PPM DATENBANKSYSTEME

ARIS Process Performance Manager 10

Oktober 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.

Inhaltsverzeichnis

1. Allgemein 4
2. Datenbankschema von PPM 5
    2.1. Namen von Kennzahlen und Dimensionen 5
    2.2. Datenbanktabellen 5
    2.3. Interaktion PPM und Datenbank 6
        2.3.1. Zugriff des PPM-Server auf die Datenbank 6
            2.3.1.1. Verbindung zur Datenbank 6
            2.3.1.2. Datenbankbenutzer 7
        2.3.2. Datenbankobjekte 7
            2.3.2.1. Behandlung von NULL-Werten 8
            2.3.2.2. Transaktionen 8
        2.3.3. Datenimport 9
            2.3.3.1. Fragmente importieren 9
            2.3.3.2. Prozessinstanzen erzeugen, Kennzahlen berechnen 9
        2.3.4. Datenanalyse 10
    2.4. Tablespaces 10
        2.4.1. Tablespace-Typen 11
        2.4.2. Tablespace-Konfiguration 11
3. Unterstützte Datenbanksysteme 12
    3.1. Oracle 12
        3.1.1. JDBC-Treiber 13
        3.1.2. Anlegen eines DB-Benutzers 13
        3.1.3. Tablespace-Konfiguration 14
            3.1.3.1. Oracle 19 14
            3.1.3.2. Oracle 18 15
            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. Anlegen eines DB-Benutzers 16
        3.2.3. Tablespace-Konfiguration 17
            3.2.3.1. DB2 17
    3.3. Microsoft SQL Server 17
        3.3.1. JDBC-Treiber 17
        3.3.2. Anlegen eines DB-Benutzers 18
        3.3.3. Tablespace-Konfiguration 19
        3.3.4. Datenbank anlegen 20
        3.3.5. Unicode-Unterstützung 22
        3.3.6. Migration 22
4. Performance Tuning 24
    4.1. Gesamt-Performance 24
        4.1.1. Hardware spezifisch 24
        4.1.2. Konfigurationsspezifisch 25
    4.2. Import-Performance 25
5. Support und rechtliche Hinweise 26
    5.1. Dokumentationsumfang 26
    5.2. Support 26
    5.3. Legal notices 27

1. Allgemein

Dieses Handbuch beschreibt die datenbanktechnischen Zusammenhänge zwischen PPM-Server und dem verwendeten Datenbanksystem. Datenbanktechnische Grundlagen sowie die grundsätzliche Arbeitsweise von ARIS Process Performance Manager sollten Ihnen bekannt sein.

Als Repository verwendet PPM ein SQL-RDBMS, das alle Konfigurationen und Daten speichert. PPM ist als Client-Server-Applikation in Java entwickelt.

Das Handbuch ist keine Installations- oder Customizing-Anleitung eines der von ARIS Process Performance Manager unterstützten Datenbanksysteme. Es vermittelt Ihnen Kenntnisse, um performance-kritische Szenarien zu erkennen und zu optimieren.

2. Datenbankschema von PPM

Die Datenbank dient PPM zur persistenten Speicherung aller Konfigurationen, administrativen Einstellungen sowie der importierten Daten. Sie bildet das Gegenstück zum flüchtigen, hauptspeicherbasierten Analyseserver, der die Analysestrukturen beinhaltet. Im Falle eines Datenverlustes im Analyseserver, kann dieser aus der Datenbank vollständig rekonstruiert werden. Zur Datensicherung eines PPM-Systems ist es also ausreichend ausschließlich das PPM-Datenbankschema bzw. den Benutzer zu sichern.

2.1. Namen von Kennzahlen und Dimensionen

Die internen Namen von Kennzahlen und Dimensionen werden auch zur internen Zuordnung von Konfigurationselementen in der Datenbank verwendet. Daher unterliegen diese Bezeichner teilweise den Beschränkungen des unterliegenden Datenbanksystems.

Die auf der PPM-Benutzeroberfläche angezeigten Namen von Kennzahlen und Dimensionen werden oberflächensprachabhängig in der Kennzahlenkonfiguration angegeben (XML-Elemente **<description>**).

Bitte beachten Sie beim Vergeben von internen Kennzahlen- und Dimensionsnamen folgende Tipps:

- Namen müssen mit einem Buchstaben beginnen.
- Verwenden Sie ausschließlich Großbuchstaben.
- Als Sonderzeichen ist nur der Unterstrich erlaubt (keine Umlaute).
- Vergeben Sie möglichst kurze Namen, vermeiden Sie überlange Namen. Die maximal zulässige Länge der Namen ist datenbankspezifisch. Wir empfehlen eine maximale Länge von 25 Zeichen.

Beispiel 1.

Die folgende Tabelle zeigt einige Kennzahlen- und Dimensionsnamen:

Typ	Name	Beschreibung
Kennzahlen	PNUM	Prozessanzahl
Dimension	D_MATERIAL	Material, zweistufig
Dimension	TIME	Dimension Zeit

2.2. Datenbanktabellen

PPM unterscheidet zwischen Datenbanktabellen mit festem Namen und Tabellen, deren Namen durch die Kennzahlenkonfiguration bestimmt wird. Die beiden folgenden Tabellen zeigen einige Beispiele:

Konfigurationsabhängige Tabellennamen

Tabellenname	Beschreibung
ATTR_INFO	Importierte Attribute

Tabellenname	Beschreibung
DBVERSIONNUMBER	Versionsnummer des Datenbankschemas
XML_CONFIGS	eingelesene Konfigurationen (**runinitdb** oder **runppmconfig**)
EPK_IMPORT_TBL	eingelesene Fragmente
EPK_TBL	zusammengeführte Prozessinstanzen

Konfigurationsabhängige Tabellennamen

Tabellenname	Beschreibung
PC_UMG_DASHBOARD	Importierte Daten der prozessunabhängigen Kennzahlreihe des Performance Dashboard

2.3. Interaktion PPM und Datenbank

In diesem Kapitel lernen Sie die Art der Zugriffe von PPM auf die verwendete Datenbank bei Datenimport und Analyse kennen. Zum besseren Verständnis sollte Ihnen die grundsätzliche Arbeitsweise von PPM vertraut sein.

2.3.1. Zugriff des PPM-Server auf die Datenbank

Der PPM-Server verwendet zum Zugriff auf die Datenbank die standardisierte Datenbankschnittstelle JDBC (Java DataBase Connectivity). Dadurch ist der verwendete Datenbanktyp für den PPM-Server weitgehend transparent. Datenbankhersteller-spezifische Unterschiede können jedoch dazu führen, dass der PPM-Server für dieselbe Aufgabe verschiedene Datenbankanfragen generiert.

