PPM OPERATION GUIDE

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. IT service and application management 6
2.1. Early alert system in case of memory shortage 6
2.1.1. Memory monitoring 6
2.1.2. System messages 7
2.1.3. Configuration 7
2.2. Workload monitoring 8
2.3. Maintenance concept 9
2.3.1. System update 9
2.3.2. Data archiving 9
2.3.3. Restore 10
2.3.4. Backup and restore user data 10
2.4. High availability systems 10
2.5. Support desk management concept 11
2.6. Data protection (GDPR) 13
2.6.1. Personal data used in PPM 13
2.6.2. How to delete and anonymize personal data 14
3. Third-party software 16
4. Performance aspects 17
4.1. Dimension indices 18
4.1.1. Updating the indices 19
4.1.2. Dependency on the process tree 19
5. Administration 20
5.1. ARIS Cloud Agent 20
5.1.1. Change the ARIS Cloud Agent user credentials 20
5.2. System status 21
5.3. Configuration 21
5.3.1. Load balancer configuration 22
5.3.2. Change the registry addresses 24
5.4. System messages 24
5.5. Proxy configuration 25
6. PPM system messages 26
6.1. Overview 26
6.2. Log file directories 28
6.3. System messages (log output) 28
6.4. Audit messages (audit logging) 32
6.5. Command line programs 36
6.6. Handle personal data in log files 39
6.6.1. Log files of client-specific runnables 39
6.6.2. Log files of system-specific runnables 40
7. PPM system monitoring 41
7.1. Using ACC commands 42
7.1.1. Monitor runnable's health 42
7.1.2. Show node 43
7.1.3. Configure health monitoring 44
7.1.3.1. Monitor the memory usage 46
7.1.3.2. Stop the system monitoring 46
8. Globalization 48
9. Job automation 49
10. Accessibility 50
11. Single Sign-On integration 51
11.1. Integrate PPM and ARIS Aware in an SSO scenario 51

11.2. Integrate PPM in an SSO scenario using Kerberos 53
11.3. Use of truststore and keystore in SSO scenario 54
12. Security aspects 55
12.1. Safe communication channels 55
12.2. Load balancer 55
12.2.1. Integrate TLS certificate 55
12.2.2. Further security measures for load balancer 56
12.3. TLS between PPM client downloader and PPM server 57
12.4. Generate and sign TLS certificates 57
13. Session timeouts 59
14. Command line programs 60
14.1. General structure and function-specific structure 60
14.2. PPM command line programs 61
14.3. Common parameters 62
14.4. runinitdb 63
14.5. runppmanalytics 64
14.5.1. runppmanalytics parameters 65
14.6. runppmconfig 66
14.6.1. runppmconfig parameters 66
14.6.2. Export the system configuration 69
14.6.3. Import and export of user data 70
14.6.4. Import and export of favorites 71
14.7. runppmcpi 73
14.7.1. runppmcpi parameters 73
14.8. runppmimport 75
14.8.1. runppmimport parameters 75
14.8.2. Recalculation 79
14.9. runppmcompress 79
14.9.1. runppmcompress parameters 80
14.10. runppmdelete 82
14.10.1. runppmdelete parameters 83
14.11. runppmexport 85
14.11.1. runppmexport parameters 85
14.12. runppmadmin 88
14.12.1. runppmadmin parameters 88
14.13. runppmsendmail 91
14.13.1. runppmsendmail parameters 91
14.14. runtableimport 92
14.15. rundimdata 93
14.16. runpikidata 93
15. Frequently asked questions 94
15.1. System integration 94
15.1.1. Database connection 96
15.1.2. SAP-2-PPM Process Extractor 97
15.1.2.1. Using Content Packages 98
15.2. System administration 98
15.2.1. PPM command line programs: Log output 102
15.2.2. Inconsistent data pools 102
16. Support and legal information 104
16.1. Documentation scope 104
16.2. Support 104
16.3. Legal notices 105

1. General

This guide provides information about integrating the PPM product components **ARIS Process Performance Manager**, **PPM Customizing Toolkit**, and **PPM** process extractors into a complex system landscape. It describes certain aspects that should make it easier for you to integrate these products into an existing system landscape.

Furthermore, this guide contains references to detailed, component-specific documents.

2. IT service and application management

ARIS Process Performance Manager is a tool designed to analyze actual working processes. The data comprising the steps of these processes is extracted from application systems and consolidated to process instances. Then, measures are calculated for these process instances. In this way, PPM provides a comprehensive overview of the processes in a company and supports you in identifying weak spots.

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.

High availability

PPM is an analytical application that imports data, calculates it, and then saves it to a database schema. The database system used must ensure the integrity of the data. You can restore a specific system status any time by importing the reference data again. PPM itself does not support high availability criteria or adaptive computing concepts.

2.1. Early alert system in case of memory shortage

When starting, the PPM analysis server can use only the amount of main memory that was specified in the client properties. If this limit is reached, out-of-memory exceptions occur. These memory errors can result in analysis server instability which is a critical issue during data import. They can also result in differences between analysis and client server data, which requires a reinitialization of the analysis server. Reinitialization of the analysis server should, however, be avoided because it can take up to several hours with large data volumes. Analysis server performance also drops with high memory load through frequent internal administration tasks (garbage collection).

2.1.1. Memory monitoring

To prevent performance loss and memory exceptions as well as the associated instability during memory shortages, memory usage of the analysis server is monitored. The system reports the risk of imminent server performance loss caused by memory shortages and errors. Memory usage checks of the analysis server's memory utilization requires specific system resources. To prevent the check from affecting overall system performance, it is cyclical and performed only for certain system activity:

- Data import

  - Upon import of new data started with the **runppmimport** command line program (or specific recalculation with paramsets)

  - Before calculated EPCs are written to the database during the import of new data. Thus, EPCs whose calculated data can no longer be saved in the analysis server are not written to the database, and analysis server and recovery files remain consistent.

  - At the end of data import
- Before and after manual input of process instance-independent measures
- Before and after importing process instance-independent measures using the **runpikidata** command line program, and before imported PIKI data sets are written to the database.

2.1.2. System messages

Checking the system memory load can have the following results:

- Normal
  Sufficient memory available for continued unlimited data import.
- Unfavorable
  Available memory is so low that performance loss is to be expected. If you continue to import data the critical threshold will soon be reached and further data import will become impossible.
- Critical
  Available memory is so low that data can no longer be imported. If this is the case, any current data import will be canceled. From this moment, the system is locked for another data import.

During monitoring, unfavorable or critical system states are output as a warning in the analysis server's logs.

If the system was locked for another data import due to critical memory load, you can unlock it by:

- Stopping the client server and analysis server and increasing the analysis server's memory utilization. Subsequently, restart the analysis server and client server.
- Aggregating or deleting a sufficient amount of data.

If the system has been locked for further data import due to critical memory load, you cannot unlock it directly by aggregating or deleting data. Instead, restart the PPM servers with sufficient memory utilization for the analysis server and start data import again by running the **runppmimport** command line program. Afterwards, you can aggregate or delete data. In the end, you can reset memory utilization to the original value.

> **Tip**
> By running the **runppmadmin** command line program with the **-memoryinfo** option, you can manually check the memory load at any time.

2.1.3. Configuration

The monitoring configuration of system memory utilization (also known as Memory Load Guard) is specified by certain key values in the configuration file **Analyseserver_settings.properties**. If one of these keys is not entered or contains invalid values, a warning is output in the analysis server logs and a default value is set for this key.

The key **MemoryLoadGuard.Enabled** specifies whether memory load monitoring is enabled or not. Valid values are **true** and **false**. The default value is **TRUE**.

You can change the configuration during analysis server runtime. The changed values will be considered the next time the system memory load is checked.

Thresholds

The two thresholds identifying the limits of unfavorable and critical memory load area are specified as a percentage of the analysis server's maximum memory available. Decisive is the memory load of the Old

Generation (memory area where Java objects are saved permanently. The Java documentation calls this "committed size", that is, the ratio of maximum and actually used size.)

Values below 50% are ignored because they are not viable.

The key **MemoryLoadGuard.Warn.PercentageOfMemoryUsed** indicates the limit at which unfavorable memory load begins. The default value is 90%. This value cooperates efficiently with the internal **Garbage Collection** component of the Java Runtime Environment. This way, imminent memory shortages can be tackled in time.

The key **MemoryLoadGuard.PreventImport.PercentageOfMemoryUsed** indicates the limit at which critical memory load begins. The default value of 95% allows the system to identify critical memory shortages early on and thus enables it to still import large data volumes safely.

Times of day

You can restrict cyclical memory load checks to specific times of day in order to save the performance of the analysis server for a current session. Start time and end time for a cyclical memory load check are specified in the keys **MemoryLoadGuard.BackgroundCheck.TimeWindow.Start** and **MemoryLoadGuard.BackgroundCheck.TimeWindow.End**. The times are specified in 24-hour notation **hh:mm** (ISO 8601). It is based on the system time of the analysis server. You can change the values during analysis server operation.

- Once a memory load check has started it will run to the end even if this takes longer than the end time specified. The memory load check is triggered by certain system activities and is independent of the cyclical check.
- We recommend that you schedule the check for a time during which the server processes only few queries.

Example 1.

The following file extract shows the default configuration of the early alert system in case of memory shortage. The system is active, the limit for unfavorable memory load is 90%, the limit for critical memory load is 95%. The cyclical memory load check is active between 20:00 and 6:00 of the following day.

```
#### Memory Load Guard ####
MemoryLoadGuard.Enabled=true
MemoryLoadGuard.Warn.PercentageOfMemoryUsed=90
MemoryLoadGuard.PreventImport.PercentageOfMemoryUsed=95
# Time window where background checks may occur.
MemoryLoadGuard.BackgroundCheck.TimeWindow.Start=20:00
# End minute is not included. However, if both Start and End are set to the
same value checks are possible all day.
MemoryLoadGuard.BackgroundCheck.TimeWindow.End=06:00
```

2.2. Workload monitoring

To monitor PPM server processes, you can use the operating system's monitoring programs. Windows systems include the standard programs **Task Manager**, **Performance Monitor**, and, from Windows Vista/ Server 2008 Windows **Resource Monitor**. The monitoring programs provided for Linux depend on the distribution and installation variant.

The platform-independent Java program **Visual VM** is well suited for monitoring the system resources used by the PPM server processes.

Java Visual VM

In various views, this program visualizes the utilization of specific system resources, for example, CPU and memory utilization. Only the system load utilized by the Java process (for example, PPM server) within the Java VM is analyzed. The program **Visual VM** is part of the Java SDK 1.8. To start under Windows, execute the file **jvisualvm.exe** in the **bin** subdirectory of the SDK installation. To start under Linux, execute the file **jvisualvm** in the **bin** subdirectory of the SDK installation. Visual VM utilizes the standard JMX functionality of Java VM. PPM does not support JMX.

Visual VM can analyze only Java processes that were started with the same user ID that the **jvisualvm** program was started with. An easy way of monitoring Java processes that were started as a Windows service is to configure the relevant service via the Windows Control Panel so that the service is executed with the ID of the user who starts the program **jvisualvm**.

With Visual VM you can also monitor Java processes running on a remote computer. To do so, start the program **jstatd** in the **bin** directory of the SDK installation on the computer on which the Java processes to be monitored are running. Visual VM programs started on remote computers can then establish a connection to **jstatd** and monitor local Java processes.

2.3. Maintenance concept

2.3.1. System update

After installation of the initial release versions of PPM by means of the ARIS PPM Setup (see documentation **PPM Installation**), you can update the products within the same service release using the ARIS PPM Patch Setup (for versions 10.5.10 and lower) and ARIS PPM Setup (for versions 10.2025.10 and higher) and apply official fixes to your PPM installation. Updates to newer service releases are possible from version ARIS PPM 10.5.0 onwards. More details on updating the system and migrating your data can be found in the **PPM Migration** guide.

We urgently recommend that you apply official fixes using the ARIS PPM Patch Setup or ARIS PPM Setup after you installed the release version.

A detailed update description is available in the **PPM Installation** manual.

2.3.2. Data archiving

All process instances created based on the basis of imported source system data, process instance-independent measures, configurations imported by the **runnppmconfig** configuration program (customizing), as well as configurations extended or edited by the user via the front-end (for example, PPM users, user-defined measures) are saved in the PPM client's database. They are automatically saved when the database schema is backed up. Therefore, you should back up the database on a regular basis.

By default, the configuration files of a PPM client are saved by client in the **config** and **custom** directories under <installation directory>\ppm\server\bin\work\data_ppm\. Save these directories each time you change the configuration. Save the **patch** directory in **<installation**

directory>\ppm\server\bin\work\data_ppm. In this directory, runtime components are saved that globally adapt the PPM functionality. The client-specific runtime components for adapting the PPM functionality are saved in **<installation directory>\ppm\server\bin\work\data_ppm\custom\<client>\patch** They are automatically archived when the **custom** directory is saved.

You must manually save the PPM directory of the Web server and all user-specific report definitions. The default path for the Web server directory is <PPM installation>\server\bin\work\data_ppm\web\.

If you want to save all global and client-specific configurations and data, create a backup copy of the **data_ppm** directory including all subdirectories.

2.3.3. Restore

In case of a crash recovery, restore the PPM installation by reinstalling it with the same options you used for the original installation. You need to recreate all clients with the original settings using PPM CTK. If required, restore the databases of all PPM clients. After this, you can copy the archived files back to their original location as described in the chapter on **Archiving [9]** , just in reverse order, and start the PPM system.

2.3.4. Backup and restore user data

To secure the PPM user data in central User Management you can back the relevant user data and restore them.

To obtain information on backing up and restoring data of central user management please see the central User Management documentation or contact the responsible system administrator.

To back up or restore data you need to log in to central user management as a user with administration privileges.

2.4. High availability systems

Overview

High availability systems are used to minimize the risk of the failure of the entire system caused by the failure of an individual component (SPOF, single point of failure). Such systems are usually based on redundant single systems, that is, multiple identical systems with identical data are operated in parallel. One of these single systems is the main system that users normally access. Another system is the backup system. If the main system fails the backup system takes over automatically. The cause of failure is irrelevant, be it sudden, unpredictable causes like hardware problems or planned maintenance activities, for example.

To identify an error a specific system monitoring message can be used (for example, the S.M.A.R.T. protocol for identifying hard drive errors). This procedure is complemented by frequent system queries that require a correct response. This type of queries is called heartbeats. Heartbeats can be active at various system levels, like specific application queries (for example, application pings) or system resource monitoring, for example, hardware monitoring. Errors occur when a system error message occurs or if there is no response to a heartbeat. The reserve system takes over automatically and a message is sent to the system administrator.

After switching to the backup system, the error in the main system can be analyzed and fixed. When the error is fixed the main system can take over again after data synchronization. High availability systems distinguish between the two strategies hot standby and cold standby.

Hot standby means that in case of failure a system remains available even during an active session. Users do not notice if the main system fails and a backup system is activated. The switch to the backup system happens without delay and without interruptions of active user sessions. This strategy is mainly used in mission-critical systems, for example, when safety and health of the general public are threatened.

If the main system fails, cold standby in contrast to hot standby provides a certain time span during which the backup system is activated and the system switches to backup system operation. During the switch phase the system is not available. This means that it is not certain that active user sessions are resumed when the system is available again after switching to the backup system.

PPM high availability system

PPM is an analytical application that imports data, calculates it, and then saves it to a database schema. The database system used must ensure the integrity of the data. You can restore a specific system status any time by importing the reference data again. PPM itself does not support high availability criteria or adaptive computing concepts, such as the restore of an interrupted session between PPM server and client. Indirectly, though, you can implement scenarios in which you can switch between the main system and an existing backup system. However, you need to accept a certain downtime during which the PPM system is unavailable (cold standby).

