PPM DATABASE SYSTEMS

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 4
2. PPM database schema 5
2.1. Measure and dimension names 5
2.2. Database tables 5
2.3. PPM and database interaction 6
2.3.1. PPM server access to the database 6
2.3.1.1. Connection to the database 6
2.3.1.2. Database user 7
2.3.2. Database objects 7
2.3.2.1. Handling of NULL values 8
2.3.2.2. Transactions 8
2.3.3. Data import 9
2.3.3.1. Import fragments 9
2.3.3.2. Create process instances and calculate measures 9
2.3.4. Data analysis 10
2.4. Tablespaces 10
2.4.1. Tablespace types 10
2.4.2. Tablespace configuration 11
3. Supported database systems 12
3.1. Oracle 12
3.1.1. JDBC driver 13
3.1.2. Create a database user 13
3.1.3. Tablespace configuration 14
3.1.3.1. Oracle 19 14
3.1.3.2. Oracle 18 14
3.1.3.3. Oracle 12c 15
3.1.3.4. Oracle 11g 15
3.2. IBM DB2 16
3.2.1. DB2 16
3.2.2. Create a database user 16
3.2.3. Tablespace configuration 16
3.2.3.1. DB2 16
3.3. Microsoft SQL Server 17
3.3.1. JDBC driver 17
3.3.2. Create a database user 17
3.3.3. Tablespace configuration 19
3.3.4. Create a database 19
3.3.5. Unicode support 21
3.3.6. Migration 22
4. Performance tuning 23
4.1. Overall performance 23
4.1.1. Hardware-related 23
4.1.2. Configuration-related 24
4.2. Import performance 24
5. Support and legal information 25
5.1. Documentation scope 25
5.2. Support 25
5.3. Legal notices 26

1. General

This guide describes the database-related context pertaining to PPM server and the database system used. You should have basic knowledge of database technology and ARIS Process Performance Manager.

PPM uses an SQL RDBMS as the repository in which all configurations and data are saved. PPM has been developed in Java as a client-server application.

This user guide does not provide installation or customizing instructions for any of the database systems supported by ARIS Process Performance Manager. It intends to help you identify and optimize performance-critical scenarios.

2. PPM database schema

PPM uses the database for the persistent storage of all configurations, administrative settings, and imported data. It is the counterpart of the volatile, main-memory based analysis server that contains the analysis structures. In case of data loss in the analysis server, you can completely restore the analysis server from the database. Therefore, for a PPM system backup it is sufficient to back up only the PPM database schema or user.

2.1. Measure and dimension names

The internal names of measures and dimensions are also used for internal allocation of configuration elements in the database. Therefore, these names are subject to the restrictions posed by the underlying database system.

The measure and dimension names displayed on the PPM user interface are specified in the measure configuration (**<description>** XML elements) and depend on the interface language.

Please take the following information into account when assigning internal measure and dimension names:

- Names must begin with a letter.
- Please use capital letters only.
- The only special character allowed is the underscore (no umlauts).
- Assign brief names, avoid long names. The maximum name length allowed depends on the database. We recommend a maximum name length of 25 characters.

Example 1.

The following table shows a few measure and dimension names:

Type	Name	Description
Measure	PNUM	Number of processes
Dimension	D_MATERIAL	Material, two-level
Dimension	TIME	Time dimension

2.2. Database tables

PPM differentiates between database tables with a fixed name and tables whose names are specified by the measure configuration. The following two tables provide a few examples.

Configuration-based table names

Table name	Description
ATTR_INFO	Imported attributes

Table name	Description
DBVERSIONNUMBER	Version number of the database schema
XML_CONFIGS	Imported configurations (**runinitdb** or **runppmconfig**)
EPK_IMPORT_TBL	Imported fragments
EPK_TBL	Consolidated process instances

Configuration-based table names

Table name	Description
PC_UMG_DASHBOARD	Imported data of the process instance-independent measure series of Performance Dashboard

2.3. PPM and database interaction

In this chapter, we describe the types of database access that the PPM software performs when importing and analyzing data. To grasp the content of this chapter, you should be familiar with the basic PPM functions and operation.