2.3.1.1. Verbindung zur Datenbank

Jede von PPM unterstützte Datenbank lässt sich über JDBC zugreifen. Der JDBC-Treiber besteht aus einem oder mehreren Java-Archivdateien (jar). Kopieren Sie das benötigte JDBC-Treiberarchiv in das folgende Verzeichnis:

<PPM-Installationsverzeichnis>\ppm\server\bin\work\data_ppm\drivers.

Die Treiber in diesem Verzeichnis werden von allen PPM-Mandanten in der Installation gemeinsam verwendet. Nach dem Hinzufügen oder Aktualisieren von Treibern ist ein Neustart des PPM-Servers erforderlich.

Warnung
Verwenden Sie nicht den ACC-Erweiterungs-Befehl, um JDBC-Treiber bereitzustellen. Der Befehl enhance wendet Treiber nur auf einzelne Laufwerke, was zu Inkonsistenzen und Treiberkonflikten führen kann, wenn mehrere Mandanten oder Laufwerke verwendet werden. Verwenden Sie immer das oben beschriebene Verzeichnis für gemeinsam genutzte Treiber.

JDBC-Treiber mit Hilfe von ACC hinzufügen

Eine weitere Möglichkeit, JDBC-Datenbanktreiber oder SAP®-Treiber auf die PPM-Installation anzuwenden, ist die Verwendung von ARIS Cloud Controller (ACC). Mit dem Befehl **enhance** in ACC werden die angegebenen Treiber ebenfalls in das oben genannte Verzeichnis kopiert:

<Installationsverzeichnis>\ppm\server\bin\work\data_ppm\drivers.

Beispiel 2.

enhance <client id> mit lokalem DB-Treiber <path to DB driver>

Bitte beachten Sie, dass die Pfadangaben im Java-Standard angegeben werden müssen. Also auch unter Windows Betriebssystemen geben Sie bitte „/" anstelle von „\" an. Außerdem muss in der PPM-Installation ein gültiger und bereits installierter Client vorhanden sein. Mit Hilfe der ID des Client-Servers können die Treiber hinzugefügt werden. Wenn Sie beispielsweise einen Client mit dem Namen **umg_en** erstellt haben, verwenden Sie **umg_en_cs** als ID für den enhance-Befehl. Die kopierten Treiber gelten für alle in der Installation enthaltenen Clients.

Verschiedene Datenbanken in einer PPM-Installation

Sie können in einer PPM-Installation für verschiedene Mandanten auch Datenbanksysteme verschiedener Hersteller ansprechen. Die entsprechenden Java-Archivdateien (jar) müssen manuell in das Verzeichnis **<Installationsverzeichnis>\ppm\server\bin\work\data_ppm\drivers** kopiert werden.

Für einen bestimmten Datenbanktyp kann immer nur ein einziger JDBC-Treiber angegeben werden. Es ist nicht möglich, unterschiedliche Versionen eines Datenbanktyps, z. B. Oracle 12 und Oracle 19, mit verschiedenen JDBC-Treiberversionen anzusprechen.

2.3.1.2. Datenbankbenutzer

Für jeden PPM-Mandanten benötigen Sie genau ein Schema eines dedizierten Datenbankbenutzers. Der Datenbankbenutzer benötigt uneingeschränkten Zugriff auf die Objekte seines Schemas.

Sie finden eine detaillierte Beschreibung zum Anlegen eines DB-Benutzers für die unterstützten Datenbanksysteme in den entsprechenden Abschnitten der unterstützen Datenbanken.

- Oracle DB-Benutzer anlegen. [13]
- IBM DB2-Benutzer anlegen. [16]
- Microsoft SQL-Benutzer anlegen. [18]

Im datenbankspezifischen Schlüssel **<RDBMS>_USE_CASE_SENSITIVE_USERNAME** der mandantenspezifischen Konfigurationsdatei **Database_settings.properties** können Sie angeben, ob das verwendetet Datenbanksystem bei der Angabe von Benutzernamen zwischen Groß- und Kleinschreibung unterscheidet (Wert **true**) oder nicht (Wert **false**). Vorgabewert ist **false**.

2.3.2. Datenbankobjekte

Unabhängig vom verwendeten Datenbanksystem benötigt der PPM-Server folgende Datenbankobjekte:

- Tabellen und Constraints
- Sequences (falls vom DB-System unterstützt, anderenfalls simuliert der PPM-Server eine vergleichbare Funktionalität)
- Indizes
- Foreign-keys (Fremdschlüssel)

Um Ihr PPM-System performant zu halten, sollten Sie regelmäßig Objekt- und Systemstatistiken berechnen lassen. Am besten ist es, wenn der Administrator Ihrer Datenbank diese Aufgabe übernimmt.

Falls Sie Oracle als Datenbanksystem für PPM verwenden, können Sie die Objektstatistiken mittels Argument **-genstats** des Kommandozeilenprogramm **runppmimport** während des Datenimports berechnen lassen. Weitere Informationen erhalten Sie in der Hilfe des Kommandozeilenprogramms, die durch den Aufruf **runppmimport -h** angezeigt wird. Insbesondere bei neueren Oracle-Versionen empfehlen wir die administrative Statistikberechnung anstelle der von PPM gesteuerten Berechnung von Statistiken.

2.3.2.1. Behandlung von NULL-Werten

Technisch wird zwischen den Aussagen **Wert ist nicht vorhanden** (NULL-Wert) und **Wert ist nicht angegeben** (Zeichenfolge der Länge Null) unterschieden. Die PPM-Datenbankextraktoren können zwischen NULL-Werten und leeren Zeichenketten unterscheiden.

Die von PPM unterstützen Datenbanksysteme verhalten sich diesbezüglich unterschiedlich. Beim Auslesen von Quellsystemdaten mit den PPM Extraktoren können diese auf das Auftreten von NULL und "" unterschiedlich reagieren.

Oracle

Leerstrings werden beim Einfügen/Aktualisieren und Fehlende Spalten beim INSERT als NULL zurückgelesen.

SQLServer und DB2

Leerstrings werden beim Einfügen / Aktualisieren auch so zurückgelesen.

Fehlende Spalten werden beim INSERT als NULL zurückgelesen.

