itself is using about 1.7 GB memory of the allocated 3.5GB memory. An external system monitor like Windows Task manager, may show a memory usage of 3.5GB as this is the memory reserved for the application.

5.5.2 Server Nodes



By opening the application server list and left clicking on one of your running servers inside the navigation tree, you open up a detailed view of the selected server. From here on out you have a wide variety of options and can view the following data:

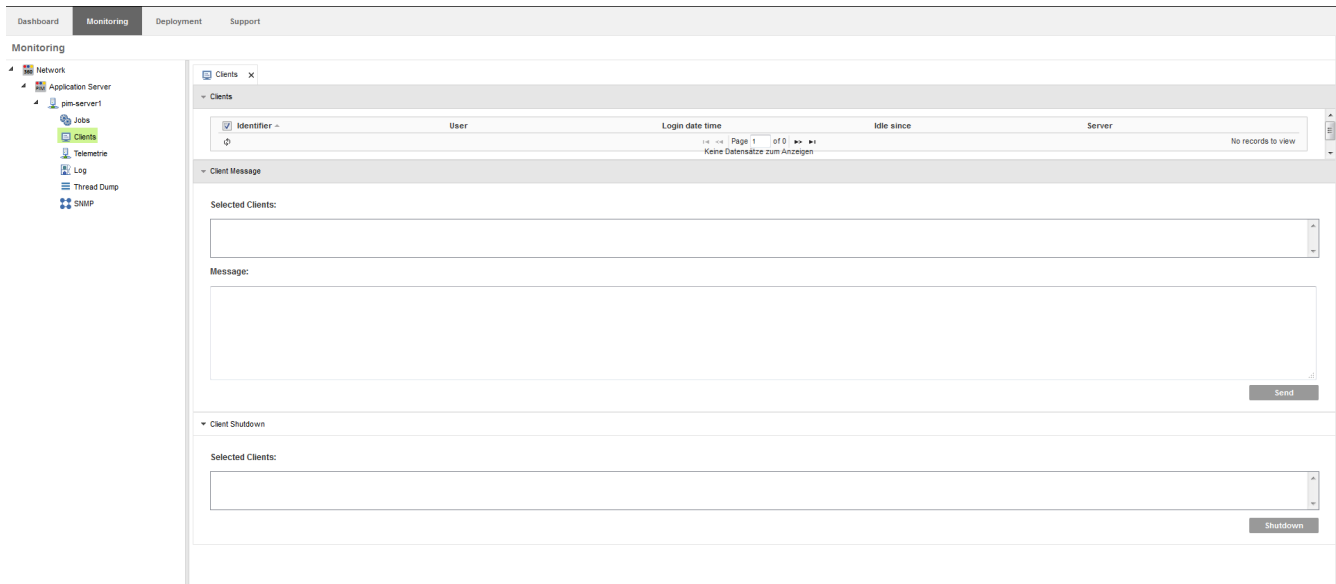
- A statistical view of the **CPU load** of the currently selected server
- A statistical view of the **Memory Usage** of the currently selected server
- A statistical view of the **Disk Usage** of the currently selected server
- A list of all currently **running jobs** and their status of the selected server
- A list of **connected clients** to the currently selected server

Just like before, you can start and stop the selected server inside the **Server Actions** tab.

5.5.3 Server Node Details

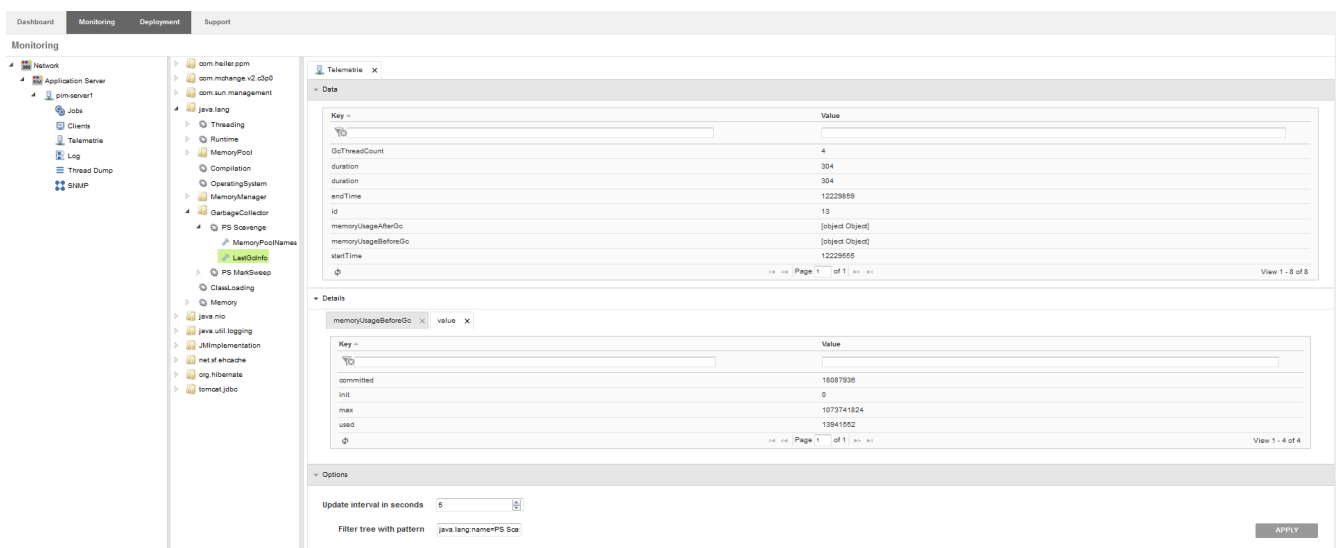
Extending the selected server node, gives you the option to have a detailed look at the servers actions and data.

Clients



The **Clients** node shows you once again, which clients are connected to the selected server. From here on out you can write messages to a selected client or shut him down.

Telemetry



Inside the **Telemetry** node, you have a wide variety of options. The navigation tree allows you to view all kind of telemetry data. By clicking on a node inside the telemetry navigation tree, you can view dynamically changing data inside the **Data** tab. This data is being updated in certain time intervals. To change these time intervals, simply change the corresponding textfield inside the actions tab and hit enter or select the **APPLY** button at the bottom right side of the screen.

The pattern field allows you to filter the navigation tree according to your pattern. Simply type in a pattern with the following form: **Domain : AttributeName1 = AttributeValue1, . . . , AttributeNameN = AttributeValueN**. Filter the tree by hitting enter or selecting the **APPLY** button.

