



InformaticaTM

Migration

Informatica MDM - Product 360

Version: 10.5 HotFix 3 SP 1

Table of Contents

1	Pre-Migration Checklist	8
2	Repository Migration	8
2.1	Required Manual Adjustments	8
2.1.1	Short Identifier	8
2.1.2	Audit Trail Settings.....	9
2.2	Repository Merger.....	9
2.2.1	Installation and Execution.....	11
2.2.2	Troubleshooting.....	11
2.2.2.1	Duplicate Elements.....	12
3	Database Migration.....	13
3.1	Media Manager Database Migration	13
3.1.1	Updating the Product 360 - Media Manager database.....	13
3.1.1.1	Automatic database update	13
3.1.1.2	Manual database update	13
3.1.1.3	Legacy versions	14
3.2	Product 360 Core Database Migration	15
3.2.1	Duplicate Article Attribute Names in the repository default language	15
3.2.1.1	Problem/ Situation	15
3.2.1.2	Solution 1	16
3.2.1.3	Solution 2	16
3.2.2	ArticleTrading entries without existing Party entry are removed	16
3.2.2.1	Problem/Situation	16
3.2.3	Invalid decimal formatting in Article Attribute Values or Structure Group Attribute Values	17
3.2.3.1	Problem/ Situation	17
3.2.3.2	Solution	17
3.2.4	Structure Migration.....	18
3.2.5	While migrating to 8.0.03.....	18
3.2.6	Qualification permission info entries in the log	19
3.2.7	Multiple Purposes for Import Mappings in the GenericData Table	21
3.2.7.1	Problem/ Situation	21
3.2.7.2	Solution	21

3.2.8	SQL scripts to check and migrate invalid decimal formatting in Article Attribute Values	22
3.2.9	SQL scripts to find and correct duplicate attribute names.....	24
4	Server and Desktop Migration.....	29
4.1	Installation of Hotfix, EBF or One-Off-EBF (=Patch) on Server	29
4.2	Installation of Hotfix, EBF or One-Off-EBF (=Patch) on Desktop	29
4.3	Server Properties(MSSQL) Database connection property	30
4.4	Repository (10.5.0.00 onwards).....	30
4.5	Structure Migration.....	31
4.5.1	Run Structure Migration Tool.....	32
4.6	XML parsing exception while loading export templates or other data graph objects.....	32
4.7	Entity IDs for ArticleLangType and ArticleExtensionType repository custom entities.....	34
4.7.1	Database.....	35
4.7.2	Repository	36
4.8	Server Jobs.....	37
4.9	Data Quality.....	38
4.9.1	Migration	38
4.9.1.1	Rule mapplets, configurations and dictionaries	38
4.9.2	Update	38
4.9.2.1	Data Quality SDK and Engine Update	38
4.9.2.2	Rule mapplets, configurations and dictionaries	39
4.10	Import mappings	39
4.11	Migration of Subentity Deletion customizing.....	40
4.12	Media Manager Integration	40
4.12.1	Adjust Media Manager database connection string (MSSQL only).....	40
4.13	Server and Desktop Pre-Migration Checklist.....	41
4.13.1	Structure Migration Preparation	41
4.13.1.1	Download Structure Migration Preparation Tool.....	41
4.13.1.2	Install Structure Migration Preparation tool	41
4.13.1.3	Run Structure Migration Preparation Tool	42
4.13.1.4	Correct invalid item/variant/product mappings.....	42
5	Unit Management.....	43
5.1	Overview.....	43
5.2	Repository	44

5.3	Data Migration	44
5.4	Enum Provider.....	45
5.5	Desktop UI	45
5.6	Import	45
5.7	Export	45
5.8	Service API	45
6	Media Manager Migration	46
6.1	Update Media Manager	46
6.1.1	Updating Media Manager File Server	46
6.1.2	Updating the database	46
6.1.3	Updating the client modules	46
6.1.3.1	Manually updating the client modules (Windows).....	47
6.1.3.2	Manually updating the client modules (OSX)	47
6.1.4	Updating Functd.....	47
6.1.5	Updating web front end.....	47
6.1.6	Deactivate old modules	48
6.1.6.1	Deactivate Session Manager.....	48
6.1.6.2	Deactivate Internet Functd	49
7	Supplier Portal Migration	49
7.1	Software upgrade (10.5.02.01 onwards)	49
7.2	On the Product 360 Supplier Portal Server.....	49
7.2.1	Adjust database connection string (MSSQL only)	50
7.3	Supplier Portal - Product 360 Integration.....	51
7.3.1	Setup Product 360 Core User groups and Permissions	51
7.3.1.1	Create/Edit Product 360 - Supplier Portal Item Editor User Group	52
7.3.1.2	Create Product 360 Supplier Portal Item Viewer User Group	55
7.3.1.3	Create/Edit Product 360 Supplier Portal Administrator Users Group	58
7.3.1.4	Example	60
8	Web Search Migration	61
8.1	Migration of Items of any Catalog index	61
8.1.1	Old Index configuration for field ArticleLang.DescriptionShort	61
8.1.2	New export template based configuration.....	62
8.1.2.1	Handling sub-entities in sub-module	66

8.2	Migration of Items of All-Supplier-Catalog index	67
8.2.1	Configurations prior to v10.0.0.0, this would be defined as:	67
8.2.2	New Configuration	68
8.2.2.1	Data Type.....	68
8.2.3	Data Sources	68
8.3	Migration of 3PPD index	69
8.3.1	Old configuration	70
8.3.2	New configuration.....	70
8.3.2.1	Exporting data for multi-PPD index	70
8.3.2.2	Identifying the parent	71
8.3.2.3	Ensuring export of single record exports it's whole hierarchy	71
8.3.2.4	Migration of 3PPD template from v10.0.0.0	72
8.4	Migration of 2PPD index	73
8.4.1	Old Search Configuration	73
8.4.2	New configuration.....	74
8.4.3	Migration of 2PPD template from v10.0.0.0	74
9	Business Process Management	74
9.1	Workflows for Product 360 Versions 10.1	74
9.2	Workflows for Product 360 versions >= 8.0.00	75
9.3	Workflows for Product 360 versions < 8.0.00	75
10	Audit Trail Migration	75
10.1	Pre-Migration Checklist	75
10.2	Part 1: Migrate to new Audit Trail module	75
10.3	Part 2: Migrate existing data	75
10.3.1	General process.....	76
10.3.2	Database preparation.....	76
10.3.3	Configuration	76
10.3.3.1	Repository	76
10.3.3.2	audittrail.migration.server.properties.....	77
10.3.3.3	plugin_customization.ini.....	77
10.3.3.4	migration_template.json.....	79
10.3.4	Rights	79
10.3.5	Start Migration	80

10.3.6	After migration	81
10.4	FAQ.....	81
10.4.1	Which is the best way to migrate the data?	81
10.4.1.1	Parallel to normal P360 server operation	81
10.4.1.2	In advance of system upgrade.....	81
10.4.2	How long will the migration take?	82
10.4.2.1	Step 1	82
10.4.2.2	Step 2	83
10.4.2.3	Step 3	84
10.4.2.4	Example	84
10.4.3	Why are there so many skipped records?	84
10.4.4	Has really all my data been migrated?.....	85
10.5	Troubleshooting.....	86
10.5.1	Re-execute migration.....	86
10.5.1.1	Delete migrated data	86
10.5.1.2	Delete progress information	87
10.5.2	Increase trace level	87
10.5.3	Configure system resources used	87
10.5.3.1	Decreasing the memory usage	87
10.5.3.2	Decreasing the number of threads.....	87
10.5.4	Elasticsearch exceptions	88
10.5.4.1	Adjustments of elastic settings (Exception: 'Unable to parse response body')	88
10.5.4.2	Elasticsearch Exception (type=circuit_breaking_exception)	88
10.5.4.3	Elasticsearch Exception (type=illegal_argument_exception)	89
10.5.4.4	Elasticsearch Exception (type=search_phase_execution_exception)	89
10.5.5	Correct object identifier.....	89
10.5.5.1	Preparation	90
10.5.5.2	Starting the database setup	90
10.5.5.3	Logging	90
10.5.5.4	Further information	91
10.5.5.5	Technical details	91
10.5.6	Why does my data look strange?.....	91
10.6	Limitations	93
10.7	Migration of Audit Trail data for Media Assets	93

11	Message Queue Migration	94
----	-------------------------------	----

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 Repository Migration

- [Required Manual Adjustments](#) (see page 8)
 - [Short Identifier](#) (see page 8)
 - [Audit Trail Settings](#) (see page 9)
- [Repository Merger](#) (see page 9)
 - [Installation and Execution](#) (see page 11)
 - [Troubleshooting](#) (see page 11)
 - [Duplicate Elements](#) (see page 12)
 - [Solution](#) (see page 12)

2.1 Required Manual Adjustments

2.1.1 Short Identifier

Starting with version 10.1 we introduced the short identifier as a new attribute in the custom area of the repository. **It is mandatory for all editable logical keys, all entities, and all fields, regardless of whether they are used with audit trail or not.**

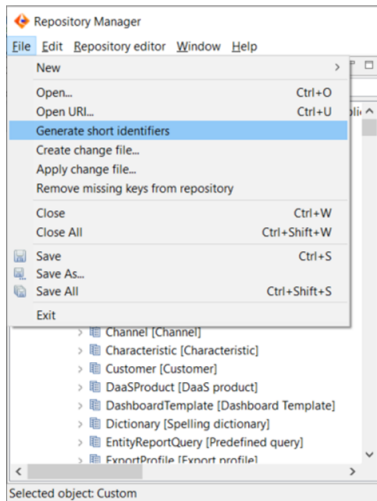
The standard repository already has a correct value for all logical keys, fields, and entities, but since this identifier should be chosen domain specific there is no automated way of providing this for customized repositories.

If you receive an error message like this: " LogicalKey:

ProductChannelCountryAttribute.LK.TargetMarket must have a short identifier" you are missing a short identifier in the repository

Please have a look at the Domain Model (Repository) documentation for details and examples of this identifier.

The repository merger also has a wizard to create short identifiers. Select the "Custom" node to generate the short identifiers for all entities, logical keys and fields, or a single entity node to add short identifiers to only one particular entity and its fields, logical keys and sub-entities.



The wizard will use the last part of the entity/field/logical key identifier and make the first character lower case.

Customers and Partners should definitely validate the generated short identifiers and adjust them in case they are not meaningful or ugly formatted etc. Remember that this identifier is used in the object API and audit trail and should never be changed once defined!