2.3.1. PPM server access to the database

To access the database, the PPM server uses the standardized JDBC (Java DataBase Connectivity) database interface. Therefore, the database type used is almost completely transparent for the PPM server. However, database manufacturer-specific differences can lead to the PPM server generating different database queries for the same task.

2.3.1.1. Connection to the database

Each database supported by PPM can be accessed via JDBC. The JDBC driver consists of one or multiple Java archive files (jar). Copy the required JDBC driver archive(s) to the following directory:

<installation **directory>\ppm\server\bin\work\data_ppm\drivers**.

The drivers in this directory are shared across all PPM clients in the installation. A PPM server restart is required after adding or updating drivers.

Warning
Do not use the ACC enhance command to deploy JDBC drivers. The enhance command applies drivers only to individual runnables, which can lead to inconsistencies and driver conflicts when multiple clients or runnables are used. Always use the shared drivers directory described above.

Using ACC to add JDBC drivers

Another option to apply JDBC database drivers or SAP® drivers to the PPM installation is to use the ARIS Cloud Controller (ACC). Using the **enhance** command in ACC will also copy the given drivers to the same directory as mentioned above:

<installation directory>\ppm\server\bin\work\data_ppm\drivers.

Example 2.

enhance <client id> with dbDriver local file <path to DB driver>

Please note that you need to enter path data in Java standard. This means that you need to use "/" instead of "\" even under Windows operating systems. Furthermore, a valid and already installed client needs to be available in the PPM installation. Use the ID of the client server to add the drivers. For example, if you have created a client named **umg_en**, use **umg_en_cs** as an ID for the enhance command. The copied drivers apply to all installed clients in the installation.

Different databases in one PPM installation

In a PPM installation, you can address database systems of different manufacturers for different clients. The relevant Java archive files (jar) must be copied manually to **<installation directory>\ppm\server\bin\work\data_ppm\drivers**.

Only one JDBC driver can be specified for a particular database type. It is impossible to address different versions of a database type, for example, Oracle 12 and Oracle 19 with different JDBC driver versions.

2.3.1.2. Database user

For each PPM client, you need a single schema of a dedicated database user. The database user requires unrestricted access to the objects of their schema.

How to create such a DB user for the supported database systems is described in detail in the corresponding sections of the supported databases.

- Create an Oracle DB user. [13]
- Create an IBM DB2 user. [16]
- Create a Microsoft SQL user. [17]

In the database-specific **<RDBMS>_USE_CASE_SENSITIVE_USERNAME** key of the client-specific **Database_settings.properties** configuration file, you can specify if the database system is to consider case-sensitive spelling of user names (value **true**) or not (value **false**). The default value is **false**.

2.3.2. Database objects

Regardless of the database system used, the PPM server needs the following database objects:

- Tables and constraints

- Sequences (if supported by DB system, otherwise the PPM server simulates a similar functionality)
- Indices
- Foreign keys

To keep up the performance of your PPM system, it is recommended that you have database system statistics calculated on a regular basis. The administrator of your database is the best person to take care of this.

If you use Oracle as a database system for PPM, you can calculate the object statistics using the **-genstats** argument of the **runppmimport** command line program during data import. Further information is available in the command line program's help which you call up with **runppmimport -h**. Particularly for newer Oracle versions, we recommend administrative statistics calculation instead of the PPM-controlled calculation.

2.3.2.1. Handling of NULL values

There is a technical difference between the statements **value does not exist** (NULL value) and **value is not specified** (string with the length null). PPM database extractors can differentiate between NULL values and empty strings.

However, the database systems supported by PPM behave differently. When extracting source system data using PPM extractors, they can exhibit different reactions to NULL and "".

Oracle

Data fields containing empty strings or without content are read as NULL.

SQL Server and DB2

Empty strings are returned when inserting/updating.

Missing columns are returned as NULL for INSERT.

2.3.2.2. Transactions

The PPM server is transaction-controlled on the database. Database queries are performed within a chain of command and finally applied (committed) in the database schema. This procedure is called transaction. Initially, transactions are executed temporarily in the runtime environment of the database system. Transactions that are not yet completed can be canceled (rolled back).

Transaction-controlled access to the database requires a minimum amount of system resources of the database instance.