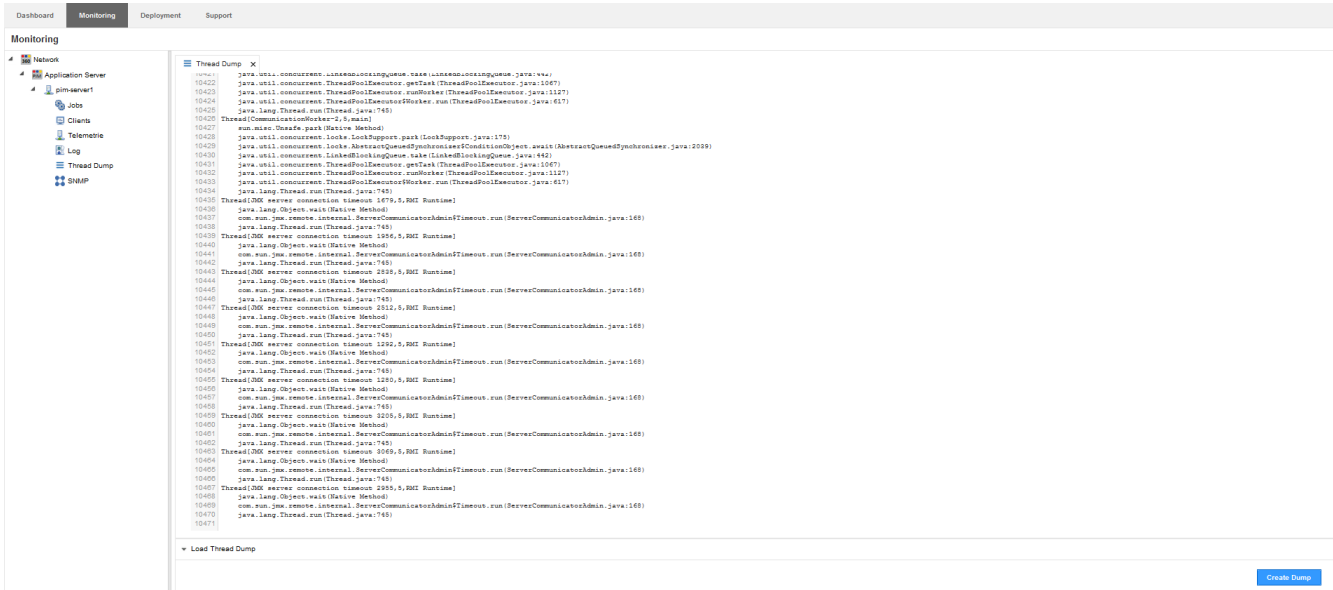
You open up the Detail View by selecting data that is marked as [Object object] inside the **Data tab**. Here you can get more detailed information about the selected entity. If you now select data that is marked as [Object object] inside the **Detail tab**, a new tab opens up and you can view that information inside of it.

Log



The **Log** node allows you to view the log of the selected server. While the **Text Content Tab** periodically updates and shows the tail of the log of the selected server, you are able to download the complete log file and the displayed log file by simply selecting the corresponding buttons at the right hand side of the **Log Actions tab**.

Thread Dump



Selecting the **Thread Dump** node lets you view the current Thread dump of the selected server. Simply select the **Create Dump** button at the right hand side inside the **Load Thread Dump tab**. The current Thread dump will then be created inside the **Thread Dump tab**.

SNMP

The screenshot shows the 'Monitoring' tab in the Informatica MDM console. On the left, a tree view shows the hierarchy: Network > Application Server > pim-server1 > SNMP. The main area displays a table of SNMP data for the selected server.

Name	Object ID	Value
remote command	1.3.6.1.4.1.200.3.0	
description	1.3.6.1.4.1.200.1.1.1	
NonHeapMemoryUsage	1.3.6.1.4.1.200.1.1.2.1	183
HeapMemoryUsage	1.3.6.1.4.1.200.1.1.3.1	1861
SystemCpuLoad	1.3.6.1.4.1.200.1.1.4.1	0.10749979627238839
ProcessCpuTime	1.3.6.1.4.1.200.1.1.5.1	0
SystemLoadAverage	1.3.6.1.4.1.200.1.1.6.1	-1.0
SystemCpuLoad	1.3.6.1.4.1.200.1.1.7.1	0.10749979627238839
TotalCompilationTime	1.3.6.1.4.1.200.1.1.8.1	106437
TotalLoadedClassCount	1.3.6.1.4.1.200.1.1.9.1	19029

To view data with the **Simple Network Management Protocol**, you need to follow the configuration instructions at the following location : Installation and Operation / Operation / Server Operation / [SNMP Monitoring](#).

Selecting the **SNMP** node, allows you to view all available data of the selected server using SNMP. You can view all available data and its corresponding **Object ID**(OID) which you can later use to monitor your system, using SNMP.

5.6 Deployment

Inside the deployment tab you have the opportunity to configure and administrate deployed systems inside your network. You can:

- administrate and install selected server installation files
- set or change the license file and license key
- edit server configuration files remotely

5.6.1 Installation

The screenshot shows the 'Deployment' tab in the Informatica MDM console. On the left, a tree view shows the hierarchy: Application Server (Deployment) > Installation. The main area displays the 'Installation' configuration page.

Server State

Identifier	Host	State
pim-server1	localhost	no state

Installation Files

Name	Time	Size
------	------	------

File Actions (Installation Files)

File upload:

Selected Files:

Install Actions

☒ Use service log on credentials

Username (domainname/username):

Password:

Selecting the Installation node allows you to inspect the installation of the running systems. You can view all running servers at the **Server State** tab. The **Installation Files** tab lets you administrate all currently used installation files. Inside the **File Actions** tab you have the options to chose installation files yourself and upload them. You can also select already used installation files and either download or delete them, using the corresponding buttons at the bottom right side. The **Install Actions** tab allows you to install PIM using the existing installation files.

For a step by step instruction see the Server Installation documentation. Be aware that the files of the server on which you are connected are used to distribute it to the others.

5.6.2 License

The screenshot shows the 'License' configuration page within the 'Deployment' tab. The left sidebar contains a tree view with 'Application Server (Deployment)' expanded, showing 'Installation', 'License', and 'Configuration'. The main content area has a 'License' tab selected. It contains a 'License File' section with a 'Drop file here' text and a 'Choose File...' button. Below it is a 'License Key' section with a 'new License Key' text and an 'Upload' button. At the bottom right is a 'Save' button.

To change the used license file and license key, the **License** node might be helpful. inside the **License** tab you can simply choose and upload a license file using the corresponding buttons at the right hand side. Pasting a license key inside the license key textfield allows you to use a new license key in your system.

5.6.3 Configuration

The screenshot shows the 'Configuration' page within the 'Deployment' tab. The left sidebar shows the 'Configuration' node selected. The main content area has a 'Configuration' tab selected. It contains a 'Server State' section with a table showing server details. Below it is a 'Configuration Update' section with a checkbox for 'Use service log on credentials', a 'Username (domain\username):' field, a 'Password:' field, and an 'Update Configuration' button.

Identifier	Host	State
pm-server1	localhost	no state

The **configuration** node allows you to configure used configuration files of your system. After putting in your credentials inside the **Configuration Update** tab, you can view and configure all kind of system properties inside your network.

5.7 Support

The Support tab features tools to support you collecting all kind of system data.