A 3-level system concept is recommended for setting up such a PPM high availability system. This means that the database server used by PPM is transferred to an independent system that already fulfills high availability criteria. Usually, database manufacturers offer high availability versions of their products. The other PPM system components are installed and operated on yet another system.

Archive the PPM system at regular intervals as described in the chapter **Archiving [9]**. In case of PPM system failure, proceed as described in the chapter **Restore [10]** for a new system with comparable hardware properties.

2.5. Support desk management concept

If unexpected behavior or errors occur during runtime you should check the system log output first. It usually provides clues as to the cause of the error or unexpected behavior. The PPM system uses the **log4J** interface to output system messages. This enables you to obtain more detailed information on the error cause by changing the configuration of the output of system messages by adjusting (refining) the log levels during PPM server runtime. Chapter **PPM system messages [26]** describes the configuration of system message output in detail. Connection to external support desk systems, such as SAP Solution Manager, is not supported by PPM.

If errors occur during the import of data or configuration the log output of the PPM system is very important for the analysis. First, you analyze the messages of the import program on the console or in the log file specified. Many times, you can already determine the cause of the error this way. The chapter on **Command line programs [36]** describes in detail how to handle system message output of command line programs. The system messages of the PPM client and analysis servers can support your error analysis. PPM client and analysis servers each write their system messages in a file. The location where these files are saved is specified in the relevant appenders in the file **Server_Log_settings.properties** or **Analysisserver_Log_settings.properties**. The default directory is **<installation directory<\ppm\server\bin\work\data_ppm\log\<client>**; the log files have the name pattern **<client>_<type>.log**, and analysis server log files have the suffix **_as** following the client name. For example,

the analysis server of the **umg_en** client writes error messages to the file **umg_en_as_error.log** in the **<installation directory>\ppm\server\bin\work\data_ppm\custom\umg_en\log**.

System messages of the PPM servers can be processed in different ways, for example, by using individual appenders. Chapter System messages (log output) [28] describes in detail the configuration of system message output.

Errors occurring during the analysis using the PPM front-end are displayed in the front-end session in a separate dialog window. You can add the output of the program's run time messages (stack traces) to the system message display. To do this, click the error dialog's **Details** button.

If you report an error to the manufacturer's support your message needs to include the following information:

- A detailed description of user and system activities being performed when the error occurs.
- Is the error reproducible? Is this a sporadic error? This information is very important for reproducing the error.
- If applicable, which project-specific runtime libraries have been added to the PPM installation?
- If applicable, log output that was output when the error occurred.
- What software versions are in use (PPM, Java, database)?

You can determine the versions of the software installed as follows:

- Java

  All applications always use the same Java version.

  To identify the version, execute the command line **<installation directory>\ppm\server\jre\bin\java -version** in a command prompt.

  Example

  C:\> ARIS\ppm\server\jre\bin\java -version
- PPM

  To determine the PPM version installed, execute the **runppmadmin** command line program with the option **-version**.

  Example

  C:\> cd ARIS\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-95.1.0-RC8-trunk-20130404.122823-4-runnable.zip\ppm\bin & runppmadmin -version

  S: 21.04.11 08:59:16: [SRV] application: 95.1.0 (10055)

  S: 21.04.11 08:59:16: [SRV] database schema: 95
- Database (and JDBC driver)

  When starting, the PPM server outputs messages stating which database and JDBC driver versions are used. You can also retrieve this information from the server's log files.

  Examples

  - Oracle

    I: ...: [SRV] Database version used: Oracle9i Enterprise Edition Release 9.2.0.7.0 - Production

    I: ...: [SRV] JDBC driver used: Oracle JDBC driver (10.2.0.4.0).
  - IBM DB2

    I: ...: [SRV] Database version used: SQL09013.

    I: ...: [SRV] JDBC driver used: IBM DB2 JDBC Universal Driver Architecture (3.4.65).
  - MS SQL Server

    I: ...: [SRV] Database version used: 8.00.2039.

    I: ...: [SRV] JDBC driver used: Microsoft SQL Server 2005 JDBC Driver (1.1.1501.101).

2.6. Data protection (GDPR)

You can delete or at least anonymize the personal data used in PPM to comply with the General Data Protection Regulation (GDPR).

The following chapters explain:

- The PPM components that save personal data [13], such as IP addresses, names, MAC addresses etc.
- How to delete personal data in PPM. [14]

2.6.1. Personal data used in PPM

User data is managed centrally in the User Management Component (UMC) of PPM. When creating users manually, the following data is mandatory:

- User name
- First name
- Last name

User data is also stored in the PPM Administration component to assign user privileges for PPM application and the imported process data. The following additional personal data can be stored:

- E-Mail address

If users imported or synchronized using LDAP, additional personal data can be stored:

- Telephone number
- LDAP DN
- ID
- Picture

The User Management Component creates audit logs in an attached database. This provides a history of changes to functions, licenses, and access rights. For this purpose, user names and IP addresses are logged.

Even if users were deleted, user names are stored in a hidden attribute together with the time of deletion in order to log the changes. The hidden attributes are automatically deleted when upgrading to a new major version of the ARIS Infrastructure. These entries can be anonymized for deleted users.

User privileges related to PPM are managed in the PPM Administration. Here the following information is available for each user:

Mandatory:

- Name (User Name from above)

Optional:

- First Name

- Last Name
- E-mail address

User names and IP addresses are also stored for several purposes in log files: audit and trace logs for all components.

In addition, user names are also available in the PPM database to store the above mentioned PPM privileges, which are not available in UMC.

As a business process monitoring and analysis tool, PPM might also import personal data from processes extracted from external source systems such as SAP, CSV files, or database systems. These processes might contain personal information from which you can identify a person involved in the process monitored. This information depends on the data extracted from the source system and is subject to the customizing model used for this source system.

2.6.2. How to delete and anonymize personal data

To comply with the GDPR, you must delete personal data of individuals who are no longer employed after a defined period of time. Since personal data may be present in the PPM database, PPM log files, and the process data itself, there are several ways to delete or anonymize this data.

PPM database

Deleting a user from UMC will not automatically delete the user in PPM. You must delete the user also in PPM. To do this, go to the PPM Administration –> **User Privileges** section, select the user, and delete it. This will automatically delete the user also in the PPM database and all his related assignments (process access rights, favorites etc.).

PPM log files

For some actions, PPM tracks the user ID (user name) and IP address of the executor. This data is used to analyze and fix potential problems that occur during the system operation.

The tracked user data is stored in client-specific as well as system-specific log files. You must delete the corresponding log files to remove the relevant personal user data.

For a detailed description of the system messages and PPM log files, see PPM Operation Guide > chapter PPM system messages [26].

Warning
If you delete the log files, all logged data is lost and cannot be restored.

Process data

You can disguise personal data embedded in process data to import into PPM. PPM provides functionality for encoding special attributes and fields that contain personal data to be extracted from the source system.

The data is not visible in plain text in PPM afterwards. You can revert this pseudonymization by providing the correct encryption key in the PPM UI. The procedure is accessible only to PPM system administrators.

Note that after importing the process data and external data into PPM, you can no longer change the included personal data. You must configure pseudonymization before importing.

3. Third-party software

PPM requires certain third-party software. PPM has been developed in the Java programming language as a client-server architecture. HTTP, HTTPS, and RMI are used as communication protocols.

Java Runtime Environment

To be able to run PPM, you need a Java Runtime Environment that is supplied when installing the product. A separate installation is not required. If you want to activate additional, performance-enhancing options of the PPM server in the runtime environment, please refer to the PPM Installation Guide. The PPM client is started using Java WebStart. This is done by downloading a JNLP file from the PPM Web server and running it in the client's Java runtime environment. You can also use the PPM client as a Java application by selecting the PPM Analysis GUI component in the installation program.

Relational database system

PPM saves imported and calculated data as well as client configurations in the database schema of one database user. PPM has been tested and approved for use with these database systems **Oracle**, **IBM DB2** and **MS SQL Server**.

Internally, PPM still uses commercial and open-source runtime libraries (for example, Apache license).

Update of 3rd party libraries

PPM is scanned on a regular basis for vulnerabilities in our own code (PPM libraries) but also in the used 3rd party libraries. The scan tools used comply with the most recent industry standards and the scans are carried out as an automated process daily within the CI/CD pipeline to build the product. Detected vulnerabilities are treaded according to the Service Level Agreements for security issues of <_sga>.

4. Performance aspects

The PPM system is designed for operation in a LAN. The volume of data to be transferred between the PPM server and client depends greatly on the action executed on the client. Depending on the number of measures and dimensions queried, analysis queries to the PPM server may produce extensive result sets, which then have to be transferred to the client. For this reason, the PPM system transfers the data in compressed form by default because the resulting improvement in system performance far exceeds the calculation effort required to compress the data.

You can also further improve system performance by using a Web server to publish certain precalculated analysis results instead of having direct communication between the PPM client and server.

System import times basically depend on hardware requirements: main memory, number of CPUs, free disk space. To estimate hardware requirements for an individual system, setup offers three predefined scenarios described below. Nevertheless, they only represent rough recommended values because the actual memory requirements depend on many more factors. Therefore, an individual hardware requirements can only be estimated in cooperation with SAG ARIS GmbH support.

When using PPM on the basis of a virtualized operation system (VMWare) you need to make sure that the main memory for this VM is exclusive for this VM when you set up the VM. If physical resources of the VMWare server are distributed to multiple VMs, the VM on which PPM is installed might run into problems. Due to using in-memory technology, we recommend reserving the main memory exclusively for the VM running PPM.

Setup differentiates between the three scenarios listed in the following table.

Hardware scenarios for PPM

PPM scenarios/ special aspects (4)		Demo scenario (1) Small setup scenario (64-bit)	Medium scenario (2) (Proof of concept) Medium setup scenario (64-bit)	Large scenario (3) (Production) Large setup scenario (64-bit)
PPM 10.2026.4	YPI	up to 200 000 PI/DR	up to 5 million PI/DR	up to 25 million PI/DR
		1 PPM client	1 PPM client	1 PPM client
		up to 2 users	up to u 10 users	up to 100 users
		2 CPU cores	4 CPU cores	8 CPU cores
		4 GB RAM	16 GB RAM	64 GB RAM
		10 GB free HD space	20 GB free HD space	50 GB free HD space

Legend	
PI	Process instances
DR	Data rows (used in Data analytics)
1	Supports only 64-bit OS, database on PPM server machine (tablespace: 1 GB)
2	Supports only 64-bit OS, database on PPM server machine or separate machine (tablespace: 50 GB)
3	Supports only 64-bit OS, database on separate machine (tablespace: 150 GB)
4	The values indicated for each scenario apply only to the defined number of PPM clients. Hardware requirements grow with the number of PPM clients and users.

4.1. Dimension indices

The analysis server can index certain process dimensions. The purpose of an index is to speed up certain queries that have a filter on this dimension. The index is held in memory so that adding more indices can make it necessary to increase the configured analysis server memory.

Data access dimensions automatically get indexed. You can select additional dimensions specific to your usage scenario. These will usually be one or more dimensions that are frequently used for selective filtering. A good example would be a highly selective filter that is always specified when accessing the main dashboard. For advice on which dimensions to index, please contact SAG ARIS GmbH Product Support.

For multi-level, two-level, variant, time, and time of day dimensions, you can also specify in which granularity (refinement) the index is to be maintained. If no refinement or an incorrect refinement is specified, the coarsest level (BY_LEVEL_1, BY_LEVEL1_N, BY_YEAR, BY_HOUR_OF_DAY) is used.

Indices are allowed for the following dimension types:

- oneleveldim
- twoleveldim (Granularity/Refinement: BY_LEVEL1, BY_LEVEL2)
- variantdim
- nleveldim (Granularity/Refinement: BY_LEVELX_Y", where X is the level and Y is the maximum level on the dimension, for example: BY_LEVEL1_4, ..., BY_LEVEL4_4)
- timedim (Granularity/Refinement: BY_DAY, BY_MONTH, BY_QUARTER, BY_YEAR)
- hourdim (Granularity/Refinement: BY_MINUTE_OF_DAY, BY_HOUR_OF_DAY)

The indices are defined in the settings file **AnalysisServer_settings.properties:**

- INDEX.PROCESS.<DIMENSION_KEYWORD>.USE=true
  enables the index on the dimension with the keyword **DIMENSION_KEYWORD**.
- INDEX.PROCESS.<DIMENSION_KEYWORD>.REFINEMENT=<REFINEMENT>
  optional, to specify a refinement for the corresponding index.

Example 2.

INDEX.PROCESS.VSTEL.USE=true -> Index on the Dimension **VSTEL**.

INDEX.PROCESS.MATERIAL.USE=true

INDEX.PROCESS.MATERIAL.REFINEMENT=BY_LEVEL2_4 -> Index on the second level of the dimension **MATERIAL**.

INDEX.PROCESS.TIME.USE=true

INDEX.PROCESS.TIME.REFINEMENT=BY_DAY -> Index on the dimension **TIME** with the accuracy **DAY**.

Currently, only indices on process dimensions are supported. In addition, only one index per dimension is currently allowed. This means that the index can only be created with one refinement, which is not a problem since the index can be used for filters of any refinement although with different performance.

If an index is switched on or off several times (.USE=true/false), the last specified value is used. If several refinements are specified, the last one is also used.

4.1.1. Updating the indices

The indices are completely regenerated at the end of the PPM import after all process instances have been updated on the analysis server. This means that the selection of dimensions on which there is an index and their configuration is updated on the analysis server with each PPM import, compression, or deletion. Even if the PPM import does not find any new data to be imported, the indices are regenerated with the currently configured settings.

Restarting the analysis server from the recovery file also updates all indices.

4.1.2. Dependency on the process tree

Only process instances whose process type has a registered indexed dimension are included in an index.

If an indexed dimension is registered or deregistered at a process tree node (at a later time), these process instances are missing from the index or are too many. The index can therefore no longer be used until regenerated. Only with the next import or restart, when all indices are regenerated, the index is then available again.

For this reason, all indices are set to "invalid" when certain configurations of dimensions in the process tree are changed. Changes to the configuration which result in the indices being set to "invalid" are:

- Import of a new measure configuration (using runppmconfig or PPM CTK)
- Importing a new process tree configuration (using runppmconfig or PPM CTK)
- Importing a new configuration into the Process Mining configurator
- Automatic process tree extension as part of the PPM import

5. Administration

Product installation comprises the individual PPM-specific product components and, as a common basis, various infrastructure components for user and license management. A detailed list of components including their description is available in the PPM Installation Guide. The following chapters include descriptions of the key administrative tasks and the tools used.

5.1. ARIS Cloud Agent

ARIS Cloud Agent uses ARIS PPM Cloud Controller to control the PPM system installed. The ARIS PPM Cloud Controller default configuration allows it to control the ARIS Cloud Agent installed on the local computer (localhost) and does not require user login. ARIS Cloud Agent, user name, and password are predefined and transferred as parameters to ARIS PPM Cloud Controller upon starting.

You can start ARIS PPM Cloud Controller in the Windows program group **Start** > **All Programs** > **ARIS** > **Administration**.

It is, however, also possible to start ARIS PPM Cloud Controller via the command line. To do so, go to **<PPM installation directory>\ppm\server\acc**

and enter the following command:

acc.bat -c "<PPM installation directory>\ppm\server\generated.apptypes.cfg" -h localhost -p 17009 -u Clous -pwd g3h31m.

