DATA ANALYTICS

ARIS Process Performance Manager 10

October 2026

This document applies to ARIS Process Performance Manager Version 10 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2000-2026 SAG Aris GmbH, Saarbrücken, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name ARIS and all SAG Aris GmbH product names are either trademarks or registered trademarks of SAG Aris GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

The Software may contain or be distributed with certain commercial third party components (“Third Party Components”), which may be subject to special license terms and conditions (“Third Party Terms”) located at https://aris.com/terms.

Third Party Terms include important licensing and warranty information and disclaimers of third party licensors. For the avoidance of doubt, if Customer uses the Software in accordance with the Documentation, the Third Party Terms do not restrict the rights granted under this Agreement. In the event that a third party manufacturer no longer provides active support for any Third Party Component, Supplier will use commercially reasonable efforts to respond to any issues with the Software related to the Third Party Components.

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with SAG Aris GmbH and/or its subsidiaries and/or its affiliates and/or their licensors.

Table of Contents

1. General 5
2. Introduction 6
    2.1. Tables 6
    2.2. Associations 6
    2.3. Analysis realm 7
3. Configure Data analytics 8
    3.1. Open the Data analytics view of a client 8
    3.2. Create a data connection 9
    3.3. Create an analysis realm 9
    3.4. Configure an analysis realm 10
        3.4.1. Add tables 10
            3.4.1.1. Define keys 11
        3.4.2. Link tables 11
        3.4.3. Define partial data extraction 12
4. Configure Cross analytics 14
    4.1. Add process tables 14
5. Import analysis realms 16
    5.1. Overwrite analysis realm configuration 16
        5.1.1. Reinitialize analysis realm in analysis server 16
    5.2. Replace analysis realm configuration 17
    5.3. Import data 18
    5.4. Delete data 18
6. Delete table data 19
7. Appendix 20
    7.1. Query logic 20
        7.1.1. Query structure 20
        7.1.2. Data filtering 21
        7.1.3. Iteration 22
        7.1.4. Aggregation 24
        7.1.5. Combination of iteration and aggregation 24
        7.1.6. Cross analytics 26
    7.2. Restrictions and special cases 26
        7.2.1. Missing links 26
        7.2.2. Missing values 26
        7.2.3. Loops 27
        7.2.4. Self-reference 27
        7.2.5. Values without assignment 28
        7.2.6. Split text dimension data 29
        7.2.7. Time dimensions with different accuracies 29
    7.3. Directory and file structure 30
    7.4. Configuration versioning 31
    7.5. Data versioning 31
    7.6. Import and export of configurations 32
        7.6.1. Export mode 32
        7.6.2. Import mode 32
            7.6.2.1. -replace parameter 33
            7.6.2.2. -overwrite parameter 33
    7.7. Data import 34
        7.7.1. Persistence 34
        7.7.2. runtableimport.bat 35
        7.7.3. Formats supported 37
        7.7.4. Number ranges supported 39
        7.7.5. Troubleshooting after data import was canceled 39

7.8. Save and restore data 39
7.9. Delete analysis realm and table data 40
7.10. Reinitialize analysis server 41
7.10.1. Reinitialize an individual analysis realm 41
7.10.2. Reinitialize entire analysis server 41
7.10.3. Automatically reinitialize analysis realm 42
7.11. XML configuration of analysis realms and data import 42
7.11.1. Document type definition (DTD) 42
7.11.2. Configuration elements 44
7.11.2.1. tables 44
7.11.2.2. associations 45
7.11.2.3. criterion-definitions 45
7.11.2.4. Example 47
7.11.3. Data sources 49
7.11.4. XML event format 49
7.11.5. Data format transformation 51
7.11.5.1. datasource.dtd extension 51
7.11.5.2. Example: The 'timestamp' transformation type 52
7.11.6. Standard deviation and percentiles 52
8. Support and legal information 54
8.1. Documentation scope 54
8.2. Support 54
8.3. Legal notices 55

1. General

This manual describes the configuration of Data analytics and Cross analytics< in ARIS Process Performance Manager (PPM). It provides the PPM system administrator with basic knowledge and configuration know-how to support him in configuration for different usage scenarios and analysis tasks.

2. Introduction

Data analytics and Cross analytics extend PPM from version 9.0 by two new topical areas for analyses.

In addition to process, function, and interaction analyses, you can now evaluate process-independent data using Data analytics. Data analytics enables the analysis of comprehensive data in table format, consisting of multiple linked tables. In Data analytics, analysis criteria, that is, dimensions and measures, are based on the table structure of the data basis, with each table column representing an analysis criterion.

Cross analytics extends the Data analytics analysis area by connecting the analysis of table data in Data Analytics with process analysis. Existing process data from process analysis are linked with existing tables from Data analytics.

With the help of PPM Customizing Toolkit (PPM CTK), you can configure all settings required (Customizing [8]) for Data analytics and Cross analytics and import the relevant data into PPM [34].

Data analytics and Cross analytics basically consist of the following elements.

2.1. Tables

A table consists of one or more columns, each of which contains the data of a single criterion (that is, a dimension or measure). In the PPM user interface, the columns are displayed as individual dimensions or measures. When customizing, you define the columns to be available as criteria (dimension or measure) and their data types for analysis in PPM. Depending on the data types of the criteria available, various analyses can be run.

2.2. Associations

Data analytics is especially practicable for comprehensive ad-hoc analyses. To enable quick and consistent ad hoc analyses, the logical links of the individual tables must be specified during customizing. The description of a logical link is also called association.

Individual tables are linked based on specific criteria, the so-called association criteria. Association criteria in the tables to be linked must have the same data type. Except for floating point numbers, you can use all data types for criteria. Two tables are linked by two criteria, that is, one criterion per table. For proper assignment, the values of the individual criteria must be identical and must not exist in different granularities, except for time stamps [29].

Example 1.

A table named **Staff** contains the data of all employees of a company, such as name, education, salary, etc. Another table called **Departments** contains the data of the company's individual departments, for example, department name, location, staff, etc. This means that each employee of the company is assigned to a specific department.

For the individual tables to be linked and the data to be assigned, both tables must each have a column containing the same criterion values. The **Departments** table, for example, could have a column called **ID**, which contains unique values for each department. The **Staff** table could include a column named **Department_ID** containing values, each of which references a department. As soon as individual values of the two criteria are identical, their records can be assigned and linked.

2.3. Analysis realm

Different business areas often require different perspectives on data. With the help of so-called analysis realms, data can be combined and the various analytical questions from one or more fields of interest can be answered interactively.

In Data analytics, an analysis realm represents a collection of tables and selected unique associations between these tables. If one analysis realm cannot answer a specific question, a new analysis realm can be created, for example, based on the same data, but with a new combination of associations. The individual analysis realms are independent and provide dimensions, measures, and processes for analysis based on your selection of tables and associations. In PPM, analysis realms are always evaluated individually.

A mixed analysis realm (Cross analytics) contains a process table in addition to the data tables (Data analytics). A mixed analysis realm enables you to combine the analysis of process-related data and table mass data.

3. Configure Data analytics

With the help of PPM CTK, you can configure the structure of analysis realms in detail and import all relevant data into PPM. You select data sources, link data, and specify the data to be available for analysis as dimensions and measures in PPM.

PPM CTK enables you to create, edit, and test analysis realms. Use PPM CTK to define and export the relevant data sources (JDBC, SAP, CSV), and specify the data tables to be used in the analysis realms. The analysis realms you configured can be saved in XML format and imported into PPM.

Please proceed as follows to configure Data analytics for a client.

1. Open the Data analytics view of a client [8]. This is where you configure all settings and transfer the configurations and data to PPM.
2. Create a data connection [9] to be used for extracting source data, such as a CSV file or a JDBC database.
3. Create an analysis realm [9].
4. Define a table (for example, from the database) and add it to the analysis realm. [10] The table is to include the columns that will be evaluated as criteria in the analysis realm.
5. Specify and configure the columns of the source table to be used as criteria in the analysis realm [10].
6. Use criteria to link various tables [11].
7. Import the analysis realm including configurations and data into PPM [16].

Various wizards and dialogs support you in performing these steps.

The Data analytics configuration files created for a client are stored in XML files located in the directory .../**custom/<client>/XML/** of your PPM installation. See chapter **Directory and file structure [30]** .

3.1. Open the Data analytics view of a client

In the Data analytics view of a client you can configure all Data analytics settings for the client. This is also where you create connections with data sources, and create and edit analysis realms.

Prerequisite

At least one client has already been created.

Procedure

1. Start PPM CTK.
2. Select a client in the **Client overview** box on the **Home** page.
3. Click **Configure Data analytics**.

The **Data analytics** view of the client is displayed showing the list of all analysis realms and data sources available for this client.

3.2. Create a data connection

You can create and configure multiple connections with different source data for a client. A data connection includes all parameters required for accessing the relevant source data and importing the data tables to be used in the analysis realms.

You can create connections to source data of the **SAP**, **JDBC**, and **CSV** types. Different settings are available depending on the type of source data selected.

Procedure

1. Click **Edit** if the button is not enabled.
2. Click **Create data connection** in the **Available data connections** box.
3. Specify your settings.
4. Click **OK**.
5. Click **Save**.

The data connection is created with the settings you specified.

The corresponding XML file, for example, JdbcConnection1.xml is saved in the directory **...\custom\<client>\xml\data_repositories** of your PPM installation.

To edit an available data connection, double-click a data connection in the **Available data connections**.

If the data connection is used in multiple analysis realms, the changes will affect all of them.

JDBC database types supported

Note the database platforms supported by PPM 10.2026.4. Further information can be found in the SAG ARIS GmbH System Requirements or in the PPM Database Systems.pdf.

3.3. Create an analysis realm

You can create any number of analysis realms [7] for an existing client.

The name of the analysis realm is language-specific and can be specified in all languages available for PPM. Based on the name, the <keyword> key is generated, which you can change when it is first created. This key is used in file names and as a parameter for the PPM command line programs, for example. Therefore, this key must be unique and contain only the characters **a** to **z**, **A** to **Z**, **0** to **9**, and "_". Invalid characters in the name are automatically removed when the key is generated.

Use the wizard to create an analysis realm or add new tables to an analysis realm, define columns, and configure the criteria required.

Procedure

1. Click **Edit** if the button is not enabled.
2. Click **Create analysis realm** in the **Analysis realms of client** box.

3. Specify your settings.
4. Click **OK**.
5. Click **Save**.

The new analysis realm is created and its overview page is displayed.

If you have enabled the **Add table to analysis realm [10]** option, a wizard opens for adding and configuring the relevant tables for the analysis realm. If you did not enable the option you can add tables to the analysis realm [10] at a later time.

Analysis realms with incorrect configurations are displayed in color in the list. A tooltip shows you further information about the invalid configuration.

Double-click an available analysis realm to edit it.

3.4. Configure an analysis realm

You configure an analysis realm [7] by adding tables [6] to it, selecting table columns to be used as criteria [6], and defining associations [6] between the tables.

3.4.1. Add tables

You can select any number of tables from different data sources and add them to an analysis realm.

First, in the **Add Table to Analysis Realm Wizard**, you use a data connection to select a table to be added to the analysis realm. Then, in the table, you select the columns whose data is to be extracted and used as criteria in PPM. At the end, you use two criteria to link the new table with a table that already exists in the analysis realm.

You can import only a completely linked analysis realm into PPM [16]. You create a completely linked analysis realm by adding individual tables one after another to the analysis realm and linking each of them with existing tables. You link two tables by selecting a criterion in one table with the same data type and the same values as the criterion in the other table. Therefore, you need to ensure that the tables you select have criteria meeting this requirement.

You can link the tables by using either the wizard or the **Table links** tab after you exit the wizard. If you want to link a new table with multiple tables you need to use the **Table links** tab.

Tables already added to the analysis realm are listed on the **Analysis realm tables** tab of the overview pate. Tables are displayed on the left including their names specified during configuration and the type of data connection. On the right, the configurations pertaining to a selected table are shown. This includes names of the table columns selected, description, data type in PPM, and the set criterion.

Procedure

1. Double-click an analysis realm in the **Analysis realms of the client** list.
   The overview page of the analysis realm is displayed.
2. Click **Edit**.
3. Click **Add table to analysis realm**.
   Use the wizard that opens for selecting tables from a data source.

4. Specify your settings.
5. Click **Finish**.

The tables you select are added to the analysis realm and displayed on the **Analysis realm tables** tab.

Double-click a table in the list to edit it.

To remove a table from the analysis realm, select the table and click **Remove table from analysis realm**.

You can also link tables at a later time.

3.4.1.1. Define keys

You can define a criterion (based on a column) as a key for each table that you add to an analysis realm [10]. With the key, identical data is identified during data import [18] and processed accordingly.