5.7.1 CSM

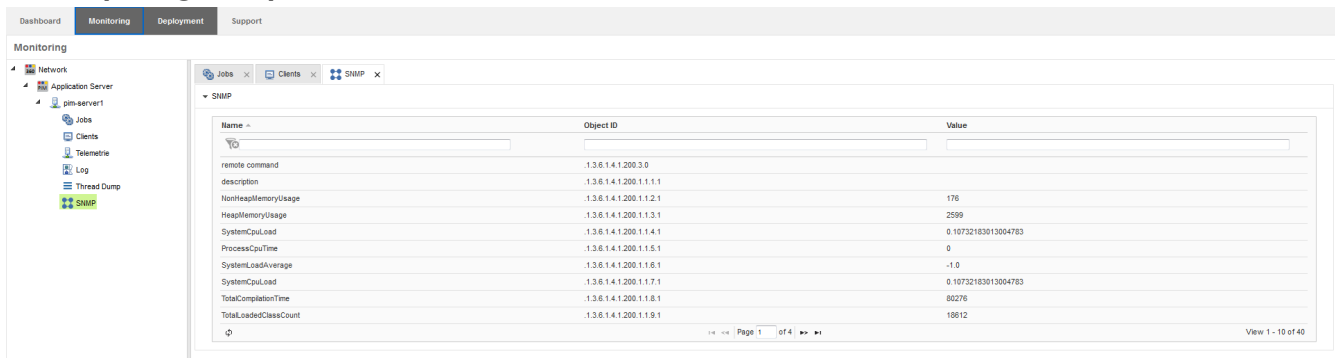
The screenshot shows the 'CSM' page within the 'Support' tab. The left sidebar shows the 'CSM' node selected. The main content area has a 'CSM' tab selected. It contains a 'Server State' section with a table showing server details. Below it is a 'CSM Action' section with a dropdown menu for 'Select the Environment Type...' and a 'Collect CSM' button.

Identifier	Host	State
pm-server1	localhost	Failed to collect the CSM. Please check the clusterx logs

The CSM node allows you to collect data about running servers in your system. Inside the **CSM Action** tab, select your desired environment type and hit the Collect CSM button. All server will automatically collect data and present it to you.

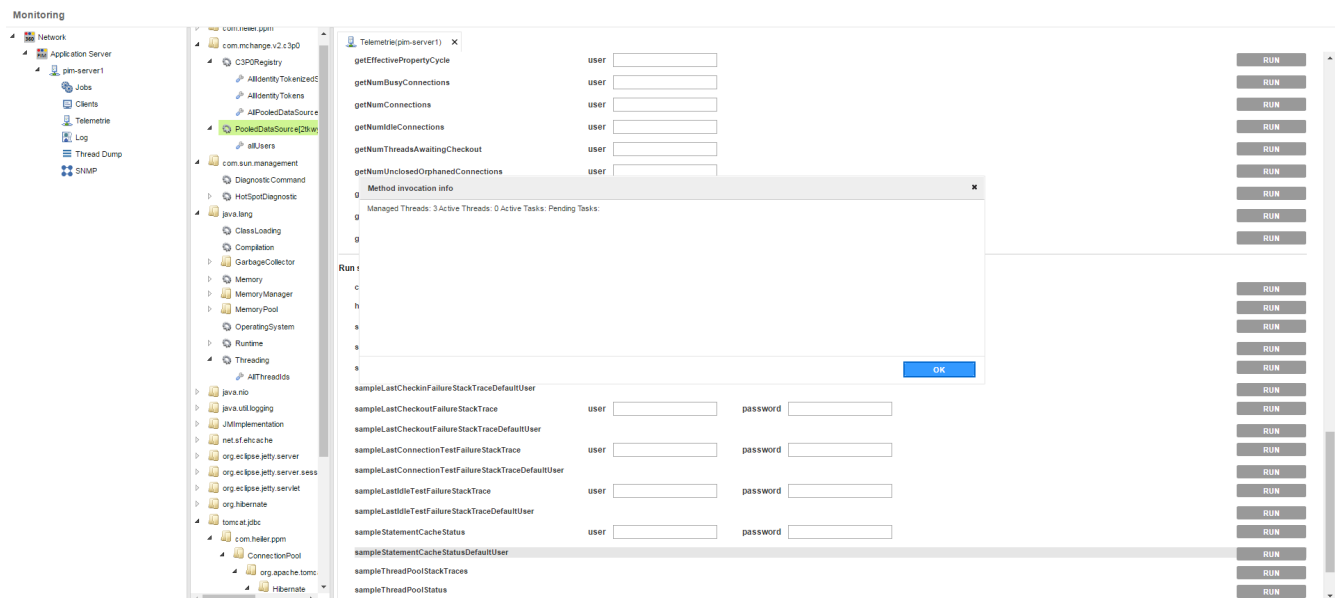
5.8 Additional Features

5.8.1 Opening multiple tabs



Holding Ctrl and selecting a node inside one of the navigation trees, opens up multiple tabs. You can switch between these tabs by simply clicking on them at the top of the screen.

5.8.2 Telemetry Actions



The **Telemetry Actions** tab allows you to change data of the selected Node. By carefully filling out the given fields with valid data, you can change system parameters immediately. There are three different kinds of operations available:

- **Retrieve specific data:** Used to get certain information about an object
- **Change specific data:** Used to change specific object values
- **Run specific data:** Used to run specific operations on the given MBean

6 Audit Trail Operation

For more details about data workflow see the system landscape diagram.

In a standard operation mode PIM - Audit Trail processor intercepts every data management request (create/update/delete) and generates a ATCS record which is sent to JMS server. If JMS server is not accessible ATCS record is saved locally on disk. When JMS server is back online saved ATCS records are sent to the server (it is not recommended that JMS server is

not accessible for a long time). JMS server is configured in a way that PIM Core writes delta changes (ATCS) into a jms topic, but audit trail server is reading these records concurrently from jms queue. It is achieved by ActiveMQ virtual topics feature. This feature is routing messages from topic into queue. On next step in the audit trail data flow the audit trail server reads ATCS records from the JMS queue. It happens asynchronously and does not affect PIM Core any more. Audit trail server converts the ATCS records into its own data structures and save them into the audit trail database. If JMS server is not available then audit trail server will wait for it and re-establish connection when it's back again. If audit trail server is down for some time then the messages will stay in the queue. This architecture guaranties stability and good performance of the audit trail feature.

ATCS - Audit Trail Change Set - PIM Core independent model of entity change.

6.1 Maintenance

6.1.1 Audit Trail access security

Audit Trail server remote access can be protected using server authentication. You can manage security credentials in server setup console (/bin/setup_console.bat).

Available console commands:

- `userAdd <username> <password>` - add new remote user
- `userDelete <username>` - remove user
- `userSetPassword <username> <old_password> <new_password>` - update user password
- `userList` - show list of existing users.

6.1.2 Audit Trail database operations

You can manage Audit Trail storage by using server setup console (/bin/setup_console.bat).

Available console commands:

dbInstall

`dbInstall [noprompt [forceReinstall]]` - perform DB installation operation and all stored procedures/triggers related it according to **server.properties**.

```
dbInstall - will ask confirmation to install, reinstall from scratch or update DB.
dbInstall noprompt - will perform DB installation without prompts. If db already exists - it will skip installation and exit
with error "Schema SCHEMANAME already exist. Setup finished."
dbInstall noprompt forceReinstall - will perform DB installation without asking confirmation. If DB already exists - rewrite
it.
```

dbDelete

`dbDelete [noprompt]` - perform DB delete operation and all stored procedures/triggers related it.

```
dbDelete - will ask confirmation to delete DB or not.
dbDelete noprompt - will perform DB delete operation without asking confirmation.
```

dbInfo

`dbInfo` - shows current size of database and detalization by tables like rows count and table size.

dbTruncate

`dbTruncate <yyyy-MM-dd>` - will truncate database data which is older than specified date.

dbTruncate 2012-05-01 - will truncate all changes which was logged to database before 1 May exclusive. For example an entry dated 2012-04-30 23:59:59 and older will be removed, and records dating 01/05/2012 00:00:00 and newer will remain

dbBackup

dbBackup [name <TARGET_DB_NAME>] [move] [<yyyy-MM-dd> [<yyyy-MM-dd>]] - will perform database backup.