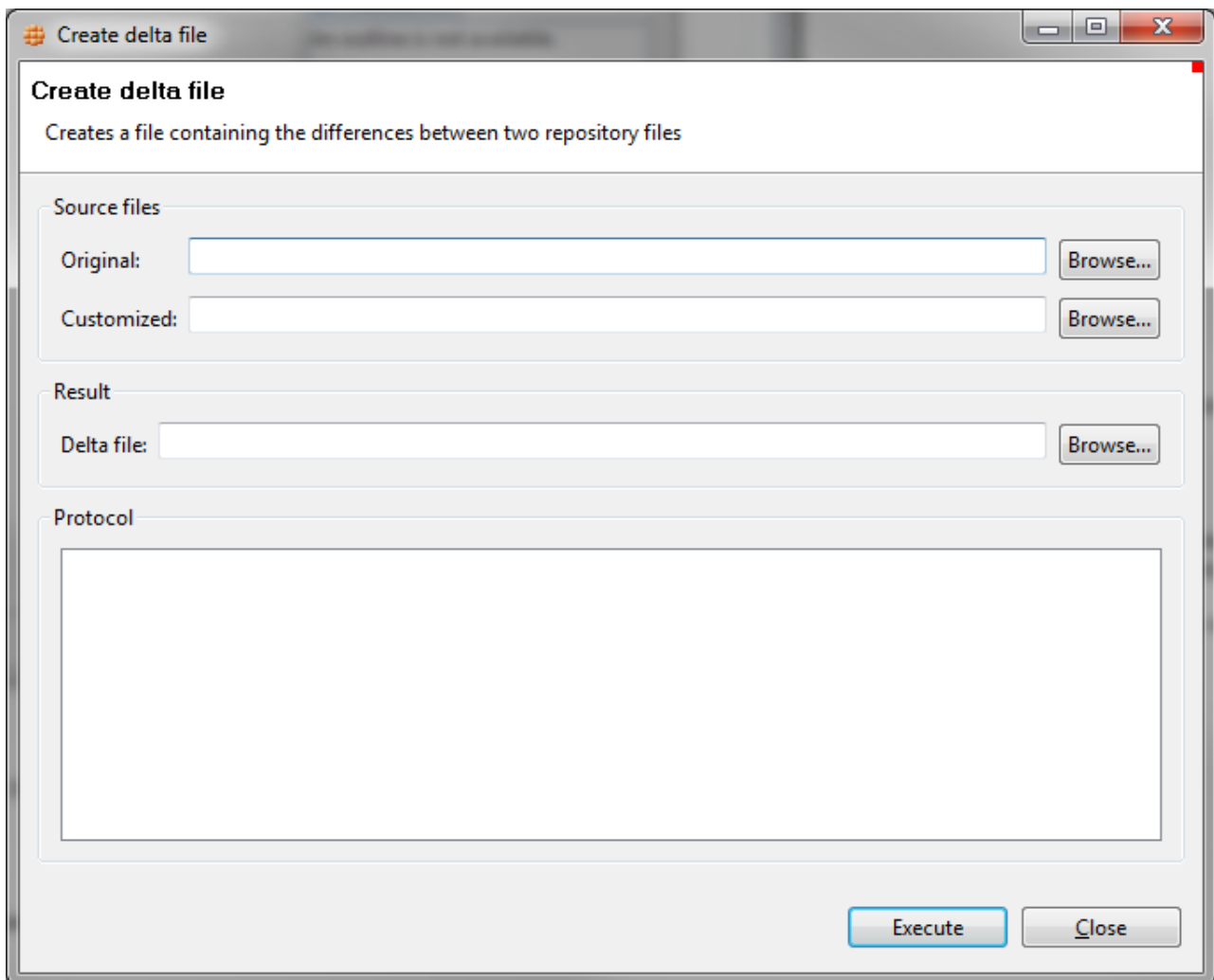
2.1.2 Audit Trail Settings

With 10.1 we introduced a new child element of entities called Audit Trail Settings. If you upgrade from an earlier version you should double check this setting for the root entities, it defines which ones will be recorded in audit trail and how they are being recorded.

Please see Audit Trail Configuration for more details.

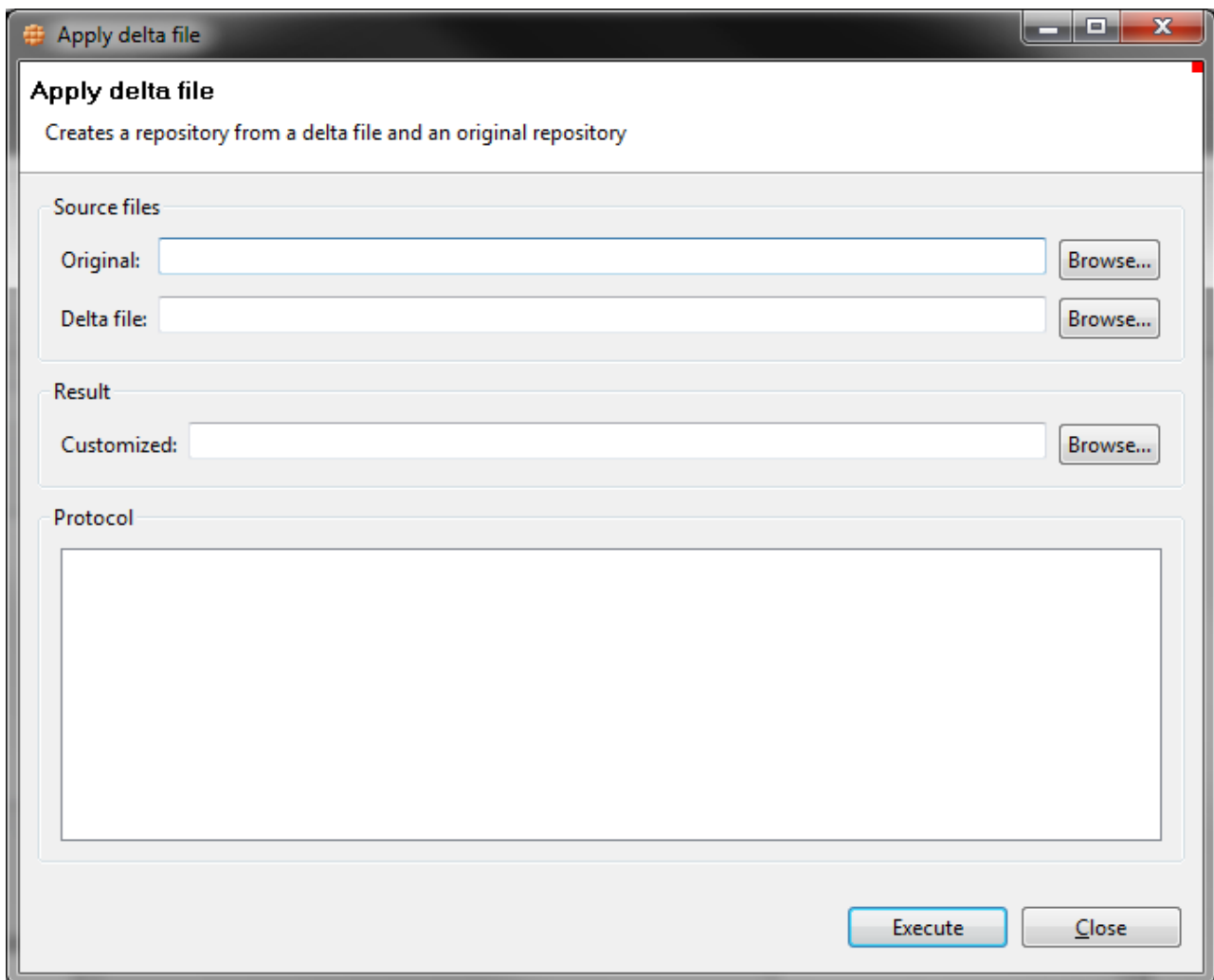
2.2 Repository Merger

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.



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.

2.2.1 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.

2.2.2 Troubleshooting

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

2.2.2.1 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.

3 Database Migration

3.1 Media Manager Database Migration

3.1.1 Updating the Product 360 - Media Manager database

3.1.1.1 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'.

3.1.1.2 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'.

3.1.1.3 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:

3.2 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:

3.2.1 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.

3.2.1.1 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 \(see page 24\)](#)" below.

3.2.1.2 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 \(see page 24\)](#)" 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 \(see page 24\)](#)" 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.

3.2.1.3 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.

3.2.2 ArticleTrading entries without existing Party entry are removed



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

3.2.2.1 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".

3.2.3 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

3.2.3.1 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.

3.2.3.2 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 \(see page 22\)](#)" below.

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

3.2.4 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 \(see page 29\)](#).

3.2.5 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. <code>Article</code> , <code>Structure</code> , <code>StructureGroup</code> , ...)	The columns <code>AcID</code> and <code>AcIFlag</code> have been moved from the root table (e.g. <code>Article</code>) to the respective revision table (e.g. <code>ArticleRevision</code>)
<code>ArticleAttribute</code>	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 <code>null</code> .
<code>Catalog</code>	The column <code>SupplierID</code> has been moved to the <code>CatalogRevision</code> table.

3.2.6 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"/>%');
```

3.2.7 Multiple Purposes for Import Mappings in the GenericData Table

3.2.7.1 Problem/ Situation

Import mappings are 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 purposes value via following sql statement:

Main database - Oracle

```
SELECT "Purposes", "GenericData",*
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.

3.2.7.2 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 - Desktop Client (recommended)

The first 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.

Manual Sanitization - Update Script

The second 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

3.2.8 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,]%')
)

```

3.2.9 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, 244) + ''' + @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."ArticleAttribut
eID"

```

```

XAAAL."DeletionTimestamp", 1, 0 ) = 1
vKeyLanguageID
vKeyLanguageID
Name"
D"

AND DECODE( XAA."DeletionTimestamp",
AND XAAAL."LanguageID" =
WHERE XAAAL."LanguageID" =
GROUP BY XAA."ArticleRevisionID", XAAAL."
HAVING COUNT(*) > 1
) x
ON x."ArticleID" = AA."ArticleRevisionI
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;
/

```

4 Server and Desktop Migration


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

 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.

4.2 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.

4.3 Server Properties(MSSQL) Database connection property

Server properties(MSSQL only) updates from 10.5.02.01 onwards

New connection property "encrypt=false" added in the jdbcUrl of main, master, supplier in the *server.properties.template.MSSQL*

Parameter	New value
db.main.pool.jdbcUrl	jdbc:sqlserver://\${db.main.server};\${db.main.port};databaseName=\${db.main.database};integratedSecurity=\${db.integrated.security};sendStringParametersAsUnicode=true;selectMethod=direct;responseBuffering=adaptive; encrypt=false
db.master.pool.jdbcUrl	jdbc:sqlserver://\${db.master.server};\${db.master.port};databaseName=\${db.master.database};integratedSecurity=\${db.integrated.security};sendStringParametersAsUnicode=true;selectMethod=direct;responseBuffering=adaptive; encrypt=false
db.supplier.pool.jdbcUrl	jdbc:sqlserver://\${db.supplier.server};\${db.supplier.port};databaseName=\${db.supplier.database};integratedSecurity=\${db.integrated.security};sendStringParametersAsUnicode=true;selectMethod=direct;responseBuffering=adaptive; encrypt=false

4.4 Repository (10.5.0.00 onwards)

The entityTypes **ArticleLangType**, **ArticlePriceType** and **ArticleReferenceType** have been modified to have **EntityId** in logical keys and fields list.