The values of the keys uniquely identify the rows of a table. During import, the key values of the preceding rows are compared to the key values of the rows to be imported. If the values match, the data is overwritten. Otherwise, the data is added to the existing table.

- Each table must contain only one key.
- Each key criterion must be able to be calculated for each row.
- Only criteria of the following types are allowed as keys: TEXT, LONG, or TIME.
- Keys based on time stamps are calculated in line with the configured accuracy of the corresponding time dimension.
- Keys are optional. If no keys are defined for a table, all rows imported into this table are added and never replace an existing row.

3.4.2. Link tables

You can configure table associations [6] for each analysis realm by linking specific criteria of the different tables.

When configuring the associations, please observe the restrictions described in chapter **Restrictions and special cases [26]** .

The associations configured are displayed in graphical form on the **Table links** tab. In the upper part of the tab, the tables are shown as labeled rectangles and tables that are already linked are combined into blocks. If the bar contains multiple blocks the analysis realm has more, independent associations. Unlinked tables of the analysis realm are shown as one block. If you properly and completely associated all tables, they should all be combined into a single block.

To create a completely linked analysis realm, you can link the individual tables with blocks, and then the blocks with each other until all tables are combined into one block.

The lower part of the tab displays the criteria that were used for linking the individual tables.

Procedure

1. Double-click an analysis realm in the **Analysis realms of the client** list.
   The overview page of the analysis realm is displayed.

2. Display the **Table links** tab.
3. In the upper part of the tab, double-click a rectangle with the name of the table that you want to link.

   The **Change table links** dialog is displayed that you can use for selecting the association criteria for linking the table.
4. Specify your settings.
5. Click **OK**.
6. Click **Save**.

Linked tables are combined in one block. Your analysis realm configuration was saved.

To delete a table link, select a table in the upper area, then click **Delete** in a link row in the lower area.

You can run a validity check for your analysis realm configuration.

3.4.3. Define partial data extraction

You can define a partial data extraction for tables with the **SAP** and **JDBC** data connection types.

The Extract all data option is enabled by default during the configuration of an analysis realm or during migration of an older client to PPM version 9.9.

After the analysis realm configuration, you can define a partial data extraction by specifying an extraction time or value from which the data is to be extracted.

Procedure

1. Double-click an analysis realm in the Analysis realms of the client list.

   The overview page of the analysis realm is displayed.
2. On the Analysis realm tables tab, select a table of the SAP or JDBC type, for which you want to define a partial data extraction.

   The required settings are displayed in the Table criteria window.
3. Enable the Extract all data option to extract all data in the table without restrictions.
4. Enable the Extract data from option, if only specific data of the table is to be extracted.
5. In the selection box, specify whether you want to extract from an extraction time (Last point in time extracted) or an extraction value (Last value extracted).

   Depending on your selection and the data connection type of the table, various options are available (see table below).
6. Specify your settings.
7. Click **Save**.

Your settings are applied.

Option	Description
Extract data from "Last point in time extracted"	The extraction starts from a specific point in time and ends with the current point in time.
Next extraction starts from	Indicates the next extraction time. Only data with a time stamp equal to or greater than the point in time defined are extracted.
Initial extraction starts from	Indicates the initial extraction date which ensures that all data of the table are going to be extracted. The default date is 01.01.1970.00.00. Enter a date and time.

Option	Description
Overlap	Time value subtracted from the current execution time as the end time during extraction. Data extraction will end earlier by the specified value (in seconds). Without specifying a time offset value, data records that have been created but not written explicitly to the database to be extracted might not be extracted in any of two subsequent extraction operations due to their offset time stamp. Detailed information is available in the **PPM Process Extractors** documentation.
Time zone	Indicates the time zone of the time stamp in the database. If you do not select a time zone, the default time zone applies.
Source system field	Field (table column) with the time stamps in the source system that is used for controlling the extraction. All source system fields are available for selection. The default selection is the first field. Select a source system field.
Source system type (applies to JDBC data sources only)	Depending on the data type of the source field, a source system type is automatically set (date, time stamp (default), and text). You can change the preset value.
Format (only available for Text source system type)	Indicates the time format of the source data, for example, yyyyMMdd. For SAP data sources, the format is automatically extracted from the source system field defined. For JDBC data sources, you can set the format only for the Text source system type.
Extract data from "Last value extracted"	Extraction starts from a specific value.
Next extraction starts from value	Value from which the next extraction starts. Enter a value. Default value is 0.
Initial extraction starts from value	Indicates the initial extraction value which ensures that all data of the table are going to be extracted. The default value is 0. Specify a value.
Overlap	Offset value subtracted from the start value during extraction (lastreadvalue from the data source file). If the result of your data is a negative value, this value is handled like a positive value. An offset value specified in the data source configuration is included in data extraction only if the lower limit is not explicitly specified. Detailed information is available in the **PPM Process Extractors** documentation.
Source system field	Field (table column) in the source system that is used for controlling the extraction. All source system fields are available for selection. The default selection is the first field. Select a source system field.

4. Configure Cross analytics

Cross analytics enables users to combine Data analytics and process analysis. Data tables are linked with a process table in a mixed analysis realm. A mixed analysis realm in Cross analytics is basically a Data analytics analysis realm with additional, specific process criteria.

Create a mixed analysis realm by adding a process table to an existing Data analytics analysis realm, or by selecting a process table and then adding data tables when creating an analysis realm. Based on the process table, you configure the process criteria for the analysis realm and link them with the Data analytics criteria. A mixed analysis realm can only have a single process table for a single specified process tree node.

The process data in Cross analytics represent a view of process analysis process data (not a copy of the data). This means that you work with the same data in each Cross analytics analysis realm as in process analysis. You import the data into process analysis and see them (or the selected section) in Cross analytics, as well.

Via a predefined data connection the process types configured in the process analysis for a client are extracted and provided in a process tree for selection. For the process tree node you select, the process criteria are transferred and displayed in the process table as measures and dimensions. The process data is not copied from process analysis to Cross analytics, but only displayed in table form. Process data is automatically transferred from process analysis so that the same process data exists in Cross analytics and process analysis.

- All process measures registered at the process tree node are available, except cardinality measures and measures of the **RELATION** and **OT_ORG** types.
- All process dimensions of the **TEXT** and **TIME** types registered at the process tree node are available.

Function measures and dimensions cannot be used for configuration.

To create a mixed analysis realm, proceed as follows. The **Add Table to Analysis Realm** Wizard supports you.

Procedure

1. Create a new analysis realm [9] or add a process table to an existing analysis realm [14].
2. Import the analysis realm into PPM [16].

The mixed analysis realm is available as an individual Cross analytics analysis realm in PPM.

4.1. Add process tables

Adding a process table in order to create a mixed analysis realm is very similar to the procedure in a Data analytics analysis realm.

Using the **Process data** data connection in the **Add Table to Analysis Realm Wizard**, import the process types configured for the client. In the next step, the available process types are displayed in a process tree. In the process tree, select the process tree node whose processes you want to analyze in the analysis realm. Then, select the process criteria, that is, process measures and process dimensions, and specify their usage. Finally, use two criteria to link the process table with a table that already exists in the analysis realm.

Prerequisite

You have configured a Data analytics analysis realm [10].

Procedure

1. Double-click an analysis realm in the **Analysis realms of the client** list.
   The overview page of the analysis realm is displayed.
2. Click **Edit**.
3. Click **Add table to analysis realm**.
   Use the wizard that opens for configuring and linking the process table.
4. Select the data connection **Process data**.
5. Click **Next**.
6. Specify your settings and click **Finish**.

The tables you select are added to the analysis realm and displayed on the **Analysis realm tables** tab.

To remove the table from the analysis realm, select the table and click **Remove table from analysis realm**.

5. Import analysis realms

You can import configured analysis realms into PPM. Analysis realm configuration and source data can be imported individually. This way, you can continue using configurations or source data existing in PPM and update them independently.

You can import only complete and properly configured analysis realms into PPM. The validity of all analysis realms of the client is checked automatically. Analysis realms with invalid configurations are highlighted in red in the analysis realm overview. A tooltip provides information on any configuration errors.

Before importing data you need to reinitialize the PPM database of the client. You initialize your database using the **initdb.bat** program in the **Programs** module in PPM CTK. For detailed information on initializing the database, please refer to the PPM CTK User Guide.

5.1. Overwrite analysis realm configuration

You can overwrite an existing analysis realm configuration in PPM. All changes are transferred to the existing analysis realm configuration.

The function **Overwrite analysis realm configuration** is run by the **runppmconfig** program in **-overwrite** mode. Further details are available in chapter Import mode [32].

Prerequisite

The PPM client service must be started.

Procedure

1. Double-click an analysis realm in the **Analysis realms of the client** list.
   The overview page of the analysis realm is displayed.
2. Click **Overwrite configuration**.
   Updates the analysis realm configuration of the current client in PPM.
   After the import, an overview of the import process of the configuration is displayed.
3. To view the log of an import process, click **Display log file** () in the **Log file** column of the dialog.
4. Click **OK**.

The configuration has been imported into PPM. An existing configuration has been updated. The updated Data analytics analysis realm is now available in PPM.

Please note that you may have to reinitialize the analysis realm in the analysis server [16] after the import. The automatically generated log file () informs you whether you need to reinitialize the analysis server after the import.

5.1.1. Reinitialize analysis realm in analysis server

To reinitialize an analysis realm in the analysis server, use the program **Eliminate inconsistencies on the analysis server**.

During reinitialization, data saved in the database are loaded to the analysis server. Reinitialization eliminates inconsistencies in database and analysis server data.

Since reinitialization can take some time, it is not automatically run when importing the configuration. You can perform the required reinitialization at a more suitable time.

However, up to reinitialization, the analysis realm configuration is inconsistent. You can still query the inconsistent analysis realm, but if the queries refer to changed configuration elements, error messages can occur in the response to the queries until you run the reinitialization.

If you import more data [18], the analysis realm will reinitialize automatically, if required.

Procedure

1. Start PPM CTK.
2. Select the relevant client in the **Client overview** box on the **Home** page.
3. Click **Configure process analysis**.
4. Open the **Programs** component in the **Client** module.
5. In the **System** program group, run the program **Eliminate inconsistencies on the analysis server**.

All inconsistent elements of the analysis realm in the analysis server are automatically reinitialized.

Alternatively, you can use the runtableimport.bat [35] program in the command line to reinitialize the analysis realm [41].

5.2. Replace analysis realm configuration

You can replace an existing analysis realm configuration in PPM with a new configuration.

The function **Replace analysis realm configuration** is run by the **runppmconfig** program in **-replace** mode. Further details are available in chapter Import mode [32].

Prerequisite

The PPM client service must be started.

The data tables of the relevant analysis realm must not contain any data.

Procedure

1. Double-click an analysis realm in the **Analysis realms of the client** list.
   The overview page of the analysis realm is displayed.
2. Click **Replace configuration**.
   Replaces the analysis realm configuration of the current client in PPM.
   After the import, an overview of the import process of the configuration is displayed.
3. To view the log of an import process, click **Display log file** ( ) in the **Log file** column of the dialog.
4. Click **OK**.

The configuration has been imported into PPM. An existing configuration has been replaced. The updated Data analytics analysis realm is now available in PPM.

5.3. Import data

You can import source data into PPM regardless of the analysis realm configuration.

If required, the analysis server is automatically reinitialized during data import.

Prerequisite

The PPM client service must be started.

Procedure

1. Double-click an analysis realm in the **Analysis realms of the client** list.
   The overview page of the analysis realm is displayed.
2. Click **Import data**.
   Extracts the source data of all tables in the analysis realm in line with the defined data connection and imports the data into PPM in addition to the existing data.
   After the import, a note on the import process of each table is displayed.
3. To view the log of an import process, click **Display log file** () in the **Log file** column of the dialog.
4. Click **OK**.

The source data of the analysis realm has been imported into PPM and existing source data has been overwritten. The updated Data analytics analysis realm is now available in PPM.

The function **Import source data into PPM** first runs the relevant extractor (for example, runjdbc2ppm.bat) regardless of the type of source data, and then the **runtableimport.bat** program with the **-add** parameter. The extractor extracts the source data and writes them into a table that runtableimport imports into PPM. See chapter **runtableimport.bat [35]** in the appendix for further information. For detailed information on extractors, please refer to the technical documentation **PPM Process Extractors**.

5.4. Delete data

You can delete all table data of an analysis realm in PPM, for example in order to prepare a new analysis realm configuration.

The **Delete data** function is run by the **runtableimport.bat** program with the **-delete** and **-alltables** parameters. Further details are available in chapter runtableimport.bat [35].

Prerequisite

The PPM client service must be started.

Procedure

1. Double-click an analysis realm in the **Analysis realms of the client** list.
   The overview page of the analysis realm is displayed.
2. Click **Delete data**.