[name <TARGET_DB_NAME>] - is optional command parameter which allows to specify backup database name. In other case will be used default name from server.properties.

[move] - is optional parameter which indicates that original data from source database should be removed after backup operation.

[<yyyy-MM-dd> [<yyyy-MM-dd>]] - is optional parameters which allows to set date period for backup operation.

- Without these arguments backup operation will be performed for entire database data.
- First argument is limiting data before specified date, second argument limiting data after specified date ([<Till> [<From>]])
- If you specify only first argument it will backup all data which is older than specified date.

dbBackup 2012-05-01 - will backup all database entries which was logged to database before 1 May exclusive.
dbBackup 2012-05-01 2012-01-01 - will backup all database entries which was logged to database from 1 January(inclusive) till 1 May (exclusive) at 2012, i.e. all data which was logged during 1st quarter of 2012.
dbBackup name AUDITTRAIL_2012Q1 move 2012-05-01 2012-01-01 - will create (if doesn't exist) database with AUDITTRAIL_2012Q1 name and move all data for 1st quarter from original database to it. Data from original database for 1st quarter will be removed.



Default backup database name can be updated at **server.properties**. If database doesn't exist it will be created during execution of first backup operation.

Dbbackup operation needs an installed SQL client (sqlcmd) on the Audit trail server.

It's possible to backup different time periods in one Audit Trail archive database.

6.1.3 Available application arguments

Commands **dbInstall**, **dbDelete**, **dbTruncate**, **dbBackup** are also available as arguments for console launcher.

You can find some examples of batch files at /bin/ directory.

- /bin/setup_db_install.bat - call **dbInstall noprompt** command.
- /bin/setup_db_install_force - call **dbInstall noprompt forceReinstall** command
- /bin/setup_db_truncate.bat - call **dbTruncate <yyyy-MM-dd>** command. Date argument should be provided as argument for this batch file. For example you can execute it at windows command line like that: setup_db_truncate.bat 2012-05-01
- /bin/setup_db_backup.bat - call **dbBackup [name <TARGET_DB_NAME>] [move] [<yyyy-MM-dd> [<yyyy-MM-dd>]]** command. All optional arguments should be provided as argument for this batch file. For security reason, You will be prompted to confirm execution of this operation.



Hint

To skip confirmation of execution for *setup_db_backup.bat* just add **noprompt** argument before other optional arguments. For example you can execute it at windows command line like that: *setup_db_backup.bat noprompt name AUDITRAIL_2012Q1 move 2012-05-01 2012-01-01*

6.1.4 Purge Audit Trail database strategy

To purge old data from AT database you should follow next steps:

1. Shutdown Audit Trail service using */bin/serviceStop.bat*. If some other instances of Audit Trail Server is running, they should be stopped also.
2. Run */bin/setupConsole.bat*
Perform purge operation, for example, we will purge all data which is older than 1 May 2012:
 - a. Backup case: execute at console command **dbBackup move 2012-05-01** - will backup to default backup database all entries which was logged to original database before 1 May (exclusive). All data from original database before 1st May 2012 will be removed.
 - b. Delete case: **dbTruncate 2012-05-01** - will truncate all changes which was logged to database before 1 May 2012 (exclusive).
4. Wait till operation will be completed.
5. Exit from console using command **exit** or just close it.
6. Run Audit Trail service using */bin/serviceStart.bat*

6.1.5 Running scheduled task

If you want running some regular task, you may set up that using the windows task scheduler.

So, what only you need is setting up the windows task that is aimed to run your bat-file with passing appropriate parameters into it. in You can find the example */bin/backupTask.bat* how create a bat-file for the periodical backup of database.

If we put aside the preparation, then remain the core of the script. This snippet of code include only main operations:

Example of running the windows task

```
rem stop the service
"%WRAPPER_EXE%" -p %WRAPPER_CONF% "wrapper.java.classpath.2=%EQUINOXJAR%" "wrapper.java.classpath.3=%OSGIJAR%"

rem run database backup task

IF NOT "%_PARAMETERS%" == "" (
    CALL "cmd /c start /D%ECLIPSEHOME% atserver.exe -console -application com.heiler.audittrail.server.setup dbBackup
    %_PARAMETERS%"
) ELSE (
    CALL "cmd /c start /D%ECLIPSEHOME% atserver.exe -console -application com.heiler.audittrail.server.setup dbBackup"
)

rem start the service
"%WRAPPER_EXE%" -t %WRAPPER_CONF% "wrapper.java.classpath.2=%EQUINOXJAR%" "wrapper.java.classpath.3=%OSGIJAR%"
```

6.1.6 Audit Trail JMS consumer

By default JMS consumer service is starting with server and reading change events from PIM Core. Consumer autostart can be controlled in the **server.properties** in Audit Trail server by setting property **jms.consumer.autostart** to **true** or **false**

You can manage JMS consumer service directly from the server console at runtime (*/bin/console.bat*).

Console command:

- `jmsConsumer start|stop` - start|stop Audit Trail JMS consumer service.

6.2 Tuning

6.2.1 PIM Audit Trail MXBean

Audit Trail server provides MXBean **audittrail** in **com.heiler.ppm** group which allow to see some statistics.

The screenshot shows the Java Monitoring & Management Console with the 'MBeans' tab selected. The left pane displays a tree view of the MBean hierarchy, with 'com.heiler.ppm.auditTrail' expanded. The right pane shows the 'Attribute values' table for the selected MBean.

Name	Value
ATCSbuiltCount	193823
AverageTime	60585.13666592716

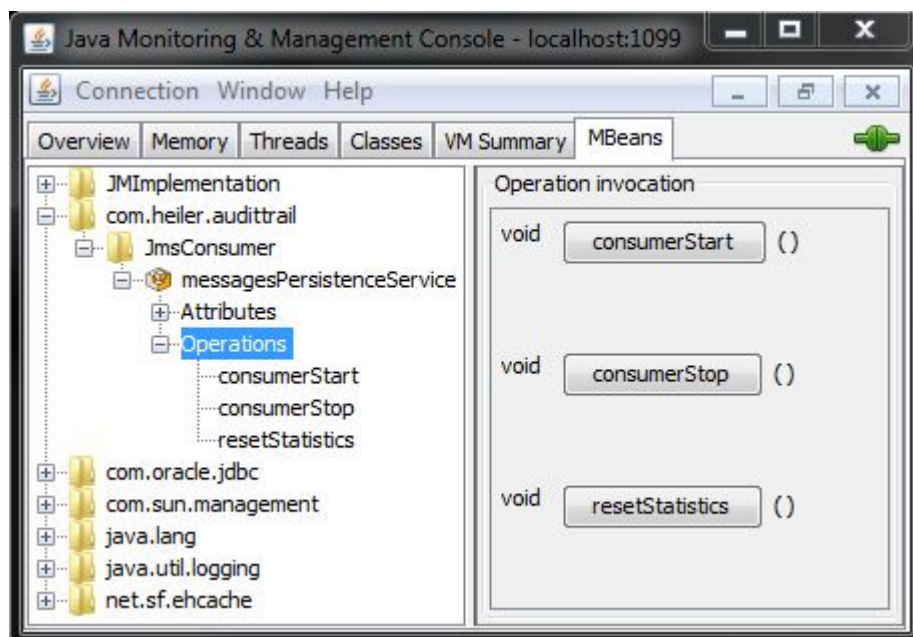
A 'Refresh' button is located at the bottom right of the console window.