The standard custom entities created on top of the above entity types, have inherited the new logical key and field.

Entity type	OOB custom entities
ArticleLangType	<ul style="list-style-type: none"> ArticleLang [Language-specific data] VariantLang [Language-specific data] Product2GLang [Language-specific data]
ArticlePriceType	<ul style="list-style-type: none"> ArticlePricePurchase [Purchase price] ArticlePriceSales [Selling price] VariantPricePurchase [Purchase price] VariantPriceSales [Selling price] Product2GPricePurchase [Purchase price] Product2GPriceSales [Selling price]
ArticleReferenceType	<ul style="list-style-type: none"> ArticleComponent [Component] ProductReference [Higher-level product] VariantReference [Higher-level variant] Article2Product2GReference [Referenced products] Article2VariantReference [Referenced variants] ArticleReference [Referenced items] SuperordinateProductReference [Higher-level product] Variant2Product2GReference [Referenced products] Variant2VariantReference [Referenced variants] Variant2ArticleReference [Referenced items] Product2GReference [Referenced products] Product2G2VariantReference [Referenced variants] Product2G2ArticleReference [Referenced items]

Any custom entities created by customer, that uses ArticleLangType, ArticlePriceType or ArticleReferenceType, will have to be adjusted explicitly to add the new logical key and field for EntityId.

4.5 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 41)). 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)

4.5.1 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 29)
- 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.

4.6 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 its 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 incl. 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:8
3)
at
org.eclipse.emf.ecore.xmi.impl.XMLHandler.createObjectFromFactory(XMLHandler.java:217
8)
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:15
7)
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)

```

4.7 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.

4.7.1 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

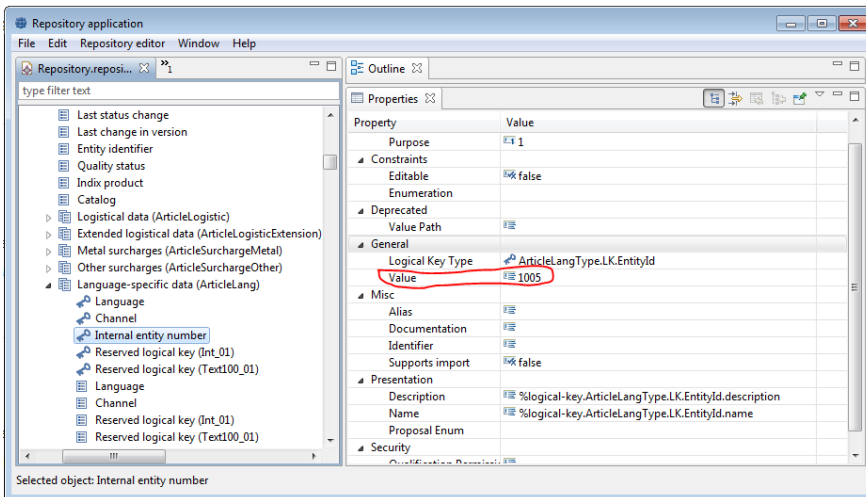
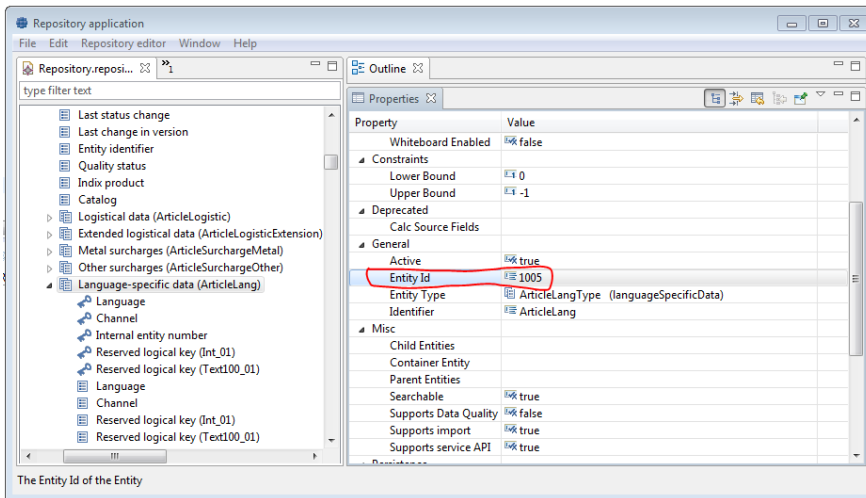
BEGIN
  EXECUTE IMMEDIATE
    'UPDATE (SELECT "ArticleExtension"."EntityID"
              FROM "ArticleExtension"
              INNER JOIN "ArticleRevision"
                ON "ArticleExtension"."ArticleRevisionID" = "ArticleRevision"."ID"
              WHERE "ArticleRevision"."EntityID" = ' || ArticleEntityID ||
/* only not converted values */
' AND "ArticleExtension"."EntityID" = 0 ' ||
/* TODO set specific values for logical key columns */
' AND "ArticleExtension"."ExtensionType" IN ('Extension1_a', 'Ext
ension1_b'))
              AND "ArticleExtension"."BuyerID" in (1, 2, 3)
              AND "ArticleExtension"."LanguageID" < 0
              AND "ArticleExtension"."Territory" IN ('WORLD', 'US') )
    SET "EntityID" = ' || ArticleExtensionEntityID;

END;

```

4.7.2 Repository

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



4.8 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
```

4.9 Data Quality

4.9.1 Migration

4.9.1.1 Rule mapplets, configurations and dictionaries

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.

4.9.2 Update

4.9.2.1 Data Quality SDK and Engine Update



Warning

Starting with version 10.1.0.02 the IDQ SDK and engine have been upgraded to version 10.5. This requires some check actions to ensure that server is starting properly and Data Quality features can be used further.

Two things to check:

1. Only if Microsoft Server is used: install MSVC Redistributable as described in the "Operating system - Data Quality" section of the Step by Step installation guide.
2. Ensure that the library paths are updated to the new IDQ version in the according wrapper.conf file

a. For Windows :

```
set.PIM_IDQ_BIN_PATH = %PIM_SERVER_PATH%/plugins/
com.informatica.sdk.dq.win32.x86_64_10.5.0/os/win32/x86_64
set.PIM_IDQ_LIB_PATH = %PIM_SERVER_PATH%/plugins/
com.informatica.sdk.dq_10.5.0/lib
```

b. For Linux:

```
set.PIM_IDQ_BIN_PATH = %PIM_SERVER_PATH%/plugins/
com.informatica.sdk.dq.linux.x86_64_10.5.0/os/linux/x86_64
```

```
set.PIM_IDQ_LIB_PATH = %PIM_SERVER_PATH%/plugins/
com.informatica.sdk.dq_10.5.0/lib
```

4.9.2.2 Rule mapplets, configurations and dictionaries

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).

4.10 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.

i 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.

i **Affected versions**

This step has to be performed whenever the previous release version was lower than 10.0 and the new version is equal or greater than 10.0.

Upgrades from versions greater than 10.0 to any newer version are not affected by this.

4.11 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.

i **Affected versions**

This step has to be performed whenever the previous release version was lower than 10.0 and the new version is equal or greater than 10.0.

Upgrades from versions greater than 10.0 to any newer version are not affected by this.

4.12 Media Manager Integration

4.12.1 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.

4.13 Server and Desktop Pre-Migration Checklist

4.13.1 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. Therefore 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).

4.13.1.1 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*



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.

4.13.1.2 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)

4.13.1.3 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:

4.13.1.4 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 \(see page 29\)](#)).

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.

5 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.

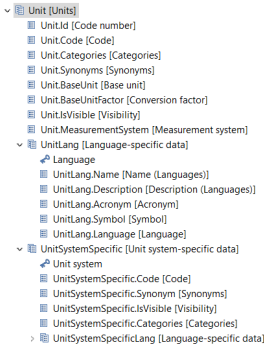
5.1 Overview

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

5.2 Repository

 <pre> Unit [Units] ├── UnitId [Code number] ├── UnitCode [Code] ├── UnitCategories [Categories] ├── UnitSynonyms [Synonyms] ├── UnitBaseUnit [Base unit] ├── UnitBaseUnitFactor [Conversion factor] ├── UnitIsVisible [Visibility] ├── UnitMeasurementSystem [Measurement system] ├── UnitLang [Language-specific data] ├── UnitSystemSpecific [Unit system-specific data] ├── UnitLang │ ├── UnitLangName [Name (Languages)] │ ├── UnitLangDescription [Description (Languages)] │ ├── UnitLangAcronym [Acronym] │ ├── UnitLangSymbol [Symbol] │ └── UnitLangLanguage [Language] └── UnitSystemSpecific ├── Unit system ├── UnitSystemSpecificCode [Code] ├── UnitSystemSpecificSynonym [Synonyms] ├── UnitSystemSpecificIsVisible [Visibility] ├── UnitSystemSpecificCategories [Categories] └── UnitSystemSpecificLang [Language-specific data] </pre>	<h3>New Fields</h3> <ul style="list-style-type: none"> Unit.Code which is now the primary, unique external identifier of the unit, independent of any unit system Unit.Categories is the field for the categorization of the unit, independent of any unit system Unit.Synonyms is the field for unit system independent synonyms of a unit <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>
---	--

5.3 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 "m i 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 "m i 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.

5.4 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]
  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]
  UnitSystem = 70
  UnitCategory = Area
  
```

5.5 Desktop UI

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

5.6 Import

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

5.7 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.

5.8 Service API

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

6 Media Manager Migration

- [Update Media Manager \(see page 46\)](#)
 - [Updating Media Manager File Server \(see page 46\)](#)
 - [Updating the database \(see page 46\)](#)
 - [Updating the client modules \(see page 46\)](#)
 - [Updating Funcd \(see page 47\)](#)
 - [Updating web front end \(see page 47\)](#)
 - [Deactivate old modules \(see page 48\)](#)

6.1 Update Media Manager

6.1.1 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.

6.1.2 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 migrate your database manually. Please refer to [Database Migration \(see page 13\)](#) > [Media Manager Database Migration \(see page 13\)](#) 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 \(see page 13\)](#) > [Media Manager Database Migration \(see page 13\)](#) for required update steps as well.

6.1.3 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.

6.1.3.1 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.


6.1.3.2 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.

6.1.4 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.

6.1.5 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.

6.1.6 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 >= 8.0. Therefore these services can be deactivated.

6.1.6.1 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.

6.1.6.2 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.

7 Supplier Portal Migration

7.1 Software upgrade (10.5.02.01 onwards)

Software	Version
Java	17
Tomcat	9.0.82

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

7.2 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".

7.2.1 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.