All table data of the analysis realm has been deleted.

If required, you can now replace an analysis realm configuration [17] or reimport the data.

6. Delete table data

You can delete data of individual tables of an analysis realm.

After deletion, the initial time of extraction or extraction value ("Next extraction starts from") of the table extractor are restored. If the table is reimported, all values from the initial time of extraction or extraction value are imported again.

Procedure

1. Double-click an analysis realm in the **Analysis realms of the client** list.
   The overview page of the analysis realm is displayed.
2. Click **Delete data**.
3. Select the tables whose data you want to delete.
4. Click **OK**.

The data is deleted.

7. Appendix

7.1. Query logic

This chapter explains how the individual Data analytics components work together and describes the options and restrictions pertaining to queries in analysis realms.

The main difference between Data analytics and Process and Interaction analysis is that the data recorded is not based on process instances. In general, however, the basic queries are the same for all three methods of analysis. By default, all available dimensions and measures can be used and calculated for analyses. The queries can contain any number of filters, iterations, and aggregations.

7.1.1. Query structure

Central interaction and query semantics in Data analytics are similar to those in Process analysis, adding various functional extensions. Queries in Data analytics consist of three parts and are modular in nature.

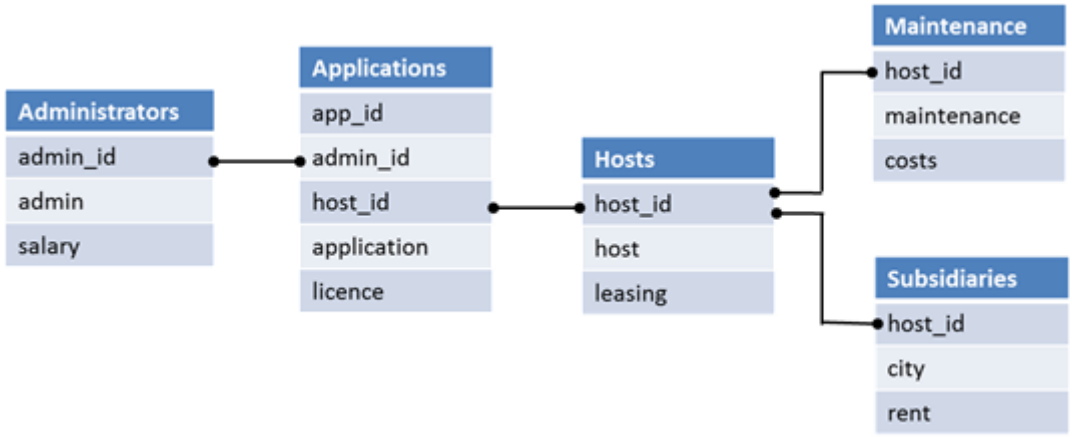
The following example illustrates the structure of queries.

Example 2.

The example describes an analysis realm pertaining to the structure of an IT department.

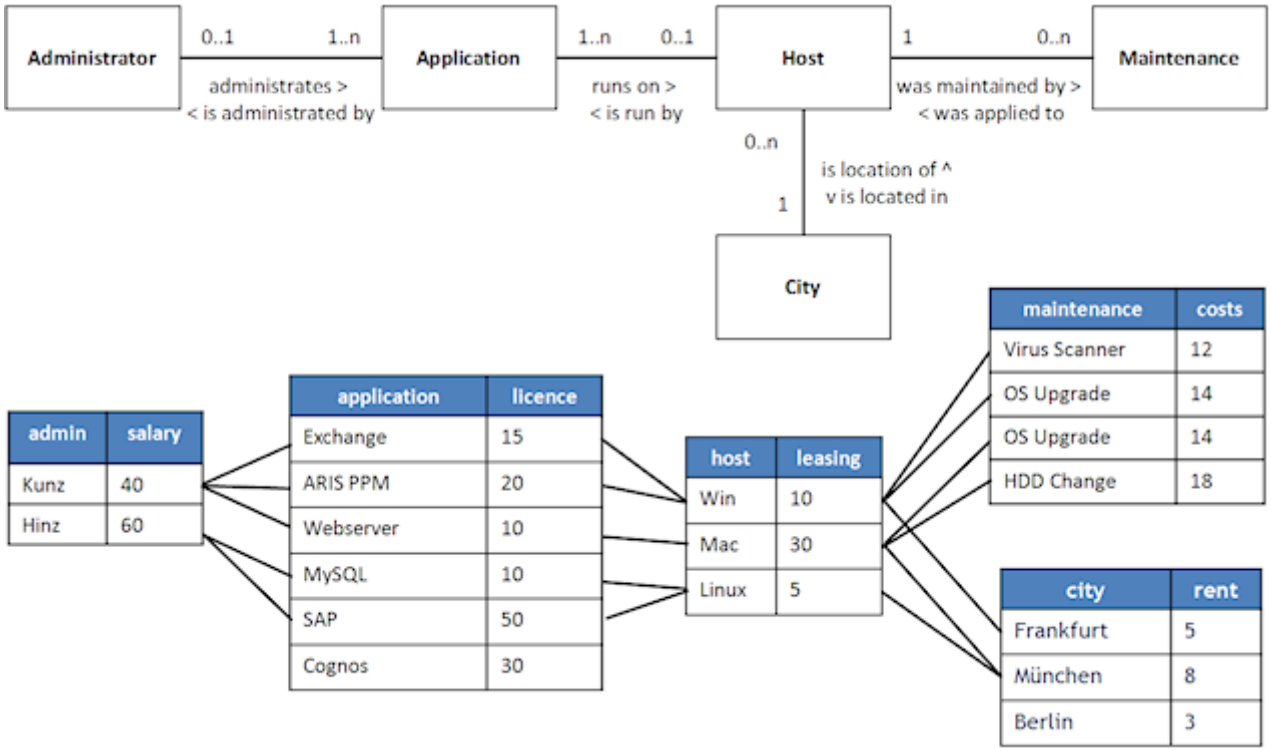
The following illustration shows the table structure including the linked association criteria. From a conceptual point of view, specific applications run on different hosts, and different administrators are responsible for the individual applications. The hosts are assigned specific maintenance and subsidiaries as a physical location.

Illustration of the analysis realm of the IT department



The following illustration shows the relationship of tables among each other and the data contained in the table. For a clearer overview, association criteria have been left out in this example. Your information is represented by the connections linking the columns of the tables assigned.

Analysis realm data of the IT department



7.1.2. Data filtering

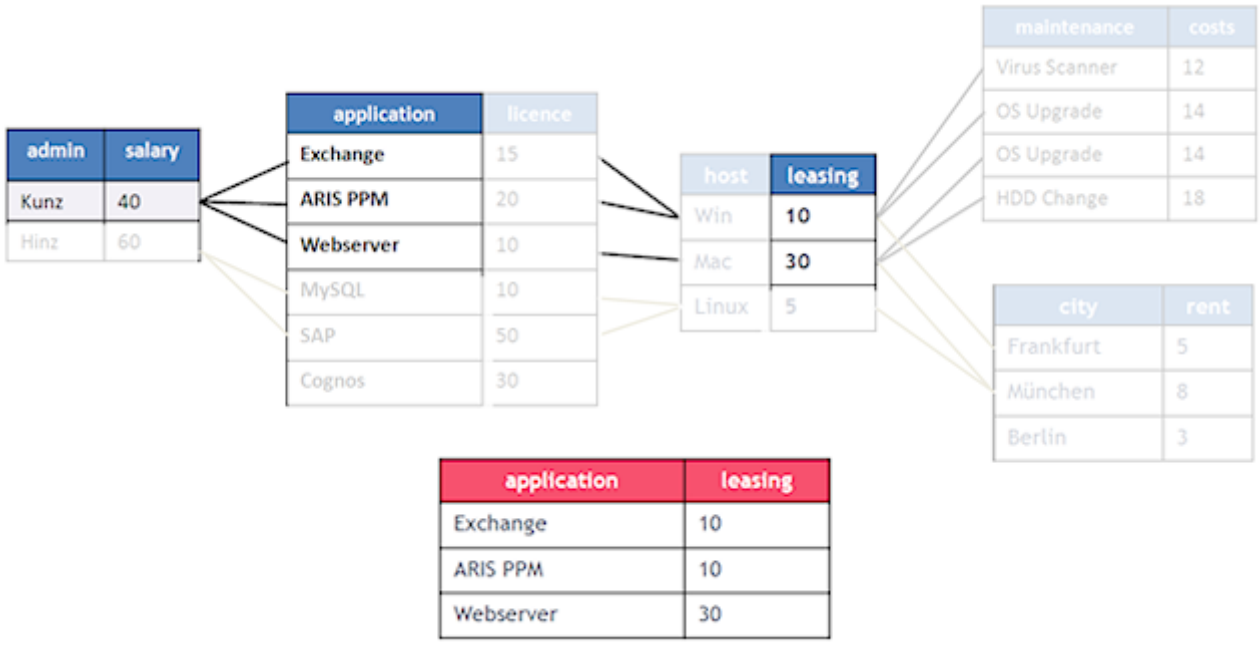
Queries extract a defined set of records. As long as no filter has been set, the records are queried without restrictions. This means that they contain all rows of the analysis realm defined. Whenever you set a filter, certain rows will be filtered out. The number of records is reduced according to the filter you set.

A filter is always applied to one criterion. The filter specifies your preferred and suitable criterion values. If a filter is set only those table rows with matching criterion values will be selected. The filter is then transferred to other tables, in which only those rows are selected that are linked to at least one selected row of the source table. The filter is transferred recursively to all tables relevant to the query.

Example 3.

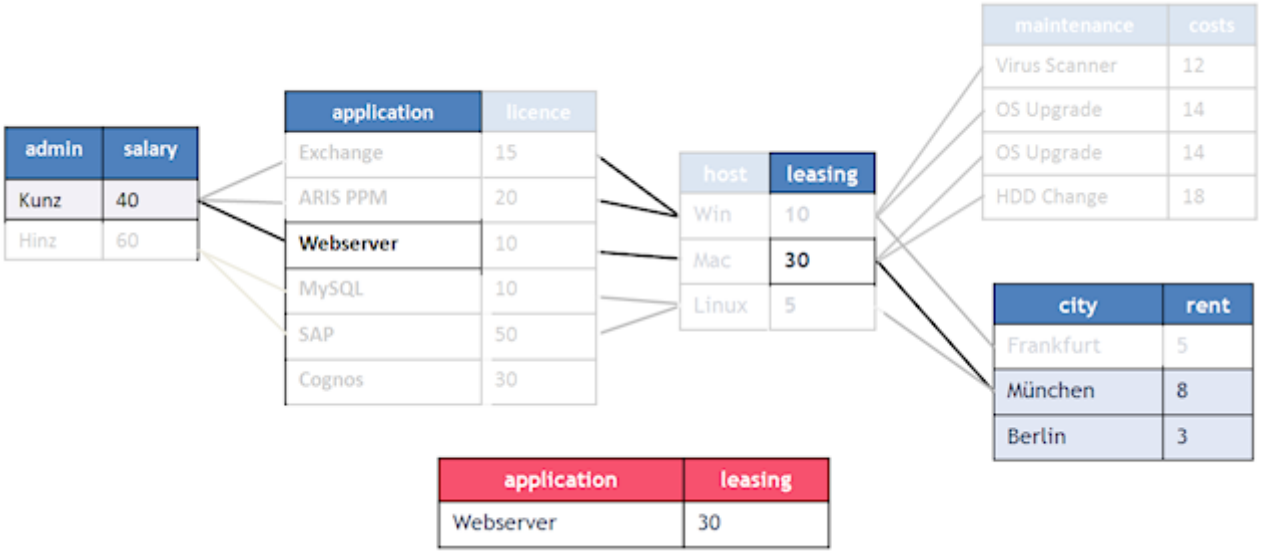
In this example, filtering for an administrator by the name of **Kunz** returns only the values of those rows that are linked with the corresponding row in the **Administrators** table.

Query of the application and the leasing rate for each application filtered by the name of the administrator **Kunz**



Rows not linked to a row selected will not be included in the record selected. Therefore, the row of the **Cognos** application will not be included in the query result as long as the query contains more than just the **Application table** criteria. If the query used the **Application** criterion for iteration and aggregated the licenses, Cognos/30 would be included in the query result.

Query result with an additional filter set to **Munich** or **Berlin**



The filters supported and their semantics in data analysis are similar to those in PPM Process analysis. For example, text or period filters are available, and you can set filters for measures. You can combine multiple filters and use result filters for iterations, etc.

7.1.3. Iteration

Iterations list the individual values of a criterion or form a grouping based on criteria values. The individual elements of a list are called iteration steps. Multiple iterations can be combined.

Example 4.