- **atcsChangeStatistics** - how many change sets generated for change type **Change** and average time in mili seconds.
- **atcsBulkChangeStatistics** - how many **bulk** change sets generated for change type **Change** and average time in mili seconds.
- **atcsBulkDeleteStatistics** - how many **bulk** change sets generated for change type **Delete** and average time in mili seconds.
- **atcsLoadEIsStatistics** - how many times Entity Items where loaded before **Delete** event and average time in mili seconds.
- **auditTrailJmsStatistics** - shows statistics of JMS producer. Allows to choose whether messages will be persisted by JMS server or not.

- **auditTrailProcessorStatistics** - shows statistics of Audit Trail processor. Allows to tweak its thread pool at run time.

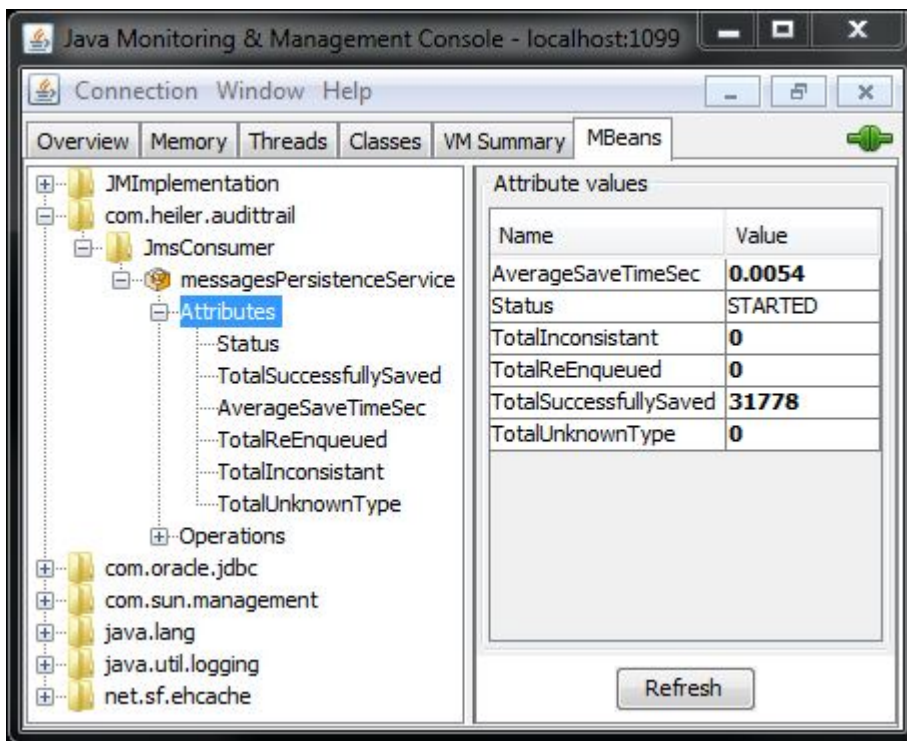
6.2.2 Audit Trail server MBean

Audit Trail server provides MBean **com.heiler.audittrail** which allow manage JMS consumer service.



Also it provides basic service statistics for all period from time when service was started or from last reset of statistic.

- **AverageSaveTimeSec** - average save time of one change set ,which was received from JMS server, into database
- **Status** - current status of service (*STOPPED* | *STARTING* | *STARTED* | *STOPPING*)
- **TotalInconsistent** - count of inconsistent messages which can't be saved at database and was skipped. Usually never happens, and can occurs only in case when database structure is not actual. In this case you should ensure that all database updates were installed. For detailed information please check server logs file to find out reason.
- **TotalReEnqueued** - count of JMS messages which AT server was unable to handle and has returned to JMS queue. It can happens while database server unreachable at current moment.
- **TotalSuccessfullySaved** - total count of change sets which was successfully saved at Audit Trail database.
- **TotalUnknownType** - total count of JMS messages which was received by AT server, which was not recognized as correct change set message and was ignored.



6.2.3 Audit Trail Server - second instance

This section describes how to start second read-only instance of the Audit Trail server on the same machine. Second instance is necessary if you need to provide access to the Audit Trail backup or archive database. *In this manual we assume that master instance of the audit trail server is already installed.*

Unpack Audit Trail server package into a separate folder (for example C:\audittrailServer-secondInstance\win32.win32.x86) and edit server.properties file as described in the audit trail installation and configuration section except several properties:

backup database properties:

```
dest.host = mssql_host_main/ mssql_host_backup
db.default.type = MSSQL2008
db.audittrail.schema = audittrail _schema_main/ audittrail _schema_backup
db.audittrail.port = 1433
db.audittrail.user = mssql _ user
db.audittrail.password = mssql _ password
```

Use other communication port to avoid conflict with the master instance:

```
server.communication.port = 2802
```

Disable Audit Trail change set JMS consumer (second instance is not allowed to save Audit Trail data):

```
jms.consumer.autostart = false
jms.consumer.jmxregistration = false
```



Make sure that JMS consumer in the second instance is deactivated. Otherwise the second instance will try to save live data from the JMS queue into the backup database.

Change JMX remote port:

In the `atserver\atserver.ini`

```
-Dcom.sun.management.jmxremote.port = 1099
```

and in the `atserver\bin\conf\wrapper.conf`

```
wrapper.java.additional.7 = -Dcom.sun.management.jmxremote.port = 1099
```

To install second instance as a windows server open `wrapper.conf` make the following changes:

```
wrapper.name = audittrailserver => wrapper.name = audittrailserverAnotherInstance  
wrapper.displayname = Heiler Enterprise PIM Audit Trail Server => wrapper.displayname = Heiler Enterprise PIM Audit Trail  
Server Another Instance  
wrapper.description = Heiler Enterprise PIM Audit Trail Server => wrapper.description = Heiler Enterprise PIM Audit Trail  
Server Another Instance
```

and run `atserver\bin\serviceinstall.bat`

7 Log File Overview

This page gives an overview about the log files and their configuration options.

Module	Default Log File Location	Log File Configuration	Remarks
Application Server Web Client	<ServerInstallationRoot>\server\logs*.log <ServerInstallationRoot>\server\wrapper.log	<ServerInstallationRoot>\server\configuration\HPM\log4j.xml	Application log files (out, error, perf) Windows Service Wrapper log
Desktop Client	<ClientInstallationRoot>\client\logs*.log	<ClientInstallationRoot>\client\configuration\HPM\log4j.xml	
Supplier Portal	<SupplierPortalInstallationRoot>\logs <SupplierPortalInstallationRoot>\tomcat\logs	<SupplierPortalInstallationRoot>\configuration\logback.xml	Application Logs Tomcat Logs

Module	Default Log File Location	Log File Configuration	Remarks
Media Manager	<pre> <MediaManagerInstallationRoot> \Informatica Media Manager\IMM_protocols <MediaManagerInstallationRoot> \OpasGWebServer\Tomcat\logs <MediaManagerInstallationRoot> \OpasGWebServer\XOBSessionMa nager\logs </pre>		
Web Search	<pre> <WebSearchInstallationRoot>\l og\server.log <WebSearchInstallationRo ot>\<tomcat-server>\logs </pre>	<pre> <WebSearchInstallationRoot>\<tomca t-server>\lib\log4j.xml <WebSearchInstallationRoot>\ <tomcat- server>\conf\logging.propert ies </pre>	Application Logs Tomcat Logs
Database Setup	<pre> <DatabaseSetupInstallationRo ot>\dbclient\logs*.log </pre>	<pre> <DatabaseSetupInstallationRoot>\db client\configuration\log4j.xml </pre>	