Start ARIS PPM Cloud Controller and in the command line, enter the command **help** to display available ARIS PPM Cloud Controller commands and related descriptions.

You can change the ARIS Cloud Agent user credentials. [20]

5.1.1. Change the ARIS Cloud Agent user credentials

For security reasons, you can change the ARIS Cloud Agent user credentials. An ARIS Cloud Agent user has also access to ARIS PPM Cloud Controller. To prevent unauthorized access, you can change the relevant user credentials using ARIS PPM Cloud Controller.

Procedure

1. Start ARIS PPM Cloud Controller. [20]
   a. To change the ARIS Cloud Agent user name, execute the following command.
      set username=<user name>
   b. To change the ARIS Cloud Agent password, execute the following command.
      set password=<password>
2. Update the properties **user** and **password** in the following properties file.
   <PPM installation>\ppm\server\bin\work\data_ppm\system\provisioning\cloudagent_settings.properties

The ARIS Cloud Agent user credentials are changed.

5.2. System status

You can query the status of the individual system components installed in ARIS Cloud Agent using the **list** command. All installed components including name, status, and version are output. The following is a list of the key commands for controlling the individual components.

Commands

ARIS PPM Cloud Controller command	Description
list	Lists all components installed on the selected ARIS Cloud Agent node. Default: localhost
start <instance id>	Starts the component specified with <instance id>
stop <instance id>	Stops the component specified with <instance id>
startall	Starts all components in a specific sequence. You can set this sequence in ARIS Cloud Agent.
stopall	Stops all components in the reverse start sequence.
killall	Closes all components regardless of the current state or existing connections. Use kill <instance id> to close a specific component. Use this command only if components no longer respond to other commands.

5.3. Configuration

Using ARIS PPM Cloud Controller you can change the configurations of individual system components. For example, you can add new database drivers for PPM or create new users in the user database. The following is a list of the key commands for configuring the individual components.

Commands

ARIS PPM Cloud Controller command	Description
show config	Displays the current configuration of ARIS Cloud Agent and ARIS PPM Cloud Controller.
show instance <instance id>	Displays the current configuration of a component. **Example 3.** show instance ppm_core Returns all configuration parameters of the **ppm_core** component.

ARIS PPM Cloud Controller command	Description
enhance [driver]	Use the **enhance** command to install required drivers. **Install database drivers** enhance <client> with dbDriver local file <path to DB driver> **Install SAP JCO drivers** enhance <client> with dbDriver local file <path to SAP JCO 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. **Example 4.** The Oracle database driver ojdbc6.jar is stored under C:\Temp. Enter the following command to install this driver for the PPM client **umg_en**. enhance umg_en_cs with dbDriver local file C:/Temp/ojdbc6.jar
enhance [user]	The command **enhance** can also be used for creating a new user in central user management. enhance <instance id> with createUser trigger only options tenant.name="default" tenant.user.name=system tenant.user.pwd=manager affected.user=user1 affected.pwd=user1 affected.first.name="John" affected.last.name="Doe" affected.email="test@test.de" affected.description="description" Creates a new user **user1** with the password **user1**. Please use the name of the component **User management** for the instance ID (default: umcadmin).
reconfigure <instance id> <parameter>	This command is used for reconfiguring components. You can change all parameters saved in the configuration file under <installation directory<\ppm\server\bin\work\work_<instance id>\runtimeinfo.properties. Changing these values in this file does not affect the instance running. Parameters can be changed only with ARIS PPM Cloud Controller. 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. Replace a backslash **** used in the password with a double backslash ****. **Example 5.** You want to change the RMI communication between the PPM server components to TLS encryption. The following commands need to be executed in ARIS PPM Cloud Controller. ▪ stop ppm_core ▪ reconfigure ppm_core ssl.enabled="true" ssl.keystorefile="<path to keystore file>" ssl.keystorepassword="<keystorepassword>" ▪ start ppm_core

5.3.1. Load balancer configuration

You can use ARIS PPM Cloud Controller to configure the load balancer parameters. The commands for configuration of the most important parameters are listed below.

The variable **<loadbalancer_x>** used below can have the values **loadbalancer_s**, **loadbalancer_m**, or **loadbalancer_l** depending on the memory model used for the installation. Refer to the **PPM Installation** documentation for details.

Stop the load balancer (ARIS PPM Cloud Controller command **stop <loadbalancer_x>**) before changing any settings and start it again after finishing (ARIS PPM Cloud Controller command **start <loadbalancer_x>**).

Configuration of the HTTP port

Change the http port (<port number>) using the **HTTPD.port** parameter. To do this, run the following command in ARIS PPM Cloud Controller:

reconfigure <loadbalancer_x> +HTTPD.port=<port number>

If you want to use HTTPS as the default communication protocol for the load balancer, run the following commands in ARIS PPM Cloud Controller:

reconfigure loadbalancer_m +zookeeper.application.instance.port=4443

reconfigure loadbalancer_m +zookeeper.application.instance.scheme=https

Configuration of the HTTPS port

By default, the HTTPS port is not enabled in an initial PPM installation. To enable or change the HTTPS port use the **HTTPD.ssl.port** parameter. In order to do this, run the following command in ARIS PPM Cloud Controller:

reconfigure <loadbalancer_x> +HTTPD.ssl.port=<port number>

To enable HTTPS protocol in the load balancer, you need a valid certificate and key pair.

Configuration of the certificate

In order to use the HTTPS protocol in the load balancer, you need a valid certificate and key. Both files can be stored in a ZIP file. The files need to be named **server.crt** and **server.key** within the package. To import the TLS certificate required for HTTPS protocol run the following command in ARIS PPM Cloud Controller:

enhance loadbalancer_x with sslCertificate local file <Certificate file>.zip

More information on how to create a certificate file can be found in chapter Generate and sign an TLS keystore.

Configuration of the FQDN (full qualified domain name)

The load balancer checks the computer name and sends a redirect if the computer name does not match the FQDN. To change the FQDN, you have to enter the following two commands in ARIS PPM Cloud Controller:

reconfigure <loadbalancer_x> +zookeeper.application.instance.host=<New FQDN>

reconfigure <loadbalancer_x> +HTTPD.servername=<New FQDN>

Disable HTTPS

To disable HTTPS in the load balancer, run the following command in ARIS PPM Cloud Controller.

reconfigure loadbalancer_m +HTTPD.ssl.port=0

In case HTTPS was set as the default communication protocol, you need to run the following commands in ARIS PPM Cloud Controller:

reconfigure loadbalancer_m +zookeeper.application.instance.port=4080

reconfigure loadbalancer_m +zookeeper.application.instance.scheme=http

reconfigure <loadbalancer_x> +zookeeper.application.instance.scheme=http

5.3.2. Change the registry addresses

You can use ARIS PPM Cloud Controller to change the ports for the RMI registry.

Change RMI registry

Enter the following command in ARIS PPM Cloud Controller.

reconfigure ppm_core +ppmrmi.zookeeper.application.instance.port=<New port> (default=17500)

Change host name

To link the RMI to a host name other than your own, you must change the RMI registry using the following command.

Reconfigure ppm_core +ppmrmi.zookeeper.application.instance.host=<FQDN of RMI registry>

This may be necessary if PPM is being operated as a master/sub-server system in the cloud and the internal host name of a sub-server does not match the name accessible externally.

Example 6.

Cloud VM has the internal host name **SAGBASE**. The computer is accessible externally under the host name **IP_xxx.xxx.xxx.xxx.doma.in**. The RMI registry for the sub-server VM must then be changed to the name accessible externally.

5.4. System messages

If there is an error in the infrastructure, you can view the system messages (log output) of the individual components to analyze the problem. The following table lists the log files of the individual infrastructure components. The PPM system messages [26] are described in detail in the subsequent chapter.

Infrastructure component	Log file path
ARIS Cloud Agent	<installation directory>\ppm\server\logs\CloudAgent.log
	<installation directory>\ppm\server\bin\log\agent.log
ARIS PPM Cloud Controller	<installation directory>\ppm\server\acc\log\acc.log
Cloud search	<installation directory>\ppm\server\bin\work\work_cloudsearch_<memory model>\defaultLog\log.txt

Infrastructure component	Log file path
Elasticsearch	<installation directory>\server\bin\work\work_elastic_<memory model>\elasticsearch\logs\elasticsearch.log
Load balancer	<installation directory>\ppm\server\bin\work\work_loadbalancer_<memory model>\httpd\logs*.log
Central User Management	<installation directory>\ppm\server\bin\work\work_umcadmin_<memory model>\base\logs*.log
PostgreSQL DB	<installation directory>\ppm\server\bin\work\work_postgres_<memory model>\pgworkdata\pg_log*.log

A tool exists to collect and package all relevant log files in a single step. This may be required for a support query when all log files are requested for the analysis of a problem. To collect the log files, switch to the following folder of your PPM installation.

<installation directory>\ppm\server\support

Execute the command **collectlogfiles.bat** under Windows and **collectlogfiles.sh** under Linux.

You can delete all log files using the batch file **deletelogfiles.bat** under Windows and the shell script **deletelogfiles.sh** under Linux.

5.5. Proxy configuration

You can specify proxy settings for the Java Virtual Machine (JVM) that can be used by the PPM download client to connect to the PPM Server.

To specify a proxy, add a semicolon separated list to the **JavaProperties** option in the **arisloader.cfg** file. The file can be found in the following directory.

<PPM Installation>\server\bin\work\data_ppm\web\config\

If this option is not available, you must create it as shown in the following example.

Example 7.

JavaProperties=java.net.useSystemProxies=true

For more information about configuring proxy settings for a JVM, see the Oracle JVM documentation.

After you have configured the **arisloader.cfg** file, you must download the PPM download client to activate the new configuration.

6. PPM system messages

6.1. Overview

To output system messages (log output), the PPM server uses the standard Java interface log4J2. log4J2 is a flexible framework for controlling output of Java application messages using various media (output channels). The log4J interface is configured either in an XML or property file. It consists of three components: logger (collection), layout (formatting), and appender (output)

PPM extracts the log4J2 configuration from a property file.

Logger

The Java application passes its messages to a certain logger (collector) that processes them depending on their type. Messages can be of these types (log levels): **ALL**, **TRACE**, **INFO**, **WARN**, **ERROR**, **FATAL**, **OFF**.

Log level	Description
ALL	Turns on all logging.
TRACE	Very detailed runtime messages, often combined with output of complete error states of the application
DEBUG	Comprehensive runtime message, often describing the internal state of the application
INFO	General information or warnings
WARN	Information about application states that are not critical for runtime, for example, missing or wrong configurations
ERROR	Errors often leading to cancelation of the active component
FATAL	Errors leading to abortion of the application, for example, lack of resources
OFF	No log output

The following lines define a logger in the log4J configuration file:

logger.<key>.name = <logger name>

logger.<key>.level = <log level>

logger.<key>.appenderRef.<appender key 1>.ref = <appender name 1>

logger.<key>.appenderRef.<appender key 2>.ref = <appender name 2>

logger.<key>.appenderRef.<appender key n>.ref = <appender name n>

Logger names are case-sensitive. The names form a hierarchical structure. The root element is the root logger. The hierarchy levels are separated by a **period** (.). The higher hierarchy level passes on its configuration to the lower level, which is then overwritten by specific settings.

The table below illustrates the inheritance hierarchy:

Logger name	Assigned log level	Inherited log level
rootLogger	INFO	none
logger.LOG	DEBUG	INFO

Logger name	Assigned log level	Inherited log level
logger.LOG.SRV	ERROR	INFO

Appender

The logger passes the messages to the appenders (recipients) that output the messages in a certain form (layout). Appenders differ in terms of their output, for example:

PPM appender example	Description
console	Messages are output on the console
logFile	Messages are written to a file
errorFile	Error messages are written to a file

The following lines define an appender in the log4J2 configuration file:

appender.<key>.name = <appender name>

appender.<key>.type = [File | Console]

You can specify multiple appenders for a logger. All appenders of a logger output the messages based on their configuration.

Some appenders write the messages to the log system of the operating system, send them as e-mails, or send them via the network to a particular log server.

Layout

You can use styles to specify the output format. These styles are specified for the relevant appender. Key placeholders for a style include:

Placeholder	Description
%c	Category/logger (corresponds to the PPM module)
%C	Class name, fully qualified
%d	Date, for example, %d{HH:mm:ss,SSS}
%xd	Date that also supports the ppmdate token, for example, %xd{ppmdate} ppmdate is a localized default pattern according to your locale.
%F	File name
%L	Line number
%m	The message itself
%M	Name of method
%n	Line break
%p	Priority, level (INFO, WARN, ERROR, etc.)
%t	Name of thread
%throwable{0}	Stacktrace output is suppressed.

Example 8.

The style **[%t] %-5p %C - %m%n** generates the following output:

[main] INFO org.apache.log4j.Action - The process was started

6.2. Log file directories

The log files of the various PPM components are stored in the following directories.

- ARIS PPM Cloud Controller:
  <installation directory>\ppm\server\acc\log
- Runnable instances:
  <installation directory>\ppm\server\bin\work\work_<instance>\ and their subdirectories
- ARIS Cloud Agent log output:
  <installation directory>\ppm\server\bin\log
- Client-specific command line programs:
  <installation directory>\ppm\server\bin\work\data_ppm\custom\ <client>\log
  <installation directory>\ppm\server\bin\work\data_ppm\config\ <client>\log
- Log files of the PPM and analysis servers:
  <installation directory>\ppm\server\bin\work\data_ppm\log\<client> *)
- Client setup log output:
  <installation directory>\ppm\server\bin\work\data_ppm\system\log
- RMI registry console output:
  <installation directory>\ppm\server\bin\work\work_ppm_core\defaultLog
- PPM server console output:
  <installation directory>\ppm\server\bin\work\work_<client>_cs\ defaultLog
- Analysis server console output:
  <installation directory>\ppm\server\bin\work\work_<mandant>_as\ defaultLog

*) Default setting: Directories and names of the log files are configured in the client-specific configuration files **AnalysisServer_Log_settings.properties** and **Server_Log_settings.properties**.

6.3. System messages (log output)

The PPM server uses the **log4J2** interface to output system messages. Each PPM component (for example, data import, merger, Measure calculator) uses own loggers. The PPM system modules (components) have the following superordinate loggers: l**og**, **trace**, and **audit**.

By default, the PPM system has the following appenders for formatted message output.

Appender	Description
console	Messages are output on the console. Only still relevant for command line programs.
logFile	Messages are written to a file. The name of the file is specified by the configuration key **appender.logFile.fileName**.

Appender	Description
errorFile	Error messages and exceptions (log output starting with E: or F:) are written to a file. The name of the file is specified by the configuration key **appender.errorFile.fileName.** Exceptions and stack traces are output to this file only.
traceFile	Messages are only written to a file. The name of the file is specified by the configuration key **appender.traceFile.fileName**. This appender is used for collecting certain messages to support troubleshooting.
auditFile	Writes audit log messages to a file. The name of the file is specified by the configuration key **appender.auditFile.fileName**.

For the client server, message output is configured in the file **Server_Log_settings.properties**, for the analysis server in the file **AnalysisServer_Log_settings.properties**. Both configuration files including their default entries are automatically created during installation.

The configuration files **Server_Log_settings.properties** and **AnalysisServer_Log_settings.properties** are checked for changes every 60 seconds (fixed setting). Therefore, the configuration of log output can be changed without having to restart the PPM servers.

Module-based logging

The familiar PPM module names are used for system message output of PPM client server and analysis server. They are configured in the files **Server_Log_settings.properties** (client server) or **AnalysisServer_Log_settings.properties** (analysis server), respectively.