Mandatory for ORACLE

Update database user rights for version management (for ORACLE only)

Starting from 10.5.02.01, existing customers(migrating from version lower than 10.5.02.01) using **ORACLE** database for Supplier Portal need to run the **setupToUpdateOracle.cmd** (*setupToUpdateOracle.sh for linux*) in the **<INSTALLATION ROOT>/database/** before starting Supplier Portal service to upgrade version management system.

7.3 Supplier Portal - Product 360 Integration

Mandatory changes

To migrate to 10.5.02.01, you need to have finished the following steps. It is mandatory to create/edit these 2 user groups in Product 360, before working on Product 360 Supplier Portal.

7.3.1 Setup Product 360 Core User groups and Permissions

There will be change in the approach of creating users and user groups for interaction between Product 360 Supplier Portal and Product 360 Core.

There will be no need to create service account users as per the parameters `web.client.hsx.supplier.login` and `web.client.hsx.readonly.supplier.login` from the `webfrondend.properties`. Having one service account, for all suppliers, would make audit of supplier catalog data in the system irrelevant. Moving forward, there would be a PIM user created for every Supplier Administrator, Supplier user and Broker user. These users would be auto-assigned to user groups on PIM, based on the access level of the **SUPPLIER** on it's catalogs(provided in P360 Supplier Portal).

We will continue to have 2 user groups, one each for Item Viewer and Item Editor. There are new properties added for this purpose in the `webfrondend.properties`.

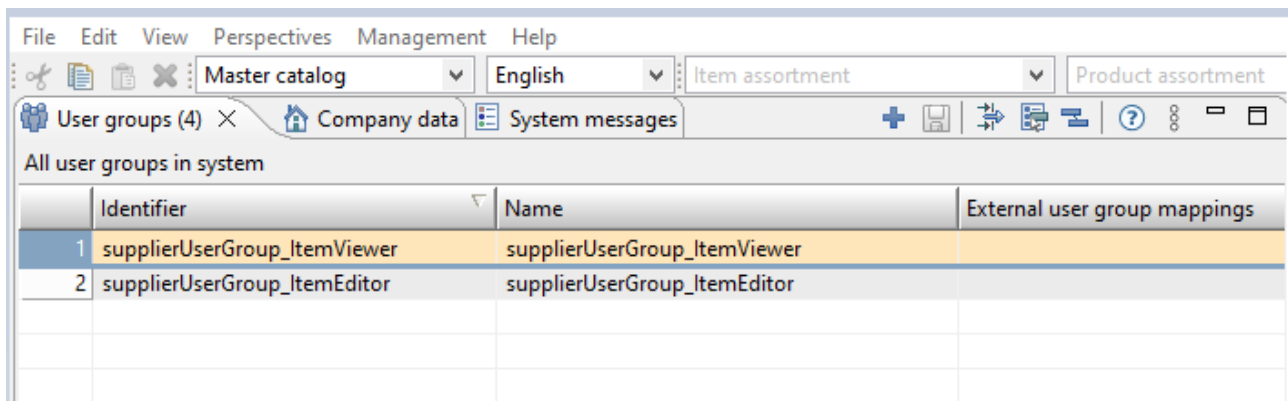
Product 360 Supplier Portal Item Editor User Group:

which means, users in this group are able to edit items within the Product 360 - Supplier Portal.
`web.client.hsx.itemEditor.userGroup = supplierUserGroup_ItemEditor` (default value)

Product 360 Supplier Portal Item Viewer User Group:

which means, users don't have the ability to edit item data within the Product 360 - Supplier Portal.
`web.client.hsx.itemViewer.userGroup=supplierUserGroup_ItemViewer` (default value)

Please ensure that the user group identifiers and the property values for `web.client.hsx.itemEditor.userGroup` and `web.client.hsx.itemViewer.userGroup` are identical



The screenshot shows the Informatica MDM User Groups interface. The top menu bar includes File, Edit, View, Perspectives, Management, and Help. Below the menu is a toolbar with icons for various actions. The main area displays a table titled 'All user groups in system' with the following columns: Identifier, Name, and External user group mappings. The table contains two rows of data.

	Identifier	Name	External user group mappings
1	supplierUserGroup_ItemViewer	supplierUserGroup_ItemViewer	
2	supplierUserGroup_ItemEditor	supplierUserGroup_ItemEditor	

Default matrix for user group details as per the properties:

Property	Property value	Corresponding user group identifier	Corresponding user group name
<i>web.client.hsx.itemEditor.userGroup</i>	<i>supplierUserGroup_ItemEditor</i>	<i>supplierUserGroup_ItemEditor</i>	<i>supplierUserGroup_ItemEditor</i>
<i>web.client.hsx.itemViewer.userGroup</i>	<i>supplierUserGroup_ItemViewer</i>	<i>supplierUserGroup_ItemViewer</i>	<i>supplierUserGroup_ItemViewer</i>

7.3.1.1 Create/Edit Product 360 - Supplier Portal Item Editor User Group

1. If not already exists, create a new Product 360 Core User Group, which manages the Product 360 Supplier Portal Item Editor permission within Product 360 Core, **assigned** with the following rights.
2. If already exists The Product 360 Supplier Portal Item Editor Users Group need to be **assigned** the extra rights marked in bold:

Rights group	Permission	Note
Web Permissions	Log in (Web)	
Catalog	Supplier catalogs, general access	

Rights group	Permission	Note
Structures	Structures, general access	Needed since version 7.1.02, while supplier is able to classify in structure tree.
Structure groups	Structure groups, general access	Needed since version 7.1.02, while supplier is able to classify in structure tree.
Import	Perform import	
Items	Items, general access	
Items	Create items	
Items	Edit items	
Item search	Item search management, general access	
Tasks	Task management, general access	Needed since version 8.1.0.00 for viewing supplier tasks
Tasks	Edit tasks	Needed since version 8.1.0.00 for working on supplier tasks

3. at least the following **action rights** have to be **revoked**:

Rights group	Permission	Note
Web Permissions	Help (Web)	

Rights group	Permission	Note
Web Permissions	Change password (Web)	
Tasks	Create tasks	
Multimedia attachments	Add multimedia attachments	
Import	revoke all permissions	Especially revoking 'Perform import' is important. Otherwise the Supplier Portal upload process would be compromised.
Merge	Merge, general access	Needed since version 8.1, since it allows to perform a catalog merge from within the Web UI now.
Merge	Perform Merge	Needed since version 8.1, since it allows to perform a catalog merge from within the Web UI now.
Flexible UI	Access Flexible UI	Needed since version 8.1 which introduces the new permission to avoid access to tasks of other suppliers and to objects from catalogs of other suppliers and master catalog using Flexible UI.

- a. at least the following **interface visibility rights** have to be **revoked**:

Category	Type	Name
Item		Tab visibility: Item, References (PIM Web)

Category	Type	Name
Context		Context visibility: Entire Context selection area
Action	Web Action	Action visibility: Import

- b. Since Product 360 8.1 it is possible to allow the assignment of Supplier Organizations to tasks setup in the system. The Supplier Organizations that are configured to work with tasks can access them similarly as their general catalog data by the item editor integration. For this setup at least the following **field rights** setup should be considered:

Data range	Permission	Note
Tasks	Revoke (visible and editable) all field rights except the following: Start date Estimated start date Anticipated completion on Progress Completed on	These adjustments are needed since version 8.1 which enables the assignment of Supplier Organizations to tasks. Revoking the field rights will guarantee that a supplier user cannot change the general definitions of a task setup by you.

All other action rights and interface visibility rights not mentioned above, as well as all field rights have to be defined individually depending on the scenario and requirements of the project and the individual use case scenarios

7.3.1.2 Create Product 360 Supplier Portal Item Viewer User Group

- If not already exists, create a new Product 360 Core User Group, which manages the Product 360 Supplier Portal Item Viewer permission within Product 360 Core.
- The Product 360 Supplier Portal Item Viewer Users Group need at least the following **action rights assigned**:

Rights group	Permission	Note
Web Permissions	Log in (Web)	
Catalogs	Supplier catalogs, general access	
Structures	Structures, general access	Needed since version 7.1.02, while supplier is able to classify in structure tree.
Structure groups	Structure groups, general access	Needed since version 7.1.02, while supplier is able to classify in structure tree.
Items	Items, general access	
Item search	Item search management, general access	

- c. at least the following action rights have to be **revoked**:

Rights group	Permission	Note
Rights group	Permission	Note
Web Permissions	Classify objects (Web)	
Web Permissions	Help (Web)	
Web Permissions	Change password (Web)	
Tasks	Create tasks	

Rights group	Permission	Note
Rights group	Permission	Note
Multimedia attachments	Add multimedia attachments	
Items	revoke all permission to edit, insert, delete or change items	
(Variants) only for 3 tier product paradigm installations	revoke all permission to edit, insert, delete or change variants	
Products	revoke all permission to edit, insert, delete or change products	
Import	revoke all permissions	Especially revoking 'Perform import' is important. Otherwise the Supplier Portal upload process would be compromised.

- d. at least the following **interface visibility rights** have to be **revoked**:

Category	Name
Item	Tab visibility: Item, References (PIM Web)
Context	Context visibility: Entire Context selection area

All other action rights and interface visibility rights not mentioned above, as well as all field rights have to be defined individually depending on the scenario and requirements of the project and the individual use case scenarios

7.3.1.3 Create/Edit Product 360 Supplier Portal Administrator Users Group

Create if not already existing, the Product 360 Supplier Portal Users Group and **assign** at least the following action rights to perform the basic actions in Supplier Portal web application. If the group already exists, assign the additional rights marked in bold

Rights group	Permission	Mandatory	Note
Catalogs	Supplier catalogs, general access	Yes	
General	Service Login	Yes	
Company Management	Company Management, general access	Yes	
Items	Items, general access	Yes	
Items	Create Items	Yes	
Items	Create Prices	Yes	
Items	Create Prices (in the past)	Yes	
Items	Delete item	Yes	
Items	Delete prices	Yes	
Items	Delete prices (in the past)	Yes	
Items	Edit items	Yes	
Items	Edit prices	Yes	

Rights group	Permission	Mandatory	Note
Items	Edit prices (in the past)	Yes	
Items	View prices	Yes	
Import	Perform import	Yes	
Suppliers	Supplier Management, general access	Yes	
Suppliers	Edit suppliers	Yes	
Structures	Structures, general access	Yes	
Structure groups	Structure groups, general access	Yes	
Users	Users, general access	Yes	
Users	Create user	Yes	
Users	Edit user	Yes	

Following **field rights** have to be defined at least as defined below.