In the example, an iteration using the **Maintenance** criterion returns a list of the following iteration steps. Iteration step **OS upgrade** is listed only once because only distinct values are shown.

Maintenance
Example 5.
Virus scanner
Example 6.
OS upgrade
Example 7.
HDD change

Example 8.

An iteration using the criteria **Host** and **Maintenance** returns the following iteration steps.

Host	Maintenance
Example 9.	**Example 10.**
Win	Virus scanner
Example 11.	**Example 12.**
Win	OS upgrade
Example 13.	**Example 14.**
Mac	OS upgrade
Example 15.	**Example 16.**
Mac	HDD change

Example 17.

The list shows which host underwent what maintenance. In line with the default iteration behavior, the **Linux** host is not displayed because Linux has not undergone any maintenance yet.

Tables not required for linking the queried criteria do not have any effect on the query result.

Iterable measures

You can mark a measure criterion as iterable in the analysis realm configuration. The same conditions as in the process analysis configuration apply.

Such a measure can be iterated using a specific step width. The step width has a defined default step width, but you can change this in a dialog. The procedure for setting the step width is the same as in process analysis and described in the PPM online help.

Only linear step widths are supported. User-defined iteration steps and logarithmic step widths are not supported in Data analytics.

7.1.4. Aggregation

An aggregation combines multiple numerical values of a criterion in one value.

Criteria have a default aggregation mode based on their data type. However, you can configure the aggregation mode during evaluation, as well.

Table of aggregation modes

Abbreviation	Description
Sum	**Sum of values, default for integers**
Avg	Average, default for floating point numbers
Min	Minimum
Max	Maximum
Count	Number of values of a criterion (in PPM, this is also called relevance)
Stddev	Standard deviation [52]
Quantiles	Quantiles, for example, median [52]

A query can consist of multiple aggregations of a criterion or of multiple criteria. Aggregations are run independently.

Example 18.

In the example, adding the individual values of the **Costs** criterion results in the total cost of maintenance (sum = 58) of all hosts.

Host	Maintenance	Costs
Example 19.	**Example 20.**	**Example 21.**
Win	Virus scanner	12
Example 22.	**Example 23.**	**Example 24.**
Win	OS upgrade	14
Example 25.	**Example 26.**	**Example 27.**
Mac	OS upgrade	14
Example 28.	**Example 29.**	**Example 30.**
Mac	HDD change	18

7.1.5. Combination of iteration and aggregation

If you are using iterations and aggregations combined, the aggregation is run for the subset of values that are part of the iteration step. The two images shown in this chapter illustrate this context.

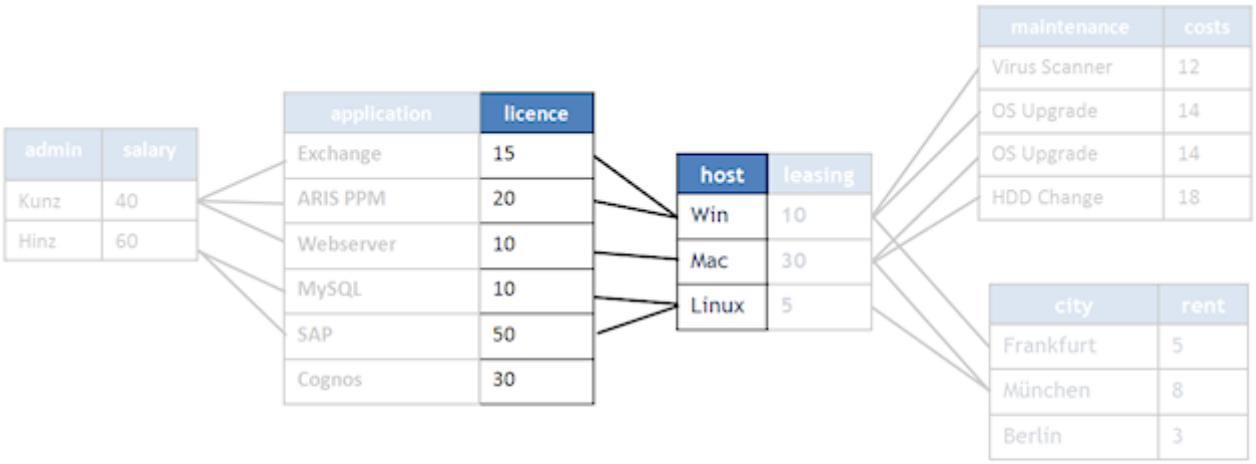
Of course, you can both combine multiple iterations for different criteria and query multiple aggregations for the same or different criteria.

The rule of thumb is that the value displayed in each iteration step is the same value that is displayed when the iteration is replaced by filters for the step values.

Example 31.

In this example, the iteration using the **Host** criterion is replaced by the filter **Linux**. The resulting average value for license costs is 30, as in the iteration.

Average license costs of the applications per host:



admin	salary
Kunz	40
Hinz	60

application	licence
Exchange	15
ARIS PPM	20
Webserver	10
MySQL	10
SAP	50
Cognos	30

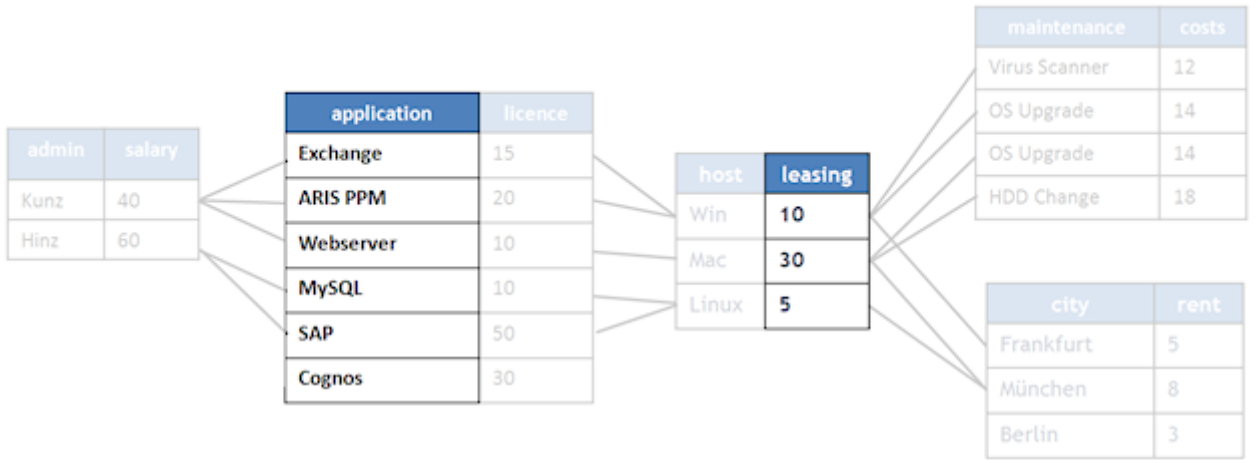
host	leasing
Win	10
Mac	30
Linux	5

maintenance	costs
Virus Scanner	12
OS Upgrade	14
OS Upgrade	14
HDD Change	18

city	rent
Frankfurt	5
München	8
Berlin	3

host	AVG(license)
Win	17,5
Mac	10
Linux	30

Host leasing rate per application:



admin	salary
Kunz	40
Hinz	60

application	licence
Exchange	15
ARIS PPM	20
Webserver	10
MySQL	10
SAP	50
Cognos	30

host	leasing
Win	10
Mac	30
Linux	5

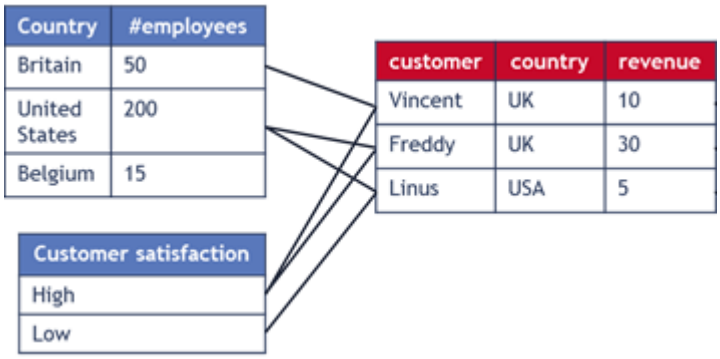
maintenance	costs
Virus Scanner	12
OS Upgrade	14
OS Upgrade	14
HDD Change	18

city	rent
Frankfurt	5
München	8
Berlin	3

application	leasing
Exchange	10
ARIS PPM	10
Webserver	30
MySQL	5
SAP	5

7.1.6. Cross analytics

The query logic in mixed analysis realms, that is, analysis realms with data tables and one process table each, is identical to the query logic in Data analytics analysis realms. Filter, iterating, and aggregation behavior remain the same.



Example 32.

Link criteria are not shown in the image.

The data of the process table (right) consists of process instances including process-level data without function or relation-level data. The data tables (left) are linked to the process data via country and customer ID. The **Country** table contains information on the number of employees of the company in each country. The other table contains information on customer satisfaction based on the latest survey.

7.2. Restrictions and special cases

7.2.1. Missing links

Links within an analysis realm are complete when all tables contained are directly or indirectly linked.

Tables in analysis realms must be completely linked because otherwise, queries are run for single, isolated components in the analysis realm. This would not lead to any plausible query results.

Analysis realms with missing links cannot be imported into PPM.

7.2.2. Missing values

By default, missing values in a database table are assigned the value **Not specified** during data import into PPM. These missing values can occur when the value of a cell in a database table is NULL. It is normally impossible in Data analytics to link two tables via values that are not specified because a link between missing values is not plausible. A table field without any value specified is usually ignored and the entire record will not be included in an association.

Text dimensions are an exception. For table cells without specified values, a custom default value can be defined that can be identical to an existing value and therefore can be linked to that value. The default value defined will always be imported if a value is missing for a text dimension. Therefore, the value **Not specified** is never ignored for text dimensions.

7.2.3. Loops

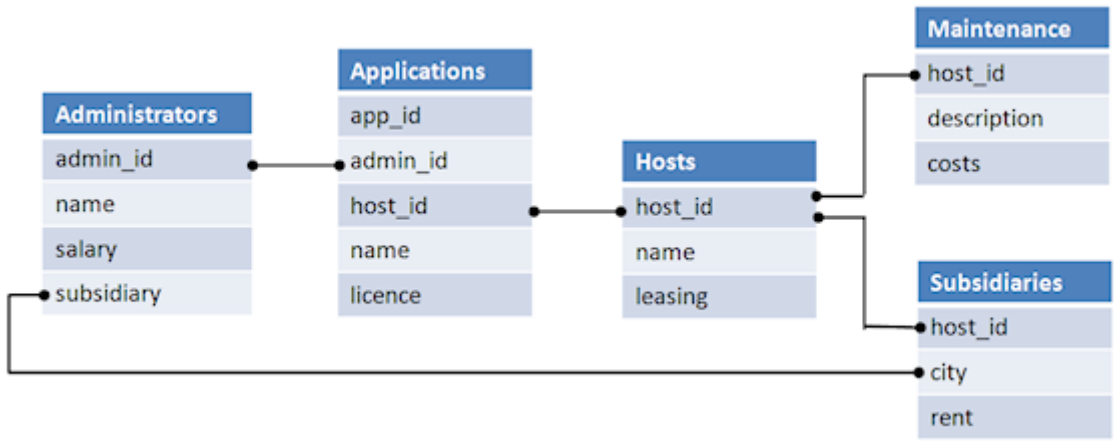
For a query to be able to return a result, the definition of an analysis realm must be unique. This means that the links and dependencies specified in an analysis realm must be consistent. The definition of a cyclical link of tables is not allowed in Data analytics because the solution path cannot be resolved unambiguously in a loop.

Example 33.

In this example, the additional association between the criteria **Administrators** and **Subsidiaries** leads to an inconsistent association in the analysis realm. If a query requests the application name (applications.name) and subsidiary (subsidiary.city) the system cannot determine unambiguously which of the associations defined is to be used. This is due to the fact that there are two solution paths, each of which leads to a different result.

If the additional association between the criteria **Administrators** and **Subsidiaries** were used, the result displayed would be the subsidiaries (subsidiaries.city) where the administrators of the application work. However, if the association was applied via the host table, the result displayed would be the physical (or organizational) locations where the hosts of the applications reside.

Analysis realm with a loop that is not allowed:



Tip
Two options for solving this problem exist.

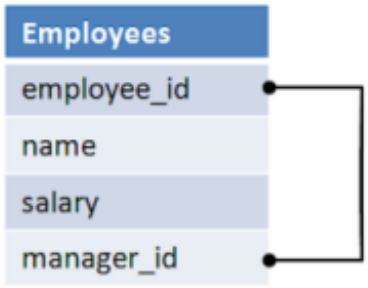
- Create a new analysis realm in which one association is removed. The selection of the analysis realm will then return the desired query result.
- Reimport the table **Subsidiaries** under a different name and define the columns of a table as new criteria under new names. The inconsistency will then be resolved by selecting the relevant criterion in the query.

