



# Informatica<sup>TM</sup>

## Migration

Informatica MDM - Product 360

Version: 10.1

# Table of Contents

<b>1</b>	<b>Pre-Migration Checklist .....</b>	<b>6</b>
<b>2</b>	<b>Control Center Migration .....</b>	<b>6</b>
2.1	Updating Control Center .....	6
<b>3</b>	<b>Repository Migration .....</b>	<b>7</b>
3.1	Repository Merger .....	7
3.1.1	Installation and Execution.....	8
3.1.2	Troubleshooting.....	9
3.1.2.1	Duplicate Elements .....	9
<b>4</b>	<b>Database Migration.....</b>	<b>10</b>
4.1	Media Manager Database Migration .....	10
4.1.1	Updating the Product 360 - Media Manager database.....	10
4.1.1.1	Automatic database update .....	10
4.1.1.2	Manual database update .....	10
4.1.1.3	Legacy versions .....	11
4.2	Product 360 Core Database Migration .....	12
4.2.1	Duplicate Article Attribute Names in the repository default language .....	12
4.2.1.1	Problem/ Situation .....	12
4.2.1.2	Solution 1 .....	13
4.2.1.3	Solution 2 .....	13
4.2.2	ArticleTrading entries without existing Party entry are removed .....	13
4.2.2.1	Problem/Situation .....	13
4.2.3	Invalid decimal formatting in Article Attribute Values or Structure Group Attribute Values .....	14
4.2.3.1	Problem/ Situation .....	14
4.2.3.2	Solution .....	14
4.2.4	Structure Migration.....	15
4.2.5	While migrating to 8.0.03.....	15
4.2.6	Qualification permission info entries in the log .....	16
4.2.7	Multiple Purposes for Import Mappings in the GenericData Table .....	17
4.2.8	SQL scripts to check and migrate invalid decimal formatting in Article Attribute Values .....	20
4.2.9	SQL scripts to find and correct duplicate attribute names.....	22

<b>5</b>	<b>Server and Desktop Migration.....</b>	<b>27</b>
5.1	Installation of Hotfix, EBF or One-Off-EBF (=Patch) on Server .....	27
5.2	Installation of Hotfix, EBF or One-Off-EBF (=Patch) on Desktop .....	28
5.3	Structure Migration.....	28
5.3.1	Run Structure Migration Tool.....	29
5.4	XML parsing exception while loading export templates or other data graph objects.....	29
5.5	Entity IDs for ArticleLangType and ArticleExtensionType repository custom entities.....	32
5.5.1	Database.....	32
5.5.2	Repository .....	34
5.6	Server Jobs.....	35
5.7	Data Quality.....	36
5.7.1	Migration .....	36
5.7.2	Update .....	36
5.8	Import mappings .....	37
5.9	Migration of Subentity Deletion customizing.....	37
5.10	Media Manager Integration .....	37
5.10.1	Adjust Media Manager database connection string (MSSQL only) .....	37
5.11	Server and Desktop Pre-Migration Checklist.....	38
5.11.1	Structure Migration Preparation .....	38
5.11.1.1	Download Structure Migration Preparation Tool.....	38
5.11.1.2	Install Structure Migration Preparation tool .....	39
5.11.1.3	Run Structure Migration Preparation Tool .....	39
5.11.1.4	Correct invalid item/variant/product mappings.....	39
<b>6</b>	<b>Unit Management.....</b>	<b>40</b>
6.1	Overview .....	40
6.2	Repository .....	41
6.3	Data Migration.....	41
6.4	Enum Provider.....	42
6.5	Desktop UI .....	42
6.6	Import.....	42
6.7	Export .....	42
6.8	Service API .....	43

<b>7</b>	<b>Media Manager Migration .....</b>	<b>43</b>
7.1	Update Media Manager .....	43
7.1.1	Updating Media Manager File Server .....	43
7.1.2	Updating the database .....	43
7.1.3	Updating the client modules .....	44
7.1.3.1	Manually updating the client modules (Windows).....	44
7.1.3.2	Manually updating the client modules (OSX) .....	44
7.1.4	Updating Funcd.....	44
7.1.5	Updating web front end.....	44
7.1.6	Deactivate old modules .....	46
7.1.6.1	Deactivate Session Manager.....	46
7.1.6.2	Deactivate Internet Funcd .....	46
<b>8</b>	<b>Supplier Portal Migration .....</b>	<b>46</b>
8.1	On the Product 360 Supplier Portal Server.....	46
8.1.1	Adjust database connection string (MSSQL only) .....	47
<b>9</b>	<b>Web Search Migration .....</b>	<b>47</b>
9.1	Migration of Items of any Catalog index .....	48
9.1.1	Old Index configuration for field ArticleLang.DescriptionShort.....	48
9.1.2	New export template based configuration.....	48
9.1.2.1	Handling sub-entities in sub-module .....	53
9.2	Migration of Items of All-Supplier-Catalog index .....	55
9.2.1	Configurations prior to v10.0.0.0, this would be defined as: .....	55
9.2.2	New Configuration .....	55
9.2.2.1	Data Type.....	55
9.2.3	Data Sources .....	55
9.3	Migration of 3PPD index .....	57
9.3.1	Old configuration .....	57
9.3.2	New configuration.....	58
9.3.2.1	Exporting data for multi-PPD index .....	58
9.3.2.2	Identifying the parent .....	58
9.3.2.3	Ensuring export of single record exports it's whole hierarchy .....	59
9.3.2.4	Migration of 3PPD template from v10.0.0.0 .....	60
9.4	Migration of 2PPD index .....	60
9.4.1	Old Search Configuration .....	61

9.4.2	New configuration.....	61
9.4.3	Migration of 2PPD template from v10.0.0.0.....	61
<b>10</b>	<b>Business Process Management .....</b>	<b>62</b>
10.1	Workflows for Product 360 Versions 10.1 .....	62
10.2	Workflows for Product 360 versions >= 8.0.00 .....	62
10.3	Workflows for Product 360 versions < 8.0.00 .....	62
<b>11</b>	<b>Audit Trail Migration .....</b>	<b>62</b>
11.1	Pre-Migration Checklist .....	63
11.2	Part 1: Migrate to new Audit Trail module .....	63
11.3	Part 2: Migrate existing data .....	63
11.3.1	General process.....	63
11.3.2	Configuration .....	63
11.3.2.1	Repository .....	63
11.3.2.2	audittrail.migration.server.properties.....	64
11.3.2.3	plugin_customization.ini.....	64
11.3.3	Rights.....	65
11.3.4	Start Migration .....	65
11.3.5	After migration .....	66
11.3.6	Troubleshooting.....	67
11.3.6.1	Why are there so many skipped records? .....	67
11.3.6.2	Re-execute migration.....	67
11.3.6.3	Increase trace level .....	68
11.3.7	Limitations .....	68
<b>12</b>	<b>Message Queue Migration .....</b>	<b>68</b>
<b>13</b>	<b>How to apply a Hotfix or EBF .....</b>	<b>69</b>
13.1	Preconditions. ....	69
13.2	Installation of Hotfix or EBF on Product 360 Server .....	69
13.3	Installation of Hotfix or EBF on Product 360 Desktop.....	70


Technical documentation about migration an installation to a new release can be found [here](#).

# 1 Pre-Migration Checklist

 **Always backup your system before updating! This especially includes the databases, file storage areas, configuration files and binaries distributables.**

Before beginning to migrate your Product 360 system, please check that:

- Your system meets the System Requirements

 Note: Please find the System Requirements in the corresponding version of PAM in MySupport portal.

- You must be able to use a command prompt to continue. If not, please contact your system administrator to assist.
- You have read the Release Notes of the version to be installed.

## 2 Control Center Migration

### 2.1 Updating Control Center

With each Hotfix and Major P360 release, there is also an update for the Control Center.

For updating the P360 Control Center follow the steps below.

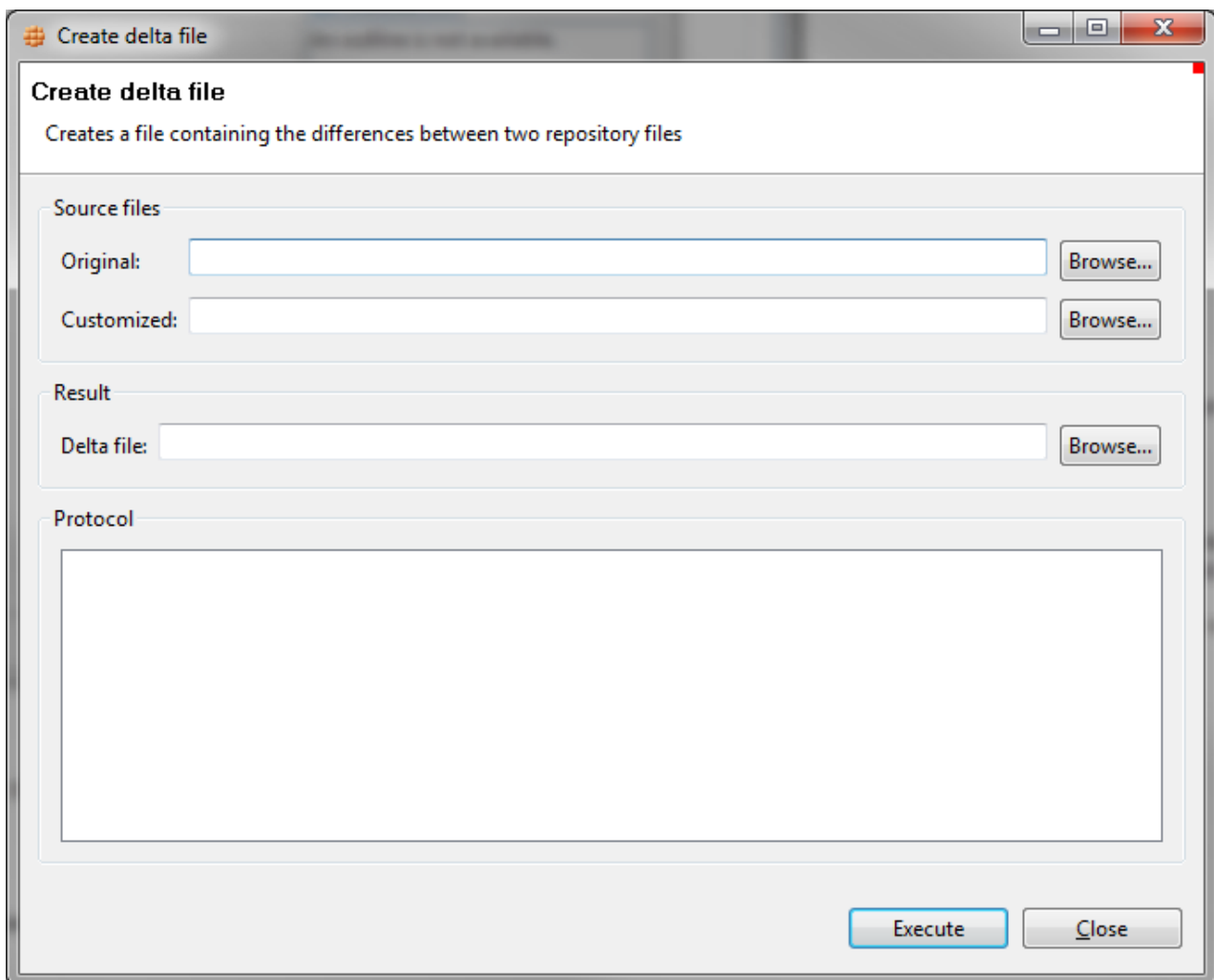
1. Stop the Control Center service if still running.
2. Uninstall the service by executing the `uninstall.cmd` (Windows) or `uninstall.sh` (Linux)
3. Rename the root folder of your current running Control Center in something like `clusterix_old`.
4. Extract the new Control Center package in the same directory as the old one.
5. If you want to encrypt the passwords using standard implementation of encryption you should replace the Java JCE policy files in `jre\lib\security` folder. For more information please refer to chapter Encryption of secure information in the Server Installation manual.
6. If you made any adjustments to configuration files like `environment.conf` or `eclipse.ini` don't forget to adjust these files also. Be careful if you just copy these files from your old installation, since there might have been made some changes. It is safer to copy only your settings instead the whole file.
7. Inside the `configuration` folder re-configure the property files with your settings. It is recommended to **not** copy the old property files on the new ones since there might be changes or new properties. This also applies to property files such as `wrapper.conf` on which usually no modification is needed for an installation, but which might have changed from the standard product.
  - `server.properties`
  - `plugin_customization.ini`
  - network configurations
  - audit trail and `hmm / hsx` properties if needed.

8. If you had changes in Repository please apply them also to the Repository provided by the update package.  
See [Repository Migration](#)(see page 7) for more information.
9. Install the service again by executing `install.cmd` (Windows) or `install.sh` (Linux)
10. Start the Control Center 360 service again

## 3 Repository Migration

### 3.1 Repository Merger

The repository merger reduces the effort to migrate a customized repository to a new standard Product 360 release by automating most of that work. It is possible to create a delta report based on an original repository file and the customized repository file. The delta creation should always be performed with the **newest** merger for which you want to merge the Repository.



**Create delta file**  
Creates a file containing the differences between two repository files

**Source files**

Original:

Customized:

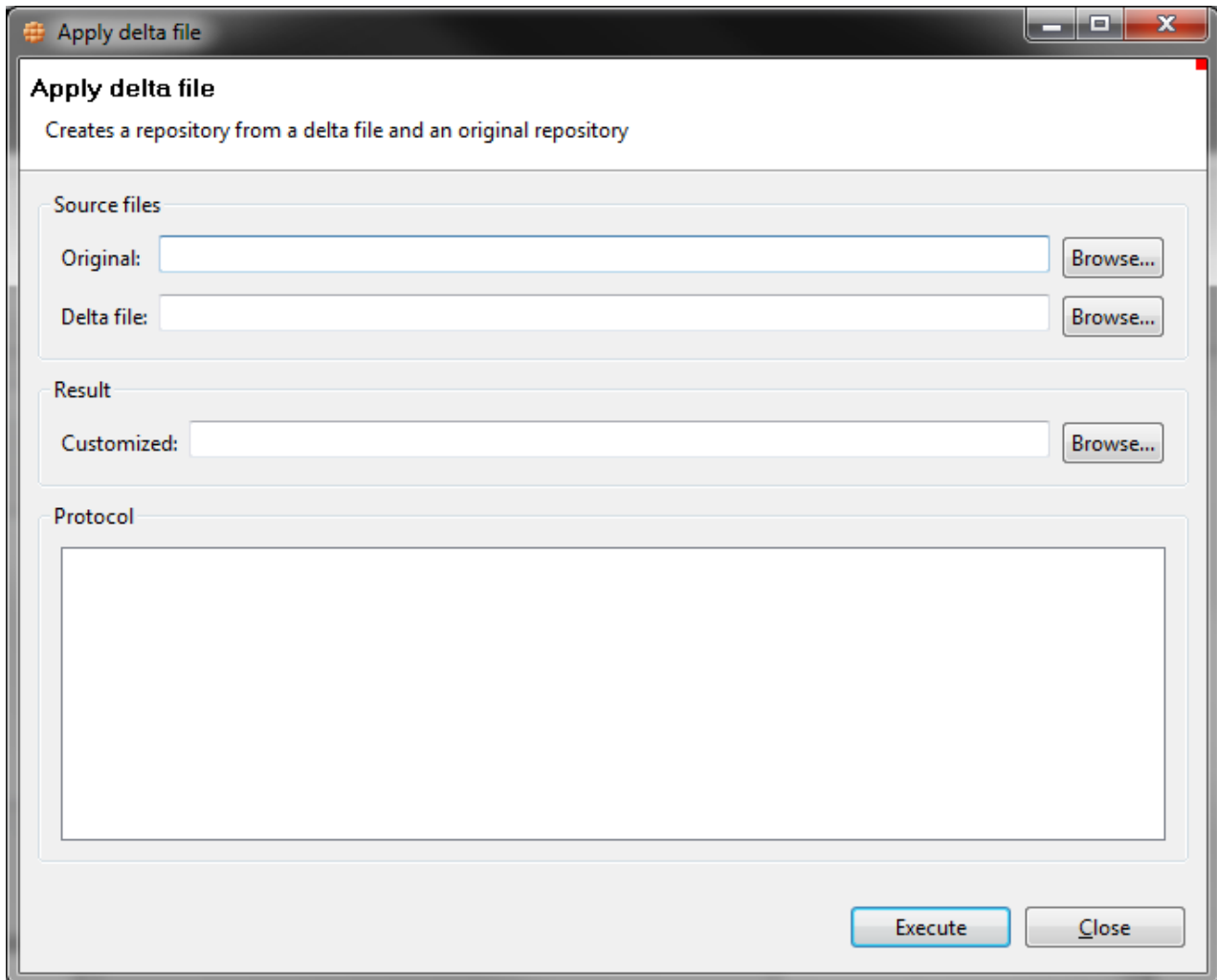
**Result**

Delta file:

**Protocol**

Dialog to create the delta file

Once you have the delta file, you can use the merger from the "target" Product Manager release to re-integrate the delta into the target's standard repository file.