PPM supports the following logger modules:

Module name	Description
LOG.XML	XML source system data import
LOG.EIM	Internal import of imported fragment instances
LOG.MGR	Merge of process fragments
LOG.TYP	Typification of process instances
LOG.KIC	Calculation of measures and dimensions
LOG.PLV	Planned value calculation
LOG.CFG	Internal configuration management
LOG.STD	PPM server in **STANDARD** mode
LOG.SRV	Standard module used for log output of the PPM server
LOG.CNV	Database converter
LOG.IMP	Import and export of configurations (for example, using **runppmconfig**)
LOG.ADM	Client administration (for example, using **runppmadmin**)
LOG.KG	Key creation for imported process fragments (key generator)
LOG.PRF	Profiler (SA)
LOG.OLA	Query interface
LOG.REP	Creation of reports
LOG.RAU	Report automation
LOG.MST	PPM server in **MASTER** mode
LOG.SUB	PPM server in **SUBSERVER** mode

Module name	Description
LOG.LOG	General log output
LOG.EAL	Early alert system
LOG.ALY	Irregularity analysis (**runppmanalytics**)
LOG.CCS	Attribute calculator in **SILENT** mode
LOG.CCD	Attribute calculator in **DEFAULT** mode
LOG.CCV	Attribute calculator in **VERBOSE** mode
LOG.PFM	sysmon client
LOG.MON	Query monitoring
LOG.PFS	Statistics of user-specific favorites
LOG.SFS	Statistics of shared favorites
LOG.CPI	Improvements, message administration
LOG.ASRV	Standard module used for log output of the analysis server
LOG.MEM	Early alert system in case of memory shortage
LOG.DAI	Data analytics import

In message output, module codes (corresponding to the module name without the prefix **LOG.**) are not localized. Therefore, for messages for the Measure calculator, the module code is **KIC**.

You must specify a log level for each logger module. The following log levels are supported: **INFO**, **WARN**, **ERROR**, **FATAL**, **OFF**. The affinity of the log levels selected increases from right to left, that is, if you set the log level to **WARN**, messages of the **ERROR** and **FATAL** levels are output, as well.

Example 9.

Information, warnings, errors, and critical errors are to be output on the console for the XML data import.

logger.LOG.XML.name=LOG.XML

logger.LOG.XML.level=INFO

logger.LOG.XML.appenderRef.console.ref = console

- Log levels are not case-sensitive, for example, logger.LOG.XML.level=error is the same as logger.LOG.XML.level=ERROR.
- Incorrect log levels, such as logger.LOG.XML.level=error are ignored. In this case, the log level of the superordinate logger is used, which is logger.LOG.level=INFO in this example.
- If no log level was specified for a logger module, the log level of the superordinate logger applies, in this example this would be logger.LOG.level=INFO.

Trace message output

In contrast to logger modules, trace modules do not have log levels. Instead, message output can only be turned on or off. This output contains further information on the error. The following trace modules exist: ASRV, SRV, CONFIG, ABT, CONPOOL, FRQ, RET, RETDS, RETQT, QPL, SELITY, DIM, IMP, RMI, MEM, MON, PPI, EIP, CLC, UPD, DST, QUE, SES, TRANS, RSAPI, RSPPMUI, RSCONFMC, RSPMC

Instead of the log level, you can specify the following settings for trace modules: **OFF**, **TRACE**.

Example 10.

Trace messages are to be output for the client server, but not for the analysis server.

logger.TRACE.SRV.level=TRACE

logger.TRACE.ASRV.level=OFF

Log output prefixes

Each log output has a prefix indicating the corresponding log level. This prefix is identical for all languages.

Log level	Prefix for all languages
Statistics	S:
FATAL	F:
ERROR	E:
WARN	W:
INFO	I:
TRACE	T:

Statistics messages are output always and can only be suppressed by the **OFF** log level.

Format log output

The log output format is specified in the two configuration files **Server_Log_settings.properties** and **AnalysisServer_Log_settings.properties**. The PPM default date format is specified by %d{ppmdate} in these files.

The date format used depends on the language that the client and analysis servers were started with. The following date formats are used for the six languages supported by PPM:

Language	Locale	Date format
English USA	EN	MM/dd/yy HH:mm:ss
English GB*	US	dd/MM/yy HH:mm:ss
German	DE	dd.MM.yy HH:mm:ss
French	FR	dd.MM.yy HH:mm:ss
Japanese	JA	dd/MM/yy HH:mm:ss
Russian	RU	dd.MM.yy HH:mm:ss
Chinese	ZH	dd/MM/yy HH:mm:ss

*The **English USA** date formate is used by PPM as default. To switch to the English GB date format, see the documentation **PPM Installation**, chapter **Adapt date format to British English language area**.

Example 11.

appender.errorFile.layout.pattern=%p %xd{ppmdate} [%c{1}] %m%n

Scaled systems

The configuration of log message output for a scaled PPM system is basically the same as for a standard system with only one PPM server. However, the following should be observed:

- All exception and trace messages are output on the corresponding sub-servers. Exceptions are passed on to the master server and written to its error log.
- Since the master server is not connected to an analysis server, master servers can output only trace messages of the client server. Trace messages of the analysis servers are written to the log output of the respective sub-servers.

6.4. Audit messages (audit logging)

Audit messages (audit logs) document important events in a system and are read by specific persons (auditors). An audit message contains all kinds of information for the auditor to be able to understand the process in detail.

Information	Description
Time stamp	Time at which the changes triggered by the process were saved
Process	Process identifier and description
Component	Component code
Users	Login ID of the user who executed the process
IP address	Computer on which the process was triggered
Application	ID of the application that triggered the process

Only those processes are logged whose execution led to data being saved permanently. Changes reverted by the user are not logged.

Example 12.

The following message shows that the user **SYSTEM** changed the user administration (USR component) at the time specified, using the GUI of the local computer of the server system (127.0.0.1) by executing the process with the ID 800. The following text describes the process in detail.

```
27.09.2010 16:36:25.527, USR, [800] User configuration was changed, SYSTEM,
127.0.0.1, GUI
```

Component ID

Each component has a unique, language-independent identifier. The following table shows the valid values and descriptions.

Component ID	Description
KIC	Measure configuration

Component ID	Description
OKI	User-defined measures
RKI	Ranking measures
ABC	User-defined dimensions
TFD	Top-flop dimensions
KID	Measure-dependent dimensions
PRT	Process tree configuration
USR	User management
REP	Report definition
RAU	Report automation
FAV	Shared favorites
MGR	Merger
EPC	Process instances
PIK	Process instance-independent measures
PLV	Planned values
ORG	Organizational units
FTC	Factory calendar
SRV	Server

Application ID

The auditor uses the application ID to find out which application caused the change. The following table shows the valid values and descriptions.

Application ID	Description
GUI	PPM user interface
EXP	Export
CIM	Configuration import
IMP	Data import (process instance-independent measures, dimension data, PPM)
CHK	Checks, for example, planned value deviations
CMP	Aggregation and deletion of instances
CVT	Database converter
INT	Internal application
SHR	Session management
ADT	Adapter
ADM	Admin tool
MST	Master-sub-server connection
OTH	Other application

Configuration

To output audit messages, PPM uses the log4J logger **AUDIT** of the logging interface described in the previous chapter. Audit logging is configured in the client-specific configuration file **Server_Log_settings.properties** based on the log4J interface (see chapter Overview [26]).

Only the PPM client server logs audit messages, the analysis server does not.

Log messages

In addition to basic information (such as time, components, process, etc.), the various PPM components output different types of messages, which are described below.

Only processes that permanently change data in the PPM system are logged. Data exports are not logged.

Measure configuration

For changes to the measure configuration made using the **runppmconfig ... -keyindicator** command line program, the following statistical data is output in addition to the basic information: Number of processed attribute calculation functions, number of processed attributes, number of processed measures, dimensions, and relations, number of processed process-independent measure series.

Process tree configuration

There are different messages depending on whether the process tree configuration has been changed using the **runppmconfig ... -processtree** command line program or automatically extended by importing new process instances with new process types. In the latter case, the names of the new process type groups and process types are output as additional information.

Extended measure configuration

Some messages are output regardless of whether the extended measure configurations (user-defined measures, ranking measures, ABC dimensions, user-defined dimensions, top-flop dimensions) were changed using the **runppmconfig** command line program or via the user interface.

User management

This component reports changes of users, user groups, group membership, data access privileges, passwords, function and access privileges, and it provides information on system login and logout actions. Login and logout actions by users logging in/out via query interface are not output.

If the user configuration is changed using the **runppmconfig ... -users** command line program, statistical data for the import process is output.

Connection to central User Management

Changes triggered by PPM which affect central User Management are not logged. Only changes taking place in PPM are output. When you activate or deactivate users or user groups in PPM a corresponding text including additional information is output.

When importing using the **runppmconfig** command line program, you can create new users in central User Management and in PPM or in just one of the two systems. If you create at least one new user in PPM a corresponding text is output. Changes in central User Management are not logged.

Central user management checks the number of users based on your license. If the number of users is exceeded a corresponding text is output.

Query monitoring

This component reports cancellations of long lasting queries that exceed the configured run time.

Organizational units

The system logs the creation, editing, and deletion of organizational units regardless of the process being triggered via the interface or the command line program.

Factory calendar

Changes of reference periods, working days, and factory calendar transformation factors are logged.

Reports

Saving report definitions and changing access privileges of a report definition are logged. Running a report definition is not logged.

Report automation

The system logs the creation, editing, and deletion of report automations regardless of the process being triggered via the interface or the command line program. The user **SYSTEM** can delete the report automations of all users by importing an overall configuration valid for all users in **replace** mode*. This process is logged with the text **Report automations of all users deleted**.

Favorites

Since shared favorites can be used in the measure configuration, the system logs changes to shared favorites (creation, editing, access privilege changes, renaming, moving, and deletion), regardless of the process being triggered via the interface or the command line program.

Changes of private favorites and the favorites cache are not logged.

Planned values

The system logs the creation, editing, and deletion of planned values regardless of the process being triggered via the interface or the command line program.

Command line programs

Some command line programs report system changes.

Aggregation and deletion of instances via **runppmcompress** and **runppmdelete** is logged only if the program runs in **execute** mode.

The messages differ depending on whether the data series for process instance-independent measures have been changed using the **runpikidata** command line program or via the user interface. Import of dimension data using the **rundimdata** command line program is logged, as well.

PPM data import

The **runppmimport** command line program manages the data existing in the PPM system (merging process fragments into process instances, anonymizing users in organizational units, typifying process instances, recalculating measures, deleting shared fragments and process keys, creating database indices, reinitializing analysis servers). A corresponding message outputs the execution of any of these processes.

The recalculation of database statistics is not logged.

Start and stop the server

Starting and stopping the PPM client server is logged.

The process is not logged if you stop the server using Task Manager or if you exit the command prompt the server runs in.

Scaled systems

If you use a scaled system, the master server and sub-servers log the same audit information as a stand-alone server. Please not that certain processes are and logged only by the master server or one or multiple sub-servers.

6.5. Command line programs

The command line programs write their messages to the console or the log file you specified with the **-protocolfile** parameter, see chapter Common parameters [62]). There is no difference between error and trace file. Exceptions are output with other log output in the file specified or on the console. The value of the key **PRINT_STACKTRACE_ON_EXCEPTION** in the configuration file **Server_settings.properties** specifies

whether the stack trace of an exception is output, as well. Valid values are **TRUE** and **FALSE**, the default value is **FALSE**.

The specified log options (**-information**, **-warning**, **-error** parameters) set the log level for each module to the specified value **INFO**, **WARN**, or **ERROR**. The log level configured for the module in the file **Server_Log_settings.properties** will be considered. Due to the hierarchical log level structure, please note the following when evaluating the log options specified. The log level to be used is determined as follows:

1. In the order of the **-information**, **-warning**, **-error** parameters specified the last **no** parameter found is determined.
2. In the order of the **-information**, **-warning**, **-error** parameters specified the first **yes** parameter found is determined.
3. From the "no" value found last in the first step and based on the module-specific default log level, the more restrictive log level is determined. If no "no" value is specified the default log level applies.
4. From the **yes** value found first in the second step and based on the log level determined in the third step, the log level with the least restrictive log level is determined.

Example 13.

The default log level for the module is INFO. When running a command line program with the parameters **–information yes –warning no –error yes** the log level is determined as follows.

- Last "no" value for –warning, log level is ERROR
- First "yes" value for –info, log level is INFO
- MAX(INFO, ERROR) is ERROR
- Log level = MIN(ERROR, INFO) is INFO

Example 14.

The resulting log level is INFO.

The default log level for the module is WARN. When running a command line program with the parameter **–information yes** the log level is determined as follows.

- The last "no" value is not set.
- First "yes" value for –info, log level is INFO
- MAX(WARN) is WARN
- Log level = MIN(WARN, INFO) is INFO

Example 15.

The resulting log level is INFO.

The default log level for the module is INFO. When running a command line program with the parameter **-warning no** the log level is determined as follows.

- Last "no" value for –warning, log level is ERROR
- First yes value not set.
- MAX(ERROR, INFO) is ERROR
- Log level = MIN(ERROR) is ERROR

The resulting log level is ERROR.

Summary

- Running a command line program with the parameter **-information yes**

  Information, warnings, errors, and fatal errors are output regardless of the settings for the other log parameters or the module-specific log level.
- Running a command line program with the parameter **-information no -warning yes**

  Warnings, errors, and fatal errors are output regardless of the settings for the other log parameters.

  Furthermore, information is output for modules whose module-specific log level is set to INFO.
- Running a command line program with the parameter **-information no -warning no -error yes**

  Errors and fatal errors are output. Furthermore, information is output for modules whose module-specific log level is set to INFO, as well as warnings for modules whose module-specific log level is set to INFO or WARN.
- Running a command line program with the parameter **-information no -warning no -error no**

  Only fatal errors are output. No messages are output for modules whose module-specific log level is set to OFF.

Format log output

The log output format of the command line programs for console output is specified by the **console** appender format. The log output format of log output to a file is specified by the appender **logFile**.

Except for the appender styles, no further configurations are evaluated for the command line programs from the files **Server_Log_settings.properties** and **Analysisserver_Log_settings.properties**. Therefore, additional appenders for the log output of the command line programs cannot be specified.

Server log output

The command line programs **runppmconfig**, **runppmreport**, and **runppmadmin** output particular log messages of the client server and analysis server. These messages appear both in the server log output and the command line program log output and are indicated by the additional module name.

Example 16.

...

I: 05.10.10 14:41:53 [IMP] Importing configuration of component "users"...

I: 05.10.10 14:41:53 [IMP] [SRV] Importing users...

I: 05.10.10 14:41:53 [IMP] [SRV] Importing groups...

I: 05.10.10 14:41:53 [IMP] [SRV] The following 0 users were added: []

I: 05.10.10 14:41:53 [IMP] The configuration of the component "users" from the file

. . .

6.6. Handle personal data in log files

For some actions PPM tracks the user ID and IP address of the executor. This data is used to analyze and fix potential problems that occur during the system operation. You can remove this personal data from your PPM System.

The tracked user data is stored in client-specific as well as system-specific log files. To remove the relevant personal user data, you must delete the corresponding log files.

Warning
If you delete the log files, all logged data is lost and cannot be restored.

6.6.1. Log files of client-specific runnables

You must delete the log files of the PPM client-specific runnables (client instances) manually.

The client-specific log files are stored in the following directories.