7.2.4. Self-reference

Recursive associations are not allowed in an analysis realm. A link between two different criteria in one and the same table would not lead to any query result.

Example 34.

In this example, the two criteria **employee_id** and **manager_id** are linked in the same table.



Tip
A solution is to import the table multiple times, each time with differing table and criteria names.

7.2.5. Values without assignment

If a query includes tables with rows without assignment to rows in other tables of the query, these rows are not included in the record queried, that is, its values will not be included in the query result.

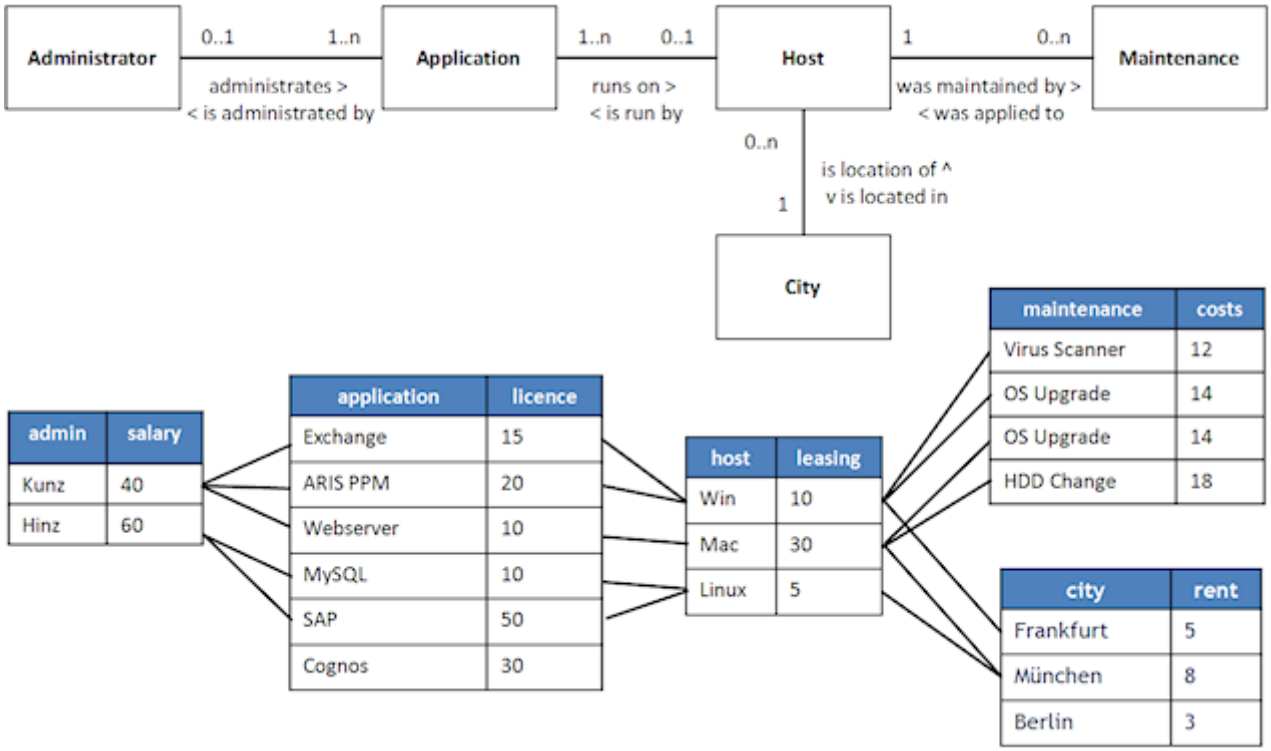
Example 35.

If the criteria **Administrators** and **Application** were queried in this example, the values **Cognos** and **Berlin** would not be included in the result. This is due to the fact that the value **Cognos** in the table **Application** and the value **Berlin** in the table **City** do not have any assignments to other rows of the tables queried. (See illustration below.)

However, if the table of the query contains a value without assignment, and the table in which the assigned value is missing is not part of the query, the values of the table queried will be included in the result.

Example 36.

If only the **application** criterion is iterated and the **licence** criterion is aggregated in the query, the iteration step **Cognos** and the license costs **30** would be included in the query result because the other linked tables would not be part of the query. As soon as another table is added to the query, for example because of a filter, iteration, or aggregation, the values **Cognos** and **30** would no longer be displayed in the result.



7.2.6. Split text dimension data

If two tables are linked via a text dimension the data of both dimensions is used. This means that in the PPM user interface, the values of both tables are displayed in filter dialogs and the **Filter** bar.

The dimension data is not deleted before all data of the analysis realm is deleted. This means that if you import various data into one of the tables and thus render the previously imported dimension data obsolete, the data continues to be displayed in PPM until the data in the analysis realm is completely deleted (for example, by replacing the analysis realm configuration).

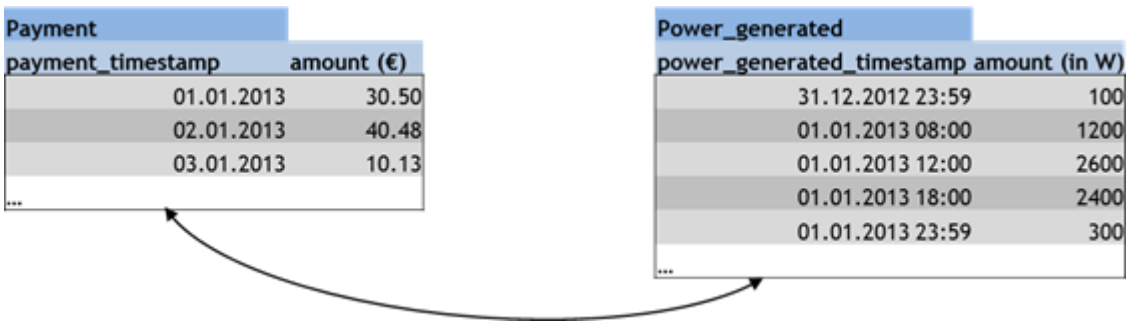
If you use a text dimension to link a data table with a process table the following special modes of behavior occur. Like in a Data analytics case, the dimension data is split. Therefore, dimension data that is used only in Data analytics, is also displayed in the user interface (filter dialogs and bar) in the process analysis, and vice versa. It is therefore difficult to remove obsolete values. The only option is to reinitialize the analysis server (runppmimport –reinitanalysisserver yes) and then reimport the Data analytics data.

7.2.7. Time dimensions with different accuracies

It is possible to link two time dimensions with different accuracies, with the rougher accuracy used for linking. Valid accuracies are **Day**, **Hour**, and **Minute**.

Example 37.

A solar energy power plant measures the solar energy generated and writes the result including time stamp (accuracy: minute) in a table. An energy provider pays daily for the energy generated (accuracy: day), which is listed in a second table.



Payment

payment_timestamp	amount (€)
01.01.2013	30.50
02.01.2013	40.48
03.01.2013	10.13
...	

Power_generated

power_generated_timestamp	amount (in W)
31.12.2012 23:59	100
01.01.2013 08:00	1200
01.01.2013 12:00	2600
01.01.2013 18:00	2400
01.01.2013 23:59	300
...	

In this example, the rougher accuracy of the two tables is **Day**, which is therefore used for the analysis. The finer measure **amount (in W)** is summarized based on the rougher accuracy. If you analyze January 1, 2013, the energy generated is summed up for this day with the following result.

payment_timestamp	amount (€)	amount (in W)
01.01.2013	30.50	6500
...		

7.3. Directory and file structure

During the configuration of Data analytics for a client, PPM CTK creates relevant files and directories. The files are stored by client in the .../**custom/<client>/XML/** directory of the PPM installation.

<client> indicates the client name and <realm> stands for the analysis realm name.

```
.../custom/
<client>/
xml/
data_repositories/
JdbcConnection.xml
SapConnection.xml
CSVConnection.xml
realm_<realm>/
datasource_CUSTOMER.xml
datasource_FILM.xml
<client>_realm_<realm>.xml
<client>_CTK.xml
```

xml directory

The **xml** directory contains the analysis realm configuration file [10] **<client>_realm_<realm>.xml** and the file **<client>_CTK.xml**. Each data connection created in PPM CTK is listed in the **<datarepositories>** XML element of the file **<client>_CTK.xml**.

```
<datarepositories>
< datarepository name="sakila_jdbc"
type="JDBC"> ..\custom\umg_en\xml\data_repositories\sakila_jdbc.xml</
```

```
datarepository >
</datarepositories>
```

The directory also contains the directories **realm_<realm>** and **data_repositories** used for storing analysis realm-specific and independent files.

data_repositories directory

The **data_repositories** directory contains the XML files for the data connections defined in PPM CTK [9]. An XML file describes a specific connection to specific JDBC, SAP, or CSV source data including the required parameters described in the relevant DTD files for JDBC (JdbcConfig.dtd), SAP (r3Config.dtd), and CSV (csvextractor).

realm_<realm> directory

The **realm_<realm>** directory contains the data sources of an analysis realm. A data source is any individual table that is used in an analysis realm The tables are assigned to the analysis realms in PPM CTK [10]. Each data source is created as a separate XML file that the relevant extractor uses for extracting the source data.

7.4. Configuration versioning

Use versioning of analysis realms to identify any configuration inconsistencies between PPM server, analysis server, and recovery files.

Each analysis realm configuration has a configuration version (ID) for the consistent identification of the configuration between PPM server, analysis server, and recovery files.

Each change in configuration leads to a gradual change of the configuration version of the analysis realm. The configuration version is saved in the PPM server, the analysis server, and the recovery files.

- The PPM server and the PPM user interface output a warning if an analysis realm configuration is inconsistent, and they write a warning to the PPM server log file.
- If the analysis server is started based on recovery files, configurations and data still existing on the analysis server but no longer on the PPM server will be deleted on the analysis server.

  When connecting the PPM server and the analysis server, missing analysis server configurations are imported. The data must be manually imported.

To render an inconsistent analysis realm configuration consistent, you need to reinitialize the analysis server [41].

7.5. Data versioning

Each table of an analysis realm has a version (ID) for the consistent identification of data between PPM server, analysis server, and recovery files.

Each change to data of a table of an analysis realm leads to an incremental change of the version of this table. The table version is saved in the PPM server, the analysis server, and the recovery files.

- Different versions of a data table between PPM server and analysis server lead to an inconsistent analysis realm and require a reinitialization of the analysis server. (See chapter Reinitialize analysis realm in analysis server [41])
- The PPM server and the PPM user interface output a warning if an analysis realm is inconsistent, and they write a warning to the PPM server log file.

7.6. Import and export of configurations

Use the **runppmconfig** configuration program for importing and exporting analysis realm configurations. You can also execute runppmconfig by calling the database initialization program **runinitb** for importing configurations from Data analytics analysis realms listed in the file **InitSystem_settings.properties** of the client.

runppmconfig supports both modes, that is, **Import** and **Export**. During the import of a configuration, the relevant configuration module is locked. You cannot run multiple imports at the same time.

Prerequisite: To execute both modes, the user must have the **Configuration import** function privilege.

7.6.1. Export mode

In **Export** mode, the current configuration of an analysis realm is exported. The configuration is saved in XML format in line with the DTD rules under a certain file name.

For exporting, runppmconfig is called with the following syntax.

runppmconfig –user <user name> -password <password> [-client <client name>] -mode export –realm <file name> -realmname <keyword>

The analysis realm name (keyword) specified must reference an analysis realm in the configuration, otherwise, an error message will be output.

7.6.2. Import mode

In **Import** mode, a configuration of an analysis realm is imported into PPM. The configuration is available in an XML file created in line with DTD rules.

During the import, you can replace (-replace) [33] or overwrite (-overwrite) [33] an existing analysis realm configuration.

For importing, runppmconfig is called with the following syntax.

runppmconfig –user <user name> -password <password> [-client <client name>] -mode import [-replace | -overwrite] –realm <file name>

The name of the imported analysis realm is the one used in the import file and does not need to be specified.

Each configuration imported is checked and validated. If they are invalid the configuration is not imported and an error message is output. If a warning is output, the import runs and a corresponding message is displayed.

After a successful import, the PPM server deletes the query cache and refills it with favorites queries.

If neither of the parameters **-overwrite** or **-replace** has been specified, the analysis realm configuration of the import file is added to the analysis realm configuration on the server. If the analysis realm already exists an error message is displayed. After successful import the new configuration is activated on the analysis server.

The **-add** parameter is not supported and results in an error.