Screen to apply previously created delta file

The result of the integration will be a CSV protocol file which can easily viewed in Excel or something similar. The protocol will only contain conflicts like "repository object not found" or "attribute value has been changed by customizing and standard". The entries in the protocol need to be checked manually, they can't be resolved automatically.

### 3.1.1 Installation and Execution

The repository merger is being distributed together with the Repository Editor (starting from version 6.0.01, from version 7 it is called Repository Manager). You can execute it using the provided UI Dialog in the Repository Editor.



### 3.1.2 Troubleshooting

Normally the repository merge works just fine - but there are some special situations. These and their solutions are described in the following:

#### 3.1.2.1 Duplicate Elements

The repository contains duplicate elements (same element type, same identifier) which is not allowed in any case

This error occurs, if you have repository elements with the same identifier and the log message looks like this:

Log			
Severity	Type	Element	Message
ERROR	FIELD	EnumEntryLang.LanguageID	The repository contains duplicate elements (same element type, same identifier) which is not allowed in any case.

In former versions of Product 360 (a.k.a. HPM 5 and 6), it was allowed to have duplicate identifiers in the repository. Since version 7 this is not allowed anymore.

In the two pictures below you can see the two repository elements, that have the same field identifier.

```
<field identifier="EnumEntryLang.LanguageID" enum-ref="Enum.Language" field-type-ref="EnumEntryLangType.Language">
  <name>%field.EnumEntryLang.LanguageID.name</name>
  <description>%field.EnumEntryLang.LanguageID.description</description>
  <editable>false</editable>
  <visible>true</visible>
  <display-by-default>true</display-by-default>
  <mergeable>false</mergeable>
  <purpose>1</purpose>
  <exportPurpose>5</exportPurpose>
  <value>7</value>
  <documentation></documentation>
</field>
```

```
<field identifier="EnumEntryLang.LanguageID" enum-ref="Enum.Language.WithLanguageIndependent" field-type-ref="EnumEntryLangType.Language">
  <name>%field.EnumEntryLang.LanguageID.name</name>
  <description>%field.EnumEntryLang.LanguageID.description</description>
  <editable>false</editable>
  <visible>true</visible>
  <display-by-default>true</display-by-default>
  <mergeable>false</mergeable>
  <purpose>1</purpose>
  <exportPurpose>5</exportPurpose>
  <value>-1</value>
  <documentation></documentation>
</field>
```

#### Solution

The solution of this error is quite simple. You have to **make the field identifiers unique** by renaming one of them. Identifiers need to be unique **in all repositories**, that are used for the migration (standard and custom).

That means if the standard repository contains duplicate identifiers too, you have to remove them as well.

*Please note that this issue is more likely to appear in migrations from HPM 5.x / 6.x to any version of Product 360.*


## 4 Database Migration

### 4.1 Media Manager Database Migration

#### 4.1.1 Updating the Product 360 - Media Manager database

##### 4.1.1.1 Automatic database update

If you haven't changed the default settings of the Product 360 Server, the Product 360 Media Manager web application and Product 360 Process Engine will perform the database update automatically. You only have to perform the following database update steps if you've turned that off.

 Automatic database update is only possible if your current version is at least Product 360 version 8.1.

If you're migrating from a version older than 8.1, you have to perform the database update steps as described in chapter 'Legacy versions'.

##### 4.1.1.2 Manual database update


The Manual database update tool can be found in the archive **PIM\_<Version>\_MediaManager.zip** at sub folder **\MAIN\_DVD\setup>manual database updater**.

##### Linux

1. Unzip **MediaManagerDatabaseUpdate-linux-<Version>.zip**.  
Run **configure.sh** once to set the +x flags.
2. Run **updateDatabaseHelp.sh** for parameter details.
3. Run **updateDatabase.sh** to perform the database update.
4. Check the protocol in the **/logs** folder.

##### Windows

1. Unzip **MediaManagerDatabase-win-<Version>.zip**.
2. Run **updateDatabaseHelp.bat** for parameter details.
3. Run **updateDatabase.bat** to perform the database update.
4. Check the protocol in the **\logs** folder.

 Manual database update is only possible if your current version is at least Product 360 version 8.1.

If you're migrating from a version older than 8.1, you have to perform the database update steps as described in chapter 'Legacy versions'.

#### 4.1.1.3 Legacy versions

Legacy versions are all versions below Product 360 - Media Manager 8.1.

To use the Automatic or Manual database update as described above the database needs to be updated to version 8.1 at least.

##### **Migration example starting at version 8.0**

If you are updating from version 8.0, you need to apply version 8.1 first. Find the required steps below.

After you have migrated the Media Manager database to version 8.1 you can use the Automatic or Manual database update to migrate to the latest version.


For updating a Product 360 - Media Manager legacy version database, you require the file **PIM\_<Version>\_MediaManager.zip** from Product 360 version 8.1 or 8.1.1 distribution.


The procedure for updating your Product 360 - Media Manager database is as follows:

1. Uncompress the file **PIM\_<Version>\_MediaManager.zip** on your Windows computer.
2. Navigate to the folder **\Setup\win\update database** of the uncompressed archive.
3. Run the program **HMM\_upd.exe**.
4. Select the file **OPAS-G.ini** from your local Product 360 - Media Manager installation.
5. Click on **Start** to update and extend your database tables.

When the process is complete, you will find a log file with the name **YYMMDD.TXT** – where **YY** represents the year, **MM** the month and **DD** the day – in the update subfolder of your local Product 360 - Media Manager installation.

 Since 8.1.1.01: It is no longer necessary to authenticate with the admin user.

 When you use an Oracle database and you want to update from version 5.4 to 5.5 it's necessary to run the long raw conversion before, otherwise the update program will refuse to commence. The conversion tool is located in the module Administration!


 When you use an Oracle database and you want to update from version 5.4 to 5.5 the update program automatically converts every **varchar2(3)** column to a short integer column. This conversion may take a long time, depending on the structure of your data. Because of that it is recommended that you plan your update window on a date when you can accept a certain downtime of your system. The procedure for updating an existing Product 360 - Media Manager database is as follows:

## 4.2 Product 360 Core Database Migration

All necessary Product 360 - Core Database migration scripts are included in the Product 360 Core Database setup, see Server Database. Normally no additional scripts need to be executed.

Nonetheless there are special situations. These and their solutions are described in the following:



### 4.2.1 Duplicate Article Attribute Names in the repository default language

 This problem only occurs, when migrating from PIM Version < 8.0.03 to PIM Version >= 8.0.03, and if the PIM database originally arises from a PIM Version <= 5.3.

#### 4.2.1.1 Problem/ Situation

In PIM Versions 5.3 and older it was possible, to have duplicate attribute names in the repository default language for the same article in the MASTER and/or SUPPLIER database. In newer PIM versions, this situation does not occur anymore, since it is prevented through business logic. But if the PIM Database arises from an older PIM Version (<=5.3), their might still exist such duplicate attribute names.

With PIM Version 8.0.03 an unique index XAK2\_ArticleAttribute on the new column nameInKeyLanguage together with the deletionTimestamp and the articleRevisionID is introduced in the PIM Database. The creation of this index fails, if duplicate attribute names in the repository language do exist (see above). This situation becomes noticeable through the following issue:

- [for MSSQL] the Database Setup shows one of the following errors for update script PCM\_MASTER\_Upd\_V3.0.5.63 or PCM\_SUPPLIER\_Upd\_V3.0.5.63:
  - Msg 1505, Level 16, State 1, ...  
The CREATE UNIQUE INDEX statement terminated because a duplicate key was found for the object name 'dbo.ArticleAttribute' and the index name 'XAK2\_ArticleAttribute'. The duplicate key value is (attribute1, Dec 31 9999 12:00AM, 1).
  - Msg 50000, Level 18, State 44, ...  
1 duplicate article attribute name(s) in the repository key language (languageID=9) is/are existing!
- [for MSSQL] the Database Setup shows one of the following errors for script 'shared\_PPM4\_Rep\_Copy':
  - Invalid column name 'Origin'.
  -  **In this special situation it is necessary to rollback to the PIM database backup you have made before you started the PIM Core Database Setup, then first of all execute the steps described under "Solution" below and after that proceed with the migration. **
  - This may also occur if the repository.default.language is another than in the old environment. In this case please see solution 2.
- [for Oracle] the Database Setup shows the following error for update script PCM\_MASTER\_Upd\_V3.0.5.68 or PCM\_SUPPLIER\_Upd\_V3.0.5.68:

- `ORA-01452: cannot CREATE UNIQUE INDEX; duplicate keys found`
- In order to proactively find and correct duplicate attribute names, please use the corresponding SQL script in the chapter "[SQL scripts to find and correct duplicate attribute names\(see page 22\)](#)" below.

#### 4.2.1.2 Solution 1

The duplicate article attribute names in the repository language have to be corrected in the PIM Database.

1. Therefor execute the SQL script to find duplicate attribute names (see chapter "[SQL scripts to find and correct duplicate attribute names\(see page 22\)](#)" below) in the MASTER and SUPPLIER database in order to find all duplicate attribute names and the corresponding Item/Variant/Product. Save the results of this SQL script, so you are able to find those Items/Variants/Products later on.
2. Then execute the SQL script to correct duplicate attribute names (see chapter "[SQL scripts to find and correct duplicate attribute names\(see page 22\)](#)" below) in the MASTER and SUPPLIER database in order to automatically rename all these duplicate attribute names to '<attributeName> (<counter>)', e.g. 'Length (2)'
3. Re execute the Database Setup. It shall now run through without the above mentioned error.
4. Correct the attribute names (that were found in step 1 and renamed in step 2) to your needs through PIM Desktop, PIM Web or Rest API.

If you have problems with the SQL scripts, please contact the Informatica Global Customer Support.


#### 4.2.1.3 Solution 2

That kind of issue is also shown if the `repository.default.language` inside the `server.properties` is not set correct.

For example: The old environment was running in `de_DE` and now there is `en_US` set.

Ensure that the `repository.default language` is still the same like in the old environment.

### 4.2.2 ArticleTrading entries without existing Party entry are removed

 This problem only occurs when migrating to a PIM version `>= 8.0.00.00`.

#### 4.2.2.1 Problem/Situation

Since PIM 8.0.00.00 an update script exists, which runs every time the Database Setup is executed and (soft-)deletes all ArticleTrading objects, where the corresponding Party is (soft-)deleted (=not existing).

This update script also deletes **ArticleTrading** objects which meet the following conditions:

The ArticleTrading logical key for **ArticleTradingType.LK.PartyMS** is defined with a not editable default value (aka "fixed value"), for example 1001 **and** in the database MAIN.Party table is no record for this ID


(This can actually only happen in case the repository has been misconfigured. to disable a Party logical key one should always use the ID for the Public party.)

#### Solution

A record in the MAIN.Party table needs to be created which has the ID which was used in the repository, BEFORE the database setup is executed! It's up to the Partner or Customer DBA to create this record with a simple Insert statement. Please make sure that the ID has the needed value from the repository (in our example: 1001)

This issue is also described in the Chapter "Domain Model (Repository)" of the Product 360 Development Guide, and here under "Logical Key".

### 4.2.3 Invalid decimal formatting in Article Attribute Values or Structure Group Attribute Values

 This problem only occurs, when migrating from Product 360 Version prior to 5.1

#### 4.2.3.1 Problem/ Situation

The attribute value is persisted as a string value in the database. In former Product 360 versions there was no datatype validation and no consistent format for decimal values. Values were persisted as they have been typed in. This means, that decimal values could have been entered (and persisted in the database) in different formats, e. g. "1,2" (with "," as decimal separator) or "1.000,23" (with "." as thousands separator) or 1,000.23 (with "," as thousands separator and "." as decimal separator).

In the current Product 360 Version, decimal values are persisted in a language-independent format, with "." as decimal separator and no thousands separator.

Having old values in the database and interpreting them with a current client leads to validation errors in the Product 360 Desktop Client and in worst cases to wrong interpretation of these values.

#### 4.2.3.2 Solution

The correct solution for this problem depends strongly on the individual customer situation. Because of this it is important in the first step to analyze the article attribute values and structure group attribute values in the customers database and then decide, how to migrate the "invalid" values in the database with a sql script.

Here you can find an example sql script for the following situation:

- there are no article attribute values and structure group attribute values with thousands separator (like "1.000,23")
- in all attribute values with comma (like "1,2"), the comma can always be interpreted as decimal separator

Find the migration sql scripts for this situation in the chapter ["SQL scripts to check and migrate invalid decimal formatting in Article Attribute Values\(see page 20\)"](#) below.

If you have problems in creating the correct migration sql scripts for your current customer situation, please contact the support department.

## 4.2.4 Structure Migration

First of all, please read the following knowledge base article, in order to understand the impacts of the new structure paradigm: Structure Types

All structures in the Product 360 System are in the first step migrated automatically through the Core Database setup in the following way:

Structure type before migration (id)	Structure type after migration (id)
Catalog structure (1)	Catalog structure (1)
Standard classification system (2)	Standard classification system (2)
User-defined structure (3)	Output structure (3)
Print structure (4)	Output structure (3)
-	Primary maintenance structure (4)
-	Secondary maintenance structure (5)


After this automatic migration, please refer to the structure migration steps described in chapter [Product 360 Server and Desktop Migration\(see page 27\)](#).

## 4.2.5 While migrating to 8.0.03

The following database model changes have been made with the hotfix 8.0.03:

Table	Change description
<all tables of versionable root entities (e.g. Article, Structure, StructureGroup, ...)>	The columns <code>AcID</code> and <code>AcFlag</code> have been moved from the root table (e.g. Article) to the respective revision table (e.g. ArticleRevision)
ArticleAttribute	The new column <code>NameInKeyLanguage</code> was added.
<all tables with <code>DeletionTimestamp</code> column>	For deleted objects, the <code>DeletionTimestamp</code> is now '9999-12-31 00:00:00' instead of null.
Catalog	The column <code>SupplierID</code> has been moved to the <code>CatalogRevision</code> table.

## 4.2.6 Qualification permission info entries in the log

 This problem can only occur when migrating Product 360 Version < 8.0.05 to Product Version >= 8.0.05 and having GDSN deactivated

With Product 360 version 8.0.05 the "PublicationStatusEntry" entity will be deactivated if GDSN is not enabled in the `application_modules.properties`. If you have maintained any qualified field rights for a user group in a version previous 8.0.05 and you are not having GDSN activated, then you might have a similar entry in the log:

```
INFO | jvm 1 | 2017/07/19 14:06:26 | 14:06:25,993 WARN [Worker 2]
[QualificationPermissionElementImpl]
INFO | jvm 1 | 2017/07/19 14:06:26 | No enum provider has been found for any
logical key with the qualification permission identifier "gdsnMessageType".
INFO | jvm 1 | 2017/07/19 14:06:26 | May be this identifier is not anymore used
for any logical key in the repository
INFO | jvm 1 | 2017/07/19 14:06:26 | or no enumeration is defined for the
logical keys that use this identifier.
INFO | jvm 1 | 2017/07/19 14:06:26 | The permissions for this identifier will be
reset when the permissions are changed the next time.
```

You can check with following SQL statements which user groups are affected:

### Main database - Oracle

```
SELECT * FROM "UserGroup" WHERE "QualiPermModel" LIKE
'%identifier="gdsnMessageType"%';
```

### Main database - MSSQL

```
SELECT * FROM [dbo].[UserGroup] WHERE [QualiPermModel] LIKE
'%identifier="gdsnMessageType"%';
```

### Solution:

There are two ways to solve this problem. The first one is by using the desktop client. As soon as you change any qualified field right for a structure group, the invalid entry "gdsnMessageType" will be removed. The second approach is to change the saved data directly on the database. Please use following SQL statements:



#### Main database - Oracle

```
UPDATE "UserGroup"
SET "QualiPermModel" = REPLACE("QualiPermModel", '<element sys-id="1" sys-type="dp"
sys-
class="com.heiler.ppm.std.core.internal.permission.QualificationPermissionElementImpl
" identifier="gdsnMessageType" key-class="java.lang.String"/>', '')
WHERE ID IN ( SELECT ID FROM "UserGroup" WHERE "QualiPermModel" LIKE
'%identifier="gdsnMessageType" key-class="java.lang.String"/>%');
COMMIT;
```

#### Main database - MSSQL

```
UPDATE [dbo].[UserGroup]
SET [QualiPermModel] = CAST(REPLACE(CAST([QualiPermModel] AS VARCHAR(MAX)), '<element
sys-id="1" sys-type="dp" sys-
class="com.heiler.ppm.std.core.internal.permission.QualificationPermissionElementImpl
" identifier="gdsnMessageType" key-class="java.lang.String"/>', '') AS NTEXT)
WHERE ID IN (SELECT ID FROM [dbo].[UserGroup] WHERE [QualiPermModel] LIKE
'%identifier="gdsnMessageType" key-class="java.lang.String"/>%');
```

## 4.2.7 Multiple Purposes for Import Mappings in the GenericData Table

Import mapping is not supposed to have more than one purpose.