However, the database systems supported by PPM behave differently in terms of transaction handling. The PPM application server optimally exploits the different transaction handling of the database systems supported.

Example 3.

- When using an Oracle-based database schema, the PPM administration components display a **Save** button. After saving, any changes apply immediately.
- When using a DB2- or SQL server-based database schema, analysis options are limited by the **READ_LOCK** transaction logic.

2.3.3. Data import

This chapter describes database-related processes when importing process instance data. Data import consists of two consecutive phases:

1. Import of fragments from application systems [9]
2. Creation and calculation of process instances [9]

2.3.3.1. Import fragments

Regardless of the data import format (system event format or graph format), process keys, hierarchy keys and shared fragment keys are calculated for the fragment instances to be imported. It is assumed that the configuration of the relevant rules is valid. If at least one process key was calculated, the imported fragment is saved in the **EPK_IMPORT_TBL** database table. Complete process instances are saved directly, that is, without calculating a process key.

Complete process instances are considered done. They can no longer be extended by merging them with imported fragments. A complete process instance is identified by the **AT_EPK_KEY** process instance attribute.

2.3.3.2. Create process instances and calculate measures

Copy EPC phase

In this step, the imported fragments are exported from the **EPK_IMPORT_TBL** table and written to the **EPK_TBL** table. The calculated values of all process-related keys (process, shared fragment, hierarchy keys) are saved with foreign key relation to the corresponding fragment in certain tables. Exactly to which tables they are saved depends on the state of the database, for example, if process instances already exist in the database. Subsequently, the fragment is deleted from the **EPK_IMPORT_TBL**.

Merger phase

Fragments with identical process key are combined in one process instance. Two fragments are processed for this, that is, all objects and connections of the fragment with the older import time stamp are copied to the fragment with the more recent import time stamp. This process is repeated until only unique process keys exist in the database.

Hierarchies phase

The sequence of calculation of hierarchically dependent process instances is specified and the state of the references is updated.

Measure calculation phase

In this phase, process instances are typified and calculated. Fragments and keys no longer needed in the database are deleted and the calculated process instances are written back into the **EPK_TBL**.

Consolidating fragments into process instances requires the process keys of all imported fragments to be held in the main memory so that they can be processed simultaneously. If you import a large number of fragments in one import process, the main memory size might not be sufficient. Use the **<IMPORT_SCENARIO>_READ_RATE_EPC** key of the **EpkImport_settings.properties** configuration file to set the maximum number of fragment instances to be imported in one process step. The default value is 100000. See also **PPM Data Import.pdf** for details on the various import scenarios provided by PPM.

2.3.4. Data analysis

Once the processes have been calculated during measure calculation and the resulting measures and dimensions have been transferred to the analysis server, the process instances are ready to be analyzed.

2.4. Tablespaces

All database systems supported by PPM use particular areas on a data carrier for permanent data storage. These areas are called **tablespaces** for Oracle and DB2 databases. Microsoft uses the term **filegroups** for its SQL server products. In the following, we will use tablespace as a uniform term for both.

The PPM system uses different classes of data:

- Imported fragment instances and process instances (binary data objects)
- Administration and structure information
- Database indices

The first step in optimizing the performance of your PPM is assigning individual tablespaces to the various data classes.

2.4.1. Tablespace types

In the client-specific **Database_settings.properties** configuration file, you specify which tablespaces are to save which PPM data.

The **Database_settings.properties** configuration file is stored in the following client-specific directory.

<PPM installation>\server\bin\work\data_ppm\config\<client>\

PPM differentiates between the following types of tablespaces:

Tablespace name	Description
STDBLOB	All database tables with binary data are saved in this tablespace, for example, the table of calculated process instances.
STDTABLE	Standard tablespace. In this tablespace, all data is saved that uses default data types of the database system.
STDINDEX	In MS SQL Server and Oracle systems, this tablespace is used for storing primary keys and index information.

Tablespace memory requirements strongly depend on the configuration and imported data of the PPM client.

If you do not change the default configuration after creating a client, all data of the PPM client is saved in the default tablespace. The default tablespace is specified upon creation of the database user. This configuration is highly unsuitable for a productive system.