2.3.2.2. Transaktionen

Der PPM-Server arbeitet auf der Datenbank transaktionsgesteuert. Datenbankanfragen werden innerhalb einer Kommandokette ausgeführt und abschließend im Datenbankschema angewendet (commit). Diese Vorgehensweise heißt Transaktion. Transaktionen werden zunächst temporär in der Laufzeitumgebung des Datenbanksystems ausgeführt. Noch nicht abgeschlossenen Transaktionen können widerrufen werden (rollback).

Der transaktionsgesteuerte Zugriff auf die Datenbank erfordert ein bestimmtes Minimum an System-Ressourcen der Datenbankinstanz.

Die von PPM unterstützten Datenbanksysteme unterscheiden sich im Transaktions-Handling.
Der PPM-Applikationsserver nutzt dieses unterschiedliche Transaktions-Handling der unterstützten Datenbanksysteme optimal aus.

Beispiel 3.

- Bei Verwendung eines Oracle basierten Datenbankschemas zeigen die PPM-Administrationskomponenten eine Schaltfläche **Speichern** an. Nach dem Speichern sind die Änderungen sofort wirksam.
- Bei Verwendung eines DB2- oder SQL-Server basierten Datenbankschemas sind die Analysemöglichkeiten durch die Transaktionslogik **READ_LOCK** eingeschränkt.

2.3.3. Datenimport

Dieses Kapitel zeigt die datenbanktechnischen Vorgänge beim Import von Prozessinstanzdaten. Der Datenimport setzt sich aus zwei, nacheinander ablaufenden Phasen:

1. Einlesen von Fragmenten aus Anwendungssystemen [9]
2. Prozessinstanzen erzeugen und berechnen [9]

2.3.3.1. Fragmente importieren

Unabhängig vom Format des Datenimports (System-Event- oder Graphformat) werden für die zu importierenden Fragmentinstanzen zunächst Prozessschlüssel, Hierarchieschlüssel und Shared-Fragment-Schlüssel berechnet. Die gültige Konfiguration der entsprechenden Regelwerke wird vorausgesetzt. Konnte mindestens ein Prozessschlüssel berechnet werden, wird das gelesene Fragment in der Datenbanktabelle **EPK_IMPORT_TBL** gespeichert. Vollständige Prozessinstanzen werden direkt, ohne Berechnung eines Prozessschlüssels gespeichert.

Vollständige Prozessinstanzen gelten als abgeschlossen. Sie können nicht mehr durch Zusammenführen mit importierten Fragmenten erweitert werden. Eine vollständige Prozessinstanz ist durch das Prozessinstanzattribut **AT_EPK_KEY** gekennzeichnet.

2.3.3.2. Prozessinstanzen erzeugen, Kennzahlen berechnen

Phase EPK kopieren (Copy EPC)

In diesem Schritt werden die importierten Fragmente aus der Tabelle **EPK_IMPORT_TBL** gelesen und in die Tabelle **EPK_TBL** geschrieben. Die berechneten Werte aller prozessbezogenen Schlüssel (Prozess-, Shared Fragment, Hierarchieschlüssel) werden mit Fremdschlüsselbeziehung auf das entsprechende Fragment in bestimmten Tabellen gespeichert. Welche Tabellen genau das ist, hängt vom Zustand der Datenbank ab, ob z. B. bereits Prozessinstanzen in der Datenbank existieren. Anschließend wird das Fragment in der **EPK_IMPORT_TBL** gelöscht.

Phase Merger

Fragmente mit identischem Prozessschlüssel werden in einer Prozessinstanz zusammengefasst. Dabei werden schrittweise immer 2 Fragmente bearbeitet, indem alle Objekte und Kanten des Fragments mit

dem älteren Importzeitstempel in das mit jüngerem Importzeitstempel kopiert werden. Dieser Vorgang läuft solange, bis ausschließlich eindeutige Prozessschlüssel in der Datenbank vorhanden sind.

Phase Hierarchien

Die Reihenfolge der Berechnung hierarchisch abhängiger Prozessinstanzen wird bestimmt, der Zustand der Referenzen wird aktualisiert.

Phase Kennzahlenberechnung

In dieser Phase werden Prozessinstanzen typisiert und berechnet. Nicht mehr benötigte Fragmente und Schlüssel in der Datenbank werden gelöscht und die berechneten Prozessinstanzen in die **EPK_TBL** zurückgeschrieben.

Das Zusammenführen von Fragmenten zu Prozessinstanzen erfordert, dass die Prozessschlüssel aller importierten Fragmente im Hauptspeicher gehalten werden, damit sie gleichzeitig bearbeitet werden können. Wenn Sie in einem Importvorgang eine große Anzahl Fragmente importieren, Größenordnung über 100000, reicht die Größe des Hauptspeichers möglicherweise nicht mehr aus. Im Schlüssel **<IMPORT_SCENARIO>_READ_RATE_EPC** der Konfigurationsdatei **EpkImport_settings.properties** können Sie einstellen, wie viele Fragmentinstanzen maximal in einem Arbeitsschritt eingelesen werden sollen. Vorgabewert ist 100000. In der Datei **PPM Datenimport.pdf** finden Sie weitere Informationen zu den einzelnen Importszenarien in PPM.

2.3.4. Datenanalyse

Nachdem im Rahmen der Kennzahlenberechnung die Prozesse berechnet und die resultierenden Kennzahlen und Dimensionen an den Analyseserver übertragen wurden, können die Prozessinstanzen analysiert werden.

2.4. Tablespaces

Alle von PPM unterstützten Datenbanksysteme benutzen zur persistenten Datenspeicherung bestimmte Bereiche auf einem Datenträger. Bei Oracle und DB2 Datenbanken werden diese Speicherbereiche **Tablespaces** genannt. Microsoft verwendet für seine SQL-Server Produkte den Begriff **Filegroups**. Im Folgenden wird einheitlich der Begriff Tablespace auch für Filegroups verwendet.

Das PPM-System verwendet verschiedene Klassen von Daten:

- Importierte Fragmentinstanzen und Prozessinstanzen (Binäre Datenobjekte)
- Verwaltungs- und Strukturinformationen
- Datenbankindizes

Sie können die Performance Ihres PPM-Systems in einem ersten Schritt optimieren, indem Sie den verschiedenen Datenklassen eigene Tablespaces zuweisen.

2.4.1. Tablespace-Typen

In der mandantenspezifischen Konfigurationsdatei **Database_settings.properties** geben Sie an, welche Tablespaces welche PPM-Daten speichern.