- <PPM_WORK_DIR>\work_<client>_as\defaultLog\
- <PPM_WORK_DIR>\work_<client>_cs\defaultLog\
- <PPM_WORK_DIR>\data_ppm\log\<client>\
- <PPM_WORK_DIR>\data_ppm\custom\<client>\log\

Warning
It is recommended that you stop the runnables for which you want to delete the corresponding log files. However, you must first evaluate whether stopping a runnable can lead to problems in the running productive system. For example, stopping a runnable during a data import can lead to data loss.

We recommend stopping the runnables for which you want to delete the corresponding log files. Otherwise, problems may occur while the PPM system is running.

Procedure

1. Start ARIS PPM Cloud Controller. (Windows program group **ARIS > PPM 10.2026.4 > Administration > Start ARIS PPM Cloud Controller 10.2026.4**)
2. Enter the **list** command in ARIS PPM Cloud Controller to list all runnables.
3. To stop a specific runnable, enter the **stop <runnable>** command in ARIS PPM Cloud Controller, for example, stop ppm_core.
4. To stop all runnables, enter the **stopall** command in ARIS PPM Cloud Controller.
5. In the Windows® Explorer, go to the directories mentioned above and delete all relevant log files.
6. Enter the **startall** command in ARIS PPM Cloud Controller to restart all runnables or enter the **start <runnable>** command to restart a specific runnable.

The user data is deleted from the PPM system.

6.6.2. Log files of system-specific runnables

The user data is also stored in the log files of the system-specific runnables (loadbalancer, zookeeper, elastic search, and umcadmin). You can delete the log files using the ARIS PPM Cloud Controller.

The log files are located in the corresponding folders in the PPM work directory.

<PPM installation>\server\bin\work\work_<runnable>\<runnable>.log

Example 17.

- <PPM_WORK_DIR>\work_elastic_m\elasticsearch.log
- <PPM_WORK_DIR>\work_elastic_m\zookeeper.log

> **Warning**
> It is recommended stopping the runnables for which you want to delete the corresponding log files. However, you must first evaluate whether stopping a runnable can lead to problems in the running productive system. For example, stopping a runnable during a data import can lead to data loss.

Procedure

1. Start ARIS PPM Cloud Controller. (Windows program group **ARIS > PPM 10.2026.4 > Administration > Start ARIS PPM Cloud Controller 10.2026.4**)
2. You can enter the **list** command in ARIS PPM Cloud Controller to list all runnables.
3. Enter the **stop <runnable>** command in ARIS PPM Cloud Controller to stop the runnable, for example, stop loadbalancer.
4. Enter the **delete log files for <runnable>** command to delete the corresponding log files, for example, delete log files for loadbalancer.
5. Enter the **start <runnable>** command in ARIS PPM Cloud Controller to restart the runnable, for example, start loadbalancer.

The user data is deleted from the PPM system.

You can delete the log files of any stopped system runnable (zoo, elastic, umcadmin) in the same way.

7. PPM system monitoring

The monitoring of physical resources is important to keep PPM running properly. You can monitor the system using the ARIS PPM Cloud Controller commands.

File space

Monitor the file system of the PPM installation. When the usage exceeds 85%, increase the disk space.

If the space is consumed by normal usage of the application, for example, many clients, many documents, etc., the available disk space must be increased. Check whether disk space can be made available. Call the **deleteLogFiles.bat** script (<PPM installation>\server\support). This ensures that current log files are written properly. A maintenance window for the runnable should be used, so that it can be stopped while the script is running. Also check, whether heap dumps have been created. They can consume a large part of disc space.

We recommend increasing the available disk space, if the used space exceeds 75 % of the available disk space.

Memory usage

Every runnable has a **JAVA-Xmx** configuration parameter. It has to be monitored, if the memory usage is rising over time.

Peaks are not a problem, if they do not break the limit. If the system is suffering over a longer time, for example, the memory usage is higher than 70% for more than one hour, check for rarely occurring use cases. They might explain the high memory consumption. This can be:

- more users accessing as usual
- expensive reporting
- running database maintenance operations
- assigning user access right over the whole database etc.

If the defined limit for memory consumption is exceeded, there are two possible measures:

- Extend the heap size for the concerned runnable. This can be done in ARIS PPM Cloud Controller with a configure command for the **JAVA-Xmx** parameter. But this step only makes sense, if enough additional memory is available.
- Scale up by increasing the resources, for example, adding a new node/runnable of the same type or assigning more memory to the virtual machine and increasing the **JAVA-Xmx** parameter for the runnable to use this memory.

CPU load

A continuous high level of CPU usage can be an indicator that it might be necessary to scale up/out the system.

For the CPU load the same parameters as for the heap usage regarding peaks are true. So if a high CPU load is detected (> 60% for more than one hour), first it should be analyzed, if short peaks or exceptional use cases are the reason. If this is the case no direct reaction is needed.

If the defined limit for CPU usage is exceeded, there are two possible measures:

- Extend the number of CPU's available for the runnable.
- Scale up by increasing the resources of the current node, for example, assigning more CPU cores to the virtual machine or scale out by adding a new node/runnable of the same type.

Thread pool size

For most runnables the **ajpNumWorkerThreads** parameter and the **httpNumWorkerThreads** parameter are set. Those values should be monitored.

If they regularly exceed 80% of the configuration parameter of the runnable, it has to be scaled out by adding a new runnable of the same type.

7.1. Using ACC commands

You can monitor the health of a specific runnable as well as the health of the whole system.

7.1.1. Monitor runnable's health

The **health** command determines a variety of health values for each runnable.

The exact list of health values available depends on the specific runnable. However, a few basic measurements are available to all runnables, in particular the CPU load and its memory usage. For all Java-based runnables, additional information is available.

Prerequisite

Health monitoring parameters are set. [44]

Procedure

1. Start ARIS PPM Cloud Controller (ACC). [20]
2. Make sure that the runnable has been started.
3. Enter, for example:
   show instance zoo_m health

The output will look like this example:

```
Status: OK
Specified IP or host name: arissrv
Resolved IP or host name: arissrv/127.0.0.1
Agent port: 9001
Monitored Resource          Current Value  Value Range    Low Thrh.  High Thrh.  State  Last msrd. at  Msrd. evry
CPU Load                    11,36%         0,00%-100,00%  0,00%      80,00%      OK     08:05:39.844   1s 000ms
Memory Usage (Percent)      36,81%         0,00%-100,00%  0,00%      80,00%      OK     08:05:39.844   1s 000ms
Memory Usage (Megabytes)    2980MB         0MB-8097MB     0MB        0MB         OK     08:05:39.844   1s 000ms
Free Disk Space (Percent)   63,40%         0,00%-100,00%  3,00%      0,00%       OK     08:05:39.845   1s 000ms
Free Disk Space (Megabytes) 302361MB       0MB-476938MB   1000MB     0MB         OK     08:05:39.846   1s 000ms
Known used ports:
   Port  Runnable  Port Parameter
   2181  zoo_m     clientPort
   9001  (AGENT)   rest.port       DEFAULT
ACC+ arissrv>
```

The first column shows the name of the measured value and its physical unit, for example, megabytes, percent, etc.

The second column shows the current value of the health value. Note that this is not measured the moment the command is executed. The agent plugin will take measurements periodically, and this column will show the result of the last measurement.

The third column shows the range of possible values, for example, minimum and maximum value. If no minimum or maximum value can be determined or the value does not have an ordinal value scale a **-** is shown instead.

The fourth and fifth columns show the minimum and maximum values, respectively. These values were observed since the runnable or the ARIS Cloud Agent had last been restarted, but only if the health value measured has an ordinal value scale. For values measured without an ordinal value scale, - will be shown.

The sixth column shows the low or high warning thresholds for the measured value, if the measured health value has an ordinal value scale and if thresholds have been configured.

The seventh column shows the state of the measured value if the measured health value has an ordinal value scale and if a low and/or high threshold has been configured. The state can be either **OK**, **LOW** or **HIGH**. If in **LOW** or **HIGH** state, the column will also show for how long the value has been in that state already.

7.1.2. Show node

The **show node** command shows machine monitoring information concerning CPU, memory and disk usage. If a threshold is violated, it will be shown.

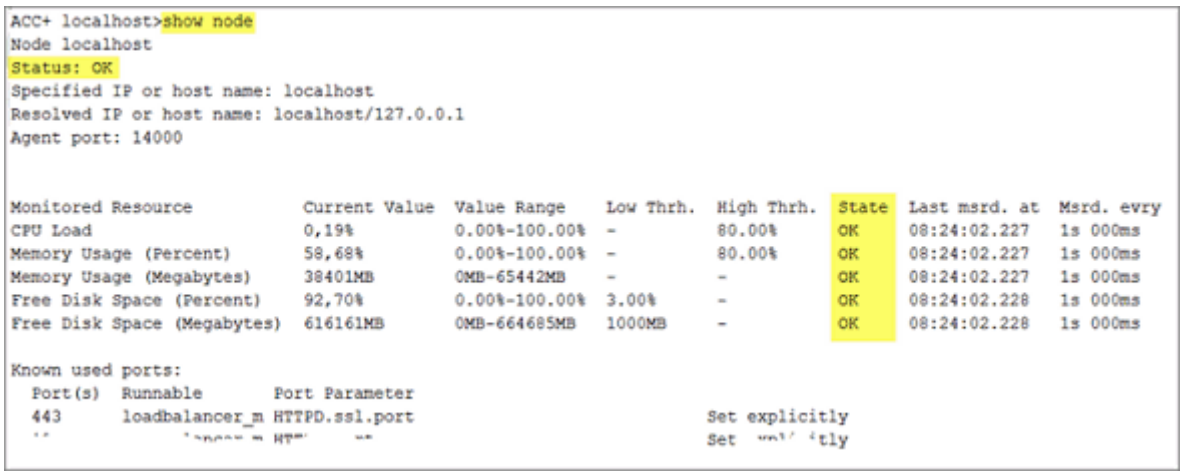
Prerequisite

Health monitoring parameters are set. [44]

Procedure

1. Start ARIS PPM Cloud Controller (ACC). [20]
2. Make sure that the runnables have been started.
3. Enter **show node**

The output will be in tabular format and look like this example:



```
ACC+ localhost>show node
Node localhost
Status: OK
Specified IP or host name: localhost
Resolved IP or host name: localhost/127.0.0.1
Agent port: 14000


Monitored Resource           Current Value  Value Range     Low Thrh.  High Thrh.  State  Last msrd. at  Msrd. evry
CPU Load                     0,19%          0.00%-100.00%   -          80.00%      OK     08:24:02.227   1s 000ms
Memory Usage (Percent)       58,68%         0.00%-100.00%   -          80.00%      OK     08:24:02.227   1s 000ms
Memory Usage (Megabytes)     38401MB        0MB-65442MB     -          -           OK     08:24:02.227   1s 000ms
Free Disk Space (Percent)    92,70%         0.00%-100.00%   3.00%      -           OK     08:24:02.228   1s 000ms
Free Disk Space (Megabytes)  616161MB       0MB-664685MB    1000MB     -           OK     08:24:02.228   1s 000ms

Known used ports:
  Port(s)  Runnable       Port Parameter
  443      loadbalancer_m HTTPD.ssl.port                    Set explicitly
```

7.1.3. Configure health monitoring

If you use the system health monitoring, using ARIS PPM Cloud Controller, make sure to configure the system.

Procedure

1. Start ARIS PPM Cloud Controller (ACC). [20]
2. Enter the parameters described below.

Thresholds

Thresholds for monitored values can be configured using agent configuration parameters:

Parameter	Description
monitoring.high.cpu.threshold.percent (0.0 – 1.0, default 0.8)	High CPU usage threshold (relative, in percent)
monitoring.relative.low.disk.space.warning.threshold.percent (0.0 – 1.0, default 0.03)	Low memory threshold (relative, in percent)
monitoring.absolute.low.disk.space.warning.threshold.megabytes (default 1000)	Low disk space threshold (absolute, in megabytes) **Warning** The ARIS agent not only warns against insufficient disk space. When the default value (1000 MB) is reached, all runnables are immediately shut down (killall) to avoid data inconsistencies. Runnables cannot be restarted. The ARIS agent autostart is disabled.

Threshold measurement

The measurement can be configured by the following parameters:

Parameter	Description
monitoring.interval (default 1000)	Interval in milliseconds, in which the measurement will be done.
monitoring.high.cpu.number.of.intervals.warning.threshold (default 10)	Number of measurements with CPU load above threshold before a warning state is reached
monitoring.high.mem.usage.number.of.intervals.warning.threshold (default 10)	Number of measurements with memory usage above threshold before a warning state is reached.
monitoring.low.disk.space.number.of.intervals.warning.threshold (default 10)	Number of measurements with free disk space below threshold before a warning state is reached.

Mail notification

Mail notification can be configured by the following parameters:

Parameter	Description
mail.notification.enabled (default: **false**)	**true** enables automatic mailing.
mail.smtp.host	Qualified hostname or IP of the mail server.
mail.smtp.port (common: **25**)	Port of the mail server.
mail.smtp.username	User name on the mail server. If this parameter is set, the applications will implicitly enable SMTP authentication. There is **NO** dedicated parameter to explicitly enable SMTP authentication.
mail.smtp.password	Mail server password for the specified user.
mail.sender.address	Sender's e-mail address.
mail.sender.name	Sender's name.
mail.recipients	Recipient's e-mail address. This parameter is multivalued, specify several recipients separated with commas, for example: **set mail.recipients = "a@abc.com","b@abc.com"**
mail.recipients.cc	Recipient's e-mail address (**C**arbon **C**opy(CC)). This parameter is multivalued (see **mail.recipients**)

Parameter	Description
mail.on.system.events	Defining events triggering e-mail notification. This parameter is multivalued, specify several event types separated by comma, for example: **set mail.on.system.events = MACHINE_STATE_CPU_LOAD_HIGH, MACHINE_STATE_** Possible values: ▪ MACHINE_STATE_LOW_DISK_SPACE_PERCENT ▪ MACHINE_STATE_OK_DISK_SPACE_PERCENT ▪ MACHINE_STATE_LOW_DISK_SPACE ▪ MACHINE_STATE_OK_DISK_SPACE ▪ MACHINE_STATE_CPU_LOAD_HIGH ▪ MACHINE_STATE_CPU_LOAD_OK ▪ MACHINE_STATE_MEMORY_USAGE_HIGH ▪ MACHINE_STATE_MEMORY_USAGE_OK

7.1.3.1. Monitor the memory usage

The memory monitoring usage, which is displayed in megabytes and percent, is disabled by default. You can enable the monitoring of the memory usage using the ARIS PPM Cloud Controller.

Procedure

1. Start the ARIS PPM Cloud Controller (ACC). [20]
2. Run the following command. Possible values are ON, OFF, AUTO.
   set monitoring.monitor.runnable.memory.usage=ON

7.1.3.2. Stop the system monitoring

You can stop the system monitoring using the ARIS PPM Cloud Controller.