2.4.2. Tablespace configuration

The assignment of database tables to tablespaces is specifically indicated in the **Database_settings.properties** client configuration file. General syntax:

<Database type>_TBLCONF_<Tablespace name> = Value

- Database type specifies one of the database types supported. Valid values: **ORACLE_11, ORACLE_12**, **ORACLE_18**, **ORACLE_19**, **DB2_10**, **DB2_11**, **SQL_Server_2016**, **SQL_Server_2017**, and **SQL_Server_2019**

If you are working with an **SQL Server** database type for a multi-byte client, you need to use one of the additional Unicode variants **SQL_Server_2016_UNICODE**, **SQL_Server_2017_UNICODE**, or **SQL_Server_2019_UNICODE**

The syntax of the tablespace specified by "value" is database-specific, too, and is described in the corresponding subchapters of the chapter on Supported database systems [12].

3. Supported database systems

This chapter describes the administrative differences between the various database systems supported by PPM.

PPM supports the following database systems:

- Oracle
- IBM DB2
- Microsoft SQL Server
  All versions in UNICODE as well, if data is to be saved in a multi-byte character set, such as Japanese.

You can find detailed version information for each platform in the ARIS System Requirements. You can find the ARIS System Requirements on the ARIS Documentation website.

When creating a client, specify the database system to be used. The database system-specific settings are specified in the subsequent configuration dialogs: Host name, port number of the database service, name of the database, and for the PPM client the name of the database user to be used, and the database user password.

The database systems supported by PPM use the term "database" with different semantics:

Oracle

Oracle links runtime processes and the database in one database instance. In their daily language use, they mostly use the term "database".

IBM DB2 and Microsoft SQL Server

With these database systems, the runtime processes of the database instance can manage any number of databases. At least the system database must exist. Microsoft calls the system database of its SQL Server products **master database**.

For **IBM DB** and **Microsoft SQL Server** database systems, we recommend that you supply each PPM client with its own database. This applies particularly to the database users required in a scaled PPM system.

3.1. Oracle

The default database of the latest Oracle releases (like 18 or 19) is a container database, which is not supported by PPM by default. A single instance database which is not configured as a container needs to be used for PPM.

To access the Oracle database, PPM uses the JDBC Thin Interface (type 4). Enter the access parameters for the Oracle database in the **URL** key of the client-specific **Database_settings.properties** configuration file.

If a single instance database is used, the syntax is as follows:

jdbc:oracle:thin:@//<host>:<port>/<database SID>

Example 4.

URL= jdbc:oracle:thin:@//pcoracle:1521/PPM

You must enter this type of URL manually in the **Database settings** dialog in PPM Customizing Toolkit.

3.1.1. JDBC driver

The JDBC drivers of newer Oracle versions are usually downward compatible. Please refer to the notes provided by the manufacturer. By default, the driver is located in the **jdbc\lib** subdirectory of your Oracle installation.

> **Warning**
> The PPM server needs a JDBC driver that implements the **JDBC-3** features used. The JDBC drivers contained in the **classes*.zip** files are obsolete and unsuitable for operation with PPM. Please use the latest **ojdbc17.jar** archive for all supported Oracle systems.

Note

With PPM version 10.5.11 and higher Java 21 Runtime environment is used. Hence, also the Oracle JDBC drivers need to be used suitable and certified for Java 21. Ojdbc8.jar versions are certified for Java 8, ojdbc11.jar is certified for Java 11 and ojdbc17.jar is certified for Java 17 and Java 21.

Copy the driver to <installation directory>\ppm\server\bin\work\data_ppm\drivers.

When starting, the PPM server outputs the interface type used as well as the exact database version and JDBC driver version to a log file or in the command line.

3.1.2. Create a database user

You can easily create a database user for PPM using suitable administration components, such as **Database control** or **EM Database Express**.

Note that user names are case sensitive in Oracle 18 and 19. PPM supports only upper case user names.

Assign the database user the roles **CONNECT** and **RESOURCE** and ensure that the user has sufficient privileges for accessing tablespaces.