Die Konfigurationsdatei **Database_settings.properties** ist im folgenden mandantenspezifischen Verzeichnis abgelegt.

<PPM-Installation>\server\bin\work\data_ppm\config\<Mandant>\

PPM unterscheidet folgende Typen von Tablespaces:

Tablespace-Name	Beschreibung
STDBLOB	Alle Datenbanktabellen mit Binärdaten werden in diesem Tablespace gespeichert, z. B. die Tabelle der berechneten Prozessinstanzen
STDTABLE	Standard-Tablespace. In diesem Tablespace werden alle Daten gespeichert, die Standarddatentypen des Datenbanksystems verwenden.
STDINDEX	Für MS SQL-Server und Oracle-Systeme wird dieser Tablespace zur Ablage der Primärschlüssel und Indexinformationen benutzt.

Der Speicherbedarf eines Tablespace ist stark abhängig von der Konfiguration und den importierten Daten des PPM-Mandanten.

Wenn Sie die Standardkonfiguration nach Anlegen eines Mandanten nicht verändern, werden alle Daten des PPM-Mandanten im Default-Tablespace gespeichert. Der Default-Tablespace wird beim Anlegen des Datenbankbenutzers bestimmt. Diese Konfiguration ist für ein produktives System äußerst ungeeignet.

2.4.2. Tablespace-Konfiguration

Die Zuordnung von Datenbanktabellen zu Tablespaces wird in der Mandantenkonfigurationsdatei **Database_settings.properties** spezifisch angegeben. Die allgemeine Syntax ist:

<Database type>_TBLCONF_<Tablespace name> = Wert

- Datenbanktyp gibt einen der unterstützten Datenbanktypen an. Zulässige Werte: **ORACLE_11, ORACLE_12**, **ORACLE_18**, **ORACLE_19**, **DB2_10**, **DB2_11**, **SQL_Server_2016**, **SQL_Server_2017** und **SQL_Server_2019**

Wenn Sie für einen MultiByte-Mandanten den Datenbanktyp **SQL Server** verwenden, müssen Sie eine der zusätzlich verfügbaren Unicode-Varianten **SQL_Server_2016_UNICODE**, **SQL_Server_2017_UNICODE** oder **SQL_Server_2019_UNICODE** verwenden

Die Syntax des mit Wert angegebenen Tablespace ist ebenfalls datenbankspezifisch und ist in den jeweiligen Unterkapitel von Kapitel Unterstützte Datenbanksysteme [12] beschrieben.

3. Unterstützte Datenbanksysteme

Dieses Kapitel informiert Sie über die administrativen Unterschiede der verschiedenen, von PPM unterstützten Datenbanksysteme.

PPM unterstützt folgende Datenbanksysteme:

- Oracle
- IBM DB2
- Microsoft SQL Server

  Alle Versionen auch in UNICODE, wenn Daten in einem Multibyte-Zeichensatz gespeichert werden sollen, z. B. Japanisch.

Detaillierte Versionshinweise für jede Plattform finden Sie in den ARIS-Systemanforderungen. Die ARIS-Systemanforderungen finden Sie auf der ARIS Dokumentations-Website.

Beim Anlegen eines Mandanten geben Sie das zu verwendende Datenbanksystem an. In anschließenden Konfigurationsdialogen werden die datenbanksystemspezifischen Einstellungen angegeben: Hostname, Port-Nummer des Datenbankdienstes, Name der Datenbank sowie für den PPM-Mandanten den zu verwendenden Datenbankbenutzername und das Datenbankbenutzerkennwort.

Die von PPM unterstützten Datenbanksysteme verwenden den Begriff Datenbank mit unterschiedlicher Semantik:

Oracle

Oracle verknüpft die Laufzeitprozesse und die Datenbank zu einer Datenbankinstanz. Im täglichen Sprachgebrauch wird fast ausschließlich der Begriff Datenbank verwendet.

IBM DB2 und Microsoft SQL-Server

Bei diesen Datenbanksystemen können die Laufzeitprozesse der Datenbankinstanz beliebig viele Datenbanken verwalten. Wenigstens die Systemdatenbank muss vorhanden sein. Microsoft nennt die Systemdatenbank seiner SQL-Serverprodukte **master-Datenbank**.

Für die Datenbanksysteme **IBM DB** und **Microsoft SQL-Server** empfehlen wir, für jeden PPM-Mandanten eine eigene Datenbank zu verwenden. Im Besonderen trifft dies für die benötigten Datenbankbenutzer eines skalierten PPM-Systems zu.

3.1. Oracle

Die Standarddatenbank der neuesten Releases von Oracle (18 oder 19) ist eine Container-Datenbank, die nicht standardmäßig von PPM unterstützt wird. Für PPM muss eine Instanzdatenbank verwendet werden, die nicht als Container konfiguriert ist.

PPM verwendet für den Zugriff auf die Oracle-Datenbank die JDBC Thin-Schnittstelle (Typ 4). In der mandantenspezifischen Konfigurationsdatei **Database_settings.properties** geben Sie im Schlüssel **URL** die Zugriffsparameter für die Oracle-Datenbank an.

Bei der Verwendung einer einzelnen Instanzdatenbank ist die Syntax wie folgt:

jdbc:oracle:thin:@//<Host>:<Port>/<SID-Datenbank>

Beispiel 4.

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

Eine solche URL müssen Sie in PPM Customizing Toolkit im Dialog **Datenbankeinstellungen** manuell eingeben.

3.1.1. JDBC-Treiber

Die JDBC-Treiber neuerer Oracle-Versionen sind in der Regel abwärtskompatibel. Bitte beachten Sie dazu die Angaben des Herstellers. Standardmäßig befindet sich der Treiber im Unterverzeichnis **jdbc\lib** Ihrer Oracle-Installation.

Warnung
Der PPM-Server benötigt einen JDBC-Treiber der verwendete **JDBC-3**-Features implementiert. Die in den Dateien **classes*.zip** enthaltenen JDBC-Treiber sind veraltet und für den Betrieb mit PPM ungeeignet. Bitte verwenden Sie das aktuelle **ojdbc17.jar**-Archiv für alle unterstützten Oracle-Systeme.

Hinweis

Bei PPM Version 10.5.11 und höher wird die Java 21-Laufzeitumgebung verwendet. Folglich müssen auch die Oracle JDBC-Treiber passend und für Java 21 zertifiziert sein. Ojdbc8.jar-Versionen sind für Java 8 zertifiziert, ojdbc11.jar ist für Java 11 zertifiziert und ojdbc17.jar ist für Java 17 und Java 21 zertifiziert.