Procedure

1. Start the ARIS PPM Cloud Controller (ACC).
2. To stop the memory usage monitoring, run the following command.
   set monitoring.monitor.runnable.memory.usage=off
3. To stop the monitoring of the network connections, enter the following command.
   set monitoring.interval.network.cxns=-1

4. To stop monitoring of a runnable, run the following command (everything in one line and separated by a space).

   reconfigure <runnable instance ID>

   +plugin.health.monitoring.value.memUsageBytes.enabled=false

   +plugin.health.monitoring.value.memUsageMegabytes.enabled=false

   +plugin.health.monitoring.value.memUsagePercent.enabled=false

   -JAVA-Dcom.sun.management.jmxremote

   -JAVA-Dcom.sun.management.jmxremote.authenticate

   -JAVA-Dcom.sun.management.jmxremote.port

   -JAVA-Dcom.sun.management.jmxremote.rmi.port

   -JAVA-Dcom.sun.management.jmxremote.ssl

   -JAVA-Dcom.sun.management.jmxremote.local.only

8. Globalization

Localization

Application

PPM is available for the following languages:

- German
- English
- French
- Russian
- Japanese
- Chinese

The PPM user interface and log output are available in the language that you specify at login. Irrespective of the interface language, any data imported is displayed in the language of the source system from which it was extracted.

Documentation

The online help supplied with PPM and the supplementing documentation in PDF format are available in **German**, **English**, and **Japanese**.

Calendar

Currently, PPM supports the Gregorian calendar only.

Bidirectional text display

The bidirectional display of RTL-based interface texts is supported only with limitations, that is, interface texts (for example, favorites, measures, dimension names) are not always represented uniformly. The layout of the controls is always uniform irrespective of the interface language that you select.

Standard date format

The **English USA** date formate is used by PPM as default. To switch to the English GB date format, see the documentation **PPM Installation**, chapter **Adapt date format to British English language area**.

9. Job automation

Job automation enables you to automatically run selected programs at specific points in time. In a job automation, you specify the programs and the execution rules based on which the programs are to be executed regularly.

You can define job automations for each client. The programs of the respective client are available.

After the job automation is completed, a message is automatically created and sent to specific recipients.

In PPM Customizing Toolkit, a wizard supports you in creating custom job automations.

For detailed information on this topic, please refer to the PPM Customizing Toolkit manual and the integrated help of the respective wizard or dialog.

10. Accessibility

This version of PPM provides only limited support for the accessibility that is recommended by **Section 508** (http://www.section508.gov) and **WCAG** (Web Content Accessibility Guidelines, http://www.w3.org/TR/WAI-WEBCONTENT).

11. Single Sign-On integration

You can integrate PPM in an Single Sign-On (SSO) scenario with other applications. This means after login to one application, the user is automatically authorized to use other integrated applications.

For this, PPM uses the SAML and Kerberos settings specified in the central User Management.

You can use the SAML settings to integrate PPM and ARIS Aware in a Single Sign-On scenario [51].

You can use Kerberos settings to
integrate PPM in a Single Sign-On scenario with any application that supports Kerberos [53] as authentication service.

11.1. Integrate PPM and ARIS Aware in an SSO scenario

You can integrate PPM and ARIS Aware installed on different servers in a Single Sign-On scenario.

From version 9.0, PPM supports Single Sign-On using SAML 2. The relevant truststore and keystore file mentioned above must be shared between the applications. We suggest using your own truststore and keystore file in PPM and ARIS Aware. For details on how to create the files and what is the difference between them, see the chapter **Use of truststore and keystore in SSO scenario [54]**.

To integrate PPM in a Single Sign-On scenario using SAML, you must configure the required parameters using central User Management (UMC). For detailed information on meaning and configuration of the parameters, please refer to the Central User Management online help.

Central User Management parameters for SAML

For an SAML connection, the HTTP query when calling PPM must be added a so-called SAML signed assertion that PPM verifies at an authentication service. The SAML signed assertion is usually provided by the application calling. Among other data, the assertion contains the data of the user who wants to log in to PPM. If the assertion is valid and the user is known to PPM and active, login is successful and the user is granted access to PPM.

Use the following entries in central User Management to configure SAML 2 access.

- com.aris.umc.saml.active = true
- com.aris.umc.saml.assertion.timeoffset = 60 // maximum allowed time difference in seconds between issuer and verifier of the signed assertion.
- com.aris.umc.saml.assertion.ttl = 120 // maximum validity of the signed assertion
- com.aris.umc.saml.keystore.alias = Alias of the keystore user
- com.aris.umc.saml.keystore.location = Keystore location (automatically entered when the keystore is uploaded into UMC)
- com.aris.umc.saml.keystore.password = Keystore user password
- com.aris.umc.saml.keystore.type = Keystore encryption type, for example, 'JKS'
- com.aris.umc.saml.truststore.alias = Truststore user alias
- com.aris.umc.saml.truststore.location = Truststore location (automatically entered when the truststore is uploaded into UMC)

- com.aris.umc.saml.truststore.password = Truststore user password
- com.aris.umc.saml.truststore.type = Truststore encryption type, for example, 'JKS'

The truststore is used for verifying the signature of a signed assertion received in a third-party system while the keystore is used for signing own signed assertions. This means that each certificate in the keystore of the issuer must exist in the truststore of the receiver so that the certificate can be verified. Ideally, the certificates are identical on both sides (truststore/keystore and issuer/receiver).

Procedure

1. Configure the User Management of PPM for SAML.
   a. Open the **Configuration** page in the User Management of PPM.
   b. On the **General** page.
      i. Select **SAML** in the drop-down menu.
      ii. Activate the **Use SAML** option.
      iii. Clear the **Identity provider ID**.
   c. On the **Signature** page.
      i. Activate the **Sign assertions**, **Sign requests**, and **Sign responses** options.
      ii. Select **RSAwithSHA512** in the **Signature algorithm** drop-down menu.
   d. On **Keystore** page.
      i. Select the **keystore** required.
      ii. Enter **ssos** in the **Alias** input box.
      iii. Enter **manage** in the **Password** input box.
      iv. Select **JKS** in the **Type** drop-down menu.
   e. On the **Truststore** page.
      i. Select the **truststore** required.
      ii. Enter **ssos** in the **Alias** input box.
      iii. Enter **manage** in the **Password** input box.
      iv. Select **JKS** in the **Type** drop-down menu.
   f. On **the Advanced settings** page.
      i. Enter **uid** in the **Keyword** input box.
      ii. Enter **99** in the **Clock skew** input box.
      iii. Enter **99** in the **Assertion lifetime** input box.
      iv. Enter **default** in the **Default** tenant input box.
   g. Set the PPM user on the **User management** page.
      i. Enter a user name (for example, ppmuser) in the **User name** input box.
      ii. Enter a name in the **First name** (for example, ppm) and **Last name** (for example, user) input box.

2. Configure ARIS Aware for SAML.

   Open the ACC Command Interface and run the following reconfigure commands, assuming that dashboarding_m is the instance ID of your dashboarding runnable:

   reconfigure dashboarding_m +JAVA-Dsaml.keystore.file.default="<ARIS Installation>\server\bin\work\work_dashboarding_m\base\conf\keystores\keystore.default.jks"

   reconfigure dashboarding_m +JAVA-Dsaml.keystore.passwd.default=manage

   reconfigure dashboarding_m +JAVA-Dsaml.keystore.alias.default=ssos

   Note that the location of the keystore file are just examples and can be set individually. But the files must be accessible by the application.

A PPM user is created in the central User Management of PPM and ARIS Aware. PPM and ARIS Aware are integrated to be used in an Single Sign-On scenario.

11.2. Integrate PPM in an SSO scenario using Kerberos

Kerberos is an authentication protocol that enables mutual authentication of applications in a network by means of key cryptography. PPM can apply the user's login data for the operating system (for example, MS Windows) to automatically log the user in to PPM in the Web browser.

To integrate PPM in an Single Sign-On scenario using Kerberos, you must configure the required parameters using central User Management. You configure the parameters on the **Configuration** page in central user administration. For detailed information on meaning and configuration of the parameters, please refer to the central user management online help.

Kerberos parameters

Use the following entries in central User Management to configure Kerberos access.

- com.aris.umc.kerberos.active
- com.aris.umc.kerberos.config
- com.aris.umc.kerberos.debug
- com.aris.umc.kerberos.kdc
- com.aris.umc.kerberos.keyTab
- com.aris.umc.kerberos.realm
- com.aris.umc.kerberos.servicePrincipalName

By default, PPM uses the MS Windows® native SSPI-API to perform a Kerberos authentication. In case that the usage of the SSPI-API causes incompatibilities you can switch the Kerberos authentication to the Java®-internal GSS-API.

To use the Java®-internal GSS-API you can edit the **Kerberos_settings.properties** file and set the **DISABLE_NATIVE_PROVIDERS=** parameter to **true**.

The **Kerberos_settings.properties** file is located in the **<PPM installation>\server\bin\work\data_ppm\config** folder.

11.3. Use of truststore and keystore in SSO scenario

In most cases, a keystore and a truststore are used when an application needs to communicate over SSL/TLS. Usually, these are password-protected files that reside on the same file system as the running application. The default format for these files was JKS until Java 8. Since Java 9, the default keystore format has been PKCS12. The biggest difference between JKS and PKCS12 is that JKS is a Java-specific format, while PKCS12 is a standardized and language-neutral method for storing encrypted private keys and certificates.

A Java keystore stores private key entries, certificates with public keys, or just secret keys that may be used for various cryptographic purposes. The keystore stores each by an alias for ease of lookup.

A truststore is the opposite of this. While a keystore typically stores certificates that identify your own server, a truststore stores certificates that identify others. A truststore is therefore used to verify the signature of a signed assertion received in a third-party system, while the keystore is used to sign one's own signed assertions. This means that each certificate in the issuer's keystore must exist in the recipient's truststore for the certificate to be verified. Ideally, the certificates on both sides (truststore/keystore and issuer/recipient) are identical.

Keystores can be created using the Java keytool (see chapter **Generate and sign TLS certificates [57]** for details) holding private and public keypairs for SSL communication and trusted certificates. Truststores can be created in the same way, but then the public/private key pairs are deleted again. Leftovers are just certificates. An example for such a truststore is the **cacerts** file and can be found here:

"$JAVA_HOME/jre/lib/security"

Note

Only the JKS format is currently supported by PPM and ARIS Aware.

12. Security aspects

This chapter describes all security-relevant aspects concerning the installation and operation of a PPM system with all required components.

12.1. Safe communication channels

The PPM client communicates with the PPM server using a load balancer via a TCP/IP network. The individual steps to secure communication with the load balancer and the PPM server are described in more detail below.

By default, PPM uses HTTP as communication channel. For security reasons, we strongly recommend that you switch off HTTP and configure HTTPS as the communication channel instead. For details on configuring HTTPS, see Load balancer configuration [22].

12.2. Load balancer

By default, the load balancer enables unencrypted access via HTTP (port 4080).

Additional configuration for activating the TLS (SSL) encryption (HTTPS) is required. You need to create a TLS (SSL) certificate and configure the load balancer to use HTTPS on a specific port.

For detailed information see chapter Load balancer configuration [22].

It's recommended to use a certificate that contains a valid root certificate of an approved certification authority. The following chapter describes how to create a valid certificate for HTTPS support of the load balancer.

12.2.1. Integrate TLS certificate

This section explains how to create a valid certificate for HTTPS support of the load balancer. The TLS (SSL) certificate to be generated must be adapted to the load balancer host name in order to avoid client warnings, especially Web browser warnings, and to ensure proper functioning.

To use TLS you need a valid certificate by a certification authority for the server hosting the load balancer. Ensure that the certificate is compatible with the Java version used by the client.

- You can purchase a certificate from an official certification authority. Most clients, and particularly Web browsers accept such a certificate.
- If your company is using a specific certification authority integrate it in the trust store of each client.

The certificate consists of two parts: the private key (file extension **.key**) to unencrypt the information returned to the client, and the server certificate (file extension **.crt**). An example on generating these files is described in chapter **TLS between PPM client downloader and PPM server [57]** .

Procedure

1. Add both files to a ZIP file. Make sure that the files are named **server.key** and **server.crt** before you add them to the ZIP file.
2. Copy the ZIP file to a location where ARIS PPM Cloud Controller can access it.
3. Start ARIS PPM Cloud Controller.
4. Stop the load balancer in ARIS PPM Cloud Controller.
5. In ARIS PPM Cloud Controller, enter the command **enhance <instanceID of the load balancer component> with sslCertificate local file "<path to ZIP file>".**

   If the load balancer instance ID is **loadbalancer_m** and your ZIP file is located at **c:\temp\lbcert.zip** , enter the command **enhance loadbalancer_m with sslCertificate local file "c:\\temp\\lbcert.zip"** .

   Please note the use of double backslashes or single slashes, for example: **"c:/temp/lbcert.zip"**.
6. Restart the load balancer.

The TLS certificate is now available.

12.2.2. Further security measures for load balancer

Central systems, such as load balancer, that are responsible for communication between client and server are often the target of attacks from the Internet. To close current and known security gaps, several measures are conceivable. They are briefly described and discussed here.

- Always keep your operating system updated. Run necessary security updates on a regular basis.
- Always keep your product installation updated. Regularly check if new fixes are available for your installation and install them.
- Do not use a self-signed certificate for load balancer, instead let your keystore sign it by an authorized institution.
- To close the so-called "Logjam" security gap you need to generate a so-called "Diffie-Hellman" group for load balancer. To do so, proceed as follows:
  - Switch to the directory

    <installation directory>\server\bin\agentLocalRepo\.unpacked\<installation_time>_httpd-run-prod-<version>-runnable.zip\httpd\bin. Replace <installation directory> with the path to your PPM installation and <version> with the version installed on your machine.
  - Open a command line and enter the following command:

    openssl dhparam -out dhparams.pem 2048

    This generates a user-defined DH group in a file called **dhparams.pem** in the same directory.
  - Now you need to attach the contents of this file to the TLS certificate of the server in the following directory.

    <installation directory>\server\bin\work\work_<loadbalancerInstanceId>\httpd\conf
  - Restart load balancer to apply the changes.
- To protect your system from unauthorized access, you should always grant direct system access (for example: remote access via RDP or directly via a management console) only to a limited group of users.
- You should also enable access only to the most necessary services on the server (such as database, PPM system, load balancer) in the firewall. Make sure that not needed ports are closed.

12.3. TLS between PPM client downloader and PPM server

The PPM client downloader uses HTTPS for its communication with the PPM server respectively with the Loadbalancer.

We recommend setting up a TLS scenario using an official certificate, as described in chapter **Integrate TLS certificate [55]** .

12.4. Generate and sign TLS certificates

Typically, SSL certificates are used on web pages that transmit and receive end-user sensitive data. They ensure the identity of a remote computer, most commonly a server, but also confirms your computer's identity to the remote computer to establish a safe connection.

SSL certificates are verified and issued by a **C**ertificate **A**uthority (**CA**). The customer applies by generating a **C**ertificate **S**igning **R**equest (**CSR**) with a key pair on your server that hold the SSL certificate. The CSR contains crucial organization details which the CA verifies.

Procedure

1. Generate a CSR and key pair locally on your server. The key pair consists of a public and private key.
2. Send the CSR and public key to a CA who will verify your legal identity and whether you own and control the domain submitted in the application. The Certificate Authority runs a check on your organization and validates if the organization is registered at the location provided in the CSR and whether the domain exists.
3. When verified, the organization receives a copy of their SSL certificate including business details as well as the public key. The organization can now install the certificate on their server.
4. When a CA issues the certificate, it binds to a certificate authority's "trusted root" certificate. Root certificates are embedded into each browser and connected to individually issued certificates to establish an HTTPS connection.

There are various tools available on the net to create the certificate and related key pair. For PPM, we just give two examples using the **Java keytool** or **OpenSSL**. Detailed information on the tools can be found here:

- **Keytool - Key and Certificate Management Tool**
  (http://docs.oracle.com/javase/7/docs/technotes/tools/windows/keytool.html)
- **OpenSSL**
  (/docs/man3.1/man1/index.html)

> **Tip**
> The following describes the generation of the keystore/keyfiles and the Certificate Signing Requests (CSR) using simple examples and demo data. The values of the individual parameters used in the examples need to be adapted to their local environment. The meaning of the individual parameters for the different tools is described in the corresponding documentation mentioned above.