Starting with version 8.1.1.07 there have been improvements with which it is possible to sanitize the purposes column of import mappings as user.

You can have a look at database entries containing a flawed purpose value via following sql statement:

#### Main database - Oracle

```
SELECT "Purposes", *
FROM "GenericData" WHERE "Classifier" = 'ImportProfile' AND "Purposes" LIKE '%;%';
```

#### Main database - MSSQL

```
SELECT [Purposes], *
FROM [dbo].[GenericData] WHERE [Classifier] = 'ImportProfile' AND [Purposes] LIKE '%;
%';
```

If there is more than one purpose in the 'Purposes' column e.g. "HSX;HSX;HSX" or "ManualImport;AutoInbox" this means that the purposes column has bad data.

#### Solution:

##### Automatic Sanitization for multiple same purposes values

In case multiple purposes of an import mapping all purposes have the same value, that kind of bad data is sanitized automatically during server start.

Meaning a Purposes column with "HSX;HSX;HSX" becomes "HSX".

##### Manual Sanitization

In case of multiple different purposes like "ManualImport;AutoInbox" there is no deterministic logic to find out the original desired purpose.

Therefore, this has to be fixed manually which can be done in two ways:

##### Manual Sanitization - Update Script

The first way is doing this by executing a SQL statement to explicitly set the single Purpose. This can already be done in all Product360 versions.

Example for setting Purposes to AutoInbox.

##### Main database - Oracle

```
UPDATE "GenericData"
SET "Purposes" = 'AutoInbox'
WHERE ID = <insert ID of desired import mapping in GenericData table>;
COMMIT;
```

##### Main database - MSSQL

```
UPDATE [dbo].[GenericData]
SET [Purposes] = 'AutoInbox'
WHERE ID = <insert ID of desired import mapping in GenericData table>;
```

Possible values for a purpose are

- 'HSX': for 'Supplier Portal' imports
- 'AutoInbox': for 'Hotfolder' imports
- 'ManualImport': for 'Manual' imports

##### Manual Sanitization - Desktop Client

The second way to do this is done as an administrative action in the Desktop Client. This is now possible with the improvements and also the recommended way to do this.

In the Manage Import Mappings Dialog look for those import mappings that show "Several purposes" in the Purpose dropdown when selecting that mapping.

Then choose the desired purpose and click 'Apply'.

Note that selecting multiple mappings now disables the purposes dropdown as opposed to previous versions. This is to prevent further possible bad multi purpose values.

## 4.2.8 SQL scripts to check and migrate invalid decimal formatting in Article Attribute Values

### for MASTER and SUPPLIER database

```
-- This script can be used after migration of a pim database older than version 5 in
-- order to correct old formatted article attribute values.
--
-- It searches for invalid formatted decimal values in ArticleAttributeValue.Value
-- and ArticleAttributeValue.ValueMax
-- which have only one comma and only digits in it, and replaces the comma with a dot
-- f.e.:
--   - "1,2" becomes "1.2"
--   - "10000,23" becomes "10000.23"
--
-- Limitations:
--   Old values with thousands separator (f.e. "1.000,2") are not migrated. Also
--   values with characters other than digits and "," (f.e. "1,2 mm") are not migrated.
-- Execute the following statements once on the MASTER database and once on the
-- SUPPLIER database.
-- 1. search for invalid values (here you have the ability to recheck, if these
-- values should be migrated)
SELECT ar.Identifier, aav.ID, aav.Value, aav.ValueMax
FROM ArticleRevision ar
INNER JOIN ArticleAttribute aa ON aa.ArticleRevisionID = ar.ID
INNER JOIN ArticleAttributeValue aav ON aa.ID = aav.ArticleAttributeID
WHERE aa.DataTypeID in (2, 3, 7) AND
      (((len(aav.Value) - len(replace(aav.Value, ',', ''))) = 1 AND aav.Value not like
      '%[^0-9,]%' ) OR
      ((len(aav.ValueMax) - len(replace(aav.ValueMax, ',', ''))) = 1 AND aav.ValueMax
      not like '%[^0-9,]%' ))
-- 2. correct invalid values in ArticleAttributeValue.Value
UPDATE ArticleAttributeValue set Value = replace (Value, ',', '.')
WHERE ID in (
SELECT aav.ID
FROM ArticleAttribute aa
INNER JOIN ArticleAttributeValue aav ON aa.ID = aav.ArticleAttributeID
WHERE aa.DataTypeID in (2, 3, 7)
AND (len(aav.Value) - len(replace(aav.Value, ',', ''))) = 1
AND aav.Value not like '%[^0-9,]%'
)
-- 2. correct invalid values in ArticleAttributeValue.ValueMax
UPDATE ArticleAttributeValue set ValueMax = replace (ValueMax, ',', '.')
WHERE ID in (
SELECT aav.ID
FROM ArticleAttribute aa
INNER JOIN ArticleAttributeValue aav ON aa.ID = aav.ArticleAttributeID
WHERE aa.DataTypeID in (2, 3, 7)
```

```
AND (len(aav.ValueMax) - len(replace(aav.ValueMax, ',', ''))) = 1
AND aav.ValueMax not like '%[^0-9,]%'
)
```

#### for MAIN database

```
-- This script can be used after migration of a pim database older than version 5 in
-- order to correct old formatted structure group attribute values.
--
-- It searches for invalid formatted decimal values in
-- StructureGroupAttributeVal.Value and StructureGroupAttributeVal.ValueMax
-- which have only one comma and only digits in it, and replaces the comma with a dot
-- f.e.:
--   - "1,2" becomes "1.2"
--   - "10000,23" becomes "10000.23"
--
-- Limitations:
--   Old values with thousands separator (f.e. "1.000,2") are not migrated. Also
--   values with characters other than digits and "," (f.e. "1,2 mm") are not migrated.
-- Execute the following statements on the MAIN database.
-- 1. search for invalid values (here you have the ability to recheck, if these
-- values should be migrated)
SELECT sgav.Identifier, sgav.ID, sgav.Value, sgav.ValueMax
FROM StructureGroupAttribute sga
INNER JOIN StructureGroupAttributeVal sgav ON sgav.StructureGroupAttributeID = sga.ID
WHERE DataTypeID in (2, 3, 7) AND
      (((len(sgav.Value) - len(replace(sgav.Value, ',', ''))) = 1 AND sgav.Value not
like '%[^0-9,]%' ) OR
      ((len(sgav.ValueMax) - len(replace(sgav.ValueMax, ',', ''))) = 1 AND
sgav.ValueMax not like '%[^0-9,]%' ))
-- 2. correct invalid values in StructureGroupAttributeVal.Value
UPDATE StructureGroupAttributeVal set Value = replace (Value, ',', '.')
WHERE ID in (
SELECT sgav.ID
FROM StructureGroupAttribute sga
INNER JOIN StructureGroupAttributeVal sgav ON sgav.StructureGroupAttributeID = sga.ID
WHERE DataTypeID in (2, 3, 7)
AND (len(sgav.Value) - len(replace(sgav.Value, ',', ''))) = 1
AND sgav.Value not like '%[^0-9,]%'
)
-- 2. correct invalid values in StructureGroupAttributeVal.ValueMax
UPDATE StructureGroupAttributeVal set ValueMax = replace (ValueMax, ',', '.')
WHERE ID in (
SELECT sgav.ID
FROM StructureGroupAttribute sga
INNER JOIN StructureGroupAttributeVal sgav ON sgav.StructureGroupAttributeID = sga.ID
WHERE DataTypeID in (2, 3, 7)
AND (len(sgav.ValueMax) - len(replace(sgav.ValueMax, ',', ''))) = 1
AND sgav.ValueMax not like '%[^0-9,]%'
)
```

## 4.2.9 SQL scripts to find and correct duplicate attribute names

### MSSQL - find duplicate attribute names

```
-- find all duplicate attributes in the key language
DECLARE @KeyLanguageID BIGINT
SET @KeyLanguageID = 9 -- TODO: change to the repository key language ID (9 = en_US,
7 = de_DE)
SELECT AR.[ArticleID] AS [ArticleID], AR.[Identifier] AS [ItemNo], AR.[CatalogID],
AR.EntityID AS [Article.EntityID], AA.[ArticleRevisionID], AAL.ID AS
[ArticleAttributeLangID], AAL.[Name]
FROM [ArticleRevision] AR
INNER JOIN [ArticleAttribute] AA
ON AR.[ID] = AA.[ArticleRevisionID]
AND AR.[DeletionTimestamp] = AA.[DeletionTimestamp]
INNER JOIN [ArticleAttributeLang] AAL
ON AA.[ID] = AAL.[ArticleAttributeID]
AND AA.[DeletionTimestamp] = AAL.[DeletionTimestamp]
INNER JOIN
(
SELECT DISTINCT XAA.[ArticleRevisionID] AS [ArticleID], XAAL.[Name] AS [Name],
MAX(XAA.[ID]) AS [MaxArticleAttributeID]
FROM [ArticleAttributeLang] XAAL
INNER JOIN [ArticleAttribute] XAA
ON XAA.[ID] = XAAL.[ArticleAttributeID]
AND XAA.[DeletionTimestamp] = XAAL.[DeletionTimestamp]
AND XAAL.[LanguageID] = @KeyLanguageID
WHERE XAAL.[LanguageID] = @KeyLanguageID
GROUP BY XAA.[ArticleRevisionID], XAAL.[Name]
HAVING COUNT(*) > 1
) x
ON x.[ArticleID] = AA.[ArticleRevisionID]
AND x.[Name] = AAL.[Name]
AND x.[MaxArticleAttributeID] <> AA.ID
WHERE AAL.[LanguageID] = @KeyLanguageID
ORDER BY AR.[CatalogID], AR.[Identifier], AAL.[Name];
GO
```

### Oracle - find duplicate attribute names

```
-- find all duplicate attributes in the key language
SELECT AR."ArticleID" AS "ArticleID", AR."Identifier" AS "ItemNo", AR."CatalogID",
AR."EntityID" AS "Article.EntityID", AA."ArticleRevisionID", AAL.ID AS
"ArticleAttributeLangID", AAL."Name"
FROM "ArticleRevision" AR
  INNER JOIN "ArticleAttribute" AA
    ON AR."ID" = AA."ArticleRevisionID"
    AND DECODE( AR."DeletionTimestamp", AA."DeletionTimestamp", 1, 0 ) = 1
  INNER JOIN "ArticleAttributeLang" AAL
    ON AA."ID" = AAL."ArticleAttributeID"
    AND DECODE( AA."DeletionTimestamp", AAL."DeletionTimestamp", 1, 0 ) = 1
  INNER JOIN
  (
    SELECT DISTINCT XAA."ArticleRevisionID" AS "ArticleID", XAAL."Name" AS "Name",
MAX(XAA."ID") AS "MaxArticleAttributeID"
    FROM "ArticleAttributeLang" XAAL
      INNER JOIN "ArticleAttribute" XAA
        ON XAA."ID" = XAAL."ArticleAttributeID"
        AND DECODE( XAA."DeletionTimestamp", XAAL."DeletionTimestamp", 1, 0 ) = 1
        AND XAAL."LanguageID" = 9 -- TODO: change to the repository key language (9 =
en_US, 7 = de_DE)
      WHERE XAAL."LanguageID" = 9 -- TODO: change to the repository key language (9 =
en_US, 7 = de_DE)
      GROUP BY XAA."ArticleRevisionID", XAAL."Name"
      HAVING COUNT(*) > 1
  ) x
    ON x."ArticleID" = AA."ArticleRevisionID"
    AND x."Name" = AAL."Name"
    AND x."MaxArticleAttributeID" <> AA.ID
  WHERE AAL."LanguageID" = 9 -- TODO: change to the repository key language (9 =
en_US, 7 = de_DE)
ORDER BY AR."CatalogID", AR."Identifier", AAL."Name"
```

## MSSQL - correct duplicate attribute names

```
-- correct all duplicate attributes in the key language
DECLARE @KeyLanguageID BIGINT
SET @KeyLanguageID = 9 -- TODO: change to the repository key language ID (9 = en_US,
7 = de_DE)
DECLARE REPLACE_CURSOR CURSOR
FOR
    SELECT AAL.ID, AA.ArticleRevisionID
    FROM [ArticleAttribute] AA
        INNER JOIN [ArticleAttributeLang] AAL
            ON AA.[ID] = AAL.[ArticleAttributeID]
            AND AA.[DeletionTimestamp] = AAL.[DeletionTimestamp]
        INNER JOIN
            (
                SELECT DISTINCT XAA.[ArticleRevisionID] AS [ArticleID], XAAL.[Name] AS [Name],
                MAX(XAA.[ID]) AS [MaxArticleAttributeID]
                FROM [ArticleAttributeLang] XAAL
                    INNER JOIN [ArticleAttribute] XAA
                        ON XAA.[ID] = XAAL.[ArticleAttributeID]
                        AND XAA.[DeletionTimestamp] = XAAL.[DeletionTimestamp]
                        AND XAAL.[LanguageID] = @KeyLanguageID
                WHERE XAAL.[LanguageID] = @KeyLanguageID
                GROUP BY XAA.[ArticleRevisionID], XAAL.[Name]
                HAVING COUNT(*) > 1
            ) x
        ON x.[ArticleID] = AA.[ArticleRevisionID]
        AND x.[Name] = AAL.[Name]
        AND x.[MaxArticleAttributeID] <> AA.ID
    WHERE AAL.[LanguageID] = @KeyLanguageID
    ORDER BY AA.ArticleRevisionID;
DECLARE @ID BIGINT, @ArticleRevisionID BIGINT, @LastArticleRevisionID BIGINT,
@Counter INT, @NameAddon NVARCHAR(5), @SQLSCRIPT NVARCHAR(1000);
SET @LastArticleRevisionID = -1
OPEN REPLACE_CURSOR;
FETCH NEXT FROM REPLACE_CURSOR INTO @ID, @ArticleRevisionID
WHILE ( @@FETCH_STATUS = 0 )
    BEGIN
        IF @LastArticleRevisionID <> @ArticleRevisionID
            BEGIN
                SET @LastArticleRevisionID = @ArticleRevisionID
                SET @Counter = 1
            END;
        SET @Counter = @Counter + 1
        SET @NameAddon = ' (' + CAST(@Counter AS NVARCHAR(3)) + ')'
        SET @SQLSCRIPT = 'UPDATE dbo.[ArticleAttributeLang] SET [Name] =
(SUBSTRING([Name], 1, 24) + ''' + @NameAddon + ''') WHERE [ID] = ' + CAST(@ID AS
NVARCHAR(25))
        PRINT @SQLSCRIPT
        EXEC sys.sp_executesql @SQLSCRIPT
        FETCH NEXT FROM REPLACE_CURSOR INTO @ID, @ArticleRevisionID
```



```
END;  
CLOSE REPLACE_CURSOR;  
DEALLOCATE REPLACE_CURSOR;  
GO
```

### Oracle - correct duplicate attribute names

```
-- correct all duplicate attributes in the key language
SET SERVEROUTPUT ON
DECLARE
    KeyLanguageID          NUMBER := 9; -- TODO: change to the repository key language
    (9 = en_US, 7 = de_DE)
    ZID                    NUMBER;
    ArticleRevisionID      NUMBER;
    LastArticleRevisionID  NUMBER := -1;
    Counter                NUMBER;
    NameAddon              NVARCHAR2(5);
    SQLSCRIPT              VARCHAR2(1000);

    CURSOR C1 (vKeyLanguageID IN NUMBER) IS SELECT AAL."ID", AA."ArticleRevisionID"
        FROM "ArticleAttribute" AA
        INNER JOIN "ArticleAttributeLang" AAL
        ON AA."ID" = AAL."ArticleAttributeID"
        AND DECODE( AA."DeletionTimestamp",
AAL."DeletionTimestamp", 1, 0 ) = 1
        INNER JOIN
        (
            SELECT DISTINCT XAA."ArticleRevisionID"
            AS "ArticleID", XAAL."Name" AS "Name", MAX(XAA."ID") AS "MaxArticleAttributeID"
            FROM "ArticleAttributeLang" XAAL
            INNER JOIN "ArticleAttribute" XAA
            ON XAA."ID" =
XAAL."ArticleAttributeID"
            AND DECODE( XAA."DeletionTimestamp",
XAAL."DeletionTimestamp", 1, 0 ) = 1
            AND XAAL."LanguageID" =
vKeyLanguageID
            WHERE XAAL."LanguageID" =
vKeyLanguageID
            GROUP BY XAA."ArticleRevisionID",
XAAL."Name"
            HAVING COUNT(*) > 1
        ) x
        ON x."ArticleID" =
AA."ArticleRevisionID"
        AND x."Name" = AAL."Name"
        AND x."MaxArticleAttributeID" <> AA.ID
        WHERE AAL."LanguageID" = vKeyLanguageID
        ORDER BY AA."ArticleRevisionID";

BEGIN
    OPEN C1(KeyLanguageID);
    LOOP
        FETCH C1 INTO ZID, ArticleRevisionID;
        EXIT WHEN C1%NOTFOUND;

        BEGIN
```

```

IF (LastArticleRevisionID <> ArticleRevisionID) THEN
  BEGIN
    LastArticleRevisionID := ArticleRevisionID;
    Counter := 1;
  END;
END IF;

Counter := Counter + 1;
NameAddon := ' (' || Counter || ')';

SQLSCRIPT := 'UPDATE "ArticleAttributeLang" SET "Name" = (SUBSTR("Name", 1,
244) || ''' || NameAddon || ''') WHERE "ID" = ' || ZID;
DBMS_OUTPUT.PUT_LINE(SQLSCRIPT);
EXECUTE IMMEDIATE SQLSCRIPT;
END;
END LOOP;
CLOSE C1;
END;
/

```

## 5 Server and Desktop Migration


### 5.1 Installation of Hotfix, EBF or One-Off-EBF (=Patch) on Server

**i** EBF packages and One-Off-EBF (=Patch) packages contain only delta plugins which have been changed. In order to recognize such packages, the suffix **delta** is contained in its name. Hotfixes and Major releases have the suffix **full** in its name.