Data range	Permission
Partner	<p>As basis grant visible and editable permissions to all fields.</p> <p>For following fields it is not mandatory to grant permissions:</p> <p>Object rights</p> <p>All fields of the field group Change Information</p>

7.3.1.4 Example

Following is a chart to illustrate how user, users-groups and audit of data would work moving forward

Supplier in Supplier Portal	Supplier's catalog access level	User under the supplier	User created in Product 360	User group in Product 360	User triggering import in Product 360 (could be for creating items as well as editing items)
Supplier A	<ul style="list-style-type: none"> Online viewing of catalog enabled Accessing catalog data online disabled 	user1@supplierA.com	user1@supplierA.com	supplierUserGroup_ItemViewer	REST user
Supplier B	Online editing of catalog enabled	user1@supplierB.com	user1@supplierB.com	supplierUserGroup_ItemEditor	user1@supplierB.com

8 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 \(see page 61\)](#)
- [Migration of Items of All-Supplier-Catalog index \(see page 67\)](#)
- [Migration of 3PPD index \(see page 69\)](#)
- [Migration of 2PPD index \(see page 73\)](#)

8.1 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.

8.1.1 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
```

8.1.2 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

Export format template properties

Items-only
You can create and configure the variables to be used by this format template here.

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

Variables:

- catalog
- language
- last modified date

Name: catalog

Note:

Data type: Catalog
A catalog, i.e. a supplier catalog or the master catalog

Value:

☒ Mandatory value ☒ Editable

New... Delete

OK Cancel

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

Export format template properties

Items-only
You can create and configure the variables to be used by this format template here.

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

Variables:

- catalog
- language
- last modified date

Name: last modified date

Note:

Data type: Date and time
A date and the time The default value is the current date. The input format is 5/28/2020 3:17 AM or 2020-05-28 03:17

Value: 1/1/1900 12:00 AM

☒ Mandatory value ☒ Editable

New... Delete

OK Cancel

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

Export format template properties

Items-only

You can create and configure the variables to be used by this format template here.

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

Variables:

- language
- last modified date
- last modified time

Name: language

Note:

Data type: Language

A language The default value is the login language for the current user.

Value:

☒ Mandatory value ☒ Editable

New... Delete

OK Cancel

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

Export format template properties

Items-only

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
- Deleted items

Name: Changed and new items

Parameter: catalog

☒ Mandatory value ☐ Editable

Assortment: last modified date

☐ Mandatory value ☐ Editable

Reference date: last modified date

☒ Mandatory value ☐ Editable

Type of change: New and changed items

☒ Mandatory value ☐ Editable

Update assortment: Working version

☐ Mandatory value ☐ Editable

Version: Working version

☒ Mandatory value ☐ Editable

New... Delete

OK Cancel

- *Deleted items* :

Export format template properties

Items-only

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

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

Data sources:

- Changed and new items
- Deleted items

Name: Deleted items

Parameter: catalog

☒ Mandatory value ☐ Editable

Date: last modified date

☐ Mandatory value ☐ Editable

Version: Working version

☒ Mandatory value ☐ Editable

New... Delete

OK Cancel

- There will always be a module which writes its 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}}
    }
  }
}
```

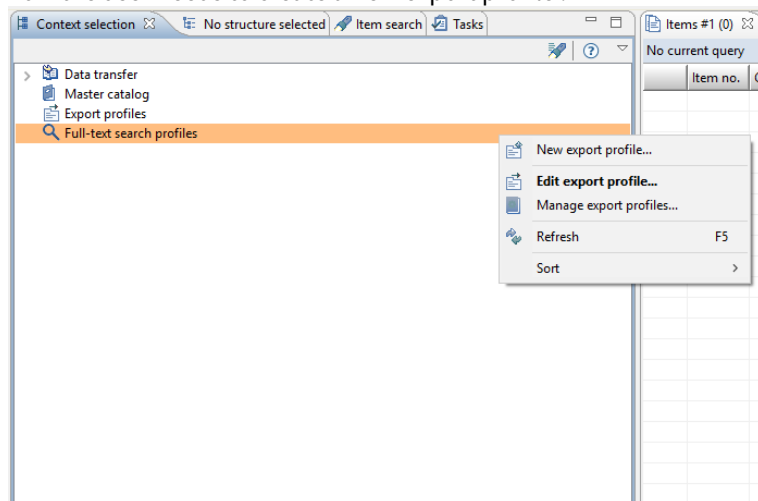

- Then the actual data part that is required to indexed. It is important that there is a strict mapping between the fields defined here and in config. **Fields Acl and Entity are the important ones, without which indexing will fail.**

```
{?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.

8.1.2.1 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}}}
```

```

    }
  }
}

```

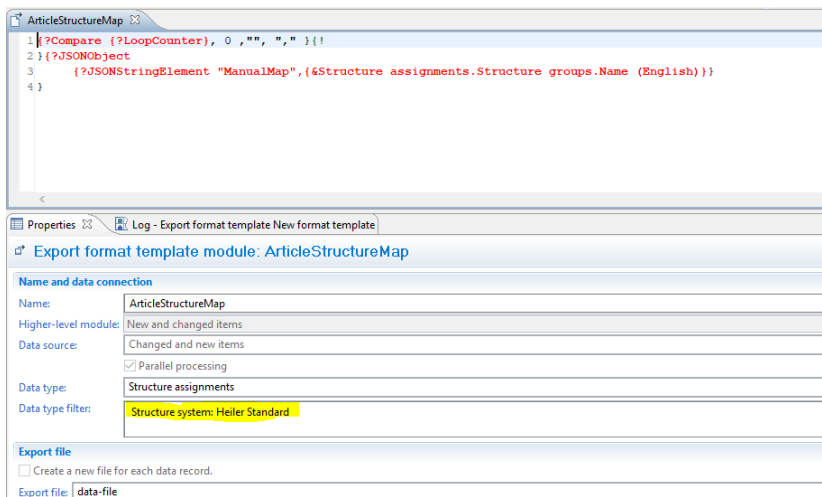
ArticleStructureMap sub-module will have following content (Note that this sub-module is qualified with the qualification provided in the config file):

```

{?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.



8.2 Migration of Items of All-Supplier-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.

Define a template similar to [Migration of Items of any Catalog index](#) (see page 61). The only difference is the data sources and catalog data type.

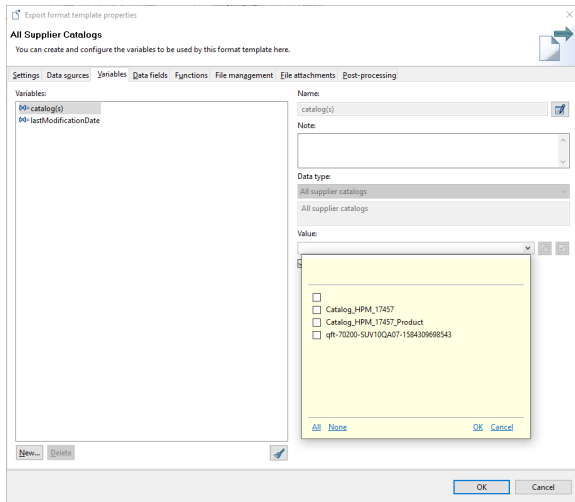
8.2.1 Configurations prior to v10.0.0.0, this would be defined as:

catalogs=Catalog1, Catalog2, Catalog3 and Catalog4

8.2.2 New Configuration

8.2.2.1 Data Type

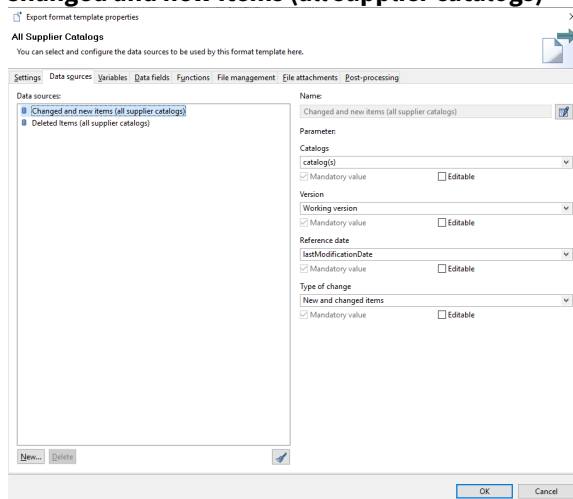
There is a new Data Type introduced which helps in creating a variable that helps in selection of multiple catalog(s). Data type is called as **All supplier catalogs**.



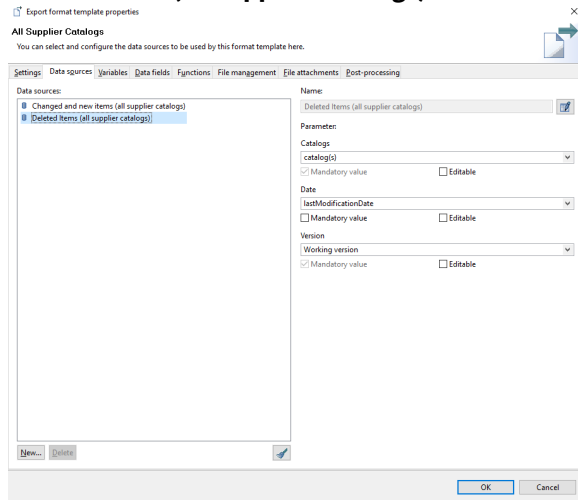
8.2.3 Data Sources

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

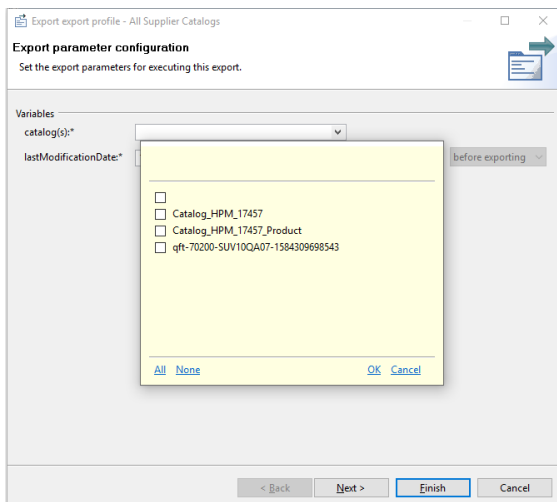
- **Changed and new items (all supplier catalogs)**



- Deleted items (all supplier catalogs)



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



8.3 Migration of 3PPD index

- Old configuration (see page 70)
- New configuration (see page 70)
 - Exporting data for multi-PPD index (see page 70)
 - Identifying the parent (see page 71)
 - Ensuring export of single record exports it's whole hierarchy (see page 71)
 - Product2G (see page 72)
 - Variant (see page 72)
 - Article (see page 72)
 - Migration of 3PPD template from v10.0.0.0 (see page 72)



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

8.3.1 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
```

8.3.2 New configuration

8.3.2.1 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 61) 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.

8.3.2.2 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.

8.3.2.3 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.

8.3.2.4 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.

8.4 Migration of 2PPD index

- [Old Search Configuration](#) (see page 73)
- [New configuration](#) (see page 74)
- [Migration of 2PPD template from v10.0.0.0](#) (see page 74)



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 69) before proceeding.