Kopieren Sie den Treiber nach <Installationsverzeichnis>\ppm\server\bin\work\data_ppm\drivers.

Beim Starten gibt der PPM-Server den verwendeten Interface-Typ sowie die genauen Versionen der Datenbank und des verwendeten JDBC-Treiber in eine log-Datei oder auf der Kommandozeile aus.

3.1.2. Anlegen eines DB-Benutzers

Sie können einen Datenbankbenutzer für PPM bequem mit Hilfe geeigneter Administrationskomponenten wie **Database Control** oder **EM Database Express** anlegen.

Beachten Sie, dass in Oracle 18 und 19 die Groß-/Kleinschreibung in den Benutzernamen berücksichtigt wird. PPM unterstützt nur Benutzernamen aus Großbuchstaben.

Weisen Sie dem DB-Benutzer die Rollen **CONNECT** und **RESOURCE** zu und achten Sie auf genügend Berechtigungen beim Zugriff auf Tablespaces.

Alternativ können Sie den Benutzer auch mit einem SQL-Skript anlegen. Unter der Voraussetzung, dass Standard-Tablespace **PPMDATA** und temporärer Tablespace **TEMP** existieren, können Sie folgendes Skript zum Anlegen von DB-Benutzers für PPM verwenden:

```
-- 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" ;
```

Warnung
Vermeiden Sie die Verwendung von System Tablespaces für einen PPM-DB-Benutzer.

3.1.3. Tablespace-Konfiguration

Die in Kapitel Tablespaces [10] beschrieben Tablespaces werden wie folgt konfiguriert:

3.1.3.1. Oracle 19

Tablespace-Name	Key
Standard	ORACLE_19_TBLCONF_STDTABLE
Binärdaten	ORACLE_19_TBLCONF_STDBLOB
-Indizes	ORACLE_19_TBLCONF_STDINDEX

Der Tablespace wird durch folgende Syntax spezifiziert:

Schlüssel = TABLESPACE <Name>

Beispiel 5.

Dateiauszug Database_settings.properties:

ORACLE_19_TBLCONF_STDTABLE=TABLESPACE PPMDATA

ORACLE_19_TBLCONF_STDINDEX=TABLESPACE PPMDATAIDX

ORACLE_19_TBLCONF_STDBLOB=TABLESPACE PPMBLOB

Beachten Sie, dass die Standarddatenbank einer Oracle 19-Installation eine Container-Datenbank ist, die nicht standardmäßig von PPM unterstützt wird. In Oracle 19 wird die Groß-/Kleinschreibung in den Benutzernamen berücksichtigt, während PPM nur Benutzernamen in Großbuchstaben unterstützt.

3.1.3.2. Oracle 18

Tablespace-Name	Key
Standard	ORACLE_18_TBLCONF_STDTABLE
Binärdaten	ORACLE_18_TBLCONF_STDBLOB
Indizes	ORACLE_18_TBLCONF_STDINDEX

Der Tablespace wird durch folgende Syntax spezifiziert:

Schlüssel = TABLESPACE <Name>

Beispiel 6.

Dateiauszug Database_settings.properties:

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
Standard	ORACLE_12_TBLCONF_STDTABLE
Binärdaten	ORACLE_12_TBLCONF_STDBLOB
Indizes	ORACLE_12_TBLCONF_STDINDEX

Der Tablespace wird durch folgende Syntax spezifiziert:

Schlüssel = TABLESPACE <Name>

Beispiel 7.

Dateiauszug Database_settings.properties:

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
Standard	ORACLE_11_TBLCONF_STDTABLE
Binärdaten	ORACLE_11_TBLCONF_STDBLOB
Indizes	ORACLE_11_TBLCONF_STDINDEX

Der Tablespace wird durch folgende Syntax spezifiziert:

Schlüssel = TABLESPACE <Name>

Beispiel 8.

Dateiauszug Database_settings.properties:

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

Der PPM-Server kann auf eine DB2-Datenbank über den Interface-Typ 4 zugreifen. Je nach verwendeter Datenbankversion benötigen Sie den entsprechende JDBC-Treiber. Die benötigten Dateien finden Sie im Unterverzeichnis **java** Ihrer DB2-Installation.

Um auf einen DB2-Server zuzugreifen, benötigen Sie den neuesten unterstützten JDBC-Treiber. Normalerweise ist dies ein Einzelarchiv für den Typ-4-Treiber: **db2jcc4.jar**. Sie können die erforderliche Datei von der Hersteller-Homepage wie oben angegeben herunterladen.

Hinweis

Bei PPM Version 10.5.11 und höher wird die Java 21-Laufzeitumgebung verwendet. Verwenden Sie bei IBM Db2 unterstützten Datenbanken den aktuellsten JDBC-Treiber, der auch für Java 21 zertifiziert ist.