Generally this is an easy process. Please consider the following steps in order to migrate your PIM to a newer version.

- Consider pre-migration checklists
- Before applying an Hotfix or Major update, you have to update the Control Center itself. See chapter "Control Center Migration" of this Migration guide.
- Login into the Control Center
- Navigate to the installation tab
- With the upload functionality, upload all packages you want to deploy. Do not rename or extract any of the packages that are delivered.
- If you want to deploy also accelerator plugins or custom plugins, you can also upload them. Custom plugins have to be uploaded separately as jar file.
- Please note that all uploaded files will be deployed and overwrite the existing files. But all files will be backedup in a folder, so you can revert them if something went wrong.
- By clicking the upload button, the update process will start and all configured servers will be updated. All configuration files on all servers will be also replaced with the ones that are located in the Control Center.
- Execute steps in chapter "Product 360 Core Database Migration" by running setup to your existing database connection in order to update existing database.

Generally, you should run your custom JUnit tests in order to ensure that your custom functionality is still working.

 If you want to encrypt the passwords using standard implementation please refer to chapter Encryption of secure information in the Server Installation manual. Updating to newest Hotfix you should replace the Java JCE policy files in `jre\lib\security` folders of all Product 360 components with recommended ones. If you install an EBF, then this step is not needed.

## 5.2 Installation of Hotfix, EBF or One-Off-EBF (=Patch) on Desktop

- Before updating the Desktop you should execute the migration of the Server first.
- For **Hotfix** or **Major** updates:
  - Remove content of PIM Desktop Installation Folder and unzip the content of the new release package
  - Update the configuration files if you have any customizings here. Instead of copying the whole file, rather copy only your changes, since there might be some new settings which then will be overwritten.
- For **EBF's** and One-Off-EBF's (=patches):
  - Remove the existing plugins with old revision and then place the plugin with new revision which are part of the EBF resp. One-Off-EBF.

## 5.3 Structure Migration

First of all, please read the following knowledge base article, in order to understand the impacts of the new structure paradigm: Structure Types

The Structure Migration Tool is part of the standard Product 360 8.0 Installation, so you do not have to install anything separately. With the Structure Migration Tool you can migrate structures to maintenance structures. Therefore this tool first of all validates the given structures in means of the new structure paradigm. The validation (and also correction of invalid items) works the same as the Structure Migration Preparation Tool (see [Product 360 Server and Desktop Pre-Migration Checklist](#)(see page 38)). If the structures are valid they are migrated to maintenance structures.

### Versioning

If you are using the Versioning functionality of Product 360, please be aware, that the selected structures will be migrated to maintenance structures only in the working version. All other versions affected by these structures will be closed, because possibly invalid object mappings in versions other than the working version are not correctable.

The following permissions are needed, when using the Structure Migration Tool:

- Interface visibility "Structure systems"
- For validation of structures:
  - Action right "Create item/variant/product assortments" (depending on which data types are used in the Product 360 system)
  - Action right "Delete item/variant/product assortments" (depending on which data types are used in the Product 360 system)
- For migration of structures:

- Action right "Edit structure systems"
- Action right "Close versions" (only when using versioning)

### 5.3.1 Run Structure Migration Tool

- Open the view "Structure systems".
- Select all the structures you want to migrate to maintenance structures.
- Right-click and choose "Migrate structures..."
- You have the possibility, to recheck your selection and probably correct it.
- Then click on "Migrate".
- The migration process is scheduled as a background server job.
- You can view the results of this job in the "Process overview" perspective. Here choose the job category ["Data maintenance"](#), the job type "Migrate structures" and then your currently scheduled job.
- In the "Log" view you can see the results of the migration job.
- If all the selected structures can be migrated to maintenance structures, you see a corresponding info and they are migrated.
- If the selected structures can not be migrated, you see a corresponding warning. Please then continue with the steps mentioned in chapter "Correct invalid item/variant/product mappings" on this page: [Product 360 Server and Desktop Pre-Migration Checklist](#)(see page 27)
- After correction of all invalid object mapping, restart the Structure Migration Tool for the same structures as before, by executing again the steps mentioned above.

## 5.4 XML parsing exception while loading export templates or other data graph objects

If you experience unexpected errors during the serialization or deserialization of data graphs after an upgrade or change of your local environment, the following hints might be helpful.

In the past we had problems with the SDK deployment mechanism, specifically with the target platform refreshing. When you define the target platform in the SDK by using the update-site, eclipse is copying all bundles of the update site to an internal cache directory.

Now, when you switch to a different target platform release, eclipse is updating it's internal cache **only** in case the bundles have changed. The fact that a bundle has changed is determined by the bundle name and the version inkl. build identifier suffix of the version. If none of these changed, the bundle is **not** refreshed in the internal cache which leads to `ClassNotFoundException` as soon as someone wants to access a new class of that bundle.

In most of our cases there is no issue with this since nearly all bundles have a build identifier which changes with every build.

Most of them, but not all. The `com.heiler.ppm.xml` bundle has no build identifier. This bundle contains the XML parser implementations which must be used with Product 360 - those parsers are also used within the Java SDK since we're using the so called "endorsed-directory" mechanism. During launch (either by cmd, by service or by launch config of the SDK) we provide the full path to the `com.heiler.ppm.xml/endorsed` directory to java.

So, in case the directory is not there, or the path is not correct in the launch scenarios, the default XML parser will be used which does currently not work with the used XML serialization of the EMF based data graph objects.

Strange XML parsing exceptions are the symptoms of this problem.

With Product 360 version 7.0.04 we increased the version number of the com.heiler.ppm.xml bundle from 5 to 6. So you need to make sure you also update all configuration files (incl. wrapper.conf) as well as your launch configs - unless you will also experience those exceptions.

A typical stacktrace of such exception can look like this:

```

Caused by: org.eclipse.emf.ecore.xmi.IllegalValueException: Value ';'is not legal.
(file:///D:/informatica/pim/client/all.datagraph,7, 21)
at org.eclipse.emf.ecore.xmi.impl.XMLHandler.setFeatureValue(XMLHandler.java:2648)
at org.eclipse.emf.ecore.xmi.impl.XMLHandler.setAttribValue(XMLHandler.java:2702)
at
org.eclipse.emf.ecore.xmi.impl.SAXXMLHandler.handleObjectAttribs(SAXXMLHandler.java:
83)
at org.eclipse.emf.ecore.xmi.impl.XMLHandler.createObjectFromFactory(XMLHandler.java:
2178)
at
org.eclipse.emf.ecore.sdo.util.DataGraphResourceFactoryImpl$DataGraphResourceImpl$Loa
dImpl$1.createObjectFromFactory(DataGraphResourceFactoryImpl.java:670)
at org.eclipse.emf.ecore.xmi.impl.XMLHandler.createObjectByType(XMLHandler.java:1316)
at
org.eclipse.emf.ecore.sdo.util.DataGraphResourceFactoryImpl$DataGraphResourceImpl$Loa
dImpl$1.handleFeature(DataGraphResourceFactoryImpl.java:554)
at org.eclipse.emf.ecore.xmi.impl.XMLHandler.processElement(XMLHandler.java:1023)
at org.eclipse.emf.ecore.xmi.impl.XMLHandler.startElement(XMLHandler.java:1001)
at org.eclipse.emf.ecore.xmi.impl.XMLHandler.startElement(XMLHandler.java:712)
at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.startElement(Unknown
Source)
at
com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanStartEleme
nt(Unknown Source)
at
com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl$FragmentConten
tDriver.next(Unknown Source)
at com.sun.org.apache.xerces.internal.impl.XMLDocumentScannerImpl.next(Unknown
Source)
at
com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanDocument(U
nknown Source)
at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(Unknown
Source)
at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(Unknown
Source)
at com.sun.org.apache.xerces.internal.parsers.XMLParser.parse(Unknown Source)
at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.parse(Unknown Source)
at com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser.parse(Unknown
Source)
at com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl.parse(Unknown Source)
at org.eclipse.emf.ecore.xmi.impl.XMLLoadImpl.load(XMLLoadImpl.java:181)
... 70 more

```

```

Caused by: java.lang.NumberFormatException: For input string:";i"
at java.lang.NumberFormatException.forInputString(Unknown Source)
at java.lang.Long.parseLong(Unknown Source)
at java.lang.Long.valueOf(Unknown Source)
at
org.eclipse.emf.ecore.impl.EcoreFactoryImpl.createELongObjectFromString(EcoreFactoryI
mpl.java:958)

```

```

at
org.eclipse.emf.ecore.impl.EcoreFactoryImpl.createFromString(EcoreFactoryImpl.java:
157)
at org.eclipse.emf.ecore.xmi.impl.XMLHelperImpl.createFromString(XMLHelperImpl.java:
1613)
at org.eclipse.emf.ecore.xmi.impl.XMLHelperImpl.setValue(XMLHelperImpl.java:1154)
at org.eclipse.emf.ecore.xmi.impl.XMLHandler.setFeatureValue(XMLHandler.java:2643)

```

## 5.5 Entity IDs for ArticleLangType and ArticleExtensionType repository custom entities

With Product 360 7.1.04.00 a new data base column "EntityID" has been introduced for data base tables "ArticleLang" and "ArticleExtension". The content of this column has to be the same as in the corresponding repository logical key for custom entities based on "ArticleLangType" and "ArticleExtensionType" repository entity types.

The "ArticleExtensionType" based standard repository entities cannot be migrated automatically because it's not possible to determine the entity of the data base entries. Those entities got a valid Entity ID but use "0" as corresponding logical key value.

### 5.5.1 Database

All "ArticleLangType" based standard and custom repository entities have been considered in corresponding data base update scripts.

If you have to migrate own custom entities based on "ArticleExtensionType" repository entity type, you have to migrate the corresponding data base entries manually. Please use the following sql scripts.



## MSSQL - Migrate ArticleExtension entries

```
-- 1st step -----
-- are there entries to be migrated?
SELECT COUNT(ID) AS 'Count of entries to be migrated' FROM ArticleExtension WHERE
    EntityID = 0

-- 2nd step -----
-- migrate
DECLARE @ArticleEntityID          varchar(10)
DECLARE @ArticleExtensionEntityID varchar(10)
/* TODO set appropriate value */
SET @ArticleEntityID              = <entity id used for your ArticleType based custom
entity> -- example: 1000 for "Article" repository entity
/* TODO set appropriate value */
SET @ArticleExtensionEntityID = <entity id used for your ArticleExtensionType based
custom entity> -- example: 1050 for "ArticleExtension.EANUCC" repository entity
EXEC ('UPDATE ae SET EntityID=' + @ArticleExtensionEntityID +
    ' FROM ArticleExtension AS ae ' +
    ' INNER JOIN ArticleRevision AS ar ON ae.ArticleRevisionID = ar.ID ' +
    ' WHERE ar.EntityID = ' + @ArticleEntityID +

    /* only not converted values */
    ' AND ae.EntityID = 0' +

    /* TODO set specific values for logical key columns */
    ' AND ae.ExtensionType IN (''Extension1_a'', ''Extension1_b'')' +
    ' AND ae.BuyerID in (1, 2, 3)' +
    ' AND ae.LanguageID < 0' +
    ' AND ae.Territory IN (''WORLD'', ''US'')')
```

### ORACLE - Migrate ArticleExtension entries

```
-- 1st step -----
-- are there entries to be migrated?
SELECT COUNT(ID) AS "Entries to be migrated" FROM "ArticleExtension" WHERE "EntityID"
= 0;

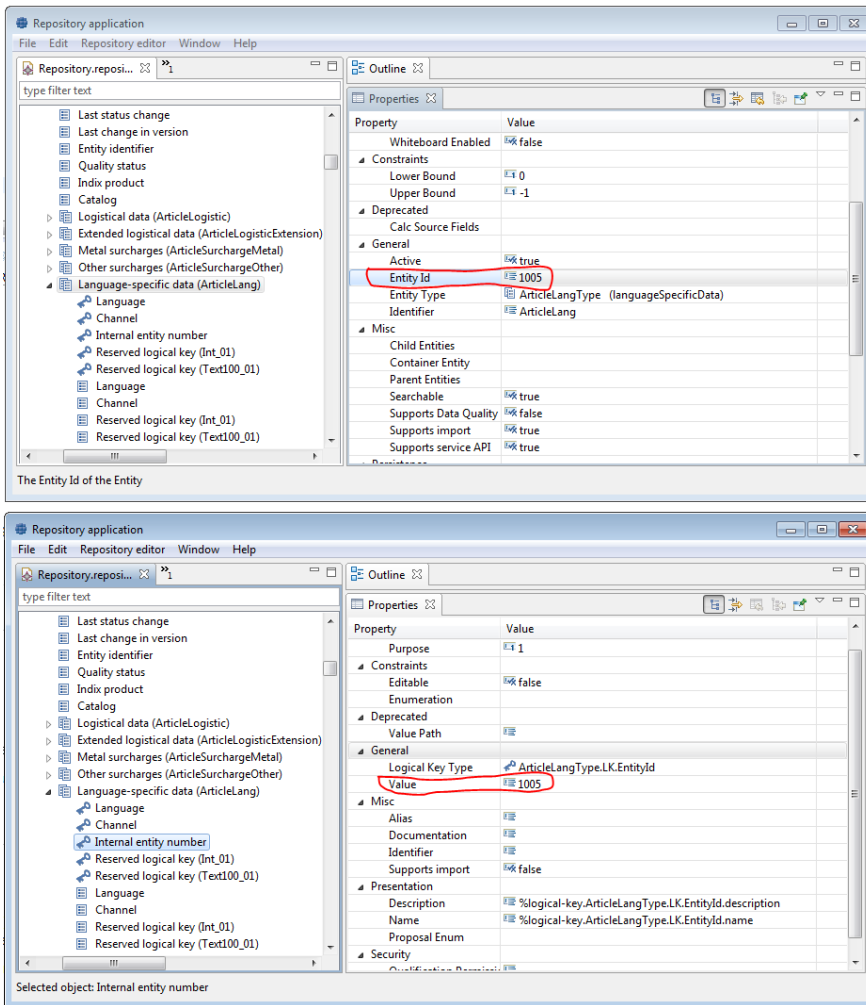
-- 2nd step -----
-- migrate
DECLARE
  /* TODO set appropriate value */
  ArticleEntityID          NUMBER          :=<entity id used for your ArticleType based
custom entity>;          -- example: 1000 for "Article" repository entity
  /* TODO set appropriate value */
  ArticleExtensionEntityID NUMBER          :=<entity id used for your
ArticleExtensionType based custom entity>; -- example: 1050 for
"ArticleExtension.EANUCC" repository entity

BEGIN
  EXECUTE IMMEDIATE
    'UPDATE (SELECT "ArticleExtension"."EntityID"
      FROM "ArticleExtension"
      INNER JOIN "ArticleRevision"
        ON "ArticleExtension"."ArticleRevisionID" = "ArticleRevision"."ID"
      WHERE "ArticleRevision"."EntityID" = ' || ArticleEntityID ||
      /* only not converted values */
      ' AND "ArticleExtension"."EntityID" = 0 ' ||
      /* TODO set specific values for logical key columns */
      ' AND "ArticleExtension"."ExtensionType" IN ('Extension1_a',
'Extension1_b'))
      AND "ArticleExtension"."BuyerID" in (1, 2, 3)
      AND "ArticleExtension"."LanguageID" < 0
      AND "ArticleExtension"."Territory" IN ('WORLD', 'US') )
    SET "EntityID" = ' || ArticleExtensionEntityID;

END;
```

## 5.5.2 Repository

Ensure all affected custom entities have the corresponding logical key with an appropriate value.