Procedure using OpenSSL

1. Create CSR (including public key) and private key

   openssl req \

   -new -newkey rsa:2048 -nodes -sha256 \

   -subj "/CN=Servername/O=Your Company AG/C=US/ST=Musterstadt/L=Musterland" \

   -days 1460 -keyout privatekey.key -out server.csr
2. Check the content of your CSR

   openssl req -in server.csr -noout -text -verify
3. CSR file can be sent to CA
4. CA provides new certificate, for example server.cer. This file and the private key file can be used to reconfigure the Loadbalancer to support HTTPS

Procedure using Java keytool

1. Use the keytool to create a keystore with the following parameters.

   keytool

   -keystore ppm_keystore -alias ppm -storepass changeit -genkey -v -keyalg RSA \

   -dname "CN=PPM Admin, OU=PPM, O=SAG, L= Saarbruecken, S=SAL, C=DE"
2. Use the keytool to create the CSR with the following parameters.

   keytool -certreq -alias ppm -file ppm_keystore.csr -keystore ppm_keystore
3. Submit the content of the file with the **.csr** extension to a CA. It will sign the certificate with a valid root certificate that is usually acknowledged by all browsers and the default Java Runtime Environments and sends back the **CSR** file.
4. The private key, which is needed for the Loadbalancer configuration can only be extracted from the above mentioned keystore by using the tool **OpenSSL**. The Java keytool does not provide this functionality. Assuming that you created a standard keystore in PKCS12 format, use the following command to extract the private key:

   openssl pkcs12 -in ppm_keystore -nodes -nocerts -out private.key
5. With step 3 and 4, you have all files that you need to reconfigure the Loadbalancer to support HTTPS.

13. Session timeouts

By default, session timeout limits are set in the User Management, that is, an initial session duration and a maximum session duration. If the user does not request data for a certain period, the initial session duration expires. This initial session duration can be extended until a maximum session duration is reached.

The initial session duration and maximum session duration can be configured in the User Management administration. PPM uses these values to refresh the user sessions automatically, until the maximum session duration is reached.

This extension of the session duration takes effect after a certain time, that can be configured using the **UMC_SESSION_REFRESH** parameter in the server configuration file.

If the initial session duration or the maximum session duration for the **system** user has been changed, the PPM server must be restarted. Otherwise, it may happen, that the user session expires before the maximum session duration has been reached.

14. Command line programs

This chapter provides you with an overview of the key command line programs that support you in managing application data in the PPM system.

14.1. General structure and function-specific structure

The command line programs (*.bat) for managing and configuring the PPM system located in the **bin** directory of the PPM installation have a four-level structure:

1. Determining and checking PPM environment variables

2. Composing an parameter character string of all transferred command line parameters

3. Creating a local class path

4. Running the Java Virtual Machine with specification of the class path and the parameter character string, and calling a command line program-specific class in the VM

PPM-specific batch files

- Windows: Windows batch files (*.bat) are stored in the directory
  <installation dirctory>\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-<version>-runnable.zip\ppm\bin.
- Unix: The corresponding directory contains the shortcuts to the **.runppmcommand.sh** shell script running the required function.

CTK-specific batch files

- Windows: Windows batch files (*.bat) are stored in the directory
  <installation dirctory>\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-<version>-runnable.zip\ppm\ctk\bin.
- Unix: The corresponding directory contains the shortcuts to the **.runctkcommand.sh** shell script running the required function.

Client-specific batch files

- Windows: Windows batch files (*.bat) are stored in the directory**<installation directory>\ppm\server\bin\work\data_ppm\ custom\<client>\bat**.
- Unix: The shell scripts are stored in the directory**<installation directory>\ppm\server\bin\work\data_ppm\ custom\<client>\bin**.

The batch files (both Windows and Unix) are also referenced in PPM CTK in the **Client** -> **Programs** module.

The batch files execute complex functions (ACC start, infrastructure start/stop, client start/stop, CTK, PPM GUI, etc.) in the following directories.

- **<installation directory>\ppm\server\bin\work\data_ppm\bin**
- **<installation directory>\ppm\server**

Under Windows, shortcuts for these batch files are created in the Start menu.

14.2. PPM command line programs

The following command line programs are available:

- runinitdb [63]
  Initializes the database schema of a PPM client
- runppmanalytics [64]
  Checks process instances for anomalies during Easy mining.
- runppmconfig [66]
  Imports configurations into the database schema of a PPM client
- runppmcpi [73]
  Automatically deletes multiple messages simultaneously during Easy mining.
- runppmimport [75]
  Controls the subsequent processing of the process instance fragments (merging, typification, measure calculation) imported using **runxmlimport** .
- runppmcompress [79]
  Aggregates imported process instances
- runppmdelete [82]
  Deletes process instances either for testing or for real
- runppmexport [85]
  Create exports by means of batch scrips
- runppmadmin [88]
  Executes various administrative functions
- runppmsendmail [91]
  Sends simple text messages
- runtableimport [92]
  Extracts data from a file in **XML** or **ZIP** (XML Event Format) format

All command line programs are executed in
<installation directory>\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-<version>-runnable.zip\ppm\bin.

For generally applicable specifications (e. g. PPM user account and client) the same parameters are used in all command line programs. Special parameters are used for controlling component-specific functions.

Calling up one of the programs described below with no parameters or with **-h** or **-?** outputs the online help on the console, providing a brief description of all available options:

14.3. Common parameters

-version

The version of the PPM software and the database schema are output on the console. Other parameters are ignored.

-user <user name> -password <password>

Enter the user name and the password of the PPM user. The specified user must have appropriate authorization for the action.

Function privilege	Command line program	Description
Data import	runxmlimport, and other source system adapters	Import process instance fragments
Data calculation	runppmimport	Calculation of process instances
Configuration import	runppmconfig runinitdb	Import XML configuration files
Process instance aggregation	runppmcompress	Aggregate process instances

The function privileges are managed in the **Users** administration component of the front-end. The system user (**system**) has all function privileges.

If you want the password to be hidden, enter only the parameter **-user** with the relevant user name in the command line. The password is then queried interactively.

-client <name>

This is where you enter the PPM client whose database schema is used by the relevant command line program. If you do not enter anything here, the **default** client is used.

-language <ISO code>

Use this option to specify the log output language. Valid values are **en** for English, **de** for German, and **fr** for French.

-protocolfile <file name>

This option writes the log output to the specified file. Only critical error messages resulting in program abortion will be output in the console.

-information {yes|no|default}, -warning {yes|no|default}, -error {yes|no|default}

These parameters control the output of messages. The command line options are described in detail in the chapter on Command line programs [36].

14.4. runinitdb

The **runinitdb** command line program initializes the database schema for the specified PPM client. For this to work, the **-init** option must also be specified. The process runs in two stages.

The database schema is recreated and populated with default values for each language referenced in the file **Locales.xml**. Any file name can be selected for the file **Locales.xml** and is specified in the file **InitDB_settings.properties**.

To initialize an existing database schema, you must specify the **-user system -password <password>** parameters. After successful initialization, the password for the **system** user is reset to the default value **manager**.

The PPM client server is started and the configuration files specified in the file **InitSystem_settings.properties** are imported.

The file **InitSystem_settings.properties** consists of various sections. Each section is made up of three lines. Each line begins with a keyword, which is assigned a value using an equals sign:

Keyword	Description
INIT_MODULE_X	Section introduction
INIT_MODULE_X_NAME	Name of the configuration component. Corresponds to the **command** parameter in the **runppmconfig** program.
INIT_MODULE_X_FILE	File name with path for the XML configuration file

X is a placeholder for the number of a section in the file. All three lines in a section must be specified. Section numbering must start from **1** and be continuous. However, the sections can appear in any order in the file.

> **Warning**
> When editing the file **InitSystem_settings.properties**, you must adhere to the sequence of the configuration components: Data types, attributes, process types, measures, process tree, others.

Example 18.

Extract from the default file **InitSystem_Settings.properties**:

```
#
# Initialization of all runppmconfig components according
# to the specified sequence
#
# INIT_MODULE_XXX_NAME : Name of the module according to
# the command line option of runppmconfig
```

```
# INIT_MODULE_XXX_FILE : Name of the XML file containing the corresponding
config data
#

INIT_MODULE_1 =
INIT_MODULE_1_NAME = datatypes
INIT_MODULE_1_FILE = D:/ppm/xml/DefaultDataTypes.xml

INIT_MODULE_2 =
INIT_MODULE_2_NAME = attributetypes
INIT_MODULE_2_FILE = D:/ppm/xml/DefaultAttributeTypes.xml

INIT_MODULE_3 =
INIT_MODULE_3_NAME = attributenames
INIT_MODULE_3_FILE = D:/ppm/xml/DefaultAttributeNames.xml
...
```

In order to import the configuration of organizational units during initialization, edit the file **InitSystem_settings.properties** in the configuration directory for the relevant client and insert the following lines at the end of the file:

```
...
INIT_MODULE_X =
INIT_MODULE_X_NAME = orgunits
INIT_MODULE_X_FILE = <Organizational units.xml>
```

For **X** you use a number **1** higher than the previous section.

In practice, it has proven useful to store the configuration files on a client-specific basis in directories of the form **<installation directory>\ppm\server\bin\work\data_ppm\custom\<client>\xml**.

14.5. runppmanalytics

Using **Easy mining**, you can check process instances for anomalies. This includes various analytical functions, such as Early alert system, alarm value deviation, Process mining, outlier analysis, and planned value infringements.

In addition to pooling various analyses in one PPM tool, messages are created for all occurring anomalies and then provided to PPM users.

Using the **runppmanalytics** command line program, easy mining analyses can be run individually or subsequently.

If you perform an analysis across all analysis types, a fixed order applies.

1. Early alert analysis
2. Planned value analysis
3. Alarm value analysis
4. Outlier analysis
5. Process mining analysis

The program is stored under
<installation directory>\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-<version>-runnable.zip\ppm\bin.

To run **runppmanalytics**, you need the **PPM login** function privilege.

> **Tip**
> Further information on **Easy mining** is available in the online help under **PPM help topics**.

14.5.1. runppmanalytics parameters

You can run runppmanalytics with the following specific parameters.

Optionally, you can specify the path to a favorites directory (**<favorites path>**) for each parameter.

Example: **runppmanalytics -earlyalert <favorites path>**

If you specify a favorites directory, the analysis is run for all favorites this directory contains.

Without path, default analyses are performed.

-earlyalert

An **early alert analysis** is run.

-plannedvalue

A **planned value deviation analysis** is run.

-alarmvalue

An **alarm value deviation analysis** is run.

-processmining

A **suspected deviation analysis** is run.

-outlier

An **outlier analysis** is run.

-all

All analyses are run in a predefined order (see runppmanalytics [64]).

Calling up the program with the **-h** parameter outputs the online help on the console, providing a brief description of all available options.

-registry <host:port>

By specifying this parameter, you can access a remote PPM system whose RMI registry can be reached via the specified computer name and port number.

-usessl <true|false|default>

If you specified the **-registry** parameter, you can use the **-usessl** parameter to specify whether the remote system is to use TLS-encrypted data communication or not.

Parameter	Description
true	The remote PPM system uses encrypted data transfer. Ensure that the local system and remote system are using the same keystore.
false	The remote system uses unencrypted data transfer.
default (default value)	The settings of the local PPM system are used.

14.6. runppmconfig

The **runppmconfig** command line program is used to import XML configuration files into the PPM system and save the configuration of a PPM client in XML files.

14.6.1. runppmconfig parameters

-mode <mode>

This is where you specify whether the configuration of the PPM component specified by the

<command> parameter is saved in the specified file or whether the configuration saved in the specified file is imported into the PPM system. The valid values are **export** and **import**.

-<command> <file name>

Specify the PPM component whose configuration is to be saved to or imported from the specified file.

command	PPM component
abcanalysis	Configuration of ABC analysis
attributenames	Names of the PPM attributes. To export, use the **-language <ISO code>** parameter to specify the relevant language for the PPM attribute names. If this parameter is missing, the attribute names are exported in the default language of the PPM client (**defaultlanguage** XML element in the file **Locales.xml**). For importing, the language code specified in the XML file is used.
attributetypes	Definition of the attribute types known in the PPM system
copyattributerules	Rules for copying the object attributes to the process instance
cpi	CPI module messages
datatypes	Extended data type definition
dimgroup	Dimension groups
epctemplate	Templates for EPC representation
factorycalendar	Factory calendar
favorites [-favoriteuser <user>]	Import or export of favorites. You can restrict the operation to a specific user's favorites with the **favoriteuser** parameter. If you enter **-favoriteuser sharedfavorites**, only all shared favorites are imported or exported.
images	Assign wallpaper and other icons for EPC objects to EPC display
keyindicator	Process instance-dependent and independent measures, as well as dimensions, attribute calculations, and relations
keyrules	Rules for key creation
kidepend	User-defined dimensions
kigroup	Measure groups
locales	PPM user interface languages
merger	Merge configuration
mysaptransactions	SAP transactions
onlinekis	User-defined measures
orgunits	Organizational units
processtree [-add]	Definition of the process tree If you specify the **-add** option, new elements from the specified file are added to the process tree. Already existing elements of the process tree are not replaced.
ranking	Configuration of ranking analysis
realm	Name of the analysis realm to be imported
reportautomations	Overall configuration of report automations
serverproperties	Export general server settings
system	System configuration. Is saved in XML files in the specified directory. (See chapter Export the system configuration [69].)
targetvalues	Planned value definitions
topflop	Top-flop dimensions
transformationfactors	Transformation factors
users	User privileges

You can also specify every possible parameter value in the file **InitSystem_settings.properties**, so that the configuration will be active immediately when the PPM database is initialized. Make sure to take account of the dependency of different configurations on one another.

-overwrite

This switch enforces the replacement of existing configurations when a configuration is imported. It is only applicable for the following configuration components:

-targetvalues, -abcanalysis, -datatypes, -factorycalendar, -keyindicator,

-ranking, -onlineki, -favorites, -reportautomations

> **Warning**
> Replacing existing configurations by specifying the
>
> **-overwrite** option can lead to unwanted effects, for example if you change the definition of data types already in use.

-replace

This switch deletes an existing configuration before the start of the import. It can only be used for the following configuration components: -targetvalues, -abcanalysis, -ranking -favorites, -reportautomations

Example 19.

The configuration of the user-defined measures of the PPM client **umg_en** is to be saved in the XML file **OnlineKI.xml** on **C** drive.

```
runppmconfig -client umg_en -user system -password manager -mode export
-onlinekis C:\OnlineKI.xml
```

> **Warning**
> Do not change the resolution (**precision**) of time dimensions (**timedim**) and time of day dimensions (**hourdim**) in an imported measure configuration later, for example, from **HOUR** to **SECOND**, as re-importing using **runppmconfig -overwrite** cannot create the required changes in the database tables.
>
> If you want to change the resolution of time and time of day dimensions for a configuration already imported, run a database initialization with the changed configuration.

-registry <host:port>

By specifying this parameter, you can access a remote PPM system whose RMI registry can be reached via the specified computer name and port number.

-usessl <true|false|default>

If you specified the **-registry** parameter, you can use the **-usessl** parameter to specify whether the remote system is to use TLS-encrypted data communication or not.

Parameter	Description
true	The remote PPM system uses encrypted data transfer. Ensure that the local system and remote system are using the same keystore.
false	The remote system uses unencrypted data transfer.
default (default value)	The settings of the local PPM system are used.