Beim Starten gibt der PPM-Server die genauen Versionen der Datenbank und des verwendeten JDBC-Treibers aus. In der log-Datei oder auf der Kommandozeile sehen Sie Meldungen der folgenden Art:

```
...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. Anlegen eines DB-Benutzers

DB2-Datenbanken verwenden das Benutzermanagement des Betriebssystems. Nach der Installation der DB2-Datenbank-Software existieren bereits der Betriebssystembenutzer **DB2ADMIN** und die Gruppe **DB2USERS**. Um einen Datenbankbenutzer für PPM zu erzeugen, legen Sie einen entsprechenden Betriebssystembenutzer an und nehmen diesen in die Mitgliedschaft der Gruppe **DB2USERS** auf. Der neue Betriebssystembenutzer erhält die für den Zugriff auf die DB2-Datenbank erforderlichen Berechtigungen durch die Gruppenzugehörigkeit. Die minimalen, datenbanktechnischen Systemberechtigungen sind folgende: **CONNECT**, **CREATETAB** und **IMPLICITSCHEMA**.

3.2.3. Tablespace-Konfiguration

Für DB2 werden die in Kapitel Tablespaces [10]beschrieben Tablespaces wie folgt konfiguriert:

3.2.3.1. DB2

Tablespace-Name	Key
Standard	DB2_11_TBLCONF_STDTABLE
Binärdaten	DB2_11_TBLCONF_STDBLOB

Bei DB2 müssen die Speicherorte der Tabellenindizes bereits beim Erzeugen der Tabelle angegeben werden. Der Tablespace wird durch folgende Syntax spezifiziert:

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

Das Schlüsselwort **DB2_<Version>_TBLCONF_STDINDEX** der Konfigurationsdatei **Database_settings.properties** wird für DB2 nicht unterstützt.

Beispiel 9. ***[en-us]***

Dateiauszug Database_settings.properties

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

Die obligatorische Erweiterung **LONG IN** <Tablespace-Name> bestimmt den Tablespace, in dem die binären Datenbankobjekte gespeichert werden, die anderen werden in dem mit Schlüsselwort **IN** angegebenen Tablespace gespeichert.

3.3. Microsoft SQL Server

PPM unterstützt verschiedene Versionen von **Microsoft SQL-Server**. Die Versionen unterscheiden sich in ihrem Verhalten und der Administration.

Detaillierte Informationen zu den **Microsoft SQL Server**-Versionen erhalten Sie in den aktuellen ARIS-Systemanforderungen.

Folgende Schritte zeigen Ihnen beispielhaft, wie Sie den Microsoft SQL-Server für PPM konfigurieren können.

3.3.1. JDBC-Treiber

Um auf einen Microsoft SQL-Server zuzugreifen, benötigen Sie den erforderlichen JDBC-Treiber. Sie können diesen Treiber von der Microsoft Homepage laden.

Beim Starten gibt der PPM-Server den verwendeten Interface-Typ sowie die genauen Versionen der Datenbank und des verwendeten JDBC-Treiber aus.

Hinweis

Bei PPM Version 10.5.11 und höher wird die Java 21-Laufzeitumgebung verwendet. Verwenden Sie bei Microsoft SQL Server unterstützten Datenbanken den aktuellsten JDBC-Treiber, der auch für Java 21 zertifiziert ist.

3.3.2. Anlegen eines DB-Benutzers

Zum Anlegen eines Datenbankbenutzers für MS SQL-Server können Sie entweder das SQL-Server Management Studio auf dem Server oder remote auf Ihrem Desktop verwenden, oder Sie verwenden das auf dem Server verfügbare SQL-Kommandozeilenprogramm

sqlcmd

.

Wenn Sie das SQL-Kommandozeilenprogramm verwenden möchten, müssen Sie sich mit der entsprechenden Datenbank als Datenbankadministrator (sa) verbinden:

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

Für beide Programme können Sie einen gültigen Datenbankbenutzer mithilfe der folgenden Befehle anlegen:

```
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>]
```

Der Name des Datenbankbenutzers und des Schemas sollten identisch sein und immer in Großbuchstaben geschrieben sein.

Wenn das anzulegende Benutzerkonto bereits existiert, können Sie es durch folgende Kommandofolge löschen:

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

Beispiel 10.

Sie möchten einen Login und Benutzer namens

PPMUSER

in der existierenden MS SQL-Server-Datenbank

PPMDB

anlegen. Öffnen Sie im MS SQL Management Studio einen Abfrage-Editor und fügen Sie die folgende Befehlsfolge hinzu:

```
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
```

Im Kapitel **Datenbank erzeugen [20]** ist beschrieben, wie Sie mithilfe von **Microsoft SQL Server Management Studio** eine Datenbank und einen Datenbankbenutzer für PPM anlegen.

3.3.3. Tablespace-Konfiguration

Für SQL-Server werden die in Kapitel Tablespaces [10]beschrieben Tablespace wie folgt konfiguriert:

Beispiel: SQL Server 2017

Tablespace-Name	Key
Standard	SQLSERVER_2017_TBLCONF_STDTABLE
Binärdaten	SQLSERVER_2017_TBLCONF_STDBLOB
Indizes	SQLSERVER_2017_TBLCONF_STDINDEX

Der Tablespace wird durch folgende Syntax spezifiziert:

Schlüssel = ON <Tablespace name>

Beispiel 11. *[en-us]*

Dateiauszug Database_settings.properties

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

3.3.4. Datenbank anlegen

Dieses Kapitel zeigt ihnen exemplarisch, wie Sie mit Hilfe von **Microsoft SQL Server Management Studio** eine SQL-Server-Datenbank für ein PPM-Demo-System erzeugen und einen Datenbankbenutzer für einen PPM-Mandantenserver anlegen.

Wenn Sie eine SQL-Server-Datenbank für ein Produktivsystem einrichten möchten, wenden Sie sich bitte an Ihren SQL-Server-Systemadministrator. Detaillierte Informationen zu Microsoft SQL Server Management Studio erhalten Sie in der entsprechenden Produktdokumentation.

Das Beispiel bezieht sich auf SQL-Server Version 2017 und das Microsoft SQL Server Management Studio 11.0.

Beachten Sie, dass Sie die Master-Datenbank in der entsprechenden SQL-Serverinstanz mit einer Sortierung konfigurieren müssen, die mit der für die PPM-Datenbank verwendeten Sortierung kompatibel ist. Eine Sortierung für die Master-Datenbank, die die Groß-/Kleinschreibung nicht berücksichtigt, ist beispielsweise nicht mit einer Sortierung für die PPM-Datenbank kompatibel, die die Groß-/Kleinschreibung berücksichtigt.

Datenbank anlegen

Starten Sie **Microsoft SQL Server Management Studio** und melden Sie sich mit einem administrativen Benutzer an, z. B. durch Verwendung eines Windows-Systemadministratorzugangs.

Wählen Sie den Eintrag **Neue Datenbank** im Kontextmenü des Knotens **Datenbanken** des im Objekt Explorer angezeigten Baums.

Seite **Allgemein**

Geben Sie einen Namen (z. B. PPM) für die neue Datenbank an.

Setzen Sie die Einstellungen für die Datenbankdateien. Zum Betrieb der PPM-Demodatenbanken genügt ein Container der Größe 1GB.

Seite **Optionen**

Überprüfen Sie die Einstellung **Sortierung**. Wir empfehlen den Wert **Latin1_General_BIN**. UTF-8-Zeichensätze werden für MS SQL-Server-Datenbanken nicht unterstützt.

Überprüfen Sie die Einstellung **Wiederherstellungsmodell**. Für eine Demodatenbank genügt die Wahl der Option **Einfach**.

Verlassen Sie das Fenster durch Klicken auf **OK**, damit die neue Datenbank erzeugt werden kann.

Anmeldung anlegen

Nach dem Anlegen der Datenbank müssen Sie eine Anmeldung für diese Datenbank anlegen. Im Objekt-Explorer wählen Sie dazu im Kontextmenü des Knotens **Sicherheit/Anmeldungen** den Eintrag **Neue Anmeldung**.

Im folgenden Fenster **wählen Sie zunächst die Seite Allgemein**, wählen Sie oben die Option **SQL Server-Authentifizierung** und geben Sie den Namen des Datenbankbenutzers, z.B. PPMDEMO (in Großbuchstaben), und das Kennwort ein. Weisen Sie dem Benutzer die gewünschte Standarddatenbank zu, z.B. die zuvor erstellte Datenbank **PPM**.

Das vergebene Kennwort benötigen Sie später beim Anlegen eines PPM-Mandanten. Deaktivieren Sie die Option **Kennwortrichtlinie erzwingen**, wenn Sie ein beliebiges Kennwort eingeben möchten.

Wählen Sie jetzt die Seite **Benutzerzuordnung**, um das Standardschema automatisch beim ersten Anmelden des neu angelegten Benutzers zu erzeugen. In der Spalte **Zuordnen** wählen Sie die Datenbank, der Sie für diesen Anmeldenamen ein Schema zuordnen möchten. In der Spalte **Benutzer** wird dann der eben eingerichtete Anmeldename angezeigt und das Feld in der Spalte **Standardschema** wird veränderbar. Geben Sie in dieser Spalte den gewünschten Namen an. Es empfiehlt sich, als Standard-Schemanamen den Anmeldenamen zu verwenden, z.B. PPMDEMO.

Der Standard-Schemanamen muss in Großbuchstaben angegeben werden, z.B. PPMDEMO.

Mit der Eingabe des Namens des Standardschemas und durch Klicken auf **OK** wird dieses Schema angelegt.

DB-Benutzer anlegen

Öffnen Sie die gewünschte Datenbank, z.B. die zuvor erstellte Datenbank **PPM**, und wählen Sie im Konfigurationsbaum des Objekt-Explorer auf dem Knoten **Sicherheit/Benutzer** den Eintrag **Neuer Benutzer**.

Geben Sie in der Seite **Allgemein** den DB-Benutzernamen und den eben angelegten Anmeldenamen ein. Der DB-Benutzername sollte identisch mit dem Anmeldenamen sein. Geben Sie in das Feld **Standardschema** den Namen des Schemas ein, das Sie beim Anlegen des Anmeldenamens dem Benutzer zugeordnet haben.

DB-Eigenschaften anpassen

Um das erzeugte Datenbankschema mit dem angelegten Benutzer für PPM verwenden zu können, müssen Sie abschließend noch bestimmte Berechtigungen des neuen Benutzers und Schemas setzen. Wählen Sie im Objekt-Explorer des **Microsoft SQL Server Management Studio** den Schemaknoten der erzeugten Datenbank, z.B. den Knoten **/Datenbanken/PPM/Sicherheit/Schemas/PPMDEMO** der zuvor erstellten Datenbank **PPM**. Wählen Sie im Kontextmenü den Eintrag **Eigenschaften**.

Im folgenden Fenster **Schemaeigenschaften** wählen Sie den Knoten **Berechtigungen** und öffnen durch Klicken auf **Datenbankberechtigungen anzeigen** das Fenster **Datenbankeigenschaften**. Der Knoten **Berechtigungen** ist dort bereits gewählt. Wählen Sie den neu angelegten Benutzer, z.B. PPMDEMO,

und weisen diesem die Berechtigungen **Schema erstellen** und **Tabelle erstellen** zu durch Aktivieren der Kontrollkästchen in der Spalte **Erteilen**.

Überprüfen der Datenbankanbindung

Starten Sie PPM Customizing Toolkit und legen einen neuen Mandanten an. Machen Sie im Dialog **Datenbankeinstellungen** die Angaben zur neu erzeugten SQL-Server-Datenbank und klicken Sie anschließend auf **Datenbankanbindung testen**.

Das Ergebnis der Überprüfung der Datenbankanbindung wird in einem eigenen Fenster angezeigt.

Wenn der Verbindungsaufbau misslingt, können Sie sich durch Klicken auf die Info-Schaltfläche ausführliche Informationen anzeigen lassen.

Achten Sie vor dem Start von PPM Customizing Toolkit darauf, dass sich der aktuelle Treiber für die SQL-Server-Datenbank im dafür vorgesehenen Verzeichnis **<Installationsverzeichnis>\ppm\server\bin\work\data_ppm\drivers** befindet.

3.3.5. Unicode-Unterstützung

PPM unterstützt Multibyte-Zeichensätzen bei der Verwendung von Microsoft SQL-Server als Datenbanksystem.

Wenn Sie PPM mit Microsoft SQL-Server Unicode-Datenbanken verwenden möchten, müssen Sie folgendes beachten.

- PPM unterstützt SQL-Server-Datenbanken, für die Collation **Latin1_General_BIN** eingestellt ist. SQL-Server-Datenbanken mit anderen Collations werden nicht unterstützt.
- Vergeben Sie die Collation auf Datenbankebene.