## 5.6 Server Jobs

With version 8.0.5 the preference for defining the number of threads for server jobs in the `quartz.properties` has become obsolete.

If changes have been made to this preference, the number should be set to the new default, which is 3.

```
org.quartz.threadPool.threadCount=3
```

The maximum number of running jobs can now be configured in the `plugin_customization.ini` with the preference:

```
com.heiler.ppm.job.server/maxRunningServerJobs = 40
```

## 5.7 Data Quality

### 5.7.1 Migration

To migrate dq rules, configurations and dictionaries from one system to another,

1. Make a backup of the dataquality shared folder in the target system before migrating the new packages.
2. ensure to copy the shared server folder 'dataquality' to the target system.
3. In order to also have a synced version between custom dictionaries in the database and the files in the dataquality/dictionaries folder, zip the complete dataquality/dictionaries folder.  
Or make a zip file containing only the custom dictionaries (latter is recommended and should be also the way customers/partners build their custom packages).  
This ensures that not only dictionaries from OOB standard are considered, but also customer dictionaries that are used for custom rules mapplets.

### 5.7.2 Update

For a full update, updating of the rule mapplet files, the dictionaries and in case of GDSN predefined rule configurations is needed.

Please refer to the appropriate sections in the New & Noteworthy and release notes for further specific details of a dataquality update.

If there is no specific information in the New & Noteworthy, it is recommended to do a full update using standard steps and default settings during dictionary update described as followed

1. **Make a backup** of the dataquality shared folder before updating the new packages.
2. **Update Rule mapplets:**  
Following rule files must be updated: 'Informatica\_PIM\_Content.xml' and depending of whether GDSN accelerator is used also 'Informatica\_PIM\_GDSN.xml'. If both files are needed these are part of the GDSN accelerator package. There is also a standalone version of the standard rule mapplet file 'Informatica\_PIM\_Content.xml' file in the dqRuleset zip artefact.  
For an update make sure that these the two rule files are deployed into the shared server folder 'dataquality/rules'.  
Do this by either  
- uploading the new version to the server via upload mapplet feature in the 'Data Quality' Perspective of the Desktop Client (strongly recommended).  
- copying the new version to that folder prior to server start
3. **Update rule configurations:**  
There are only OOB predelivered rule configurations for GDSN Accelerator package. The file 'StandardDataQualityMappingProfile.xml' is located in the GDSN accelerator package.  
For an update make sure to copy that file to the shared server folder 'dataquality/config' prior to server start. Overwrite the file if already existing.
4. **Update dictionaries:**  
Following dictionaries must be updated: Dictionary information contained in 'Informatica\_PIM\_Content.zip' and depending of whether GDSN accelerator is used also 'Informatica\_PIM\_GDSN.zip'. If both files are needed these are part of the GDSN accelerator package. There is also a standalone version of the 'Informatica\_PIM\_Content.zip' file.  
Customer dictionaries are updated to the target system by providing a zip file containing the complete content (see migration steps above) or only the custom dictionary files (latter is recommended).

## 5.8 Import mappings

By introducing subentity deletion on import new checks were put in place in order to ensure import results are correct and consistent.

This requires existing import mappings to be validated again manually, because the error tolerance is slimmer than before.

### **Import mapping has to be valid**

**If an import mapping is invalid, e.g. due to a non-matching repository, the import will always fail. This is also true for already existing import mappings. In order to fix this, the import mapping has to be loaded once into the Product 360 Desktop Client's "Import" perspective and to be saved again. By doing this, the invalid mappings are automatically removed.**

### **Affected versions**

This step has to be performed whenever the previous release version was lower than 8.2 and the new version is equal or greater than 8.2.

Upgrades from versions greater than 8.2 to any newer version are not affected by this.

## 5.9 Migration of Subentity Deletion customizing

For customers with an existing customization for the "Subentity deletion on import" feature it is required to remove the customization from their Product 360 System and migrate the Subentity Deletion Configuration to this new standard feature, via the Import Perspective in the Product 360 Desktop Client. For more details, see subchapter "Subentity deletion on import" in the Knowledge Base chapter of the Technical Documentation.

### **Affected versions**

This step has to be performed whenever the previous release version was lower than 8.2 and the new version is equal or greater than 8.2.

Upgrades from versions greater than 8.2 to any newer version are not affected by this.

## 5.10 Media Manager Integration

### 5.10.1 Adjust Media Manager database connection string (MSSQL only)

Since 10.0.0.01 a different database connection string is required to connect to the Media Manager database. The database connection is defined in the `hmm.properties` file.

The new connection string pattern looks like

#### hmm.properties

```
...
hmm.db.url=jdbc:sqlserver://localhost:1433;databaseName=opasdb
...
```

Previous versions used the JTDS database driver pattern. JTDS has been replaced by Microsoft's JDBC driver.

## 5.11 Server and Desktop Pre-Migration Checklist

### 5.11.1 Structure Migration Preparation

First of all, please read the following knowledge base article, in order to understand the impacts of the new structure paradigm: Structure Types

It is important, that the following steps are executed **before** the migration to Version 8.0 of the Product 360 system itself. The reason for this is, that with Product 360 8.0 former existing structures are handled as output structures by default, and output structures in Product 360 8.0 do not have the same characteristics and behaviour as structures in former Product 360 Versions (especially concerning the inheritance of structure features to items/variants/products, see also the linked article about the new structure paradigm above).


With the Structure Migration Preparation Tool you can validate structures to be migrated to maintenance structures in Product 360 8.0. Therefor this tool generates assortments for invalid item/variant/product mappings. In the next step any user can use these assortments for correcting the invalid object mappings (also on those clients without having the Structure Migration Preparation Tool installed).

#### 5.11.1.1 Download Structure Migration Preparation Tool

Structure Migration Preparation Tool is available for the newest hotfix release of Product 360 7.1. Find the download packages on the corresponding Hotfix wiki pages. The following artifacts are needed:

- *HPM\_x.x.xx.xx\_Rev-xxxxx\_server\_migration\_structure.zip*
- *HPM\_x.x.xx.xx\_Rev-xxxxx\_client\_migration\_structure.zip*

**i** Please be aware that this tool was tested and released only for the newest hotfix release of Product 360 7.1. You can use this tool also with older Product 360 hotfix releases 7.0 and 7.1, it should be downwards compatible for these major versions. If you run into problems thereby, please contact Product 360 Support. Please also contact Product 360 Support, if you need this tool for Product 360 5.3 or Product 360 6.0. Thanks.

 For custom ArticleType-based entities please ensure, that there is also a corresponding AssortmentType-based entity existing.

#### 5.11.1.2 Install Structure Migration Preparation tool

The Structure Migration Preparation Tool consists of a Product 360 server feature and a Product 360 Desktop client feature. It is a tool for administrative purpose only. Because of that it is recommended to install the client feature only on those Product 360 client(s), which are used by the users performing the structure migration itself.

- The Product 360 server feature can simply be installed by copying content of the *HPM\_x.x.xx.xx\_Rev-xxxx\_server\_migration\_structure.zip* package into Product 360 Server folder (into features and plugins folders respectively). After this you have to restart the Product 360 Server.
- The Product 360 client feature can simply be installed by copying content of the *HPM\_x.x.xx.xx\_Rev-xxxx\_client\_migration\_structure.zip* package into Product 360 Desktop Client folder (into features and plugins folders respectively). After this you have to restart the Product 360 Desktop Client.

The following permissions are needed, when using the Structure Migration Preparation tool:

- Interface visibility "Structure systems"
- Action right "Create item/variant/product assortments" (depending on which data types are used in the Product 360 system)
- Action right "Delete item/variant/product assortments" (depending on which data types are used in the Product 360 system)

#### 5.11.1.3 Run Structure Migration Preparation Tool

- Open the view "Structure systems".
- Select all the structures you want to migrate to maintenance structures.
- Right-click and choose "Validate for migration"
- The validation process is scheduled as a background server job.
- You can view the results of this job in the "Process overview" perspective. Here choose the job category ["Data maintenance"](#), the job type "Migrate structures" and then your currently scheduled job.
- In the "Log" view you can see the results of the validation job.
- If all the selected structures can be migrated to maintenance structures, you see a corresponding info and everything is fine. You can then proceed with the migration of the Product 360 System.
- If the selected structures can not be migrated, you see a corresponding warning. Please then continue with the following step:

#### 5.11.1.4 Correct invalid item/variant/product mappings

If objects (items/variants/products) are existing, which have invalid mappings to structure groups and structure group features in the selected structures (see last step above), several assortments with these objects are created.

The assortment name has the following format:

**<catalog\_label>\_<object\_type>\_<cause>**

For each supplier catalog and the master catalog separate assortments are created. Also for each object type (**item**, **variant**, **product**) separate assortments are created.

And at last for each of the following causes (for invalid mappings) separate assortments are created:

- **attributemapping**: objects, that have attribute(s) with multiple mappings to structure group features within the given structures.
- **nodelevel**: objects, that are classified to node-level structure groups of the given structures.
- **multiclassified**: objects, that are multi-classified to one of the given structures.

All assortments are created in an assortment category named "\_\_MIGRATION\_STRUCTURE\_VALIDATION" and are visible for all users (also on those clients without having the Structure Migration Preparation Tool installed).

**Now, the task is, to go through each assortment, and correct the invalid mappings.**


### Versioning

Structures are only validated in the working version, because the migration is only done in the working version. This is because possibly invalid object mappings in a version other than the working versions are not correctable. All affected versions (other than the working version) will later be closed by the Structure Migration Tool (see also [Product 360 Server and Desktop Migration](#)(see [page 27](#))).

You can recheck the structures as often as you want by simply running the Structure Migration Preparation Tool again with the same structures (see last chapter above) until eventually you get a positive result of the validation job. But please be aware, that all existing assortments in the category "\_\_MIGRATION\_STRUCTURE\_VALIDATION" will be automatically deleted, when executing the "Validate structures for migration" action again.

## 6 Unit Management

With version 10.1 a major refactoring of the Unit Management has been implemented which provides several advantages.

- 

Note that in order to realize this feature the release comes with integrated update scripts that will adjust your existing units in the system.  
**It is strongly recommended to perform a data backup before and revise all units maintained in the system after the upgrade for consistency.**

### 6.1 Overview

The data model entity of type unit has been refactored to allow for export, import and API based access from now on.

The following list describes the main enhancements in more detail:

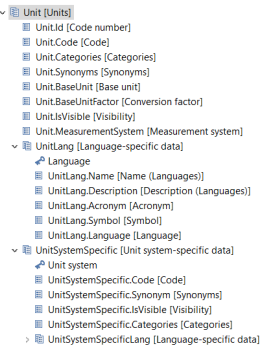
- A mandatory, globally unique code has been introduced which also acts as the external identifier of a unit. Migration scripts will use the so-called "System Units" unit system's codes for this. In case duplicates are



recognized during migration, they are named accordingly.

- All unit system-specific fields are renamed to "Alternative ..." (e.g. "Alternative Code", "Alternative Symbol", etc.)
- Unit fields have been adjusted and correct categories are assigned to the fields
- The generic import has been activated for the unit entity
- The generic List API has been activated, hence units can now also be created and adjusted via the Service API
- Export data providers have been adjusted to support the unit root entity and all its child entities

## 6.2 Repository

	<h3>New Fields</h3> <ul style="list-style-type: none"> <li>• Unit.Code which is now the primary, unique external identifier of the unit, independent of any unit system</li> <li>• Unit.Categories is the field for the categorization of the unit, independent of any unit system</li> <li>• Unit.Synonyms is the field for unit system independent synonyms of a unit</li> </ul> <h3>New Labels</h3> <p>All fields within the UnitSystemSpecific sub-entity are now treated as "alternative". Alternative Code, Alternative Categories etc.</p> <p>Those fields need to be qualified by a unit system, no change here except the label of the fields.</p> <h3>New Categories</h3> <p>All fields have been assigned meaningful categories</p>
------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 6.3 Data Migration

Multiple update scripts will migrate the existing units and create the new fields.

- The new Unit.Code field is initialized with the Code of the unit from the "System Units" unit system
  - In case a Unit had no mapping to the "System Units" unit system (which is actually an invalid state to some sort, see Enum Provider), the code of the Unit System with the smallest Unit System ID is used. A prefix "mi g" and a suffix with the Unit System's identifier is added to the code.
  - In case there were duplicate codes within a unit system (probably due to custom scripts or other situations), the code is prefixed with "mi g\_dup", followed by the code and the database id of the duplicate unit.
- The "System Units" unit system which existed in previous versions is removed

⚠ As the new Unit.Code field is also the unique external identifier, there is no way for the system to know which of the duplicates should be active and kept, the customer and project need to manually open the Unit Maintenance and adjust the code of those units.

## 6.4 Enum Provider

The Unit Enum Provider can be used with a unit system context, or without. The default is without. Such as for fields like Order Unit or others.

Within the GDSN area it is used with the GDSN unit system context. The unit system to use can be provided in the repository as enum parameter.

So the "System Units" have been in the past, is now just "all units", by omitting the UnitSystem parameter. Here an example, no unit system context, but a filter for only the order units:

```
▼ Enum.OrderUnits [Order units]
  Enum.UnitCategory = orderUnits,packagingUnits,scaleUnits,miscellaneous
```

Example of a GDSN specific enum provider configuration. Context set to GDSN (ID = 70), only Units of the Category "Area":

```
▼ Enum.GDSNAreaUnits [GDSN Area units]
  Enum.UnitSystem = 70
  Enum.UnitCategory = Area
```

## 6.5 Desktop UI

The basic maintenance perspective has not changed much, except that the "System Units" unit system is gone now.

## 6.6 Import

Units can now be imported just like every other entity. The Unit.Code needs to be provided as external identifier.

## 6.7 Export

Most export templates using units will be automatically migrated, however, some special cases apply. Please see the Export Compatibility documentation of the Knowledge Base for this.

## 6.8 Service API

The generic ListAPI functionality works now also for units. This includes read and write calls.

# 7 Media Manager Migration

- [Update Media Manager](#)(see page 43)
  - [Updating Media Manager File Server](#)(see page 43)
  - [Updating the database](#)(see page 43)
  - [Updating the client modules](#)(see page 44)
  - [Updating Funcd](#)(see page 44)
  - [Updating web front end](#)(see page 44)
  - [Deactivate old modules](#)(see page 46)

## 7.1 Update Media Manager

### 7.1.1 Updating Media Manager File Server

To update the Product 360 - Media Manager File Server, you require the file **PIM\_<Version>\_MediaManager.zip** from your Product 360 distribution.

1. Uncompress the file **PIM\_<Version>\_MediaManager.zip** on your computer.
2. Navigate to the folder **\Setup\HMM** of the uncompressed archive.
3. Copy the **help%0** and **update** directories to **Volume0** on your file server, to the **opasdata** directory.

### 7.1.2 Updating the database

By default Product 360 Server, the Product 360 Media Manager web application and Product 360 Process Engine will perform the database update automatically.

If you have disabled the automatic database update, you have to migrated your database manually. Please refer to [Database Migration](#)(see page 10) > [Media Manager Database Migration](#)(see page 10) for required update steps.




#### Migration from versions older than 8.1

If you're migrating from a version older than 8.1, please refer to [Database Migration](#)(see page 10) > [Media Manager Database Migration](#)(see page 10) for required update steps as well.

### 7.1.3 Updating the client modules

For hotfix updates client modules are updated automatically on the first start after updating Product 360 - Media Manager. If problems occur the following steps can be performed manually.


 Changed behavior: For major and minor updates (e.g. from 10.0.0.02 to 10.1.0.0.00) the following steps need to be executed manually.

#### 7.1.3.1 Manually updating the client modules (Windows)

1. Switch to the directory **opasdata/update/win/ocln** on your Volume0 (located on Media Manager file server).
2. Run **OPAS\_cln.exe**.
3. Follow the subsequent instructions.