Import a Cross analytics analysis realm

Under certain circumstances, the import process can take a long time, especially when importing a Cross analytics analysis realm. In this case, the analysis server is creating an internal data structure for the imported process table and perform indexing. The new structure is created when the Cross analytics analysis realm is imported for the first time or if the process table association has changed during another import.

7.6.2.1. -replace parameter

The **-replace** parameter replaces the analysis realm configuration on the server with the new configuration from the import file. If the analysis realm does not exist a new analysis realm is created.

If the import file contains an analysis realm without tables and criteria, the analysis realm is considered empty and will be deleted from the configuration.

If the analysis realm already exists and contains saved data, the process is canceled without changing the existing configuration, and an error message is displayed. Only data tables are checked for existing data. This means that process data is allowed to exist in Cross analytics analysis realms.

If the analysis realm already contains data, you need to delete the data tables by means of the runtableimport [35] program before you replace the analysis realm configuration. After successful import the new configuration is activated on the analysis server. You need to reimport that data.

The import file with the following content deletes the analysis realm with the keyword **Employee_Positions**.

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE realm SYSTEM "analysisrealm.dtd">

<realm keyword="Employee_Position" />

7.6.2.2. -overwrite parameter

The **-overwrite** parameter retains the existing analysis realm configuration and applies only the changes in the analysis realm configuration to be imported. In contrast to the **-replace** parameter, most changes are applied without deleting the data from the tables first. Before any changes are applied, the analysis realm configuration to be imported is analyzed and validated. Not all changes are allowed, or they require a subsequent reinitialization of the analysis realm.

Certain changes are applied only if the table involved no longer contains any data. In the following cases, you need to delete the table data and restart the import of the analysis realm configuration:

- Deletion of a table
- Deletion of a table column

  Can only be done if the table data has been deleted. Otherwise, this change is ignored without any further prompt, and the configuration change is applied without it.
- Changed data type of a table column
- Add or remove a key, or change structural properties of a table column used as a key

When changing the retriever type of a measure, all analysis realm data must be deleted before the import.

After structural changes to the configuration, the analysis server configuration must be reinitialized to apply the changes. A corresponding message is displayed when running runppmconfig.

If the analysis realm configuration is inconsistent already before the import of the changed configuration, no changes are transferred to the analysis server. The analysis server must be reinitialized in this case, as well.

For detailed information on reinitializing the analysis server, please refer to chapter
Reinitialize analysis server [41].

7.7. Data import

You import data from various heterogeneous data sources, such as XML files or relational databases and combine and represent them in different analysis realms. The analysis realms are saved in their entirety in the analysis server's memory and thus enable quick access to data and their efficient evaluation.

Data import consists of two steps. In the first step, an extractor extracts data from the source system and stores them in PPM system event format in the file system. In the second step, the **runtableimport** command line program extracts the data from the files stored in the file system and imports them into the PPM system. Both extractors and **runtableimport** program work with the same data source file that specifies where the data comes from and how they are to be imported into the analysis realms. PPM CTK automatically generates the data source file and stores it as described in chapter **Directory and file structure [30]** . Detailed information on handling the PPM extractors is available in the documentation **PPM Process Extractors**.

7.7.1. Persistence

The imported data is also saved persistently in the PPM database. The structure of the analysis realm tables in which the data is saved is defined in the analysis realm configuration [8].

- Each table can contain a maximum of 2^{31} (2.147.483.648) rows.
- Each analysis realm table can have a key [11] based on which an existing table row can be identified.
- A versioning mechanism checks if the current data in the database and in the analysis server are consistent. Different existing versions result in data inconsistency and require reinitialization of the analysis server.
- Only completely calculated data rows (measures and dimensions) are imported into the database and the analysis server.
- All data is transformed (according to the specifications in the data source or the file **AdapterConfig_settings.properties**) before being saved in the database. Changes to the file AdapterConfig_settings.properties are therefore not transferred when the analysis server is reinitialized.

- Data of the **FACTORYTIMESPAN** type are saved in the original format in the database, for example, 5 FACTORY_DAYS.

  A change to the transformation factors (<client>_transformationfactors.xml) is therefore transferred when the analysis server is reinitialized.

To avoid having to initialize data after every server start, the analysis realms of the analysis server are saved in recovery files and loaded automatically.

Data import with duplicate keys

If a table row is imported and has the same key [11] as an existing row, the existing row is replaced by the new row.

If the import file contains multiple rows with ab identical key, only the last row is imported and saved in the database.

Data import with missing criteria or criteria that cannot be calculated

If a criterion for a row cannot be calculated, an error message is output and the row is not imported. The same applies to key and non-key criteria.

If a column is missing in the import file or if a column is empty, the criterion based on this column is not imported. In case the criterion is used as a key and it is not a text dimension, the entire row is skipped because the missing key must be an error.

Reinitialization behavior differs from data import behavior: If a measure or dimension cannot be calculated during reinitialization, an error message is output, but the relevant row is transferred to the analysis server. A text dimension is assigned the default value. All other criteria are handled as "not specified". This does not apply to key criteria.

7.7.2. runtableimport.bat

The **runtableimport** program extracts data from a file in **XML** or **ZIP** format (XML event format) and updates a table in an analysis realm. Import file, table, and analysis realm are specified by parameters.

Prerequisite: The user has the **Data import** function privilege.

You run the **runtableimport** program in the command prompt by entering the parameters listed.

runtableimport.bat

-user <user name>

-password <password>

-realm <realm name>

[-mode (add | replace | delete)]

[(-datasource <file name> |

-table <table name>

-file <path>)]

[-client <client name>]

[-recoveryfile {yes | no}]

[-version]

[-protocolfile <file name>]

[-language <ISO code>]

[-information {yes | no | default}]

[-warning {yes | no | default}]

[-error {yes | no | default}]

[-alltables]

[-reinitanalysisserver {yes | no | auto}]

Parameter	Description
user <user name>	User name for logging in to a client Default: system Specification: Mandatory
password <password>	Password for logging in to a client Default: manager Specification: Mandatory
realm <realm name>	Name of the analysis realm to be imported Specification: Mandatory
mode (add | replace | delete)	Specifies whether the data imported is to be added (add) to the existing data, or whether the existing data will be replaced by the new data (replace), or whether the existing data will be deleted (delete). If the "add" mode is selected and the table has a key [11], existing table rows are replaced by new ones. (See chapter Configure analysis realm [10]) In delete mode, the data of one or all tables of an analysis realm are deleted in the database and the analysis server [40]. This mode is applied in case of inconsistencies in the analysis server and database. Default value: add Specification: Optional
datasource <file name>	Name and path of the data source; contains references to the importing file, table names, and data connection Specification: Mandatory if import file and table are not specified
table <table name>	Name of the table to be imported. The table is part of the analysis realm. Specification: Mandatory if no data source is specified
file <path>	Path including name of the file to be imported. Specification of the file format is optional, the default format is XML. You can also specify a ZIP file containing multiple XML files. By including a placeholder (*) in the file name you can import multiple files simultaneously. Specification: Mandatory if no data source is specified
-client <client name>	Name of the PPM client into which the data is to be imported. If no client is specified, the default client is used. Specification: Optional

Parameter	Description
-recoveryfile {yes|no}	Indicates whether the Data analytics relevant recovery files are to be saved after the import is complete. Default value: Yes Specification: Optional By default, all data (that is, all tables in all analysis realms) is saved, even if only one table is imported. If an import is canceled, a recovery file will not be created.
-version	Outputs the version without importing
-protocolfile <file name>	Name of the log file to which the program output is to be written Specification: Optional
-language <ISO code>	Language of data imported. The language is specified in ISO code. If no language is specified the current operating system language is used. Specification: Optional
-information {yes|no|default}	Specifies whether information messages are to be output. Specification: Optional
-warning {yes|no|default}	Specifies whether warnings are to be output. Specification: Optional
-error {yes|no|default}	Specifies whether error messages are to be output. Specification: Optional
-alltables	Indicates whether the action is to be performed for all tables of an analysis realm, for example, deletion of all tables. Specification is mandatory depending on the context
-reinitanalysisserver {yes | no | auto}	Indicates whether the analysis realm in the analysis server is to be reinitialized. With the **auto** setting, the analysis realm is reinitialized only if required. The **no** (default) setting prevents reinitialization in general. See also chapter Reinitialize analysis server [41].

VM settings

The program is started with its own VM settings configured in the batch file **_vmparam.bat**. This file is located in the folder **config/<client>/bin**.

Default values are

- LOCAL_PPM5_VM_MIN_MEM_RT_TABLEIMPORT=64
- LOCAL_PPM5_VM_MAX_MEM_RT_TABLEIMPORT=512

These values cannot be set in PPM CTK.

7.7.3. Formats supported

A system event file to be imported must use the formats supported by PPM, or a data format transformation must be configured. Use PPM CTK to configure the data format of the source data. The data format is transformed into the required PPM format during the import.

See also chapter Data format transformation [51].

PPM supports the following formats.

- Numbers of the **LONG** type
  must not contain commas (,) or periods (.) as separators (for example, 9000).
- Numbers of the **FLOAT** type
  must contain only a period (.) as a separator (for example, 3.14).
- Time and date formats supported
  - d.M.yyyy H:mm:ss
  - d.M.yyyy H:mm
  - d.M.yyyy H
  - dd.M.yyyy
  - w.'WEEK'.yyyy
  - M.'MONTH'.yyyy
  - M.'QUARTER'.yyyy
  - yyyy

The character strings **WEEK**, **MONTH**, and **QUARTER** are not interpreted. They only serve to interpret the numbers that follow them.

Please refer to this page for detailed explanations of the date formats:

http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html

The number of characters defines the minimum number of numbers. For example, the characters **dd** define a date with two numbers, such as 02, 08, 13, 25. The character **d**, however, defines a date with at least one number, such as 2, 8, 13, 25.

Each date format imported is set to the second-most detailed level of accuracy by selecting a neutral, smaller time unit for missing units. For example, if a month is missing, the value is set to January, and if a day is missing, the value is set to the first day of the month. Hours, minutes, and seconds are set to zero.

For imported time stamps, the time zone of the server applies.

Define custom time stamps

You can define additional time stamp formats. For the following keys common time stamp formats are specified in the file **AdapterConfig_settings.properties**.

- XML_DATEFORMAT
- XML_DATEFORMAT_ALTERNATIVE

These settings apply to the entire client, the XML import (graph format only), and all Data analytics data imports (including other tables, analysis realms, and Process analysis).

During date format conversion, the format stored in the **XML_DATEFORMAT** key is applied first. If this fails, the **XML_DATEFORMAT_ALTERNATIVE** is applied. If this fails, as well, the date is interpreted as a PPM default format. If no format can be applied, an error message is displayed and the event is skipped for import.

7.7.4. Number ranges supported

The following table describes the supported number ranges and accuracies.

Data type	Supported values
Floating point number	All finite numbers mappable in IEEE-754 double format **binary64**. Rounding to the next 64-bit binary number during import, if required.
Integer	All integer numbers between -9×10^{18} and 9×10^{18}. Values outside the supported range are rejected during the import and a corresponding error message is output.

7.7.5. Troubleshooting after data import was canceled

Before data import, the analysis realm tables involved are locked for the duration of the import procedure. If data of a locked table is queried, an error message is output. Import abort can result in an inconsistent analysis realm. The error message may continue to be displayed and querying the table data may continue to be impossible.

This behavior also affects the import of process data into process analysis, if they are associated with the process table of the Data analytics analysis realm.

Error importing a data table

- If no table key exists, it is possible to restore data consistency by reinitializing the analysis server [41] (runtableimport –reinitanalysisserver auto). You can also do this in combination with rerunning the data import that was previously canceled.
- If a table key exists, you can restore data only by importing and adding the same file while reinitializing it (runtableimport -add -reinitanalysisserver auto).

  If the original data no longer exists, you cannot restore that data. In that case you can either leave the data incomplete and reinitialize the analysis server, or delete the data [40].

Error importing process data

When the import of process data fails, run the **runppmimport** program.

- runppmimport -reinitanalysisserver auto

7.8. Save and restore data

Data analytics enables you to save all content and meta data of your analysis realms in preconfigured recovery files. This recovery files allow you to restore your data.

By default, the client recovery files are stored in the directory **\server\bin\work\data_ppm\recovery\<client>** of the PPM installation.

The path to the recovery files is defined by the key **RECOVERY_FOLDER** in the client-specific file **AnalysisServer_settings.properties**.

Save data

Saving the Data analytics recovery files is triggered automatically after import, if this behavior was not explicitly disabled by the **–recoveryfile no** parameter during import.

See chapter **runtableimport.bat [35]**

The process analysis recovery files are not affected by this.

With the **runppmadmin** program, you can create recovery files for the analysis server, regardless of the data import. Recovery files are created for all process analysis and Data analytics recovery files.