Alternatively, you can create the user with an SQL script. Provided that the **PPMDATA** default tablespace and the **TEMP** tablespace exist, you can use the following script to create database users for PPM:

```
-- USER SQL
CREATE USER "UMG_EN" IDENTIFIED BY "umg_en"
DEFAULT TABLESPACE "PPMDATA"
TEMPORARY TABLESPACE "TEMP";

-- ROLES
GRANT "CONNECT" TO "UMG_EN" ;
GRANT "RESOURCE" TO "UMG_EN" ;
```

```
-- SYSTEM PRIVILEGES
GRANT UNLIMITED TABLESPACE TO "UMG_EN" ;
```

Warning
Please avoid using system tablespaces for a PPM database user.

3.1.3. Tablespace configuration

The tablespaces described in the chapter on Tablespaces [10] are configured as follows:

3.1.3.1. Oracle 19

Tablespace name	Key
Default	ORACLE_19_TBLCONF_STDTABLE
Binary data	ORACLE_19_TBLCONF_STDBLOB
Indices	ORACLE_19_TBLCONF_STDINDEX

The tablespace is specified by the following syntax:

Key = TABLESPACE <name>

Example 5.

Database_settings.properties file extract:

ORACLE_19_TBLCONF_STDTABLE=TABLESPACE PPMDATA

ORACLE_19_TBLCONF_STDINDEX=TABLESPACE PPMDATAIDX

ORACLE_19_TBLCONF_STDBLOB=TABLESPACE PPMBLOB

Please note that the default database in an Oracle 19 installation is a container database that is not supported by PPM by default. User names are case sensitive in Oracle 19 and only upper case user names are supported by PPM.

3.1.3.2. Oracle 18

Tablespace name	Key
Default	ORACLE_18_TBLCONF_STDTABLE
Binary data	ORACLE_18_TBLCONF_STDBLOB
Indices	ORACLE_18_TBLCONF_STDINDEX

The tablespace is specified by the following syntax:

Key = TABLESPACE <name>

Example 6.

Database_settings.properties file extract:

ORACLE_18_TBLCONF_STDTABLE=TABLESPACE PPMDATA

ORACLE_18_TBLCONF_STDINDEX=TABLESPACE PPMDATAIDX

ORACLE_18_TBLCONF_STDBLOB=TABLESPACE PPMBLOB

3.1.3.3. Oracle 12c

Tablespace name	Key
Default	ORACLE_12_TBLCONF_STDTABLE
Binary data	ORACLE_12_TBLCONF_STDBLOB
Indices	ORACLE_12_TBLCONF_STDINDEX

The tablespace is specified by the following syntax:

Key = TABLESPACE <name>

Example 7.

Database_settings.properties file extract:

ORACLE_12_TBLCONF_STDTABLE=TABLESPACE PPMDATA

ORACLE_12_TBLCONF_STDINDEX=TABLESPACE PPMDATAIDX

ORACLE_12_TBLCONF_STDBLOB=TABLESPACE PPMBLOB

3.1.3.4. Oracle 11g

Tablespace name	Key
Default	ORACLE_11_TBLCONF_STDTABLE
Binary data	ORACLE_11_TBLCONF_STDBLOB
Indices	ORACLE_11_TBLCONF_STDINDEX

The tablespace is specified by the following syntax:

Key = TABLESPACE <name>

Example 8.

Database_settings.properties file extract:

ORACLE_11_TBLCONF_STDTABLE=TABLESPACE PPMDATA

ORACLE_11_TBLCONF_STDINDEX=TABLESPACE PPMDATAIDX

ORACLE_11_TBLCONF_STDBLOB=TABLESPACE PPMBLOB

3.2. IBM DB2

3.2.1. DB2

The PPM server can access a DB2 database via interface type 4. Depending on the database version used, you need corresponding JDBC drivers. The required files are located in the **java** subdirectory of your DB2 installation.

To access a DB2 server, you need the latest supported JDBC driver. Usually, it is a single archive for the type 4 driver: **db2jcc4.jar**. You can download the required file from the manufacturer's homepage as mentioned above.

Note

With PPM version 10.5.11 and higher, the Java 21 Runtime environment is used. For IBM DB2 supported databases, use the latest available JDBC driver, which is also certified with Java 21.