#### 7.1.3.2 Manually updating the client modules (OSX)

1. Switch to the directory **opasdata/update/osx** on your Volume0 (located on Media Manager file server).
2. Mount the image **IMM.install.dmg**.
3. Run the **included package installer**.
4. Follow the subsequent instructions.

 If you are installing new client modules or updating the version, you must re-activate Product 360 - Media Manager. For more details, refer to Activating Product 360 - Media Manager, defining volumes & setting up Funcd.

### 7.1.4 Updating Funcd

To update the Product 360 - Media Manager Funcd, you require the file **PIM\_<Version>\_ThirdPartySoftware.zip** from your Product 360 distribution.

 You should always update the Funcd to get the last versions of the third party tools. To update the Funcd and the third party tools proceed as follows:

1. Uninstall Funcd and the third party tools.
2. Do the installation like described in Installing Funcd.

### 7.1.5 Updating web front end

To update the Product 360 - Media Manager web application, you require the file **PIM\_<Version>\_MediaManager.zip** from your Product 360 distribution.

The procedure for updating your web front end is as follows:


1. Uncompress the file **PIM\_<Version>\_MediaManager.zip** on your computer.
2. Stop the Tomcat server.
3. Make a backup of the web application by renaming e.g. **C:\OpasGWebServer** to **C:\OpasGWebServer\_backup**.
4. Navigate to the folder **\Setup\webapp package\full** of the uncompressed archive.
5. Unpack the **OpasGWebServer.zip** to **C:\**.
6. Copy and overwrite the configuration files from **C:\OpasGWebServer\_backup\XOBSessionManager\conf** to **C:\OpasGWebServer\XOBSessionManager\conf**.
7. Copy and overwrite configuration files from **C:\OpasGWebServer\_backup\Tomcat\conf** to **C:\OpasGWebServer\Tomcat\conf**.
8. Delete **C:\OpasGWebServer\Tomcat\webapps**.
9. Copy **C:\OpasGWebServer\_backup\Tomcat\webapps** to **C:\OpasGWebServer\Tomcat\webapps**.
10. Delete the directory **C:\OpasGWebServer\Tomcat\webapps\opas\WEB-INF\lib**.
11. Delete the directory **C:\OpasGWebServer\Tomcat\webapps\opas\WEB-INF\classes**.
12. Delete the directory **C:\OpasGWebServer\Tomcat\webapps\opas\script**.
13. Navigate to the folder **\Setup\webapp package\update** of the uncompressed archive.
14. Unpack **opas.zip** to **C:\OpasGWebServer\Tomcat\webapps\opas** and replace existing files.
15. When you are updating from version < 10.0.0.01 and using MSSQL database you have to apply the following changes in the file **C:\OpasGWebServer\Tomcat\webapps\opas\Base.cfg** to define the database connection parameters.  
Replace

```
<DATABASE_URL>jdbc:jtds:sqlserver://localhost:1433/opasdb</DATABASE_URL>
```


with


```
<DATABASE_URL>jdbc:sqlserver://localhost:1433;databaseName=opasdb</DATABASE_URL>
```

16. Restart the Tomcat server.

 Product 360 Media Manager web application will update the Media Manager database to the latest version on the first startup.

If you want to prevent the database update it is necessary to set the property  
<DB\_AutomatedUpdateEnabled> to false in the Base.cfg file

 Do copy and not replace the existing opas directory with the archive, because the update package contains only the changed parts of the web front end.

 When changing the layout you have to update the layout profiles and re-assign them to the users. You can use the layout distribution in the "Administration" area in the web front end.

## 7.1.6 Deactivate old modules

The modules: Informatica Media Manager Session Manager, Informatica Media Manager Functd and Informatica Media Manager Internet are no longer needed for versions  $\geq 8.0$ . Therefore these services can be deactivated.

### 7.1.6.1 Deactivate Session Manager

Open the "**Windows Services**" and search the service "Informatica Media Manager Session Manger" (Depending on you preinstalled version, the name can differ for example Heiler Media Manager Session Manager). Stop the service if it is running and afterwards set the Startup type to disabled or manual.

### 7.1.6.2 Deactivate Internet Functd

- If the "Informatica Media Manager Process Watcher" is running as Window service you have to stop this service.
- Open the Process Watcher module.
- Stop the automatic mode by clicking cancel.
- Select **Processwatcher > Processwatcher default settings**
- Disable the monitoring of the Internet functd in the second pane.
- Save and close the application.
- If the "Informatica Media Manager Process Watcher" was running as Window service you have to start this service again.

## 8 Supplier Portal Migration

To update an existing installation of Product 360 Supplier Portal, please run the following steps:

### 8.1 On the Product 360 Supplier Portal Server

- Stop your Product 360 Supplier Portal Windows service "Informatica Product 360 - Supplier Portal"
- Uninstall the old service by running following command in a Windows console ( replace <SERVICE\_NAME> with your Services Service Name, default: **ISP** )

```
<INSTALLATION ROOT>/uninstall.bat <SERVICE_NAME>
```

- Make a backup of your current **<INSTALLATION ROOT>/configuration** and **<INSTALLATION ROOT>/filestorage** directories.
- Unzip the installation package "PIM\_<Version>\_SupplierPortal.zip" into your new NEW\_INSTALLATION ROOT directory.
  - a. After unzipping merge the new **<INSTALLATION ROOT>/configuration/ configuration.properties** with the configuration file backed up previously. See Product 360 Supplier Portal Release Notes for new or updated configuration options that may be added, changed or removed. Database migrations are applied during the first server start, so please ensure your database connection parameters are still valid.

- b. Replace the new **<INSTALLATION ROOT>/configuration/logback.xml** with the one you back up previously.
- c. Replace the new **<INSTALLATION ROOT>/filestorage** directory with the one you back up previously.
- Install the new service by running following command in a Windows console ( replace <SERVICE\_NAME> with your Services Service Name, default: **ISP** )

```
<INSTALLATION ROOT>/install.bat <SERVICE_NAME>
```

Check the windows services control panel and start the service "Informatica Product 360 - Supplier Portal" if not starting automatically. Verify the startup type is set to "Automatic".

### 8.1.1 Adjust database connection string (MSSQL only)

Since 10.0.0.01 a different database connection string is required to connect to the Supplier Portal database. The database connection is defined in the **<INSTALLATION ROOT>/configuration/configuration.properties** file.

Adjust the value for property `database.driverClassName.mssql` with value `com.microsoft.sqlserver.jdbc.SQLServerDriver`


and `database.url.mssql` with value `jdbc:sqlserver://${database.server}:${database.port};databaseName=${database.name}`.

#### configuration.properties

```
...
database.driverClassName.mssql=com.microsoft.sqlserver.jdbc.SQLServerDriver
...
database.url.mssql=jdbc:sqlserver://${database.server}:${
{database.port}};databaseName=${database.name}
...
```

Previous versions used the jTDS database driver pattern. jTDS has been replaced by Microsoft's JDBC driver.

## 9 Web Search Migration

-  The Web Search has undergone complete revamp with Product 360 version 10.0. All indices needs to manually created afresh using export templates and data needs to be synchronized into Elasticsearch.



**Tip**

Go through Configuration Guide Chapter 9 ( Web Search Configuration ) before proceeding.

In-order to help the migration process, there are the following run-books available.

- [Migration of Items of any Catalog index\(see page 48\)](#)
- [Migration of Items of All-Supplier-Catalog index\(see page 55\)](#)
- [Migration of 3PPD index\(see page 57\)](#)
- [Migration of 2PPD index\(see page 60\)](#)

## 9.1 Migration of Items of any Catalog index



Step-by-step guide - How to convert an existing text file based search index configuration into the new export template based search index configuration.

Below example illustrates how to define an export template equivalent to old configuration in order to create a search index.

### 9.1.1 Old Index configuration for field ArticleLang.DescriptionShort

```
field.Article.ArticleLang.DescriptionShort.autocompletable=false
field.Article.ArticleLang.DescriptionShort.facetable=false
field.Article.ArticleLang.DescriptionShort.filterable=true
field.Article.ArticleLang.DescriptionShort.searchable=true
field.Article.ArticleLang.DescriptionShort.sortable=true
field.Article.ArticleLang.DescriptionShort.sourcefield=ArticleLang.DescriptionShort(eng)
field.Article.ArticleLang.DescriptionShort.returnfield=ArticleLang.DescriptionShort(eng)
field.Article.ArticleLang.DescriptionShort.stored=true
field.Article.ArticleLang.DescriptionShort.type=descriptionshort
```

### 9.1.2 New export template based configuration

- Create a new template with purpose as Full-text Search
- Take care of the points defined in Web Search Configuration
- In the configuration of template, define following variables



- *catalog* variable that can uniformly set the catalog for all the data sources

The screenshot shows the 'Export format template properties' dialog box with the 'Items-only' tab selected. The 'Variables' list on the left contains 'catalog', 'language', and 'last modified date'. The 'catalog' variable is selected, and its configuration is shown on the right. The 'Name' field is 'catalog'. The 'Data type' is 'Catalog'. The 'Value' field is empty. The 'Mandatory value' and 'Editable' checkboxes are both checked.

- *last modification date* variable which would be responsible for providing delta changes from the data source.

The screenshot shows the 'Export format template properties' dialog box with the 'Items-only' tab selected. The 'Variables' list on the left contains 'catalog', 'language', and 'last modified date'. The 'last modified date' variable is selected, and its configuration is shown on the right. The 'Name' field is 'last modified date'. The 'Data type' is 'Date and time'. The 'Value' field is '1/1/1900 12:00 AM'. The 'Mandatory value' and 'Editable' checkboxes are both checked.

- *language* variable, used for qualification of logical keys of different entities which are language dependent.

The screenshot shows the 'Export format template properties' dialog box with the 'Items-only' tab selected. The 'Variables' list on the left contains 'catalog', 'language', and 'last modified date'. The 'language' variable is selected, and its configuration is shown on the right. The 'Name' field is 'language'. The 'Data type' is 'Language'. The 'Value' field is empty. The 'Mandatory value' and 'Editable' checkboxes are both checked.

- Define 2 data sources for making sure all the data is picked up
  - Changed and new items* :

- Deleted items* :

- There will always be a module which writes it's output to config-file.json. This module defines the meta data about all the repository based entities and fields that are to be indexed (along with language qualification using the above defined variable) and contains all the other configurations specific to elasticsearch.
- In the config module, define the required field as below:

```
{
  "rootEntities": [
    {
      "identifier": "Article",
      "fields": [],
      "subEntities": [
        {
          "identifier": "ArticleLang",
          "fields": [
            {
              "identifier": "DescriptionShort",
              "dataType": "text",
              "searchProperties": {
```

```

        "searchable": true,
        "sortable": false,
        "facettable": false
      },
      "qualifications": [
        "{%language}"
      ]
    }
  ]
}

```

- The above step just acts as meta data for elastic and it does not do anything for the "data" part. For that user needs to define the data modules which generate data via export using the above defined data sources in advanced configuration of template.
- Since there are 2 data sources available here, user will have to define 2 different modules to utilize both the data sources (New and changed items and Deleted items).
- Users can use the new JSON methods available in the export templates when they define the data fields for exporting. This ensures the formatting required to index data and gives certain level of legibility.
- Associate each of these modules and their sub-modules if any, with export file "data-file.json".
- New and changed items module will be defined as follows :
  - Meta data that defines the operation and identifier for Elastic

```

{?JSONObject
  {?JSONObjectElement "upsert",
    {?JSONObject
      {?JSONStringElement "_id", {?DatasetIdWithContainer}}
    }
  }
}

```

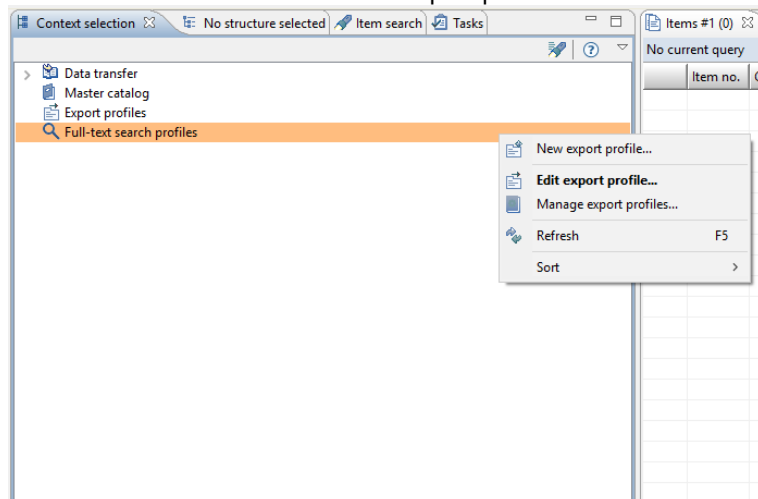
- Then the actual data part that is required to indexed. It is important that there is a strict mapping between the fields defined here and in config. **Fields Acl and Entity are the important ones, without which indexing will fail.**

```
{?JSONObject
  {?JSONStringElement "Acl",{?FormatDecimal {&Item.Object
rights.Internal identifier},,,}},
  {?JSONStringElement "Entity","Article"},
  {?JSONObjectElement "Article",
    {?JSONObject
      {?JSONArray
        {?JSONObject
          {?JSONStringElement "DescriptionShort",{&Item.Short description
(English)}}
        }
      }
    }
  }
}
```

- Deleted items module is defined as follows:

```
{?JSONObject
  {?JSONObjectElement "delete",
    {?JSONObject
      {?JSONStringElement "_id",{?DatasetIdWithContainer}}
    }
  }
}
```

- Now the user needs to create a new export profile :



- Running an export job from here or scheduling it will create an index and upload the data to Elasticsearch server.

### 9.1.2.1 Handling sub-entities in sub-module

- In case of sub-entities that can only be handled inside the sub-module of export template. First a field for the corresponding sub-entity with it's qualification(s) needs to be defined in the config module for elastic:

```
{
  "rootEntities": [
    {
      "identifier": "Article",
      "fields": [],
      "subEntities": [
        {
          "identifier": "ArticleStructureMap",
          "fields": [
            {
              "identifier": "ManualMap",
              "dataType": "text",
              "searchProperties": {
                "searchable": true,
                "sortable": true,
                "facettable": true
              },
              "qualifications": [
                "HeilerStandard"
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

- The New and changed items module will now have following content to accommodate the sub-entity. Notice that ArticleStructureMap is a sub-module which is defined below :

```

{?JSONObject
  {?JSONObjectElement "upsert",
    {?JSONObject
      {?JSONStringElement "_id",{?DatasetIdWithContainer}}
    }
  }
}
{?JSONObject
  {?JSONStringElement "Acl",{?FormatDecimal {&Item.Object rights.Internal
identifier},,,},
  {?JSONStringElement "Entity","Article"},
  {?JSONObjectElement "Article",
    {?JSONObject
      {?JSONObjectElement "ArticleStructureMap",{?JSONArray
{$ArticleStructureMap}}}
    }
  }
}
}

```

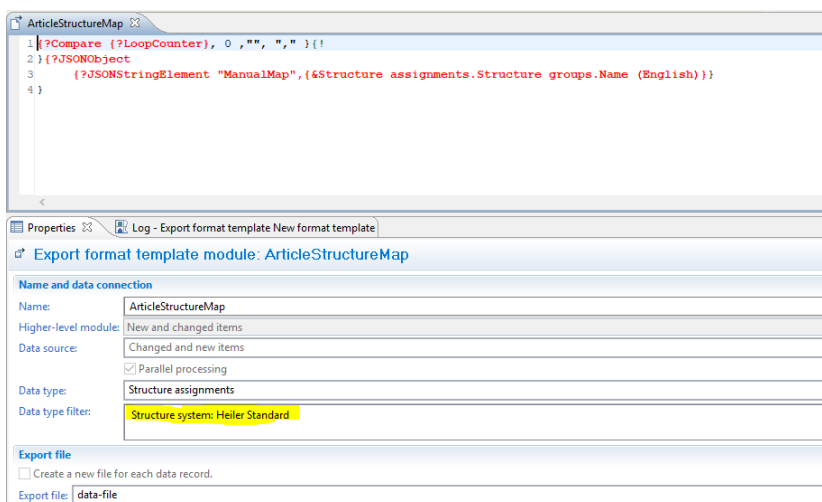
ArticleStructureMap sub-module will have following content (Note that this sub-module is qualified with the qualification provided in the config file):