For detailed information on the **runppmadmin** program, please refer to the **PPM Operation Guide**.

Restore data

Data analytics data (together with the process data) is restored automatically whenever the analysis server is started.

To prevent data from being restored, you start the analysis server by adding the parameter **-startempty**.

7.9. Delete analysis realm and table data

Use the runtableimport.bat [35] program to delete data from individual tables or of an entire analysis realm.

Delete analysis realm data

To delete all data of an analysis realm (<realm name>), run runtableimport with the **-alltables** parameter as follows.

runtableimport.bat –user <user name> -password <password> -realm <realm name> -mode delete -alltables

Parameters **–file** and **–table** or **–datasource** are not allowed in this context.

Delete data of a table

To delete data of a table (<table name>) in an analysis realm (<realm name>), run runtableimport with the **-table** parameter as follows.

- runtableimport.bat –user <user name> -password <password> -realm <realm name> -table <table name> -mode delete

To delete data of an analysis realm table in the data source(<data source>), run runtableimport with the **-datasource** parameter as follows.

- runtableimport.bat –user <user name> -password <password> -realm <realm name> -datasource <data source> -mode delete

Parameters **–file** and **–alltables** are not allowed in this context. Parameters **–datasource** and **–table** cannot be run in parallel.

7.10. Reinitialize analysis server

Use the **runtableimport** program and the **-reinitanalysisserver** argument to reinitialize a specific analysis realm in the analysis server [41] or the entire analysis server [41] independently of an import.

During reinitialization, data saved in the database are loaded to the analysis server. Reinitialization eliminates inconsistencies from the analysis server's data when compared to the database, for example, after a failed data import.

It is possible to reinitialize individual analysis realms in the analysis server [41] or the entire analysis server [41]. It is impossible to reinitialize individual tables.

If an analysis realm table has a key [11], a canceled import can only be corrected by reimporting the same data into the table. Reinitialization is not sufficient in this case.

7.10.1. Reinitialize an individual analysis realm

You can reinitialize an individual analysis realm in the analysis server. This applies to the analysis realm configuration and the Data analytics data of all analysis realm tables of the analysis realm specified.

- runtableimport –user <user name> -password <password> -realm <realm name> -reinitanalysisserver [yes | no | auto]

With the **yes** parameter, the analysis server is reinitialized regardless of whether the configuration or data of the analysis realm are consistent or not. That way you can eliminate potential inconsistencies.

If you use the **auto** argument, the analysis server is reinitialized only if required. That way you can eliminate potential inconsistencies.

With the default setting **no** (default) you prevent reinitializing completely.

Dimensions in associations between processes and tables are reinitialized only if you run ppmimport –reinitanalysisserver yes (see Reinitialize entire analysis server [41]).

7.10.2. Reinitialize entire analysis server

By means of the following parameters you can reinitialize the entire analysis server including all analysis realms, table data, process instance-independent measures, and process data.

runppmimport.bat –user <user name> -password <password> -reinitanalysisserver yes|auto

- With reinitanalysisserver **yes**, all parts of the analysis server are unconditionally reinitialized.
- With reinitanalysisserver **auto**, only inconsistent parts of the analysis server are reinitialized.

Consistency of analysis realm tables is irrelevant for importing process instance-independent measures and process data.

Dimension tables used in associations between processes and data tables are reinitialized only with the **-reinitanalysisserver yes** parameter. If you are using the **-reinitanalysisserver auto** parameter, text dimension values that no longer exist in data rows may still be displayed for these dimensions in filter dialogs.

7.10.3. Automatically reinitialize analysis realm

With the **reinitanalysisserver auto** parameter an analysis realm is automatically reinitialized in the analysis server, if any of the following conditions apply.

- The analysis realm configuration is inconsistent.
- An analysis realm table is inconsistent.
- An analysis realm table is incomplete.

A warning is output, stating that an analysis realm is inconsistent or that an inconsistent or incomplete table was found. For a table with a key the data needs to be reimported.

7.11. XML configuration of analysis realms and data import

The Data analytics configuration includes the definition of data sources and the data import, as well as the specifications of the analysis realms based on the data source tables. The configuration is saved, imported, and exported in XML format.

With PPM CTK, you can configure your analysis realms [8]. The following XML specifications are for reference only. They are based on the configurations' XML representation.

7.11.1. Document type definition (DTD)

The following document type definition provides you with an overview of the configuration elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % paramset.dtd SYSTEM "paramset.dtd">
%paramset.dtd;
<!ELEMENT realm (descriptions?, tables?, criterion-definitions?, as-
sociations?, standardquery?)>
<!ATTLIST realm
keyword NMTOKEN #REQUIRED
>
<!ELEMENT descriptions (description+)>
<!ELEMENT tables (table | process-table)>
<!ELEMENT table (column+)>
<!ATTLIST table
keyword NMTOKEN #REQUIRED
>
<!ELEMENT process-table (process-type-filter)>
<!ATTLIST process-table
keyword NMTOKEN #REQUIRED
>
```

```
<!ELEMENT process-type-filter (filter?)>
<!ELEMENT column EMPTY>
<!ATTLIST column
keyword CDATA #REQUIRED
datatype (LONG|DOUBLE|TEXT|TIME|TIMESPAN|FACTORYTIMESPAN) #RE-QUIRED
>
<!ELEMENT criterion-definitions (kidef|oneleveldim|timedim|process-criterion)+>
<!ELEMENT process-criterion (description*) >
<!ATTLIST process-criterion
process-table-keyword NMTOKEN #REQUIRED
process-keyword NMTOKEN #REQUIRED
keyword NMTOKEN #REQUIRED
relation-name NMTOKEN #IMPLIED
>
<!ELEMENT kidef (description+)>
<!ATTLIST kidef
name NMTOKEN #REQUIRED
comment CDATA #IMPLIED
table NMTOKEN #REQUIRED
column CDATA #REQUIRED
distribution (TRUE|FALSE) "FALSE"
standarddeviation (TRUE|FALSE) "FALSE"
retrievertype (KEYINDICATOR|NUM_KEYINDICATOR) "KEYINDICATOR"
kigroup NMTOKEN #IMPLIED
internal (yes|no) "no"
assessment (POS|NEG) "NEG"
scale NMTOKEN #IMPLIED
refinement CDATA #IMPLIED
>
<!ELEMENT timedim (description+)>
<!ATTLIST timedim
name NMTOKEN #REQUIRED
comment CDATA #IMPLIED
table NMTOKEN #REQUIRED
column CDATA #REQUIRED
precision (DAY|HOUR|MINUTE) "DAY"
internal (yes|no) "no"
dimgroup NMTOKEN #IMPLIED
scale NMTOKEN #IMPLIED
refinement CDATA #IMPLIED
>
<!ELEMENT oneleveldim (description+, leveldesc)>
<!ATTLIST oneleveldim
name NMTOKEN #REQUIRED
comment CDATA #IMPLIED
internal (yes|no) "no"
dimgroup NMTOKEN #IMPLIED
>
<!ELEMENT leveldesc (dimitem)>
<!ELEMENT dimitem (description+, defaultvalue?)>
<!ATTLIST dimitem
table NMTOKEN #REQUIRED
column CDATA #REQUIRED
length CDATA #IMPLIED
>
```

```
<!ELEMENT defaultvalue (#PCDATA)>
<!ELEMENT associations (table-association+)>
<!ELEMENT table-association (criterion-association)>
<!ELEMENT criterion-association (criterion-reference, criterion-reference)>
<!ELEMENT criterion-reference EMPTY>
<!ATTLIST criterion-reference
name NMTOKEN #REQUIRED
>
<!ELEMENT standardquery (paramset)>
```

7.11.2. Configuration elements

At the main level, the **realm** element represents the analysis realm configuration. To identify the analysis realm, the element has a unique keyword. The configuration of the analysis realm comprises five XML elements: descriptions, tables, criterion-definitions, associations, and standardquery.

The **descriptions** element contains language-specific names and descriptions, which will be displayed in the interface. Each element to be displayed in the user interface must contain the relevant language-specific names and descriptions.

As in Process analysis, the **standardquery** element contains a default paramset and represents a default query in the analysis realm.

The remaining three main elements are described in detail in the following chapters.

7.11.2.1. tables

The **tables** element describes only the table structure of the analysis realm. This area contains all tables with the associated columns.

There are two table types. One is intended for Data analytics data only, the other contains only process data.

Each Data analytics table is specified by a unique keyword and contains only columns. Each column is specified by a unique keyword and a specific data type.

The following data types are supported:

- TEXT: Unidimensional text dimension
- LONG: Integer measure
- DOUBLE: Floating point measure
- TIME: Time stamp/time dimension
- TIMESPAN: Time interval
- FACTORYTIMESPAN: Time interval

  Transformation factors are included in measure calculation. Transformation factors are described in the file **<client>_transformationfactors.xml**.

Each process data table is identified by a unique keyword. In contrast to data tables, they contain a process type filter that selects a specific node of the process tree from process analysis, that is, all processes (root nodes), a process type group, or a process type. This renders all criteria registered at this process tree node usable for the configuration of process criteria in the Cross analytics analysis realm.

7.11.2.2. associations

The **associations** element describes how the individual tables are linked. Each table link describes the association between two tables and contains criterion associations. The latter describe the link between two columns of two tables.

7.11.2.3. criterion-definitions

The **criterion-definitions** element describes the criteria used in Data analytics. Four criteria are supported: Measures, one-level dimensions, time dimensions, and process criteria.

Each criterion except for the process criterion must include a unique reference to a column of a table in the analysis realm. The process criterion must specify the source criterion from the process analysis (by password and relation) and the target table in the analysis realm.

Data analytics table criteria

kidef element

```
<!ELEMENT kidef (description+)>
<!ATTLIST kidef
name NMTOKEN #REQUIRED
comment CDATA #IMPLIED
table NMTOKEN #REQUIRED
column CDATA #REQUIRED
distribution (TRUE|FALSE) "FALSE"
standarddeviation (TRUE|FALSE) #IMPLIED
retrievertype (KEYINDICATOR|NUM_KEYINDICATOR) "KEYINDICATOR"
kigroup NMTOKEN #IMPLIED
internal (yes|no) "no"
assessment (POS|NEG) "NEG"
scale NMTOKEN #IMPLIED
refinement CDATA #IMPLIED
>
```

The **kidef** element defines a measure. Attributes not listed here are described in the **kidef** and **useki** sections in the document type definitions **KeyindicatorConfiguration.dtd** and **KeyindicatorProcesstree.dtd**.

- name: Unique measure name
- table: Table containing data
- column: Column containing data
- internal: Measure not to be displayed in the analysis, such as the ID column for associations

timedim element

```
<!ELEMENT timedim (description+)>
<!ATTLIST timedim
```

```
name NMTOKEN #REQUIRED
comment CDATA #IMPLIED
table NMTOKEN #REQUIRED
column CDATA #REQUIRED
precision (DAY|HOUR|MINUTE) "DAY"
internal (yes|no) "no"
dimgroup NMTOKEN #IMPLIED
scale NMTOKEN #IMPLIED
refinement CDATA #IMPLIED
>
```

The **timedim** element defines a time dimension. Attributes not listed here are described in the **timedim** section in the document type definitions **KeyindicatorConfiguration.dtd** and **KeyindicatorProcesstree.dtd**.

- name: Unique dimension name
- table: Table containing data
- column: Column containing data

oneleveldim element