When starting, the PPM server outputs the exact versions of the database and the JDBC driver. In the log file or on the command line, messages like this are displayed:

```
...Establishing connection between user UMG_EN and jdbc:db2://ppmdbsrv:50001/
PPM...
...Database version used: SQL09013.
...JDBC driver used: IBM DB2 JDBC Universal Driver Architecture (3.4.65).
```

3.2.2. Create a database user

DB2 databases use the user management of the operating system. After the installation of the DB2 database software, the **DB2ADMIN** operating system user and the **DB2USERS** group already exist. To generate a database user for PPM, you create a corresponding operating system user and add this user to the **DB2USERS** group. Through the group membership, the new operating system user receives the privileges required for accessing the DB2 database. The minimum database-related system privileges are: **CONNECT**, **CREATETAB**, and **IMPLICITSCHEMA**.

3.2.3. Tablespace configuration

For DB2, the tablespaces described in the chapter on Tablespaces [10] are configured as follows:

3.2.3.1. DB2

Tablespace name	Key
Default	DB2_11_TBLCONF_STDTABLE
Binary data	DB2_11_TBLCONF_STDBLOB

For DB2, the locations where the table indices are saved must be specified when creating the table. The tablespace is specified by the following syntax:

Key = IN <Tablespace name> INDEX IN <Index tablespace name>

The keyword **DB2_<version>_TBLCONF_STDINDEX** of the configuration file **Database_settings.properties** is not supported for DB2.

Example 9.

File extract Database_settings.properties

```
DB2_11_TBLCONF_STDTABLE=IN PPMDATA INDEX IN PPMINDX
DB2_11_TBLCONF_STDBLOB=IN PPMBLOB INDEX IN PPMINDX LONG IN PPMLOB
```

The mandatory extension **LONG IN** <tablespace name> determines the tablespace in which the binary database objects will be saved, all others are saved in the tablespace identified with the keyword **IN**.

3.3. Microsoft SQL Server

PPM supports various versions of **Microsoft SQL Server**. The versions differ in terms of behavior and administration.

Detailed information on the **Microsoft SQL Server** versions supported is available in the current ARIS System Requirements.

The following steps show an example of how to configure Microsoft SQL Server for PPM.

3.3.1. JDBC driver

To access a Microsoft SQL Server, you need the required JDBC driver. This driver is available for download on the Microsoft homepage.

When starting, the PPM server outputs the interface type used as well as the exact database version and JDBC driver version.

Note

With PPM version 10.5.11 and higher, the Java 21 Runtime environment is used. For Microsoft SQL Server supported databases, use the latest available JDBC driver, which is also certified with Java 21.

3.3.2. Create a database user

To create a database user for MS SQL Server, you can either use the SQL Server Management Studio on the server or remotely on your Desktop, or you can use the SQL command line program

sqlcmd

on the server directly.

Using the SQL command line tool you need to connect to the relevant database as database administrator (sa):

```
sqlcmd -d <database name> -U sa
```

For both programs, you can create a valid database user using the following commands:

```
CREATE LOGIN [<DB User>] WITH PASSWORD=N'<some password>',
DEFAULT_DATABASE=[<DB Name>], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF

CREATE USER [<DB User>] FOR LOGIN [<DB User>]

ALTER USER [<DB User>] WITH DEFAULT_SCHEMA=[<Schema>]

CREATE SCHEMA [<Schema>] AUTHORIZATION [<DB User>]
GRANT CREATE SCHEMA TO [<DB User>]
GRANT CREATE TABLE TO [<DB User>]
GRANT CREATE VIEW TO [<DB User>]
```

The database user name and schema name should be identical and always created in upper case letters.

If the user account already exists, you can delete it using the following sequence of commands:

```
DROP SCHEMA <schema name>
DROP USER <PPM user name>
DROP LOGIN <PPM login name>
```

Example 10.