8.4.1 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
```

8.4.2 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.

8.4.3 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.

9 Business Process Management

9.1 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 BPM Queue

9.2 Workflows for Product 360 versions $\geq 8.0.00$

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).

9.3 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.

10 Audit Trail Migration

The Audit Trail migration consists of two parts, software migration and data migration.

10.1 Pre-Migration Checklist

- P360 Server and Desktop are updated to the latest version
- Legacy audit trail database is still available and preparation DB script has been executed on this audit trail database
- The repository is completely configured for the new audit trail

10.2 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.

10.3 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.

10.3.1 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 10,000 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.

10.3.2 Database preparation

There are update scripts for Oracle and MS SQL Server database to speed up the migration. Scripts are available in the configuration folder of the audit trail migration: `<PIM_SERVER_INSTALLATION_ROOT> \configuration\HPM\audittrail\migration\dbPreparationScripts`

The corresponding script has to be executed on the legacy audit trail database before migration can be started. During the script execution the unnecessary audit trail indexes will be deleted and new optimized index(es) will be created. Please note that this can take a few hours, depending on the database size.


Oracle_AdjustATForMigration.sql

SQLServer_AdjustATForMigration.sql

10.3.3 Configuration

10.3.3.1 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.

In addition to the root entity audit trail settings, the "Supports Audit Trail" setting for repository fields will also be checked. Only audit trail enabled fields and logical keys will be migrated.

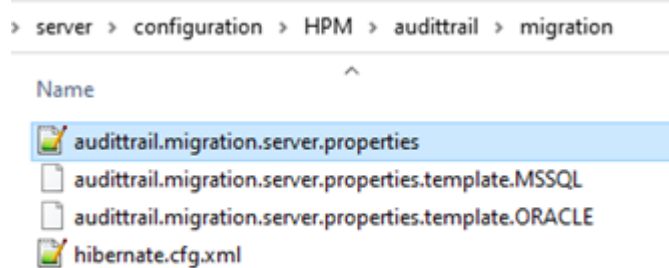
Furthermore, the mandatory attribute "short identifier" must have a valid value for all active entities, logical keys and fields. You can check your repository by starting a validation in the repository editor.

10.3.3.2 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

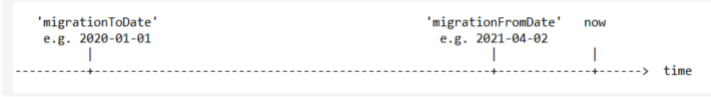
...

### 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
```

10.3.3.3 plugin_customization.ini

In order to use the migration you have to configure the time period for which you want to migrate changes. In addition to that, you can configure system resources to be used. The affected configuration file is

<PIM_SERVER_INSTALLATION_ROOT>\configuration\HPM\plugin_customization.ini .

Property	Description
<p>com.heiler.ppm.audittrail.migration.server/migrationToDate</p> <p>com.heiler.ppm.audittrail.migration.server/migrationFromDate</p>	<p>Both dates define the time span for which legacy audit trail data is to be migrated:</p> <ul style="list-style-type: none"> all data created on or after migrationToDate and on or before migrationFromDate will be migrated all data older than migrationToDate will NOT be migrated all data newer than migrationFromDate will not be migrated <p>migrationToDate is mandatory, migrationFromDate is optional, "today" is used as default value</p> 
com.heiler.ppm.audittrail.migration.server/locale	<p>Locale to be used for the audit trail migration job.</p> <p>The locale is needed to resolve labels of the objects to be migrated. When the data was written, the locale configured for the attribute <code>audittrail.atcsbuilder.locale</code> in the <code>audittrail.properties</code> file was used. The same value should be used for the data migration.</p> <p>Default value: en_US</p>
com.heiler.ppm.audittrail.migration.server/migration.fetch.size	<p>Fetch size used for database read access when retrieving migration data from the legacy audit trail database. In case of memory issues during the migration, this value should be decreased.</p> <p>Maximum value: database-specific value of <code>db.default.rowPrefetchSize</code> defined in <code>server.properties</code></p> <p>Default value: 10000</p>

Property	Description
com.heiler.ppm.audittrail.migration.server/migration.maxThreads	<p>Number of threads used to retrieve and process migration data from the legacy audit trail database.</p> <p>Dependent on the environment the migration is running, this number can be increased up to the number of CPU cores of the database server or, in case of running the migration in parallel with normal operation of the P360 system in production, it could be decreased.</p> <p>Default value: one half of the value for <code>db.available.cpu</code> configured in <code>server.properties</code></p> <p>Minimum value is 1, maximum value is number of CPU cores of the database server</p>

10.3.3.4 migration_template.json

In addition to the audit trail index policies and templates used for Elasticsearch in

`<PIM_SERVER_INSTALLATION_ROOT>\configuration\HPM\audittrail`, there is a similar template that is used for audit trail migration indexes, `migration_template.json`. For details, please refer to the Audit Trail Configuration documentation, chapter "Index policies and templates".

10.3.4 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"

Grouping	Allowed	Permission	Rights group	Description
1	<input checked="" type="checkbox"/>	Audit Trail, show deleted meta informati...	Audit Trail	Permission to show deleted meta information i
2	<input checked="" type="checkbox"/>	Audit Trail, general access	Audit Trail	Permission for general audit trail access
3	<input checked="" type="checkbox"/>	Audit Trail, migration	Audit Trail	Permission to start audit trail migration job
4	<input checked="" type="checkbox"/>	Audit Trail, show users	Audit Trail	Permission to show user names in audit trail
5	<input checked="" type="checkbox"/>	Audit Trail, search for users	Audit Trail	Permission to search for users in audit trail

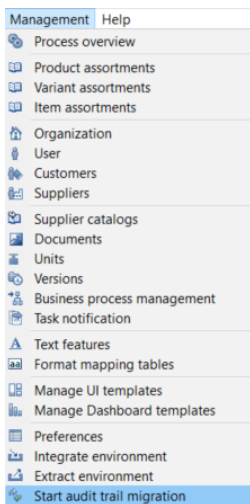
Other rights are not considered in the migration process.

10.3.5 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".

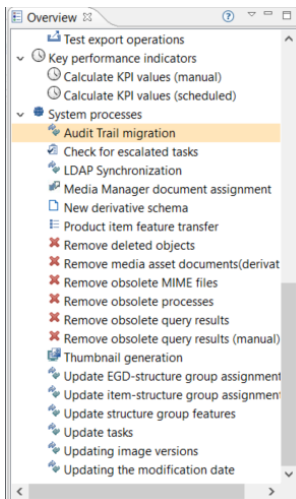
If the entry is missing, the user probably does not have the action right.

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.

10.3.6 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.

10.4 FAQ

10.4.1 Which is the best way to migrate the data?

The audit trail migration is a process that will take several days. Therefore, you should consider in advance what options you have. In general, there are two ways the migration can be done.

10.4.1.1 Parallel to normal P360 server operation

The first option is to start the migration after the P360 system has been updated. Because the migration first processes the latest data and then goes backwards in time, the data most likely to be needed is available almost immediately. It is important to be aware that not all of the system resources should be used for the migration, otherwise the P360 server will not run smoothly for your everyday tasks. To limit the system resources used, you can configure the values for the attribute `migration.maxThreads` and `migration.fetch.size`, see [plugin_customization.ini](#) (see page 77).

Alternatively, for a multi-server system, one of the job servers could be used exclusively for the migration. Please have a look at the priority job server feature (available since 10.1.0.00.05). With this feature you can define certain jobs as priority jobs, so those will only be executed on the priority job servers.

10.4.1.2 In advance of system upgrade

The second option is to do the migration in advance of the system upgrade. For this purpose, a separate P360 system (including a copy of the database) is needed, for example the QA or test system. As the migration would be the only process running on that system, you can use all available system resources for it, see [plugin_customization.ini](#) (see page 77).

First step

First, you have to configure a start date, which must be a day in the past, e.g., yesterday. In this step, the new Elasticsearch indexes will be created and all data from yesterday until the `migrationToDate` back in the past will be migrated.

Second step

After the update of the system, the second, much smaller part of the data will be migrated using the previously created audit trail indexes. For this, the migration period must be configured correctly, which is now the current time back to the day after the start of the first part.

The assumption here is that the second step is done on a different system than you used for the first step. If you want to do both steps on the same system, you need to remove the progress information of the first step. Remove the `progress.json` file from the subfolder `audittrailMigration` of the configured shared folder.

Example

If you want, for example, to migrate all data from the first of January 2020 until now, your first step would use

```
migrationFromDate = 2021-04-13 (this may be yesterday or another day in the past)
migrationToDate = 2020-01-01 (this is the date of the oldest data to migrate, all older legacy audit trail data will not be migrated).
```

The second step covers the day after 13rd of April until now, you would use

```
migrationToDate = 2021-04-14
migrationFromDate=<empty>
```

10.4.2 How long will the migration take?

An assumption on the duration can be made based on following data

- number of root entity changes to be migrated totally => **root_entities_total**
- number of root entity changes to be migrated in a time frame => **root_entities_in_time_frame**
- duration of migration in the same time frame => **time_for_migration_in_time_frame**

You can determine these numbers by executing statements on the customer database and observing the migration progress on the customer system.

10.4.2.1 Step 1

Determine the number of root entity changes to be migrated totally and the number of root entity changes to be migrated in a time frame (e.g. first month). Use following statements:

Statements to determine the number of root entity changes

```
-- MS SQL
---- root_entities_total
SELECT COUNT(1) root_entities_total
```

```

FROM EntityChange
WHERE ParentEntityChangeID IS NULL

---- root_entities_in_time_frame
SELECT COUNT(1) as root_entities_in_time_frame
FROM EntityChange
WHERE ModificationDate < '2020-09-19 00:00:00'
AND ModificationDate > '2020-08-19 00:00:00'
AND ParentEntityChangeID IS NULL

-- Oracle
---- root_entities_total
SELECT COUNT(1)
FROM "EntityChange"
WHERE "ParentEntityChangeID" IS NULL

---- root_entities_in_time_frame
SELECT COUNT(1)
FROM "EntityChange"
WHERE "ModificationDate" < TO_TIMESTAMP('2020-09-19 00:00:00', 'yyyy-mm-dd
HH24:MI:SS')
AND "ModificationDate" > TO_TIMESTAMP('2020-08-19 00:00:00', 'yyyy-mm-dd
HH24:MI:SS')
AND "ParentEntityChangeID" IS NULL

```

10.4.2.2 Step 2

Analyze the migration protocol for the same time frame as in SQL statement above and determine the duration of the migration process.

In our example it took 133 minutes.