```

{?Compare {?LoopCounter}, 0 ,"" ,"," }{!
}{?JSONObject
  {?JSONStringElement "ManualMap",{&Structure assignments.Structure groups.Name
(English)}}
}

```

**Take care that every sub-module is qualified as shown below as per the config file.**



## 9.2 Migration of Items of All-Supplier-Catalog index

**i** Step-by-step guide - How to convert an existing text file based search index configuration into the new export template based search index configuration.

Define a template similar to [Migration of Items of any Catalog index](#)(see page 48). The only difference is the data sources and catalog data type.

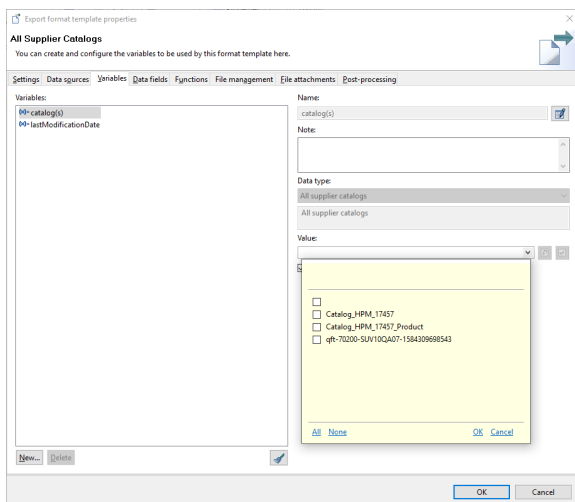
### 9.2.1 Configurations prior to v10.0.0.0, this would be defined as:

*catalogs=Catalog1, Catalog2, Catalog3 and Catalog4*

### 9.2.2 New Configuration

#### 9.2.2.1 Data Type

There is a new Data Type introduced which helps in creating a variable that helps in selection of multiple catalog(s). Data type is called as **All supplier catalogs**.



### 9.2.3 Data Sources

Also there are 2 new data sources available in the system for this purpose:

- **Changed and new items (all supplier catalogs)**

**Export format template properties**

**All Supplier Catalogs**

You can select and configure the data sources to be used by this format template here.

Settings: Data sources | Variables | Data fields | Functions | File management | File attachments | Post-processing

Data sources:

- Changed and new items (all supplier catalogs)
- Deleted items (all supplier catalog)

Name: Changed and new items (all supplier catalogs)

Parameter:

Catalog(s): catalog(s) ☐ Mandatory value ☐ Editable

Version: ☐ Mandatory value ☐ Editable

Working version: ☐ Mandatory value ☐ Editable

Reference date: lastModificationDate ☐ Mandatory value ☐ Editable

Type of change: New and changed items ☐ Mandatory value ☐ Editable

Buttons: New... Delete OK Cancel

- **Deleted items (all supplier catalogs)**

**Export format template properties**

**All Supplier Catalogs**

You can select and configure the data sources to be used by this format template here.

Settings: Data sources | Variables | Data fields | Functions | File management | File attachments | Post-processing

Data sources:

- Changed and new items (all supplier catalogs)
- Deleted items (all supplier catalog)

Name: Deleted items (all supplier catalogs)

Parameter:

Catalog(s): catalog(s) ☐ Mandatory value ☐ Editable

Date: lastModificationDate ☐ Mandatory value ☐ Editable

Version: Working version ☐ Mandatory value ☐ Editable

Buttons: New... Delete OK Cancel

This data source lets user choose catalog(s) while export job is triggered as follows:

**Export export profile - All Supplier Catalogs**

**Export parameter configuration**

Set the export parameters for executing this export.

Variables:

catalog(s):\*

lastModificationDate:\*

before exporting

Modal Dialog:

- ☐ Catalog\_HPM\_17457
- ☐ Catalog\_HPM\_17457\_Product
- ☐ qft-70200-SUV10QA07-1584309698543


Buttons: All None OK Cancel

Navigation: < Back Next > Finish Cancel



## 9.3 Migration of 3PPD index

- [Old configuration](#)(see page 57)
- [New configuration](#)(see page 58)
  - [Exporting data for multi-PPD index](#)(see page 58)
  - [Identifying the parent](#)(see page 58)
  - [Ensuring export of single record exports it's whole hierarchy](#)(see page 59)
    - [Product2G](#)(see page 59)
    - [Variant](#)(see page 59)
    - [Article](#)(see page 59)
  - [Migration of 3PPD template from v10.0.0.0](#)(see page 60)

 Step-by-step guide - How to convert an existing text file based search index configuration into the new export template based search index configuration.

### 9.3.1 Old configuration

Below is the configuration for current status field

*entity.Article.pageable= true*  
*entity.Article.parent= Variant*

*entity.Variant.pageable= true*  
*entity.Variant.parent= Product2G*

*field.Article.CurrentStatus.facetable= true*  
*field.Article.CurrentStatus.facetordervalue=2*  
*field.Article.CurrentStatus.searchable= true*  
*field.Article.CurrentStatus.sortable= true*  
*field.Article.CurrentStatus.stored= true*  
*field.Article.CurrentStatus.type= infotext*

*field.Variant.CurrentStatus.facetable= true*  
*field.Variant.CurrentStatus.facetordervalue=2*  
*field.Variant.CurrentStatus.searchable= true*  
*field.Variant.CurrentStatus.stored= true*  
*field.Variant.CurrentStatus.sortable= true*  
*field.Variant.CurrentStatus.type= infotext*

*field.Product2G.CurrentStatus.facetable= true*  
*field.Product2G.CurrentStatus.facetordervalue=2*  
*field.Product2G.CurrentStatus.searchable= true*  
*field.Product2G.CurrentStatus.stored= true*  
*field.Product2G.CurrentStatus.sortable= true*  
*field.Product2G.CurrentStatus.type= infotext*

## 9.3.2 New configuration

### 9.3.2.1 Exporting data for multi-PPD index

Just like 1PPD index, here users need to define configuration and data modules.

- In the configuration users would need to configure each entities and their fields. Under the array *rootEntities*, users just need to define all the entities. **What is important here is that the relationship between entities should be defined using the field *parentEntityIdentifier*.**
- Users would require data sources for all these entities and a module corresponding to each. Contents of these modules would be pretty much be similar to what was already explained in [Migration of Items of any Catalog index](#)(see page 48) except that they would be entity specific.

What differentiates a 1PPD index from multi-PPD index is that, there are relations present between different entities. The way 1PPD template was defined, exported data would not have enough information to create a document with the parent and children information in every record. To fulfill such a requirement, primarily 2 things are required:

1. Some way to identify the parent or children record in the exported data.
2. Ensure that if any record from the hierarchy is getting exported then it's parent and all it's children and their children and so on are also exported. In short, full family gets exported each time a member gets exported.

### 9.3.2.2 Identifying the parent

For the purpose of identifying the parent, a new field is introduced for multi-PPD configuration. This field is called *RecordJoin*.

- *RecordJoin* would have a string value at the root of the hierarchy.
- *RecordJoin* would contain object as value otherwise, this object contains two fields:
  - *name* This field specifies the entity identifier for that record.
  - *parent* This field specifies the parent record's internal id along with it's container in the format of *id@container*

For example,


If there are 3 entities present in the system namely, Product2G, Variant, and Article. They are related as Product2G parent of Variant which is a parent of Article.

- *RecordJoin* for Product2G would look like this

```
{?JSONStringElement "RecordJoin","Product2G"}
```

- *RecordJoin* for Article would look like this

```
{?JSONObjectElement "RecordJoin",
  {?JSONObject
    {?JSONStringElement "name","Article"},
    {?JSONStringElement "parent",{?FormatDecimal {&Higher-level
variant.Referenced variant.Internal variant number}, ".", 0}@1}
  }
}
```

 There are data types available like Higher-level variant for Article which is used above to extract such an information.

### 9.3.2.3 Ensuring export of single record exports it's whole hierarchy

In order to achieve this goal, all we need is a few more data sources. The good thing is all that user needs to do is, create a module based off of these data sources and just copy paste contents from the data source which represents it's entity.

For 3PPD paradigm, modules are as mentioned below:

#### Product2G


- New Products
- Changed Products
- Products by changed variants
- Products by variants by changed items

#### Variant

- New Variants
- Changed Variants
- Variants by changed product
- Variants by changed items

#### Article

- New Items
- Changed Items
- Items by variant by changed products
- Items by changed variants

 Deleted Items/Products/Variants modules are not mentioned here as they have nothing to do with exporting the hierarchy. However they need to present in the template.


In order to make these modules, following approach was taken

- Create *Changed Product*

- Select *Variants by product assignments*, select *Changed Product* as value of Product assortment. This data source is named *Variants by changed product*
- Select *Items by variant assignments*, select *Variants by changed product* as value of Variant assortment.

This cascading effect ensures that everything that is related to the "Changed Products" gets exported as well. Similar was done for "Changed Items" and "Changed Variants".

As it can be observed from the names of the data source, they very well take care of exporting any related record in the hierarchy. For example, if an item was changed and then it's corresponding parent variant will be exported from the data source *Variants by changed items* and it's corresponding parent product will be exported using the data source *Products by variants by changed items*.

 Users need not worry about redundancy of data in the export. That is internally taken care of.


#### 9.3.2.4 Migration of 3PPD template from v10.0.0.0

Templates are much simpler than they were in v10.0.0.0. Following steps would help a user in migration:

- Config stays the same.
- New language variable needs to be defined and logical keys based off of language need to be qualified using this one. (Optional)
- Users would need all the data sources that are defined above and a corresponding module for each.
- Deleted products and deleted variants stay the same.
- Copy the contents of *New and changed products* to all the above Product2G modules. (along with the sub module).
- From *New and Changed Variants*, remove the routing field. Remove the Higher-level Product sub module. Copy all it's contents to all the new Variant modules as mentioned above. (along with sub modules)
- Remove all the other modules.
- Define new modules for item. Copy the contents from an items-only template and paste it here on all the modules along with sub-modules. Add the *RecordJoin* field as mentioned above.

## 9.4 Migration of 2PPD index

- [Old Search Configuration](#)(see page 61)
- [New configuration](#)(see page 61)
- [Migration of 2PPD template from v10.0.0.0](#)(see page 61)

 Step-by-step guide - How to convert an existing text file based search index configuration into the new export template based search index configuration.

 Go through [Migration of 3PPD index](#)(see page 57) before proceeding.

### 9.4.1 Old Search Configuration

Old configuration will be defined as (If Article.CurrentStatus and Product2G.ManufacturerAID needs to be indexed):

```
field.Article.CurrentStatus.facetable= true
field.Article.CurrentStatus.facetordervalue=2
field.Article.CurrentStatus.searchable= true
field.Article.CurrentStatus.sortable= true
field.Article.CurrentStatus.stored= true
field.Article.CurrentStatus.type= infotext

field.Product2G.ManufacturerAID.searchable= true
field.Product2G.ManufacturerAID.type= referencenumber
field.Product2G.ManufacturerName.searchable= true
field.Product2G.ManufacturerName.sortable= true
field.Product2G.ManufacturerName.facetable=true
field.Product2G.ManufacturerName.facetordervalue=-1
field.Product2G.ManufacturerName.type= infotext
```

### 9.4.2 New configuration

In case of 2PPD index, only 2 root entities need to be configured in the config file. In a standard system these would be Product2G and Article. As already mentioned in 3PPD migration, **parentEntityIdentifier** needs to be added.

- *RecordJoin* field needs to be added to the record as explained in 3PPD migration.
- In order to ensure all relevant records in hierarchy always gets exported following modules are required.
  - **Product2G**
    - New Products
    - Changed Products
    - Products by Item assignments (It has assortment specified as Changed Items)
  - **Item**
    - New Items
    - Changed items
    - Items by product assignments (It has assortment specified as Changed Products)
- It is very clear from the module names, all the records in hierarchy are exported whenever any of the record goes through a change. For example, a product was exported from *Changed Products*. Then all it's items will be exported from *Items by product assignments*.
- Note that as already mentioned all the modules in an entity will always have the identical data in it. This can introduce redundancy in exported data but that is taken care of internally.

### 9.4.3 Migration of 2PPD template from v10.0.0.0

Follow the following steps to easily migrate from v10.0.0.0

- Introduce a language variable and qualify all logical keys which are based on language using this variable. (Optional)
- Copy the contents of New and Changed products and all it's sub-modules to above specified modules under Product
- Deleted products and deleted items stay as is.
- Remove modules *Deleted item product reference* and *HandleRelationshipChanges* along with all it's sub-modules.
- Remove the *routing* field from Changed and New Items.
- Remove the *Higher-level product* sub-module.

## 10 Business Process Management

### 10.1 Workflows for Product 360 Versions 10.1

A new best practice workflow example is available with Version 10.1.

We strongly encourage to check out the corresponding Informatica BPM Accelerator.

This best practice workflow, if followed, provides significant better overall performance and reduction on the load for all system components, especially the BPM server itself. However, no automatic migration to this workflow is possible, but a manual adjustment is required.

Previously existing workflows should still work fine, with one exception. In case the fieldChangeInfo elements of the ItemModified trigger payload is used, the workflow needs to be adjusted as this payload has changed. See also Trigger Payload

### 10.2 Workflows for Product 360 versions >= 8.0.00

Workflow processes designed with the Informatica BPM Designer for the Product 360 versions 8.0.00 to 8.0.03 do not have to be migrated.

However, a new default workflow is required for the new "Terminate workflow" functionality and needs to be deployed to the Informatica BPM server (see Informatica BPM Installation).

### 10.3 Workflows for Product 360 versions < 8.0.00

Workflow processes designed with the jBPM workflow engine for Product 360 versions earlier than version 8.0.00 cannot automatically be migrated and deployed to the Informatica BPM server.

## 11 Audit Trail Migration

The audit trail migration consists of two parts, software migration and data migration.

## 11.1 Pre-Migration Checklist

- P360 Server and Desktop are updated to the latest version
- Legacy Audit Trail database is still available
- The repository is completely configured for the new Audit Trail

## 11.2 Part 1: Migrate to new Audit Trail module

As the Audit Trail feature is delivered as part of the Server and Desktop Client package, nothing needs to be done here.

## 11.3 Part 2: Migrate existing data

With Product 360 10.1 the audit trail component has been redesigned. Instead of storing changes in a relational database, they are now stored in Elasticsearch in the form of a JSON document.

With the data migration it is possible to convert old database entries into such JSON documents to keep them accessible in the system.

### 11.3.1 General process

The migration has been implemented as a simple "read - transform - write" cycle.


Beginning with today and going backwards up to the defined migration date, data chunks of 1,000 changes will be processed. They are read from the old audit trail database and if the root entity is configured to support audit trail it is transformed into the new JSON document structure and written to Elasticsearch.

The user can cancel the job at any time. The job will write a file to the shared folder indicating where it stopped so that it can continue with its work if the job is started again.

### 11.3.2 Configuration

#### 11.3.2.1 Repository

Please ensure the repository has all needed Audit Trail configurations in place. This is important because only data for audit trail enabled root entities will be migrated, all other data will be skipped.

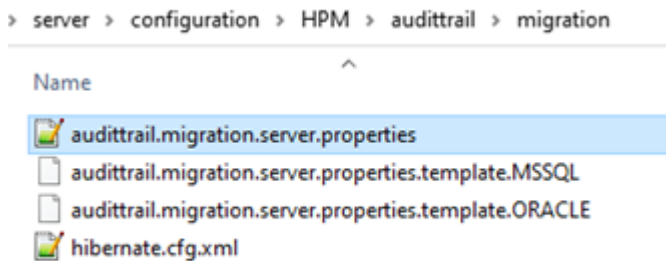
 The entities in the 10.1 repository enabled for Audit Trail do not exactly match those in previous versions. Please check all settings and adapt them to your needs.

### 11.3.2.2 audittrail.migration.server.properties

The access to the legacy Audit Trail database should be configured in

<PIM\_SERVER\_INSTALLATION\_ROOT>\configuration\HPM\audittrail\migration\audittrail.migration.server.properties file.

This file can be created from a corresponding template file. The essential settings here are the host and schema configuration. You can take the settings from the old configuration for audit trail.



#### audittrail.migration.server.properties

```
### General Host
dest.host                = host

### Default database settings
db.default.type          = MSSQL

...