You want to create a login and user called

PPMUSER

in the existing MS SQL Server database

PPMDB

. Within the MS SQL Management Studio, open a query editor and add the following command sequence:

```
USE [PPMDB]
GO
DROP SCHEMA <ppmschema>;
DROP USER ppmuser;
DROP LOGIN ppmuser;
GO
USE [PPMDB]
GO
CREATE LOGIN [PPMUSER] WITH PASSWORD=N'ppmuser', DEFAULT_DATABASE=[PPMDB],
CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO
USE [PPMDB]
GO
CREATE USER [PPMUSER] FOR LOGIN [PPMUSER]
GO
USE [PPMDB]
GO
ALTER USER [PPMUSER] WITH DEFAULT_SCHEMA=[PPMUSER]
GO
USE [PPMDB]
GO
CREATE SCHEMA [PPMUSER] AUTHORIZATION [PPMUSER]
```

```
GO
use [PPMDB]
GO
GRANT CREATE SCHEMA TO [PPMUSER]
GO
use [PPMDB]
GO
GRANT CREATE TABLE TO [PPMUSER]
GO
use [PPMDB]
GO
GRANT CREATE VIEW TO [PPMUSER]
GO
```

The chapter **Create database [19]** describes how to use **Microsoft SQL Server Management Studio** to create a database and database user for PPM.

3.3.3. Tablespace configuration

For SQL servers, the tablespaces described in the chapter on Tablespaces [10] are configured as follows:

Example: SQL Server 2017

Tablespace name	Key
Default	SQLSERVER_2017_TBLCONF_STDTABLE
Binary data	SQLSERVER_2017_TBLCONF_STDBLOB
Indices	SQLSERVER_2017_TBLCONF_STDINDEX

The tablespace is specified by the following syntax:

Key = ON <Tablespace name>

Example 11.

File extract Database_settings.properties

```
SQLSERVER_2017_TBLCONF_STDTABLE=ON PPMDATA
SQLSERVER_2017_TBLCONF_STDINDEX=ON PPMINDX
SQLSERVER_2017_TBLCONF_STDBLOB=ON PPMBLOB
```

3.3.4. Create a database

This chapter illustrates by example how you use **Microsoft SQL Server Management Studio** to set up an SQL server database for a PPM demo system and create a database user for a PPM client server.

If you want to set up an SQL server database for a productive system, please contact your SQL server system administrator. Detailed information on Microsoft SQL Server Management Studio is available in the relevant product documentation.

The example refers to SQL server version 2017 and Microsoft SQL Server Management Studio 11.0.

Note that you must configure the master database in the respective SQL Server instance with a collation that is compatible with the collation to be used for the PPM database. For instance, a case-insensitive collation for the master database is not compatible with the case-sensitive collation required for a PPM database.

Create a database

Start **Microsoft SQL Server Management Studio** and log in with an administrative user, for example by using a Windows system administrator access.

In the pop-up menu of the node **Databases** in the tree displayed in Object Explorer, select **New Database**.

General page

Specify a name (for example, PPM) for the new database.

Specify the settings for the database files. To operate the PPM demo databases, a 1 GB container is sufficient.

Options page

Check the **Collation** setting. We recommend that you use the value **Latin1_General_BIN**. UTF-8 character sets are not supported for MS SQL Server databases.

Check the **Recovery Model** setting. For a demo database, the **simple** option is sufficient.

Click **OK** to exit the window and create a new database.

Create a login

After having created the database, you need to create a login for it. To do so, select the entry **New Login** in the pop-up menu of the **Security/Logins** node in Object Explorer.

In the subsequent window, select **General** under **Select a page**, then enable the **SQL Server authentication** option, and enter the name of the database user, for example, PPMDEMO (in uppercase letters) as well as the password. Assign the user the relevant default database, for example, the database you just created, **PPM**.

You will need the assigned password later when creating a PPM client. Disable the **Enforce password policy** option if you want to specify any password.

Now, select the **User Mapping** page in order to generate the default schema automatically upon the first login of the newly created user. In the **Map** column, select the database you want to map a schema to for this login name. The **User** column then shows the login name you just created and the field in the **Default schema** column becomes editable. Specify the required name in this column. We recommend that you use the login name as the default schema name, for example, PPMDEMO.

The default schema name must be in upper-case letters, for example, PPMDEMO.

Entering the name of the default schema and clicking **OK** creates this schema.

Create DB user

Open the required database,for example, the database **PPM** you just created, and select the entry **New User** under the node **Security/Login** node in the Object Explorer configuration tree.