- Wenn Sie die Einstellung der Collation nachträglich ändern, werden vorhandene Inhalte der Datenbank nicht umgesetzt. Damit die geänderten Datenbankeinstellung wirksam wird, müssen Sie das Datenbankschema des entsprechenden PPM-Mandanten durch Ausführen des Kommandozeilenprogramm **runinitdb** neu initialisieren.

Warnung
Beim Initialisieren der Datenbank werden alle importierten Daten gelöscht.

3.3.6. Migration

Wenn Sie Ihre vorhandenen SQL-Serverdatenbanken nach Microsoft SQL-Server migrieren möchten, überprüfen Sie, dass für diese Datenbanken die Collation **Latin1_General_BIN** eingestellt ist.

Warnung
Wenn Sie Datenbanken nach Microsoft SQL-Server migrieren, für die eine andere Collation als **Latin1_General_BIN** eingestellt ist, ist nicht gewährleistet, dass die migrierten Datenbanken weiterhin fehlerfrei von PPM verwendet werden können.

4. Performance Tuning

Dieses Kapitel zeigt Ihnen Performance beeinflussende Zusammenhänge zwischen der PPM-Server-Software und dem Datenbanksystem.

4.1. Gesamt-Performance

Die Laufzeiteffizienz des PPM-Serversystems insbesondere des Imports wird zu großen Teilen von dem Datenbanksystem bestimmt.

Beachten Sie folgende Performance beeinflussende Hinweise für Ihr PPM-System:

- Beobachten Sie die Laufzeitkonfiguration der Datenbankinstanz in regelmäßigen Abständen und passen Sie diese bei erkennbarem Verbesserungspotential entsprechend an.
- Halten Sie die Datenbankstatistiken auf aktuellem Stand.
- Reorganisieren Sie nach dem Verdichten oder Löschen einer großen Menge von Prozessinstanzen die Tabellen und lassen Sie alle Indizes des Datenbankschemas neu berechnen.
- Das Kommandozeilentool **runppmimport** unterstützt Sie beim Neuberechnen der Indizes. Geben Sie das Argument **-index new** an, z. B.:
- runppmimport -client <Mandantenname> -user system -password <Kennwort> -index new

Brechen Sie diesen Vorgang nach Möglichkeit nicht ab, um zeitaufwändige Konsolidierungsprozesse im Datenbankschema zu vermeiden.

4.1.1. Hardware spezifisch

Die eingesetzte Hardware beeinflusst direkt die Gesamtperformance des PPM-Serversystems.

- PPM-Server und -Datenbanksystem sollten nach Möglichkeit auf demselben Rechner installiert sein oder über ein ausreichend schnelles Netzwerk verbunden sein.
- Wählen Sie als Speicherort der Tablespace-Dateien nach Möglichkeit ein Dateisystem, das auf einem RAID 5 oder RAID 10 Array basiert.
- Eine weitere Steigerung der Performance erreichen Sie, indem Sie die Tablespace der PPM-Tablespace-Typen (siehe Kapitel Unterstützte Datenbanksysteme [12]) auf physikalisch unabhängige Dateisysteme verteilen.
- Konfigurieren Sie eine MS SQL-Server-Datenbank so, dass Transaktions-Logs und Datenbankdateien in physikalisch unabhängigen Dateisystemen gespeichert werden.

4.1.2. Konfigurationsspezifisch

- Melden Sie selbst definierte Dimensionen und Kennzahlen an einem bestimmten Prozesstyp oder Prozesstypgruppe an, damit diese Dimensionen und Kennzahlen nur für Prozessinstanzen dieses Prozesstyp oder dieser Prozesstypgruppe berechnet werden.

  Warnung
  Die Standardkennzahlen Prozess- und Funktionsanzahl, die Standarddimensionen Prozesstyp, Zeit und Funktion sowie alle Datenzugriffsdimensionen müssen an der Prozessbaumwurzel angemeldet sein, da sie für interne Berechnungen verwendet werden.

- Verwenden Sie wenn möglich Prozesskennzahlen und -dimensionen. Anfragen von Funktionskennzahlen und -dimensionen sind aufgrund der mehrfach größeren Datenmenge berechnungsintensiver. In der Regel werden Funktionskennzahlen nur benötigt, wenn eine Funktion in einer Prozessinstanz mehrfach vorkommt und Sie jede Funktion unabhängig voneinander auswerten möchten.

4.2. Import-Performance

Beim Einlesen der Fragmente und Berechnen der Prozessinstanzen werden große Datenmengen in die Tablespaces des Datenbanksystems geschrieben. Die Performance des Dateisystems, auf dem die Tablespace-Dateien gespeichert sind, bestimmt hauptsächlich die Performance des Datenimports.

5. Support und rechtliche Hinweise

In diesem Abschnitt finden Sie einige allgemeine Informationen zum Produktsupport und zu rechtlichen Belangen.

5.1. Dokumentationsumfang

Die zur Verfügung gestellten Informationen beschreiben die Einstellungen und Funktionalitäten, die zum Zeitpunkt der Veröffentlichung gültig waren. Da Software und Dokumentation verschiedenen Fertigungszyklen unterliegen, kann die Beschreibung von Einstellungen und Funktionalitäten von den tatsächlichen Gegebenheiten abweichen. Informationen über solche Abweichungen finden Sie in den mitgelieferten Release Notes. Bitte lesen und berücksichtigen Sie diese Datei bei Installation, Einrichtung und Verwendung des Produkts.

Wenn Sie technische und/oder geschäftliche Systemfunktionen ohne Nutzung der Service-Leistungen der SAG ARIS GmbH installieren möchten, benötigen Sie umfangreiche Kenntnisse hinsichtlich des zu installierenden Systems, seines Zwecks sowie der Zielsysteme und ihrer Abhängigkeiten untereinander. Aufgrund der Vielzahl von Plattformen und sich gegenseitig beeinflussender Hardware- und Softwarekonfigurationen können nur spezifische Installationen beschrieben werden. Es ist nicht möglich, sämtliche Einstellungen und Abhängigkeiten zu dokumentieren.

Beachten Sie bitte gerade bei der Kombination verschiedener Technologien die Hinweise der jeweiligen Hersteller, insbesondere auch aktuelle Verlautbarungen auf deren Internet-Seiten bezüglich Freigaben. Für die Installation und einwandfreie Funktion freigegebener Fremdsysteme können wir keine Gewähr übernehmen und leisten daher keinen Support. Richten Sie sich grundsätzlich nach den Angaben der Installationsanleitungen und Handbücher der jeweiligen Hersteller. Bei Problemen wenden Sie sich bitte an die jeweilige Herstellerfirma.

Wenn Sie Hilfe bei der Installation von Drittsystemen benötigen, wenden Sie sich an Ihre örtliche ARIS Vertriebsorganisation. Bitte beachten Sie, dass diese Art von herstellerspezifischer beziehungsweise kundenspezifischer Anpassung nicht vom Standard ARIS Software Wartungsvertrag abgedeckt ist und nur auf besonderen Wunsch und mit entsprechender Zustimmung geleistet werden kann.

5.2. Support

Produktsupport

Wir bieten allen Kunden mit einem gültigen Supportvertrag Unterstützung für ARIS Produkte.

Informationen zur Verfügbarkeit und zum Support von Produktversionen sowie zu den Zeiträumen für den fortlaufenden Support finden Sie in der ARIS Community.

Wenden Sie sich an unsere Global Support Service unter ARIS Community um Support-Anfragen zu stellen und zu aktualisieren oder Ideen und Funktionsanforderungen zu äußern.

Community

Registrieren Sie sich bei ARIS Community, um Produkte, Updates und Fixes herunterzuladen, Experteninformationen zu nutzen und sich mit anderen Benutzern auszutauschen.

Dokument

Für Informationen über die Verwendung und Aktivierung von Produkten und für andere Fragen zum technischen Support besuchen Sie die ARIS Dokumentations-Website.

Wenn Sie zu Dokumentationsinhalten Feedback geben möchten, schreiben Sie an 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.