### AuditTrail database/schema
db.audittrail.schema     = HPM_AUDITTRAIL
db.audittrail.schema.backup = ${db.audittrail.schema}_BAK
db.audittrail.server     = ${dest.host}
db.audittrail.port       = 1433
db.audittrail.user       = user
db.audittrail.password   = password
```

### 11.3.2.3 plugin\_customization.ini

In order to use the migration you have to configure the time period for which you want to migrate changes, in the <PIM\_SERVER\_INSTALLATION\_ROOT>\configuration\HPM\plugin\_customization.ini



**plugin\_customization.ini**

```
# -----
# Audit trail migration preferences
# -----
# All old audit trail entries between this date and today will be transformed to the
new audit trail
# format and stored in elastic.
# This value is mandatory
# Date is included
# Format: yyyy-MM-dd
# com.heiler.ppm.audittrail.migration.server/migrationdate =
#
# Locale to be used for the audit trail migration job
# This should be the same value as it was for the "audittrail.atcsbuilder.locale"
property
# in the audittrail.properties file of older versions
# Default is en_US
# com.heiler.ppm.audittrail.migration.server/locale=en_US
```

The migration date is the end date of the migration. All changes between the given date and today are migrated to Elasticsearch. This value is mandatory.

The locale is needed to resolve labels of the objects to be migrated. When the data was written, the locale configured for the attribute `audittrail.atcsbuilder.locale` in the `audittrail.properties` file was used. The same value should be used for the data migration.

### 11.3.3 Rights

There is a new action right the user must have in order to execute the migration:

"Audit Trail, migration" - "Permission to start audit trail migration job"

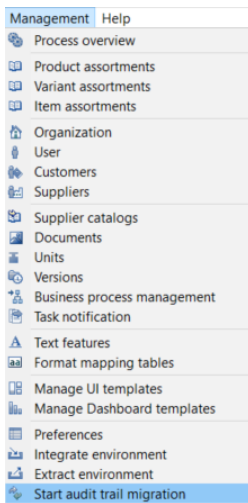
Other rights are not considered in the migration process.

### 11.3.4 Start Migration

In order to start the migration, start the Desktop Client and open the Management menu. You will find a new entry at the bottom "Start audit trail migration".

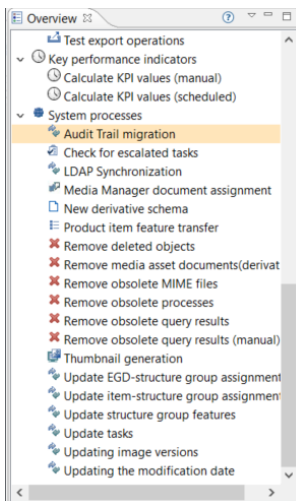
If the entry is missing, the user probably does not have the action right.

Only one migration job can run at a time, so if the menu entry is disabled, a job instance might already be running.



You can find the executed server job in the process overview perspective under System processes.

The problem log of the job will show the progress as well as errors.



The number of migrated changes is logged for every processed day. This includes also the number of errors or skipped changes. A change will be skipped if it could not be migrated, details will also be logged, or if its entity is not configured to support audit trail.

### 11.3.5 After migration

After the migration you can revoke the migration right from your users.

The customer won't need to start the audit trail server anymore. This also means the old audit trail database is not used anymore. If you didn't migrate all data and you are not sure if you might need the data again sometime, think about storing a backup from the database.

## 11.3.6 Troubleshooting

### 11.3.6.1 Why are there so many skipped records?

If you have a look at the migration protocol you might wonder why there are so many skipped records.

37	i	Summary	2020-09-30	Processed day: 2020-09-30, successfully migrated records: 249, erroneous records: 0, skipped records: 215, problems: 0
----	---	---------	------------	------------------------------------------------------------------------------------------------------------------------

As we only migrate records for audit trail enabled root entities, there might be a problem in your repository configuration. But in most cases it should be one of the following reasons:

- In our standard repository, we deactivated Audit Trail for MediaAsset since it is mapped to the objects (e.g. item, structure group) which already record the corresponding Audit Trail information. That's why records for media assets will not be migrated.
- By default, we don't log changes for User entity, that means "Last login" and "Last login with" will be skipped in the migration. Depending on user activity in the past, there could be many such records.

If you want to count the records for each entity for a specific day, you can use the following SQL statement:

#### Count records of one day grouped by entity

```

1  -- (MSSQL) list all entities and the corresponding record count for one
2  day, e.g. 2020-09-30
3  SELECT EntityName, count(EntityName)
4  FROM EntityChange
5  WHERE ModificationDate < '2020-09-30 23:59:59' and ModificationDate >
6  '2020-09-30 00:00:00'
7  AND ParentEntityChangeID is null
8  GROUP BY EntityName

```

### 11.3.6.2 Re-execute migration

If something goes horribly wrong, what we - of course - don't anticipate, don't worry. The data in the old data base is not destroyed by the migration, so you can do it again. In order to remove the already migrated entries from elastic you have to delete the migration indexes.

The index names follow a pattern, dependent on the settings in the repository. The migration indexes have the same names, but with the suffix '\_migrate', e.g.

- Changes of products of the Master catalog are stored in <system\_name>.audit\_product2g\_master\_migrate
- Changes of items of supplier catalogs are stored in <system\_name>.audit\_article\_supplier\_migrate
- Changes of other objects are usually stored in <system\_name>.audit\_longterm\_main\_migrate

You can find the system name in the server.properties.

To delete the indexes, either use an interface for elastic like kibana or use the following commands:

```
DELETE /<index>
DELETE /_template/<index>_template
```

For further information see [Elastics Delete index API](#) and [Delete index template API](#).

Remove the progress.json file from the configured shared folders subfolder audittrailMigration and you can try again.

### 11.3.6.3 Increase trace level

There is also a possibility to increase the trace level for audit trail migration job to see more information about records migrated with warnings. By default, only errors will be logged.

Enable the logging of migration warnings to the Process overview log in the log4j2.xml file. Be aware of very technical information which will be logged and use this trace level carefully.

```
<Logger name="AT_MIGRATION" level="DEBUG" />
```

### 11.3.7 Limitations

Because in the old audit trail some data hasn't been stored that is available in the new audit trail, there are some limitations to the migrated changes

- Attribute and Characteristic values are not formatted according to their data type (which is a capability introduced with the new audit trail)
- Changes of characteristic values are missing the characteristic hierarchy to which they belong (which is a capability introduced with the new audit trail)

## 12 Message Queue Migration

We have to do following changes in server.properties in order to enable batching.

- Remove default batch threshold and batch timeout parameters.

```
queue.default.message.batch.threshold = 100
queue.default.message.batch.timeout = 5000
```

- Remove all properties related to Data Quality queue.

```

queue.dq.type = ${queue.default.type}
queue.dq.writer.count = ${queue.default.writer.count}
queue.dq.consumer.count = ${queue.default.consumer.count}
queue.dq.url = ${queue.default.url}
queue.dq.username = ${queue.default.username}
queue.dq.password = ${queue.default.password}
queue.dq.message.format = XML
queue.dq.name = P360_DATA_QUALITY
queue.dq.label = Data Quality
queue.dq.delivery.delay = ${queue.default.delivery.delay}
queue.dq.message.batch.threshold = ${queue.default.message.batch.threshold}
queue.dq.message.batch.timeout = ${queue.default.message.batch.timeout}

```

- Add new properties for batch queue. Threshold and Timeout values are not required here.

```

queue.batchapi.type = ${queue.default.type}
queue.batchapi.writer.count = ${queue.default.writer.count}
queue.batchapi.consumer.count = ${queue.default.consumer.count}
queue.batchapi.url = ${queue.default.url}
queue.batchapi.username = ${queue.default.username}
queue.batchapi.password = ${queue.default.password}
queue.batchapi.message.format = XML
queue.batchapi.name = P360_BATCH_API
queue.batchapi.label = Batch API
queue.batchapi.delivery.delay = ${queue.default.delivery.delay}

```

## 13 How to apply a Hotfix or EBF

### 13.1 Preconditions.

- You must be able to use a command prompt to continue. If not, please contact your system administrator to assist.
- Please read the Release Notes of the version to be installed.
- **It is highly recommended to backup your system before updating. This especially includes the databases, file storage areas, configuration files and binaries distributables.**

### 13.2 Installation of Hotfix or EBF on Product 360 Server

**i** EBF packages contain only **delta** plugins which have been changed. In order to recognize such packages, the suffix delta is contained in its name. Hotfixes and Major releases have the suffix **full** in its name.

Before applying an Hotfix or Major update, you have to update the Control Center itself. See [P360 - Control Center Migration\(see page 6\)](#)

- Login into the Control Center
- Navigate to the installation tab
- With the upload functionality, upload all packages you want to deploy. Do not rename or extract any of the packages that are delivered.
- If you want to deploy also accelerator plugins or custom plugins, you can also upload them. Custom plugins have to be uploaded separately as jar file.
- Please note that all uploaded files will be deployed and overwrite the existing files. But all files will be backedup in a folder, so you can revert them if something went wrong.
- By clicking the upload button, the update process will start and all configured servers will be updated. All configuration files on the server will be also replaced with the ones that are located in the Control Center package.
- Execute steps in [PIM - Core Database Migration\(see page 12\)](#) by running setup to your existing database connection in order to update existing database.

Generally, you should run your custom JUnit tests in order to ensure that your custom functionality is still working.

### 13.3 Installation of Hotfix or EBF on Product 360 Desktop

- Before updating the Desktop you should execute the migration of the Server first.
- For **Hotfix** or **Major** updates:
  - Remove content of Product 360 Desktop Installation Folder and unzip the content of the new release package (EBF only apply delta.)
  - Update the configuration files if you have any customizings here. Instead of copying the whole file, rather copy only your changes, since there might be some new settings which then will be overwritten.
- For **EBF's**:
  - Remove the existing plugins with old revision and then place the plugin with new revision which are part of the EBF.

# Copyright

Copyright (c) 1993-2020 Informatica LLC. All rights reserved.

This Software and documentation contain proprietary information of Informatica LLC and are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright law. Reverse engineering of the software is prohibited. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC. This Software may be protected by [U.S. and/or](#) international Patents and other Patents Pending.

Use, duplication, or disclosure of the Software by the U.S. Government is subject to the restrictions set forth in the applicable software license agreement and as provided in DFARS 227.7202-1(a) and 227.7702-3(a) (1995), DFARS 252.227-7013©(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

The information in this product or documentation is subject to change without notice. If you find any problems in this product or documentation, please report them to us in writing.

Informatica, Informatica Platform, Informatica Data Services, PowerCenter, PowerCenterRT, PowerCenter Connect, PowerCenter Data Analyzer, PowerExchange, PowerMart, Metadata Manager, Informatica Data Quality, Informatica Data Explorer, Informatica B2B Data Transformation, Informatica B2B Data Exchange Informatica On Demand, Informatica Identity Resolution, Informatica Application Information Lifecycle Management, Informatica Complex Event Processing, Ultra Messaging and Informatica Master Data Management are trademarks or registered trademarks of Informatica LLC in the United States and in jurisdictions throughout the world. All other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved.

Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright © ej-technologies GmbH. All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved. Copyright © International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright (c) University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright

© EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jqWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright © MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved.

This product includes software developed by the Apache Software Foundation ( <http://www.apache.org/>), and/or other software which is licensed under various versions of the Apache License (the "License"). You may obtain a copy of these Licenses at <http://www.apache.org/licenses/>. Unless required by applicable law or agreed to in writing, software distributed under these Licenses is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Licenses for the specific language governing permissions and limitations under the Licenses.

This product includes software which was developed by Mozilla ( <http://www.mozilla.org/>), software copyright The JBoss Group, LLC, all rights reserved; software copyright © 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under various versions of the GNU Lesser General Public License Agreement, which may be found at [http:// www.gnu.org/licenses/lgpl.html](http://www.gnu.org/licenses/lgpl.html). The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (©) 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at <http://www.openssl.org> and <http://www.openssl.org/source/license.html>.

This product includes Curl software which is Copyright 1996-2013, Daniel Stenberg, < [daniel@haxx.se](mailto:daniel@haxx.se) >. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at [http://curl.haxx.se/docs/ copyright.html](http://curl.haxx.se/docs/copyright.html). Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (©) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at [http://www.dom4j.org/ license.html](http://www.dom4j.org/license.html).

The product includes software copyright © 2004-2007, The Dojo Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at [http://dojotoolkit.org/ license](http://dojotoolkit.org/license).

This product includes ICU software which is copyright International Business Machines Corporation and others. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://source.icu-project.org/repos/icu/icu/trunk/license.html>.

This product includes software copyright © 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at [http://www.gnu.org/software/kawa/ Software-License.html](http://www.gnu.org/software/kawa/Software-License.html).

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at <http://www.opensource.org/licenses/mit-license.php>.

This product includes software developed by Boost ( <http://www.boost.org/>) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at [http:// www.boost.org/LICENSE\\_1\\_0.txt](http://www.boost.org/LICENSE_1_0.txt).

This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at <http://www.pcre.org/license.txt>. This product includes software copyright © 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at [http:// www.eclipse.org/org/documents/epl-v10.php](http://www.eclipse.org/org/documents/epl-v10.php) and at <http://www.eclipse.org/org/documents/edl-v10.php>.

This product includes software licensed under the terms at <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib/?License>, [http:// www.stlport.org/doc/ license.html](http://www.stlport.org/doc/license.html), <http://asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqLicense.html>, <http://unit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt> , [http://www.gzip.org/zlib/zlib\\_license.html](http://www.gzip.org/zlib/zlib_license.html), [http://www.openldap.org/software/release/ license.html](http://www.openldap.org/software/release/license.html), <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, [http://fusesource.com/downloads/license- agreements/fuse-message-broker-v-5-3- licenseagreement](http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-licenseagreement); [http:// antlr.org/license.html](http://antlr.org/license.html); <http://aopalliance.sourceforge.net/>; [http://www.bouncycastle.org/ licence.html](http://www.bouncycastle.org/licence.html); <http://www.jgraph.com/jgraphdownload.html>; <http://www.jcraft.com/jsch/LICENSE.txt>; [http://jotm.objectweb.org/bsd\\_license.html](http://jotm.objectweb.org/bsd_license.html); . [http://www.w3.org/Consortium/Legal/ 2002/copyright-](http://www.w3.org/Consortium/Legal/2002/copyright-)



software-20021231; <http://www.slf4j.org/license.html>; <http://nanoxml.sourceforge.net/orig/copyright.html>; <http://www.json.org/license.html>; <http://forge.ow2.org/projects/jaservice/>, <http://www.postgresql.org/about/licence.html>, <http://www.sqlite.org/copyright.html>, <http://www.tcl.tk/software/tcltk/license.html>, <http://www.jaxen.org/faq.html>, <http://www.jdom.org/docs/faq.html>, <http://www.slf4j.org/license.html>; <http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License>; <http://www.keplerproject.org/md5/license.html>; <http://www.toedter.com/en/jcalendar/license.html>; <http://www.edankert.com/bounce/index.html>; <http://www.net-snmp.org/about/license.html>; <http://www.openmdx.org/#FAQ>; [http://www.php.net/license/3\\_01.txt](http://www.php.net/license/3_01.txt); <http://srp.stanford.edu/license.txt>; <http://www.schneier.com/blowfish.html>; <http://www.jmock.org/license.html>; <http://xsom.java.net>; <http://benalman.com/about/license/>; <https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>; <http://www.h2database.com/html/license.html#summary>; <http://jsoncpp.sourceforge.net/LICENSE>; <http://jdbc.postgresql.org/license.html>; <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>; <https://github.com/rantav/hector/blob/master/LICENSE>; <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>; <http://jibx.sourceforge.net/jibx-license.html>; <https://github.com/lyokato/libgeohash/blob/master/LICENSE>; <https://github.com/hjiang/jsonxx/blob/master/LICENSE>; <https://code.google.com/p/lz4/>; <https://github.com/jedisct1/libsodium/blob/master/LICENSE>; <http://one-jar.sourceforge.net/index.php?page=documents&file=license>; <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>; <http://www.scala-lang.org/license.html>; <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>; and <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>.

This product includes software licensed under the Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), the Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>) the Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (<http://www.opensource.org/licenses/bsd-license.php>), the new BSDLicense (<http://opensource.org/licenses/BSD-3-Clause>), the MIT License (<http://www.opensource.org/licenses/mitlicense.php>), the Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) and the Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

This product includes software copyright © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://xstream.codehaus.org/license.html>. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit <http://www.extreme.indiana.edu/>.

This product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.

This Software is protected by U.S. Patent Numbers 5,794,246; 6,014,670; 6,016,501; 6,029,178; 6,032,158; 6,035,307; 6,044,374; 6,092,086; 6,208,990; 6,339,775; 6,640,226; 6,789,096; 6,823,373; 6,850,947; 6,895,471; 7,117,215; 7,162,643; 7,243,110; 7,254,590; 7,281,001; 7,421,458; 7,496,588; 7,523,121; 7,584,422; 7,676,516; 7,720,842; 7,721,270; 7,774,791; 8,065,266; 8,150,803; 8,166,048; 8,166,071; 8,200,622; 8,224,873; 8,271,477; 8,327,419; 8,386,435; 8,392,460; 8,453,159; 8,458,230; 8,707,336; 8,886,617 and RE44,478, International Patents and other Patents Pending.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.iv>

## NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY,

MISREPRESENTATION AND OTHER TORTS.