On the **General** page, enter the DB user name and the login name you just created. The DB user name is recommended to be the same as the login name. In the **Default schema** field, enter the name of the schema that you had assigned to the user when creating the login name.

Adjust DB properties

To be able to use the database schema with the user for PPM, you need to specify certain permissions for the new user and schema. In the **Microsoft SQL Server Management Studio** Object Explorer, select the schema node of the created database, for example, the node **/Databases/PPM/Security/Schemas/PPMDEMO** of the previously created **PPM** database. Select **Properties** from the pop-up menu.

In the **Properties** window, select the **Permissions** node and click **View Database Permissions** to open the window **Database Properties**. The **Permissions** node is already selected there. Select the newly created user, for example, PPMDEMO and assign it the permissions **Create Schema** and **Create table** by enabling the check boxes in the **Assign**.

Check database connection

Start PPM Customizing Toolkit and create a new client. In the **Database settings** dialog, enter the data for the new SQL server database and click **Test database connection**.

The test result is displayed in a separate window.

If the connection fails, you can click the Info button to display detailed information.

Before starting PPM Customizing Toolkit, make sure that the current driver for the SQL server database is located in the correct directory **<installation directory>\ppm\server\bin\work\data_ppm\drivers**.

3.3.5. Unicode support

PPM supports multi-byte character sets when using Microsoft SQL Server as a database system.

If you want to use PPM with Microsoft SQL Server Unicode databases, please note the following:

- PPM supports SQL Server databases set up for collation **Latin1_General_BIN**. SQL Server databases with other collations are not supported.
- Assign the collation at the database level.
- If you change collation settings at a later time, existing database content is not changed. For the changed database settings to become effective, you need to reinitialize the database schema of the corresponding PPM client by running the command line program **runinitdb**.

Warning
All imported data is deleted when you initialize the database.

3.3.6. Migration

If you want to migrate your existing SQL server databases to Microsoft SQL Server, please ensure that the collation **Latin1_General_BIN** is set for these databases.

> **Warning**
> If you migrate databases to Microsoft SQL Server with collations other than **Latin1_General_BIN** it is not sure that PPM can use the migrated databases without errors.

4. Performance tuning

This chapter provides performance-relevant information pertaining to PPM server software and the database system.

4.1. Overall performance

Runtime efficiency of the PPM server system, and especially the import largely depends on the database system.

Please observe the following performance-related instructions for your PPM system:

- Monitor the runtime configuration of the database instance in frequent intervals and, if required, adjust the configuration if you see room for improvement.
- Frequently update the database statistics.
- After aggregating or deleting a large number of process instances, reorganize the tables and recalculate all database schema indices.
- The **runppmimport** command line tool supports you in recalculating the indices. Indicate the argument **-index new**, for example:
- runppmimport -client <client name> -user system -password <password> -index new

If at all possible, do not interrupt this procedure to avoid time-consuming consolidation processes in the database schema.

4.1.1. Hardware-related

The hardware you use directly influences the overall performance of the PPM server system.

- PPM server and database system should be installed on the same computer or connected via a network with sufficient capacity.
- If possible, select a RAID 5 or RAID 10 array-based file system for tablespace file storage.
- You can further increase performance by distributing the tablespace of the PPM tablespace types (see chapter Supported database systems [12]) to physically independent file systems.
- Configure an MS SQL Server database so that transaction logs and database files are saved in physically independent file systems.

4.1.2. Configuration-related

- Register user-defined dimensions and measures at a particular process type or process type group so that these dimensions and measures will be calculated for process instances of this process type/ process type group only.

  Warning
  The standard measures (number of processes and number of functions, time, function) and all data access dimensions must be registered at the process tree root because they are required for internal calculations.

- If possible, use process measures and process dimensions. Function measure queries and function dimension queries require more calculation capacity due to the significantly larger data volume. Normally, function measures are required only if a function exists multiple times in a process instance and if you want to analyze the functions individually.

4.2. Import performance

When importing fragments and calculating process instances, large volumes of data are written into the database system's tablespaces. Therefore, the performance of the file system on which the tablespace files are saved determines the performance of the data import.

5. Support and legal information

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

5.1. 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.

5.2. 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.

5.3. 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.