Current and past processes (12)						
Audit trail migration						
No.	User name	Date	Last change	Scheduled for	Step	Status
15,051	infalser	11/18/2020 3:13 PM	11/18/2020 3:26 PM	11/18/2020 3:13 PM		Completed
1 element selected						
Log - Log entries: AudittrailMigration_15051614_15051614 (33) 32 Quality check						
Log for:						
Status	Category	Date	ID	Date	Time	Message
1	Summary	2020-09-18		11/18/2020	3:13 PM	Migrationprozess gestartet, konfiguriertes Mig
2	Summary	2020-09-17		11/18/2020	3:29 PM	Verarbeite Tag: 2020-09-17, erfolgreich migrie
3	Summary	2020-09-16		11/18/2020	3:29 PM	Verarbeite Tag: 2020-09-16, erfolgreich migrie
4	Summary	2020-09-15		11/18/2020	3:34 PM	Verarbeite Tag: 2020-09-15, erfolgreich migrie
5	Summary	2020-09-14		11/18/2020	3:38 PM	Verarbeite Tag: 2020-09-14, erfolgreich migrie
6	Summary	2020-09-13		11/18/2020	3:44 PM	Verarbeite Tag: 2020-09-13, erfolgreich migrie
7	Summary	2020-09-12		11/18/2020	3:52 PM	Verarbeite Tag: 2020-09-12, erfolgreich migrie
8	Summary	2020-09-11		11/18/2020	4:03 PM	Verarbeite Tag: 2020-09-11, erfolgreich migrie
9	Summary	2020-09-10		11/18/2020	4:13 PM	Verarbeite Tag: 2020-09-10, erfolgreich migrie
10	Summary	2020-09-09		11/18/2020	4:22 PM	Verarbeite Tag: 2020-09-09, erfolgreich migrie
11	Summary	2020-09-08		11/18/2020	4:29 PM	Verarbeite Tag: 2020-09-08, erfolgreich migrie
12	Summary	2020-09-07		11/18/2020	4:30 PM	Verarbeite Tag: 2020-09-07, erfolgreich migrie
13	Summary	2020-09-06		11/18/2020	4:31 PM	Verarbeite Tag: 2020-09-06, erfolgreich migrie
14	Summary	2020-09-05		11/18/2020	4:31 PM	Verarbeite Tag: 2020-09-05, erfolgreich migrie
15	Summary	2020-09-04		11/18/2020	4:35 PM	Verarbeite Tag: 2020-09-04, erfolgreich migrie
16	Summary	2020-09-03		11/18/2020	4:38 PM	Verarbeite Tag: 2020-09-03, erfolgreich migrie
17	Summary	2020-09-02		11/18/2020	4:39 PM	Verarbeite Tag: 2020-09-02, erfolgreich migrie
18	Summary	2020-09-01		11/18/2020	4:44 PM	Verarbeite Tag: 2020-09-01, erfolgreich migrie
19	Summary	2020-08-31		11/18/2020	4:52 PM	Verarbeite Tag: 2020-08-31, erfolgreich migrie
20	Summary	2020-08-30		11/18/2020	4:53 PM	Verarbeite Tag: 2020-08-30, erfolgreich migrie
21	Summary	2020-08-29		11/18/2020	4:54 PM	Verarbeite Tag: 2020-08-29, erfolgreich migrie
22	Summary	2020-08-28		11/18/2020	4:56 PM	Verarbeite Tag: 2020-08-28, erfolgreich migrie
23	Summary	2020-08-27		11/18/2020	5:04 PM	Verarbeite Tag: 2020-08-27, erfolgreich migrie
24	Summary	2020-08-26		11/18/2020	5:06 PM	Verarbeite Tag: 2020-08-26, erfolgreich migrie
25	Summary	2020-08-25		11/18/2020	5:07 PM	Verarbeite Tag: 2020-08-25, erfolgreich migrie
26	Summary	2020-08-24		11/18/2020	5:07 PM	Verarbeite Tag: 2020-08-24, erfolgreich migrie
27	Summary	2020-08-23		11/18/2020	5:09 PM	Verarbeite Tag: 2020-08-23, erfolgreich migrie
28	Summary	2020-08-22		11/18/2020	5:11 PM	Verarbeite Tag: 2020-08-22, erfolgreich migrie
29	Summary	2020-08-21		11/18/2020	5:16 PM	Verarbeite Tag: 2020-08-21, erfolgreich migrie
30	Summary	2020-08-20		11/18/2020	5:18 PM	Verarbeite Tag: 2020-08-20, erfolgreich migrie
31	Summary	2020-08-19		11/18/2020	5:26 PM	Verarbeite Tag: 2020-08-19, erfolgreich migrie
32	Summary	2020-08-19		11/18/2020	5:26 PM	Verarbeite Tag: 2020-08-19, erfolgreich migrie

10.4.2.3 Step 3

Determine the total duration using the following formula

$$\text{time_total} = \text{root_entities_total} * \text{time_for_migration_in_time_frame} / \text{root_entities_in_time_frame}$$

10.4.2.4 Example

In our example migration we determined following numbers:

Changes on	Root entities	Duration	Space on disk (DB)	Space on disk (Elasticsearch index)
1 month 2020-08-19 to 2020-09-18	1,760,830	133 min	n/a	~ 1 GB
Total	295,152,230	~ 15 days	2.26 TB	~170 GB



There are database preparation scripts and some configuration settings available to adjust the performance and as a result the duration of the migration process. Please read the whole migration documentation and use these possibilities to adjust the migration to your needs.

10.4.3 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 themselves will not be migrated. Read more on [how to activate the MediaAsset entity to support Audit Trail \(see page 93\)](#) and migration of records.
- 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 < '2020-10-01 00:00:00' and ModificationDate >
6      '2020-09-30 00:00:00'
7      AND ParentEntityChangeID is null
8  GROUP BY EntityName
9
10 -- (Oracle) list all entities and the corresponding record count for one
11 day, e.g. 2020-10-14
12     SELECT "EntityName", count("EntityName")
13     FROM "EntityChange"
14     WHERE "ModificationDate" < TO_TIMESTAMP('2020-10-15 00:00:00', 'yyyy-
mm-dd HH24:MI:SS')
    AND "ModificationDate" > TO_TIMESTAMP('2020-10-14 00:00:00', 'yyyy-
mm-dd HH24:MI:SS')
    AND "ParentEntityChangeID" is null
    GROUP BY "EntityName"

```

10.4.4 Has really all my data been migrated?

Our first goal in migrating the audit trail records was not to lose any data. This is not always easy because legacy audit trail data can be very different depending on the repository configuration. Sometimes they may be incomplete, or changes may have been made to the repository so that individual fields or logical keys no longer exist. For other change records the used enumeration values might not exist anymore.

So we had to consider many different cases and maybe there are some data constellations for which the displayed migrated audit trail data does not look as nice as for changes logged with the new audit trail.

But if you take a closer look at the data you will see that no information has been lost.

Examples:

- Sometimes it can happen, that values for the qualification are not available, then we try to find the corresponding field values. In this case the data is displayed like this:

▼ Sabine Beutler Thursday, 11/26/2020 7:29:38 PM (6 days ago) Initiated by Migration Created		
Field	Old value	New value
▶ Header data		
▼ Logistics		
Extended logistical data (Gemüselieferant, Language independent)	unknown (Packing unit)	
Code128	The "Packing unit" qualification value could not be resolved during migration but the corresponding field value could	128
Packing unit		four pack

- If it was not possible to resolve a logical key during migration, it will be put with its original name to the change document. In case of such non-resolved logical keys, we show the name of the logical key in addition to its value:

<div> Sabine Beutler <div> Thursday, 11/26/2020 7:29:38 PM (6 days ago) </div> <div> Initiated by Migration <div>Created</div> </div> </div>		
Field	Old value	New value
▶ Header data		
▼ Logistics		
▼ Extended logistical data (Gemüselieferant, Language independent, unknown (Packing unit))		
Code128		128
Packing unit		four pack
▼ Extended logistical data (Language independent, Holzlieferant (Supplier), unknown (Packing unit))		
Code39		39
Packing unit		box
▼ Logistical data (Germany)		
Packing unit		bag

10.5 Troubleshooting

10.5.1 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. Two things are required to bring your system back to the starting point: delete the Elastic migration indexes and clean up the progress information.

10.5.1.1 Delete migrated data

In order to remove the already migrated entries from elastic you have to delete the corresponding 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` file.

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

```
DELETE /<index>
```

Related CURL commands are:

```
curl -X DELETE server:port/<index>
```

To find out related indices in Elasticsearch you can use:

```
curl -X GET "server:port/_cat/indices?pretty" -u user
```

You have to replace: server, port and user with Elasticsearch URL and credentials.

For further information see Elasticsearch Delete index API.



Don't remove any of the index templates associated with Product 360

10.5.1.2 Delete progress information

Finally, remove the `progress.json` file from the subfolder `audittrailMigration` of the configured shared folder and you can try again. The P360 server doesn't need to be restarted.

10.5.2 Increase trace level

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

Enable the logging of migration warnings 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" />
```

10.5.3 Configure system resources used

10.5.3.1 Decreasing the memory usage

There is a possibility to decrease the memory consumption of the audit trail migration process by adjusting the value of `com.heiler.ppm.audittrail.migration.server/migration.fetch.size`, see [plugin_customization.ini](#) (see page 77).

Please note that the time of migration will increase, if you decrease the fetch size.

10.5.3.2 Decreasing the number of threads

Retrieving and processing migration data from the old audit trail database is the crucial part of the audit trail migration process, so all available CPU cores on the audit trail migration server are used by default. If the audit trail migration is running in parallel with a running P360 system on the same machine, the number of threads used by the migration must be limited to ensure fluid use of the P360 system.

This can be done by changing the property `com.heiler.ppm.audittrail.migration.server/migration.maxThreads`, see [plugin_customization.ini](#) (see page 77).

10.5.4 Elasticsearch exceptions

When working with Elasticsearch, you are likely to get errors from time to time. Elasticsearch provides detailed documentation, there is also a large community with a huge amount of knowledge. Therefore, while we briefly list the most common errors here, we strongly recommend that you read this documentation for further details and problems not listed here.

10.5.4.1 Adjustments of elastic settings (Exception: 'Unable to parse response body')

There can be huge change documents created and stored in Elasticsearch, e.g. during migration of changes made by import or merge.

To avoid `HTTP/1.1 413 Request Entity Too Large` errors that are logged to the migration protocol as "Exception during migration: Unable to parse response body" errors, add and adjust the setting `http.max_content_length` in the Elasticsearch configuration. The default value is 100MB which might not be enough for some migration cases.

Please note: Before you make any changes, you should always read the Elasticsearch Installation documentation.

If PIM runs in a hosted environment like AWS, an AWS ES limitation can cause such issue. e.g. `m5.large.elasticsearch` instance has a Maximum Size of HTTP Request Payloads of 10MB. That may be not enough for any audit trail migration request to Elasticsearch.

See 'Network limits' in <https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/aes-limits.html>

10.5.4.2 Elasticsearch Exception (type=circuit_breaking_exception)

If the audit trail migration stops with an Elasticsearch exception with

```
type=circuit_breaking_exception and reason=Data too large, data for
[<http_request>] ...
```

then the memory settings (-Xmx) in Elasticsearch `config/jvm.options` file is too low.