```
<!ELEMENT oneleveldim (description+, leveldesc)>
<!ATTLIST oneleveldim
name NMTOKEN #REQUIRED
comment CDATA #IMPLIED
internal (yes|no) "no"
dimgroup NMTOKEN #IMPLIED
>
```

The **oneleveldim** element defines a unidimensional text dimension. In Data analytics, the values of oneleveldim are simply keywords, not pairs consisting of key value and description. Attributes not listed here are described in the **oneleveldim** section in the document type definition **KeyindicatorConfiguration.dtd**.

Process table criteria (Cross analytics)

- process-table-keyword

  Analysis realm process table that the criterion is linked with.
- process-keyword

  Unique key referencing the source criterion imported from process analysis into the analysis realm.
- keyword

  The unique key of the criterion in the analysis realm. It does not have to match the key of the source criterion.
- relation-name

  Relation of the criterion imported into the analysis realm from process analysis

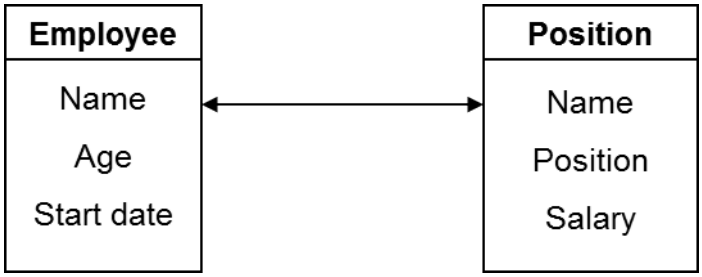
Process criteria do not require a description because it is inherited from the source process criterion.

7.11.2.4. Example

This chapter includes a brief configuration example for a simple analysis realm.

Example 38.

The analysis realm contains the two tables **Employee** and **Position**. The names in the table are unique and can be used as primary keys. The analysis realm scheme looks as follows.



The analysis realm is configured in XML as follows.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE realm SYSTEM "analysisrealm.dtd">
<realm keyword="Employee-Positions">
<descriptions>
<description
name="Employee positions"
language="de">Positionen der Mitarbeiter...
</description>
<description name="Employee positions" language="en">
Positions of employees...
</description>
</descriptions>
<tables>
<table keyword="employee">
<column keyword="name" datatype="TEXT"/>
<column keyword="age" datatype="LONG"/>
<column keyword="start_date" datatype="TIME"/>
</table>
<table keyword="position">
<column keyword="name" datatype="TEXT"/>
<column keyword="position" datatype="TEXT"/>
<column keyword="salary" datatype="DOUBLE"/>
</table>
</tables>
<criterion-definitions>
<kidef name="employee_age" table="employee" column="age"
comment="an optional comment"
distribution="FALSE"
standarddeviation=#IMPLIED
retrievertype="NUM_KEYINDICATOR" kigroup="SomeGroup"
assessment="POS">
<description name="Alter" language="de"></description>
<description name="Age" language="en"></description>
</kidef>
<kidef name="position_salary" table="position" column="salary"
```

```
comment="an optional comment" internal="no"
distribution="FALSE"
standarddeviation="FALSE"
retrievertype="NUM_KEYINDICATOR" kigroup="SomeGroup"
assessment="POS">
<description name="Gehalt" language="de"></description>
<description name="Salary" language="en"></description>
</kidef>
<oneleveldim name="employee_name"
comment="an optional comment" internal="no" dimgroup="SomeGroup">
<description name="Mitarbeitername" language="de"/>
<description name="Employee name" language="en"/>
<leveldesc>
<dimitem table="employee" column="name">
<description name="Mitarbeitername" language="de"/>
<description name="Employee name" language="en"/>
<defaultvalue>"SomeName"</defaultvalue>
</dimitem>
</leveldesc>
</oneleveldim>
<oneleveldim
name="position_name"
comment="an optional comment" internal="no" dimgroup="SomeGroup">
<description name="Mitarbeitername" language="de"/>
<description name="Employee name" language="en"/>
<leveldesc>
<dimitem table="position" column="name">
<description name="Mitarbeitername" language="de"/>
<description name="Employee name" language="en"/>
<defaultvalue>"SomeName"</defaultvalue>
</dimitem>
</leveldesc>
</oneleveldim>
<oneleveldim name="position_title"
comment="an optional comment" internal="no" dimgroup="SomeGroup">
<description name="Position" language="de"/>
<description name="Position" language="en"/>
<leveldesc>
<dimitem table="position" column="title">
<description name="Position" language="de"/>
<description name="Position" language="en"/>
<defaultvalue>"Software Developer"</defaultvalue>
</dimitem>
</leveldesc>
</oneleveldim>
<timedim name="start_date" table="employee" column="start_date"
comment="optional comment" precision="DAY" scale="DAYSCALE"
refinement="BY_MONTH" dimgroup="SomeGroup">
<description name="Anfangsdatum" language="de"></description>
<description name="Start date" language="en"></description>
</timedim>
</criterion-definitions>
<associations>
<table-association>
<criterion-association>
```

```
<criterion-reference name="employee_name "/>
<criterion-reference name="position_name"/>
</criterion-association>
</table-association>
</associations>
</realm>
```

7.11.3. Data sources

You can import data from various sources of the **EVENT**, **JDBC**, **SAP**, and **CSV** type for Data analytics. The data source for Data analytics is identified by a special attribute in the relevant XML file.

```
<!ATTLIST datasource
...
analysistype ( PROCESS | DATA_ANALYTICS) "PROCESS"
...
>
```

To import data for Data analytics, the attribute must have the value **DATA_ANALYTICS**. Otherwise, it will be interpreted as a data source for Process analysis.

The **runtableimport** program also extracts two other fields from the data source.

- XML import file path
  <!ELEMENT data (#PCDATA)>
- **tablename** attribute of the **realmtable** element
  <!ELEMENT realmtable (sourcetable?)>
  <!ATTLIST realmtable
  tablename CDATA #REQUIRED
  >

The **tablename** attribute contains the table name specified in the configuration of the analysis realm including the data source. The data referenced in the data source is imported into this table.

7.11.4. XML event format

Only data in XML system event format can be imported for data analysis.

Existing meta data in the XML import file is ignored.

Each system event represents the data of one table row.

```
<event>
<attribute type="VBTYP">M</attribute>
<attribute type="POSNR">000010</attribute>
<attribute type="VBELN">0090018105</attribute>
</event>
<event>
```

```
<attribute type="VBTYP">C</attribute>
<attribute type="POSNR">000010</attribute>
<attribute type="VBELN">0090018147</attribute>
</event>
```

The attribute type corresponds to the column name.

Example 39.

In the example, columns named **VBTYP**, **POSNR**, and **VBELN** are created. The two events result in the following data series which will be imported in the analysis server.

VBRK-VBTYP	POSNR	VBELN
M	000010	0090018105
C	000010	0090018147

If a value is missing or an attribute contains an empty string, the analysis server uses a default value instead. The default value can be changed for text dimensions. Measures and time dimensions have the default value **Not maintained**.

```
<event>
<attribute type="VBTYP">M</attribute>
<attribute type="POSNR">000010</attribute>
</event>
<event>
<attribute type="VBTYP">C</attribute>
<attribute type="POSNR"></attribute>
<attribute type="VBELN">0090018147</attribute>
</event>
```

VBTYP	POSNR	VBELN
M	000010	<DEFAULT_VALUE>
C	<DEFAULT_VALUE>	0090018147

If an attribute is used multiple times, one of the values is randomly selected for import.

```
<event>
<attribute type="VBTYP">M</attribute>
<attribute type="POSNR">000010</attribute>
<attribute type="VBELN">0090018105</attribute>
<attribute type="VBELN">0090018106</attribute>
<attribute type="VBELN">0090018107</attribute>
</event>
<event>
<attribute type="VBTYP">C</attribute>
<attribute type="POSNR">000010</attribute>
<attribute type="POSNR">000020</attribute>
<attribute type="POSNR">000030</attribute>
<attribute type="VBELN">0090018147</attribute>
</event>
```

VBTYP	POSNR	VBELN
M	000010	0090018105
C	000030	0090018147

During import, the analysis realm configuration is used for identifying the columns. Only attributes whose type is the same as a column name are imported. If an event does not contain any attribute types or contains attribute types not listed in the table configuration an empty table row is imported. Text dimensions receive a definable default value, and measures and time dimensions receive the value **Not maintained**.

After the import, the following values are output on the console:

- Number of rows imported
- Number of empty rows imported
- Number of rows imported with at least one empty cell (including the completely empty rows)

XML system event files can be imported in ZIP format, as well. If the ZIP file contains only one XML file, the file is imported by default. If the ZIP files contains multiple files the column names are identified only once during the import of the first XML file. Therefore, all XML files in a ZIP file should contain the same columns.

The XML files are processed according to their order in the ZIP file.

7.11.5. Data format transformation

7.11.5.1. datasource.dtd extension

The file **datasource.dtd** was extended for defining the data format transformation.

```
…
<!ENTITY % description.dtd SYSTEM "_description.dtd">
%description.dtd;
<!--
       attribute transformations
-->
<!ENTITY % attributetransformation.dtd SYSTEM "_attributetransformation.dtd">
%attributetransformation.dtd;
…
<!ELEMENT realmtable (sourcetable?, realmvaluetransformations?)>
…
<!ELEMENT realmvaluetransformations (realmvaluetransformation+)>
<!ELEMENT realmvaluetransformation (eventattribute, transformation)>
<!ELEMENT eventattribute (#PCDATA)>
```

New DTD **_attributetransformation.dtd**

```
…
<!ELEMENT transformation EMPTY>
<!ATTLIST transformation
       type NMTOKEN "timestamp"
```

```
        format CDATA #REQUIRED
>
```

For an analysis realm data source, the transformation of the data format can be defined using the XML element **realmvaluetransformations**. At least one **realmvaluetransformation** element must be defined, comprising the elements **eventattributetype** and **transformation**.

The **eventattributetype** element contains the type of the attribute in the XML source event, file to which the transformation belongs.

The transformation rule consists of the transformation type (default: **timestamp** because this is transferred from the file **eventmapping.dtd**). For each event attribute type, only the **realmvaluetransformation** element can be defined.

The **format** element defines the format to be used for transformation.

7.11.5.2. Example: The 'timestamp' transformation type

By means of the transformation type **timestamp** you can transform imported values into a valid PPM data format of the **TIME** type.

Example

```
                  <event>
<attribute type="VBTYP">C</attribute>
<attribute type="POSNR"></attribute>
<attribute type="VBELN">0090018147</attribute>
<attribute type="ERZET">20121105160956</attribute>
</event>
```

Used with the data source configuration:

```
…
<realmvaluetransformations>
<realmvaluetransformation>
<eventattributetype>ERZET</eventattributetype>
<transformation type="timestamp" format="yyyyMMddHHmmss"/>
</realmvaluetransformation>
</realmvaluetransformations>
...
```

For Data analytics, the value **05.11.2012 16:09:56** is imported into the column **ERZET** of the analysis realm table.

7.11.6. Standard deviation and percentiles

From PPM version 9.7, you can statistically evaluate data in Data analytics and display the corresponding standard deviation and percentiles, such as median, in the analysis.

Standard deviation is available by default for measures with the **DOUBLE**, **TIMESPAN**, and **FACTORYTIMESPAN** data types. Data type **LONG** is not supported, the option is disabled. If you want

to deactivate standard deviation for a measure, set the **standarddeviation (TRUE|FALSE)** attribute in the criterion-definitions [45] configuration element to FALSE.

Percentiles can be calculated and displayed in PPM for all measures.

8. Support and legal information

This section provides you with some general information regarding product support and legal aspects.

8.1. Documentation scope

The information provided describes the settings and features as they were at the time of publishing. Since documentation and software are subject to different production cycles, the description of settings and features may differ from actual settings and features. Information about discrepancies is provided in the Release Notes that accompany the product. Please read the Release Notes and take the information into account when installing, setting up, and using the product.

If you want to install technical and/or business system functions without using the consulting services provided by SAG ARIS GmbH, you require extensive knowledge of the system to be installed, its intended purpose, the target systems, and their various dependencies. Due to the number of platforms and interdependent hardware and software configurations, we can describe only specific installations. It is not possible to document all settings and dependencies.

When you combine various technologies, please observe the manufacturers' instructions, particularly announcements concerning releases on their Internet pages. We cannot guarantee proper functioning and installation of approved third-party systems and do not support them. Always follow the instructions provided in the installation manuals of the relevant manufacturers. If you experience difficulties, please contact the relevant manufacturer.

If you need help installing third-party systems, contact your local ARIS sales organization. Please note that this type of manufacturer-specific or customer-specific customization is not covered by the standard ARIS software maintenance agreement and can be performed only on special request and agreement.

8.2. Support

Product support

We provide support for ARIS products to all customers with a valid support contract.

Information on product version availability and support, as well as sustained support dates, is available at ARIS Community.

Contact our Global Support services at ARIS Community to raise and update support incidents, or post ideas and feature requests.

Community

Register with ARIS Community to download products, updates and fixes, find expert information, and interact with other users.

Documentation

For information on how to use and activate products and for other technical support questions, visit the ARIS Documentation website.

If you want to give feedback on any documentation topic, write to documentation@aris.com.

8.3. Legal notices

This document applies to ARIS Process Performance Manager Version 10 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2000-2026 SAG Aris GmbH, Saarbrücken, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name ARIS and all SAG Aris GmbH product names are either trademarks or registered trademarks of SAG Aris GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

The Software may contain or be distributed with certain commercial third party components (“Third Party Components”), which may be subject to special license terms and conditions (“Third Party Terms”) located at https://aris.com/terms.

Third Party Terms include important licensing and warranty information and disclaimers of third party licensors. For the avoidance of doubt, if Customer uses the Software in accordance with the Documentation, the Third Party Terms do not restrict the rights granted under this Agreement. In the event that a third party manufacturer no longer provides active support for any Third Party Component, Supplier will use commercially reasonable efforts to respond to any issues with the Software related to the Third Party Components.

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with SAG Aris GmbH and/or its subsidiaries and/or its affiliates and/or their licensors.