7.1 Additional Logging of Dataquality

If you want verbose logging of the execution of a dataquality rule by the rule engine, go to the `log4j.xml` file and change the priority of the IDQ category like below:

```

<category name="IDQ">
  <priority value="DEBUG"/>
</category>

```

The log files can also be found in the server log folder `<ServerInstallationRoot>\server\logs\`. Each rule configuration will be executed in it's own worker thread, so a log file will have following name pattern:

`<JobWorkerName>-<RuleConfigurationName>-<StartDate>.log`

7.2 Additional Logging of Change Summary and Datagraph

To get the insides of datagraphs and change summaries before store them to the database you can enable the following logger. You can define the logger for each subclass of

```
com.heiler.ppm.std.server.entity.EntityPersistenceManagerImpl
```

This sample enables the logging for `ArticlePersistenceManager`.

```

<category name="com.heiler.ppm.article.server.internal.entity.ArticlePersistenceManager">
  <priority value="TRACE"/>
</category>

```

Use this logger wisely. The logging output is big.

7.3 Additional Logging of Media Asset Processing

If you want to fetch the logging of JMS processing, go to the `log4j.xml` file and add the following setting:

```
<category name="PPM.MEDIA_ASSET_NOTIFICATION">
  <priority value="DEBUG"/>
</category>
```

If you want to fetch the logging of Media Manager connector API methods, go to the `log4j.xml` file and add the following setting:

Note: Connector logging via PIM is available since PIM 8.0.01

```
<category name="com.heiler.imm.connector">
  <priority value="DEBUG"/>
</category>
```

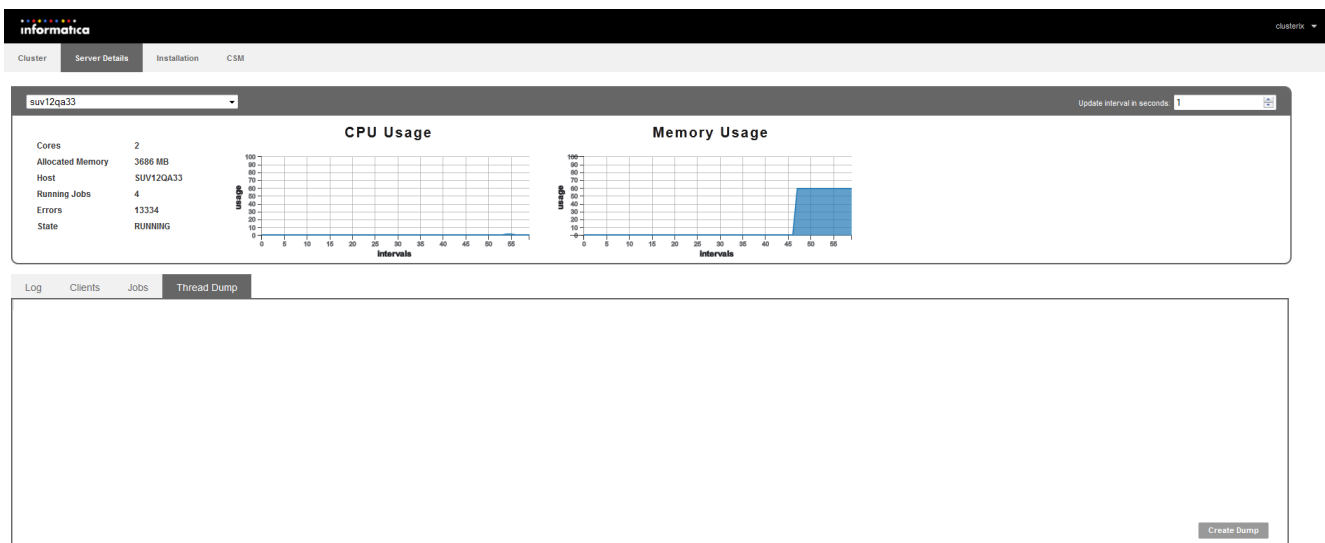
For more details information please visit the page [Logging for Media Manager Connector within PIM](#).

8 How to create a thread dump

This page describes how to obtain a thread dump for the product manager's client and server. A thread dump returns the stack traces of **all threads** of the java virtual machine it's executed on, at the moment of execution.

8.1 Control Center

The easiest way to create a thread dump of a running server is by using the Control Center. In the `Server details` view you can find the `Thread Dump` tab. By clicking on `Create Dump`, the dump will be created for the selected server and displayed in the view where it can be analyzed.



8.2 Console Window

When the client or server is being executed in debug mode (using the start_debug.cmd files) you can easily perform a thread dump to the console window. Just press CTRL + BREAK. This will print the stack trace to the console window.



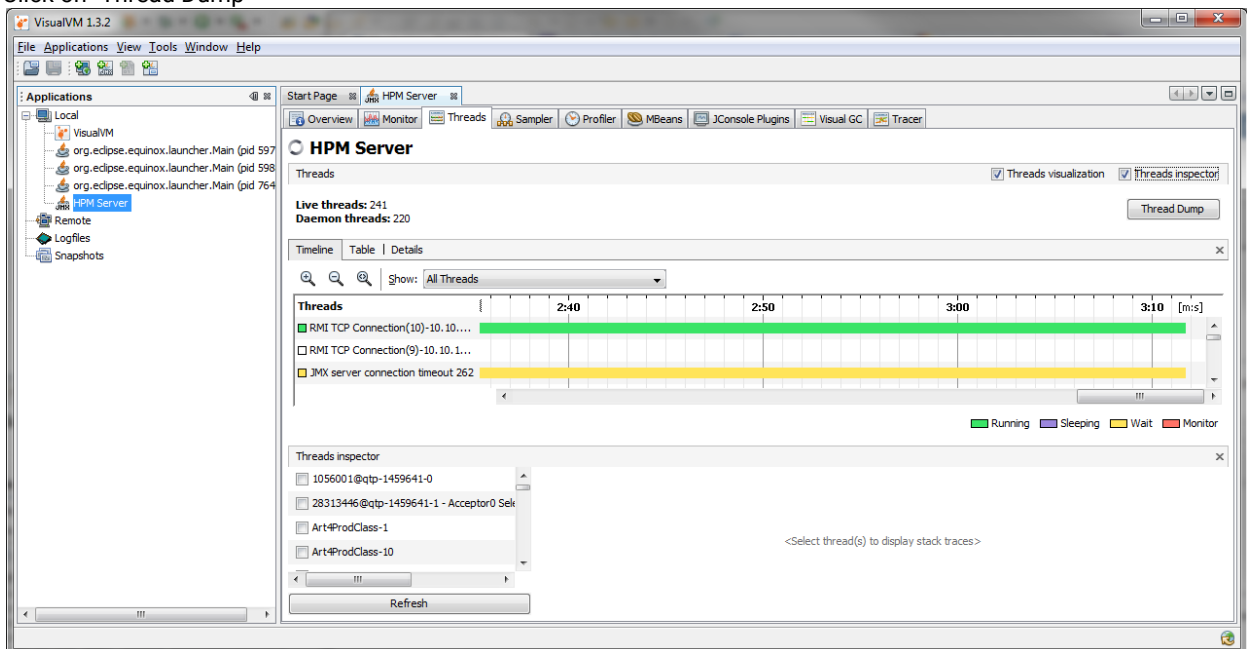
Starting with PIM - Server version 5.x this also performs a memory dump - which might take a while.