Please note, if the memory for Xmx is > 1GB and you are using JDK8 on windows, then you should use an odd number (1, 3, 5... GB) as value (<https://github.com/elastic/elasticsearch/issues/47384#issuecomment-543952442>).

10.5.4.3 Elasticsearch Exception (type=illegal_argument_exception)

If the audit trail migration stops with an Elasticsearch exception with `type=illegal_argument_exception` and `reason=all shards failed`, please look for any nested exceptions. If the nested exception is also an Elasticsearch exception with `type=search_phase_execution_exception`, please have a look to the section below.

10.5.4.4 Elasticsearch Exception (type=search_phase_execution_exception)

If the audit trail migration stops with an Elasticsearch exception with `type=search_phase_execution_exception` and `reason=Result window is too large`, from + size must be less than or equal to ... then the Elasticsearch setting `max_result_window` has to be increased.

The maximum number of search results for an index is limited to 10000 by default. We recommend to use 50000 for this property. But be careful, search requests take heap memory and time proportional to `max_result_window` value, so it should not be increased too much.

You can change the value for the corresponding index directly, for example using Kibana (Management → Index Management → Edit settings) or by using the update index settings API of Elasticsearch (see <https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-update-settings.html>).

10.5.5 Correct object identifier

If you have started the audit trail data migration with a version before 10.1.0.00.06, the object identifiers of migrated data are different from those of the data changed (by a user) in the new audit trail.

The screenshot shows the Informatica Product 360 interface. On the left is a navigation sidebar with icons for Structures, Catalogs, Media, Tasks, Queries, Search, Assortments, and Lookups. The main area is divided into a search filter section on the left and a search results section on the right.

Search filter section:

- Identifier:** `Article_8526274370611;Article_8526274370611'@MASTER'`
- Data type:** (empty)
- Change types:**
 - ☒ Created
 - ☒ Changed
 - ☒ Deleted
- Timeframe:** (empty)
- Initiator(s):** (empty)
- Search:** (button)

Search result (2) table:

	Data type	Identifier	Container	Event type	Date time
1	Item	Article_8526274370611	MASTER	Changed	Wednesday, 4/21/2021 5:26
2	Item	'Article_8526274370611'@MASTER'	MASTER	Changed	Tuesday, 4/20/2021 3:00:33

Below the table is a "Detail view" section with a placeholder for more information.

Most likely, rerunning the audit trail data migration is not an option. Therefore we provide a fix to update all affected object identifiers in the elasticsearch migration indexes. This update is optional.

We have integrated that update into the Database Setup Client. Note that this update will not be executed when you start a regular database update.

10.5.5.1 Preparation

During the database setup you specify a `server.properties` file. This usually is the one you use for your sever. However, if you want to update the object identifiers in the audit trail migration indexes, you have to prepare a special `server.properties` file. The best way is to copy the original `server.properties` file and adjust it. The following changes will cause the Database Setup Client to run in "single version only" mode, which means that only one database update will be processed. If it is finished successfully, the corresponding version will be written to the database so that the same update script will not run a second time. (As this update is optional, your server will startup without the update as well.)

Add the following lines to your copy of the `server.properties` file:

- For MS SQL Server

```
db.update.only.schema = MAIN
db.update.only.version = 794
```

- For Oracle

```
db.update.only.schema = MAIN
db.update.only.version = 795
```

10.5.5.2 Starting the database setup

Now you can start the Database Setup Client and choose the adjusted `server.properties` file.

10.5.5.3 Logging

By default the logging will only tell you that the update is executed.

You can get more information if you adjust the settings in the `log4j2.xml` file of the Database Setup Client by setting the log level to `DEBUG`:

```
<Logger name="com.heiler.ppm.dbsetup.core.auditTrailIdentifier" level="DEBUG" />
```

The log will contain the names of the indexes that were found, the number of processed documents and the task id of the task updating the documents in elasticsearch.

If you want to get more information about the progress of the update, you can activate the performance log:

```
<Logger name="com.heiler.ppm.dbsetup.core.PERF" level="DEBUG" />
```

In this mode, a log entry will be written every five minutes that provides information about the number of documents that have already been processed.

```
Index '<systemName>.audit_article_master_migrate', updated: 7529000/27830242, not
touched documents: 0, duration: 1:00:30.828
```

Whenever elasticsearch has finished processing a single index, you will see a summary in the log as well.

10.5.5.4 Further information

Please note that the duration of the update depends on the number of documents in the migration indexes. For our tests we found that 100000 documents take around one minute to be processed. The overall time of the update is determined by the biggest index because all indexes are processed in parallel.

Not touched documents mean these documents match our query for the update but couldn't be processed by the script to correct the identifier because it is something really unusual we cannot handle (e.g 'item'@'123'@'MASTER'). These identifiers won't be changed.

10.5.5.5 Technical details

For developers, there is a way to further enhance the log. Set `com.heiler.ppm.dbsetup.core.PERF` to log level TRACE.

For the index updates this will add the average duration for 100000 documents. For the finish summary of each index it will add the average duration of 100000 processed documents as well as the number of updated and not touched documents. Note that for technical reasons the number of updated documents in the finish summary is the number from the last update, not the total number of processed documents.

You can use the Kibana Console to execute the following REST request to see the status of the tasks in elasticsearch:

```
GET _tasks?actions=*byquery&detailed
```

10.5.6 Why does my data look strange?

Some customers might see migrated data in the History tab that looks different from data that has been changed by a user in the UI after the migration. This section should give you an idea why that happens and how you can fix it by configuring your repository differently.

Let's have a look at an example. This is the changed data when it has been changed in the UI in the new P360 version (without migration):

▾ Sabrina Ehlers Thursday, 10/28/2021 11:06:26 AM (21 hours ago) Initiated by User		
Field	Old value	New value
▾ NPA data		
↻ ▾ Design		
↻ Buyer Authorisation Indicator	Buyer Auth Ind 1	Buyer Auth Ind 2

This, however, is the the same changed field when it has been migrated:

Field	Old value	New value
▼ NPA data		
▼ Design (0 (Reserved Logical Key (Int_01)), 20026 (Internal Entity number), DEFAULT (Reserved Logical Key (Text_01)), DEFAULT (Reserved Logical Key (Text_02)), Default Channel (Channel Name), TME (Target Market))		
Buyer Authorisation Indicator		Buyer Auth Ind 1

The big difference is in the logical keys. In the first picture no logical keys or their values are shown. This is because they are all not editable. We would expect the same for the migrated data but in the migrated data we see the values for the logical keys followed by the display name of the respective logical key.

Why?

During the migration the data that's found in the old Audit Trail database is transformed in a json document. This json document contains a 'qualification' section that uses the short identifier to identify a logical key. Therefore the logical key (and its field) has to be found in the repository during migration. In this case the short identifier wasn't found. Instead the display name was written in the json document and the UI has no chance to find out information like visibility or editability and therefore simply shows what it found in the json document.

How can it be fixed?

It can be fixed by some changes or adjustments in the repository.

Add fields belonging to logical keys

Check if there is a field that belongs to the logical key and if not add it.

The short identifier is retrieved from the field that belongs to the logical key. If it doesn't exist, it won't work correctly.

In order to find out which field type your new field needs to have, find the logical key type in the types area and in the properties have a look at 'Field Type'. The logical key and the field must have the same short identifier.

After you added the fields, it may look something like this:

▼ Sabrina Ehlers Thursday, 10/28/2021 9:14:31 AM (23 hours ago) Initiated by Migration		
Field	Old value	New value
▼ NPA data		
▼ Design (20,026, TME, 0, DEFAULT, DEFAULT, DEFAULT, Default Channel (Channel Name))		
Buyer Authorisation Indicator		Buyer Auth Ind 1

Make values convertible

'Default Channel (Channel Name)' still doesn't look as expected. This is because the value couldn't be converted properly. The field does not have an enumeration. Setting an enumeration should fix this. If no enumeration is available but the value is an EntityProxy, make sure that the 'Proxy Transition Entity' is set.

Adjust visibility

The logical key values are still visible. This can be fixed with repository settings as well.

When is a logical key value visible? They are not visible when they haven't been migrated. This means they are not part of the json document in elastic search.

This is also defined by the field belonging to the logical key.

The logical key is skipped during the migration if

- the fields 'Supports Audit Trail' is set to false
- the fields short identifier is empty
- the field is set to visible = false, editable = false and the logical key is set to editable = false

Now the migrated data and changes that are made later look the same in the History tab.

10.6 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)

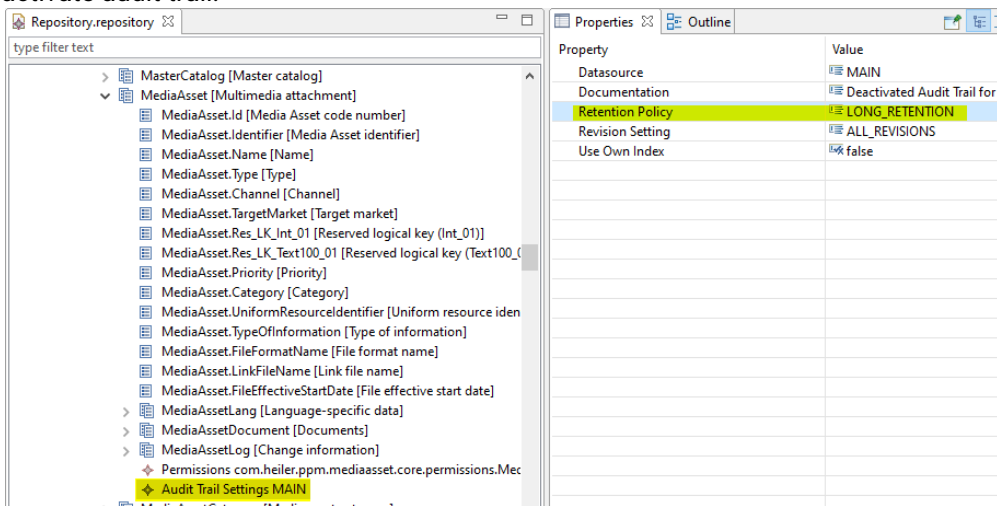
10.7 Migration of Audit Trail data for Media Assets

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.

However, in some cases depending on your business processes, it can be useful to activate the `MediaAsset` entity to support audit trail.

For example if the media assets "Auto path resolution by reference" functionality is in use on your system. In this case the audit trail logs the delayed creation of media asset derivatives only for media asset documents themselves and not for the object (e.g. item, structure group) to which this document belongs. So the audit trail data for creation of derivatives won't be migrated and shown in the change history if `MediaAsset` entity is not configured to support audit trail.

You only need to adjust the setting "Retention Policy" in the repository for the `MediaAsset` entity to activate audit trail.



If it is not necessary to log the change history for Media Assets separately and in its entirety on your system, you can reset this configuration to NO_RETENTION again.

11 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}
```

Copyright

© Copyright Informatica LLC 1993, 2024

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. 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.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.