Of course, this is no feasible way for a productive system, which should not be run in debug mode at all!

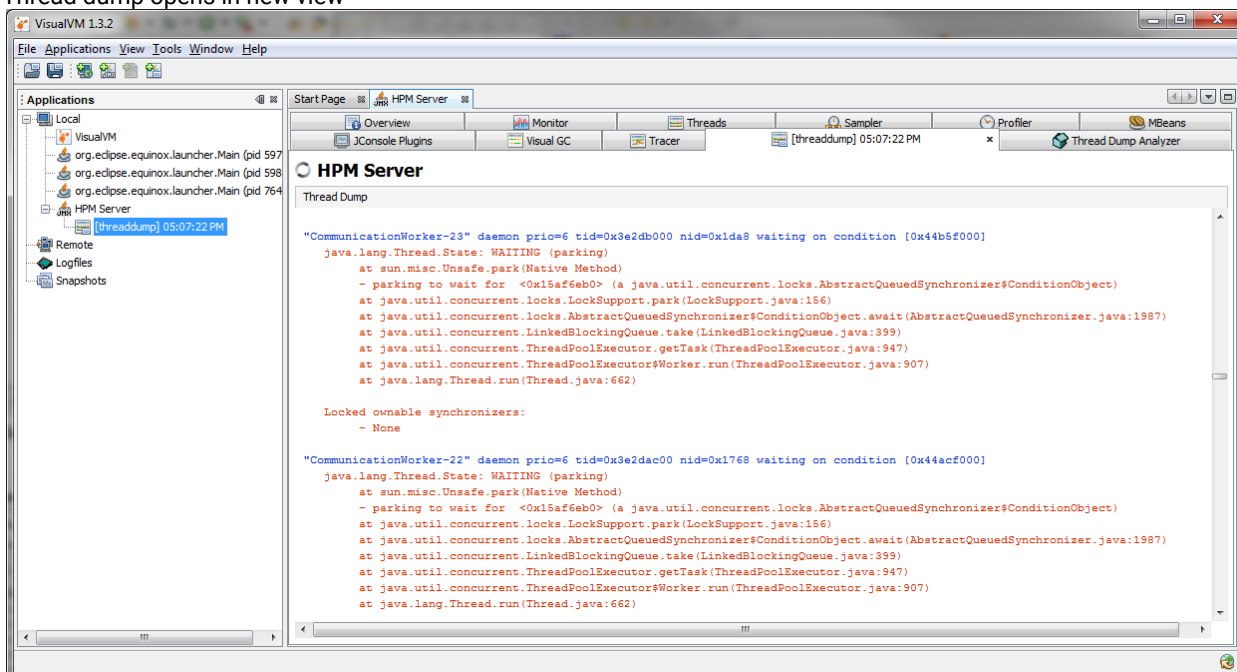
8.3 VisualVM (JMX Interface)

With VisualVM you can connect to a running java virtual machine and perform a thread dump at any given time. Please see the [Tuning Advisory](#) for details on how to enable the JMX (Java Management eXtension) for the server.

1. Open Visual VM
2. [Connect to the server](#)
3. Switch to "Threads" view
4. Click on "Thread Dump"



5. Thread dump opens in new view



If you have the choice, use Visual VM - it's much easier to work with 😊

If the tool hangs during Thread dump loading, you can also find the written DUMPS in the local user directory at C:\Users\<userId>\AppData\Roaming\VisualVM\8u40\repository\jmxapplications

8.4 Analyzing Thread Dumps

Analyzing Thread Dumps requires some internal knowledge of PIM Server architecture. Here are some useful hints:

- The purpose of thread groups can be usually determined by the thread's name.
 - CommunicationWorker are used by the internal communication framework to process requests from clients
 - http-* are managed by Jetty web server and used for both PIM Web as well as REST calls
- A useful tool for analyzing ThreadDumps is ThreadDumpAnalyzer <https://github.com/spotify/threaddump-analyzer>
 - It's JavaScript based and can be executed locally on a html page.
 - Threads with identical stacks are grouped
 - Locks between threads are visualized

Online Java Thread Dump Analyzer

file:///C:/Tools/threaddump-analyzer-gh-pages/index.html#thread-105474

Online Java Thread Dump Analyzer

Paste your thread dump here:

```
"Reference Handler" - Thread t@2
  java.lang.Thread.State: WAITING
    at java.lang.Object.wait(Native Method)
    - waiting on <5c019e1f> (a java.lang.ref.Reference$Lock)
    at java.lang.Object.wait(Object.java:502)
    at java.lang.ref.Reference$ReferenceHandler.run(Reference.java:157)

Locked ownable synchronizers:
- None
```

AnalyzeClear

This page is client-side only, and no data will leave your computer when you click Analyze.

Top Methods From 71 Running Threads

```
55 java.net.SocketInputStream.socketRead0(Native Method)
8 sun.nio.ch.WindowsSelectorImpl$SubSelector.poll0(Native Method)
5 java.net.DualStackPlainSocketImpl.accept0(Native Method)
1 java.io.FileInputStream.readBytes(Native Method)
1 sun.management.ThreadImpl.dumpThreads0(Native Method)
1 sun.nio.ch.ServerSocketChannelImpl.accept0(Native Method)
```

406 threads found

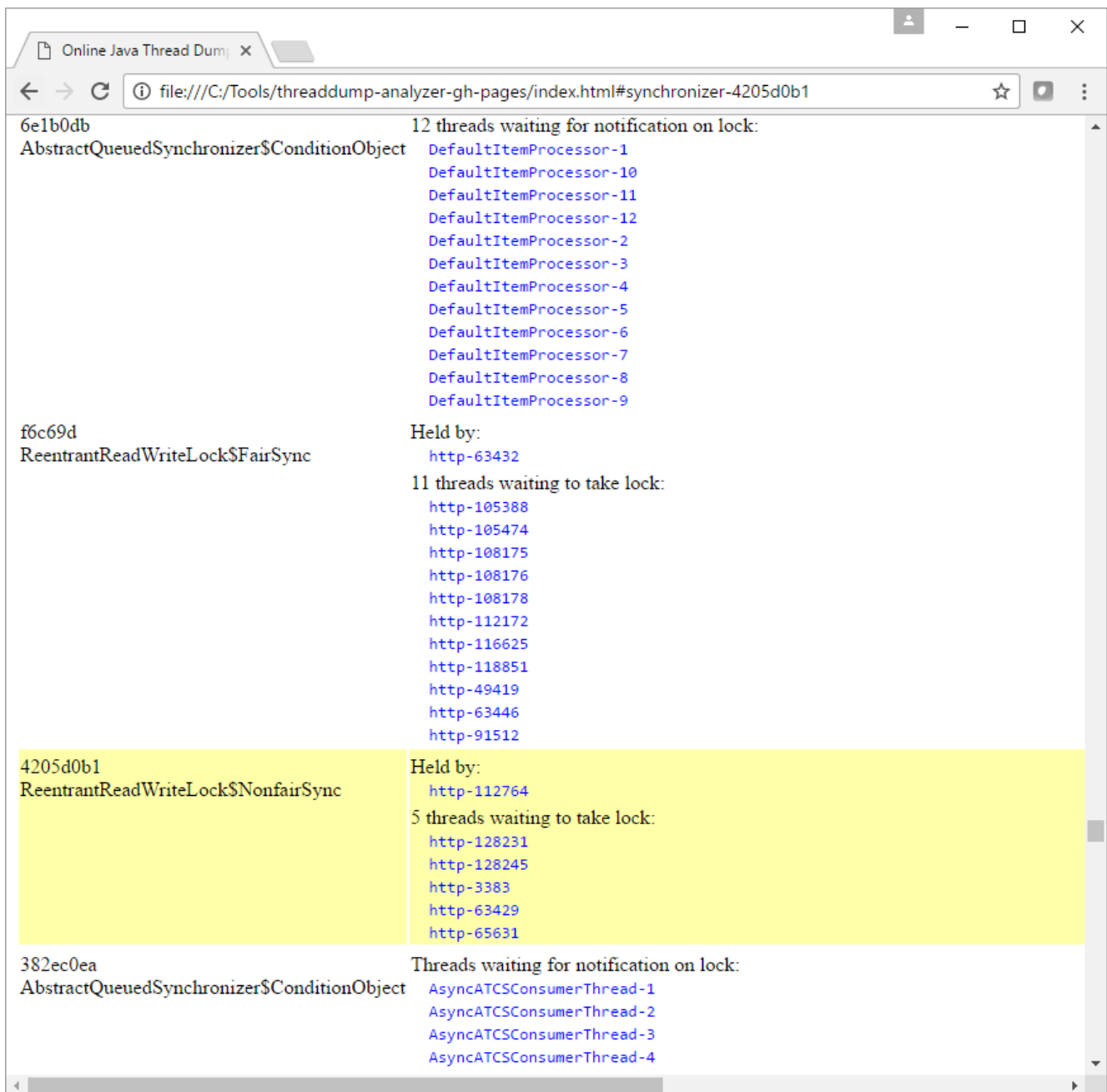
79 threads with this stack:

```
"AsyncATCSConsumerThread-1": awaiting notification on [382ec0ea]
"AsyncATCSConsumerThread-2": awaiting notification on [382ec0ea]
"AsyncATCSConsumerThread-3": awaiting notification on [382ec0ea]
"AsyncATCSConsumerThread-4": awaiting notification on [382ec0ea]
"CommunicationWorker-1": awaiting notification on [5799a2ec]
"CommunicationWorker-10": awaiting notification on [5799a2ec]
```

Informatica MDM - Product 360

Operation

88



9 How to enable Java Management Extensions (JMX)

The Product 360 application server has various performance and management extensions which are available through the standard Java Management Extensions (JMX) interface. By default, the application server is started with the needed JMX parameters in place. This page describes those parameters for older versions or just as a reference.

Development and Test environment

To enable PIM - Server jmx you need to set jmx java properties. In the development environment use eclipse launch config. In test environment set them in the `console_debug.cmd`:

How to enable JMX local access

```
-Dcom.sun.management.jmxremote
```

9.1 Production environment

In production use `wrapper.conf` file:



Tanuki Wrapper

Because of bugs in the tanuki wrapper the jmv processes sometimes are not visible when started as a windows service (see Tanuki FAQ). In this case you need to configure jmx remote access. If you are running a production system make sure that firewall protects the jmx port from unauthorized access. See oracle JMX configuration manual for more details

How to enable JMX remote access

```
#This is not safe from security point of view!  
-Dcom.sun.management.jmxremote.port=55555  
-Dcom.sun.management.jmxremote.authenticate=false  
-Dcom.sun.management.jmxremote.ssl=false
```

Add this lines to the `wrapper.conf` file in the Block # VM OPTIONS. Continue the existing lines (ending with "wrapper.java.additional.25"):

```
...  
wrapper.java.additional.26 = -Dcom.sun.management.jmxremote.port=55555  
wrapper.java.additional.27 = -Dcom.sun.management.jmxremote.authenticate=false  
wrapper.java.additional.28 = -Dcom.sun.management.jmxremote.ssl=false
```

9.2 Download and Installation of JMX Tooling

To monitor managed beans you can use `jconsole.exe` (available in the JDK 1.5 and higher) or `visualvm` (available in the JDK 1.6 and higher or as a standalone application) with JMX plugins. Please note that the development department will need these tools on the corresponding machine if you want us to have a look on the system - it always streamlines the help processes if those tools are already installed.



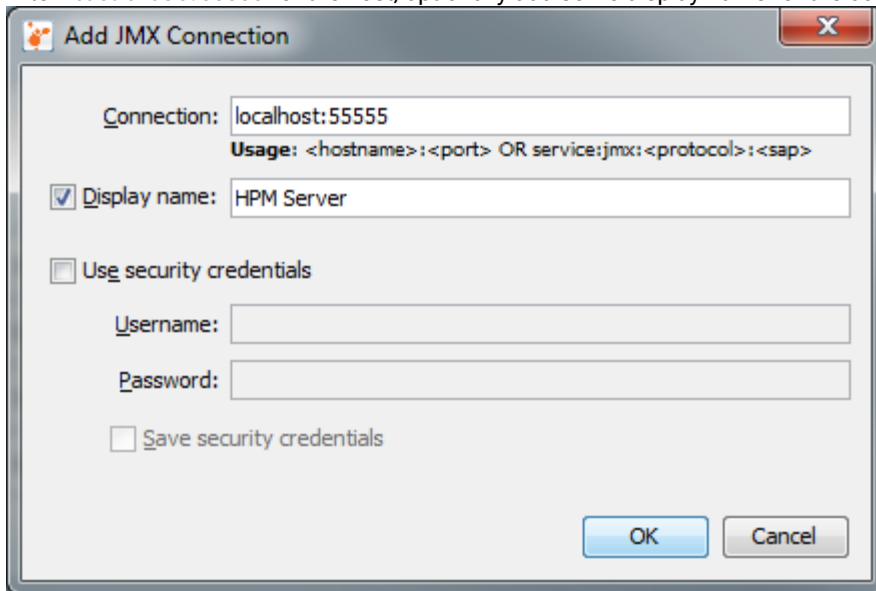
User permissions

In some cases you will need to start `jconsole`/`VisualVM` with admin permissions (in windows file explorer "Start as Administrator") in order to see java process started as services

9.2.1 Connect on productive system

1. Open Visual VM
2. Go to File / Add JMX Connection

3. Enter localhost:55555 for the host, optionally add some display name for the connection



The image shows a dialog box titled "Add JMX Connection". It has a standard Windows-style title bar with a close button (X) in the top right corner. The dialog contains the following fields and options:

- Connection:** A text field containing "localhost:55555". Below it, a usage hint reads: "Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap>".
- Display name:** A checked checkbox followed by a text field containing "HPM Server".
- Use security credentials:** An unchecked checkbox.
- Username:** A text field, currently empty.
- Password:** A text field, currently empty.
- Save security credentials:** An unchecked checkbox.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

4. Press ok
5. Connect to the application HPM Server by double click on the application name in the Applications view