14.6.2. Export the system configuration

The **-system** parameter in the **runppmconfig** command line program enables almost the entire configuration (except report automation, locales and cpi configuration) of a client to be extracted and saved to XML files. This creates the file **InitSystem_settings.properties**, which contains the names of the configuration files with path information. To initialize a PPM database, copy the file **InitSystem_settings.properties** to the configuration directory for the relevant client and then execute the **runinitdb** command.

Example 20.

Execution of the command line

runppmconfig -user system -password manager -mode export -system <installation directory>\ppm\server\bin\work\data_ppm\config\default\xml

Example 21.

extracts the configuration of the **default** client and saves it under **<installation directory>\ppm\server\bin\work\data_ppm\config\default\xml** in these XML files:

```
default_abcanalysis.xml
default_attributenames_de.xml
default_attributenames_en.xml
default_attributenames_fr.xml
default_attributetypes.xml
default_copyattributerules.xml
default_datatypes.xml
default_dimgroup.xml
default_epctemplate.xml
default_factorycalendar.xml
default_favorites.xml
default_keyindicator.xml
default_keyrules.xml
default_kidepend.xml
default_kigroup.xml
default_merger.xml
default_mysaptransactions.xml
default_onlinekis.xml
default_orgunits.xml
```

```
default_processtree.xml
default_ranking.xml
default_realm_REALMNAME.xml (optional, a file for each existing analysis realm)
default_targetvalues.xml
default_topflop.xml
default_transformationfactors.xml
default_users.xml
InitSystem_settings.properties
```

14.6.3. Import and export of user data

You import and export PPM user data with the **runppmconfig** command line program.

The format for user data import and export is defined in the file **userconfig.dtd**.

You can create new users and user groups only in central User Management. In PPM, you can only manage specific PPM privileges, such as data access or process access privileges. Further information is available in the online help under PPM help topics.

Import user data

The program is called as follows.

runppmconfig –user system –password manager -client <name> –mode import –users <file name>

Users or user groups already existing in PPM are not created again during the import. The **-replace**, **-overwrite**, and **-add** options available for the command line program **runppmconfig** are not available for importing users. A corresponding error message is output.

Users and user groups that are imported into PPM but do not exist in central User Management are marked as to be deleted.

When running runppmconfig, only specific PPM user data is imported into PPM, for example, data and process access privileges, if they are not known in PPM yet.

General user data, such as group assignment, first name, last name, e-mail address, etc. are not saved in PPM. This data can be managed only in central User Management.

The list of user attributes comprises the following elements and default settings:

```
<!ATTLIST user
login CDATA #REQUIRED
password CDATA #IMPLIED
name CDATA #IMPLIED
lastname CDATA #IMPLIED
email CDATA #IMPLIED
isldapuser (yes | no) "no"
isumcadmin (yes | no) "no"
>
```

It is impossible to specify in the XML import file whether the data is to be imported only in PPM or only in central User Management.

Only users with the **Configuration import** function privilege and the **Privilege management** privilege can import user data.

Export user data

The program is called as follows.

runppmconfig –user system –password manager -client <name> –mode export –users <file name>

During an export, only specific PPM user data is exported, such as data and process access privileges. Data of active users and users to be deleted are exported.

Import and export of passwords

Passwords can be managed only in central User Management.

14.6.4. Import and export of favorites

Use the command line program **runppmconfig** to import private and shared favorites into PPM.

Import of favorites

The import of favorites via the command line supports the options **Replace** (parameter **-replace**) and **Overwrite** (parameter **-overwrite**). If no parameters are specified, the import is additive by default. Only new favorites and folders are created. Existing favorites and folders remain unchanged.

Prerequisite: You have the **Favorites administration** function privilege.

Import and export of shared favorites

If you specify the option **sharedfavorites** of the parameter **–favoriteuser**, only the shared favorites tree is imported or exported. If you do not specify the parameter **–favoriteuser**, the shared favorites tree is exported like a private favorites tree with the login name together with the private trees.

During the export of shared favorites, the access privileges for the shared favorites folders are exported, as well. Inherited or implicit access privileges are not exported.

You do not need special privileges for exporting favorites.

Import and export of template favorites

Importing and exporting template favorites is similar. However, you can import or export template favorites and favorites individually (parameter **-favoritetemplates**) or together (parameter **-favoritesall**).

runppmconfig -user <user name> -password <password>

-mode {import|export}

-favorites <file name>

[-client <client name>]

[–favoriteuser <ppm user>]

[-language <ISO code>]

[-overwrite]

[-replace]

[protocoloptions]

[-version]

[-favoritetemplates]

[-favoritesall]

List of specific parameters of favorites import and export

Parameter	Description
-favorites <file name>	File name with path to the XML file to which favorites are exported or from which favorites are imported. Specification: Mandatory
[–favoriteuser <ppm user>]	Specifies the PPM user that the favorites are imported or exported for. If you do not specify this option, import or export will be executed for all PPM users. Specification: Optional
[-overwrite]	Adds new favorites and folders, and changes existing favorites and folders, that is, content of the favorites and access privileges of the shared favorites folders. Specification: Optional
[-replace]	Deletes all folders and favorites and then imports the new favorites and folders including access privileges of the shared favorites folders. Specification: Optional
[-favoritetemplates]	Exports or imports only template favorites of all users. Specification: Optional
[-favoritesall]	Exports or imports favorites and template favorites of all users. Specification: Optional

Example 22.

- Export of favorites of all users
  runppmconfig -user <user name> -password <password> -mode export –favorites <file name>
- Export of favorites of a specific user
  runppmconfig -user <user name> -password <password> -mode export –favorites <file name> -favoriteuser <user>
- Import of favorites for all users
  runppmconfig -user <user name> -password <password> -mode import –favorites <file name>
- Import of favorites for a specific user
  runppmconfig -user <user name> -password <password> -mode import –favorites <file name> -favoriteuser <user>

- Export of all template favorites of the "system" user to the file TemplatesSystem.xml

  runppmconfig -user <user name> -password <password> -mode export –favoritetemplates TemplatesSystem.xml -favoriteuser system
- Export of all template favorites of all PPM users to the file Templates.xml

  runppmconfig -user <user name> -password <password> -mode export –favoritetemplates Templates.xml
- Export of all favorites and template favorites of the "system" user to the file FavoritesAllSystem.xml

  runppmconfig -user <user name> -password <password> -mode export –favoritesall FavoritesAllSystem.xml –favoriteuser system

14.7. runppmcpi

During **Easy mining**, a lot of messages including notes and warnings can be generated in PPM. To avoid significant system performance loss due to a large number of messages (from about 800-1000), you can delete existing messages via the user interface or automatically using the command line.

With the **runppmcpi** command line program, you can delete multiple messages simultaneously. You can either delete the messages permanently or you can assign them the status **To be deleted**.

To run **runppmcpi**, you need the **CPI administration** function privilege.

> **Tip**
> Further information on Easy mining is available in the online help under **PPM help topics**.

14.7.1. runppmcpi parameters

-deletecpimessages

Parameter to start the deletion procedure of CPI messages. As a stand-alone parameter with no effect. At least one of the following parameters must be transferred as a filter: **-excluding**, **-messagetype**, and **–owner**.

-mode <delete|mark_deleted>

Specifies if messages will be deleted (**delete** mode) or are set to be deleted (**mark_deleted** mode).

In **mark_deleted** mode, all CPI messages are set to be deleted without actually being deleted permanently. These messages can be restored. Default value for **–mode** (without –mode parameter) is **mark_deleted**.

The **–deletecpimessages** parameter is absolutely necessary for starting the deletion procedure. The parameters **-excluding**, **-messagetype** and **-owner** are AND-linked and restrict the set of messages to be deleted.

Example 23.

runppmcpi … - deletecpimessages –excluding 5w –mode mark_deleted

All CPI messages older than 5 weeks are set to the **to_delete** status.

Example 2

runppmcpi … -deletecpimessages –excluding 5w –mode delete

Example 24.

All CPI messages older than 5 weeks are deleted.

-excluding <n>[y|q|m|w|d]

Excluding filter with current reference date: y=year, q=quarter, m=month, w=week, d=day. "n" must be an integer.

All CPI messages older than the excluding filter are permanently deleted/set to the **to_delete** status.

Example 25.

runppmcpi … -deletecpimessages –excluding 5w –mode delete

All CPI messages older than 5 weeks are deleted.

-messagetype <system_message, activity, comment, outlier_message, earlyalert_message, alarm_message, target_message, processmining_message>

The parameter filters by the type of CPI message. All CPI messages of the transferred **messagetype** are deleted. Multiple **messagetypes** can be transferred.

Example 26.

runppmcpi … -deletecpimessages -messagetype comment –mode delete

All comments will be deleted.

runppmcpi … -deletecpimessages -messagetype comment earlyalert_message

All comments AND early alert messages will be set to the **to_delete** status.

-owner <owner>

The CPI messages of the user specified by **owner** will be deleted.

Example 27.

runppmcpi … -deletecpimessages –owner mueller -messagetype comment –mode delete

All comments with owner=mueller will be deleted.

-registry <host:port>

By specifying this parameter, you can access a remote PPM system whose RMI registry can be reached via the specified computer name and port number.

-usessl <true|false|default>

If you specified the **-registry** parameter, you can use the **-usessl** parameter to specify whether the remote system is to use TLS-encrypted data communication or not.

Parameter	Description
true	The remote PPM system uses encrypted data transfer. Ensure that the local system and remote system are using the same keystore.
false	The remote system uses unencrypted data transfer.
default (default value)	The settings of the local PPM system are used.

- After calling runppmcpi, the number of deleted CPI messages is output.
- After calling runppmcpi, the owner of the CPI message (CURRENT_OWNER) is set to the user specified in the **-user** parameter.
- The **–excluding** parameter is exact to the day at the most (for example, **–excluding 1d**). It is impossible to specify hours (for example, **–excluding 1h**).

Master-sub-server system

In a scaled system, CPI messages can be created on the master and sub-servers. When deleting messages via **runppmcpi**, master and sub-servers are considered stand-alone servers. No deletion will take place.

14.8. runppmimport

The **runppmimport** command line program starts the subsequent processing of the process instance fragments imported through the XML import into the PPM system. EPC import, process merge, typification and measure calculation are executed consecutively for each process instance.

14.8.1. runppmimport parameters

-merger anonymizing

All process instances in the PPM database that have not yet been anonymized are now anonymized.

Normally, only new process instance fragments added are anonymized when importing process instance fragments. Existing instances that are not processed by the merger are not anonymized.

It is not possible to change the anonymization of process instances already anonymized, as the information about the actual processor is overwritten when an instance is anonymized.

-typifier new

All process instances in the PPM database (new and those already typified) are retypified using the specified process type definition.

Retypifying process instances also requires the recalculation of the measures and dimensions, as they depend on the process type.

-keyindicator new

The measures and dimensions for all process instances are re-calculated. You should use this option if you have imported a new Measure calculator configuration (see also chapter Recalculation [79]).

-reinitanalysisserver {yes|no|auto}

With **-reinitanalysisserver yes** you specify that the analysis server is to be reinitialized after a successful PPM import. If you use the **auto** parameter, the analysis server is reinitialized only if required. With the default setting **no** (default) you prevent reinitializing completely. This option applies to process instance data, process instance-independent data, and Data analytics data.

If recovery files are recreated during the reinitialization of the analysis server (runppmimport –reinitanalysisserver [yes|auto] –recoveryfile yes), all recovery files will be generated.

The technical documentation **PPM Data Analytics** provides details on the reinitialization of Data analytics analysis realms of the analysis server.

-ps <paramset>

Specifies the XML file containing the paramset to limit the quantity of process instances for which one of the options **-typifier new** or **-keyindicator new** are to be used (see also chapter Recalculation [79]). If necessary, specify the complete path for the file.

Even if you specify a paramset, error instance inclusion is controlled by the **-errorinstances** parameter and needs to be turned off explicitly, if required.

Example 28.

In the analysis component, the Number of processes measure has been displayed over time with filters for the Material/MACH (machines) and Time (2nd quarter 2002) dimensions for the Cash sales process type in the Order processing process type group. The corresponding paramset was saved in the file paramset.xml.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE paramset SYSTEM "paramset.dtd">
<paramset>
```

```
<analysewizparam selectedrow="0"/>
<chartparam>
<chart-chartposition>0.16388889,0.20876624
</chart-chartposition>
<chart-legend-visible> false</chart-legend-visible>
<chart-zoom>75</chart-zoom>
<chart-scale>1.0,10.00.10.0</chart-scale>
<chart-colinfo/>
</chartparam>
<cpiparam/>
<epkparam/>
<favoritparam usecache="yes"/>
<kiquery usecache="yes" showzero="auto">
<keyindicator>
<criterion name="PNUM">Number of processes</criterion>
</keyindicator>
<iteration>
<criterion name="PROCESSTYPE">Process type</criterion>
<refinement name="BY_LEVEL2">Detailed</refinement>
</iteration>
<iteration>
<criterion name="TIME">Time</criterion>
<refinement name="BY_MONTH">By month</refinement>
</iteration>
<iteration>
<criterion name="MATERIAL">Material</criterion>
<refinement name="BY_LEVEL2">Detailed</refinement>
</iteration>
<filter>
<criterion name="PROCESSTYPE">Process type</criterion>
<filteritem operator="or">
<datalist usewildcard="no">
Order processing
<dataitem>
Order processing\cash sales
<datatype name="TEXTPAIR">Text pair</datatype>
<scale name="LEVEL1SCALE">Rough</scale>
</dataitem>
</datalist>
</filteritem>
</filter>
<filter>
<criterion name="TIME">Time</criterion>
<filteritem operator="or">
<datarange operator="range">
2. Quarter 02
<dataitem value="2ND QUARTER.2002">
2. Quarter 02
<datatype name="TIME">Time</datatype>
<scale name="QUARTERSCALE" factor="7776000.0">
Quarter(s)
</scale>
</dataitem>
<dataitem value="2ND QUARTER.2002">
2. Quarter 02
```

```
<datatype name="TIME">Time</datatype>
<scale name="QUARTERSCALE" factor="7776000.0">
Quarter(s)
</scale>
</dataitem>
</datarange>
</filteritem>
</filter>
<filter>
<criterion name="MATERIAL">Material</criterion>
<filteritem operator="or">
<datalist usewildcard="no">
Machines (MACH)
<dataitem>
MACH{Machines}
<datatype name="TEXTPAIR">Text pair</datatype>
<scale name="LEVEL2SCALE" factor="2.0">
Detailed
</scale>
</dataitem>
</datalist>
</filteritem>
</filter>
</kiquery>
<navparam>
<nav-view>chart</nav-view>
</navparam>
<tableparam view="listtable">
<tbl-cross totalresult="yes"/>
<tbl-instance split="horizontal"/>
</tableparam>
</paramset>
```

Example 29.

Executing the following command line only retypifies and calculates those process instances, which match the filter specifications in the above paramset:

```
runppmimport -user system -password manager -typifier new -ps paramset.xml
```

-genstats {before|after|keyindicator|merger|delete}

Updates or deletes the database statistics before the planned values are checked. **before** executes the function at the start, **after** at the end, **merger** before each call of the merger, and **keyindicator** before each measure calculation. You can use **delete** to delete the database statistics at the start of the PPM import. The parameter values can be combined. The generation of statistics as a standard database user is only supported by Oracle. For all other DB systems, the calculation of statistics is a task for administrators and must be executed by users with DB administrator rights.

-errorinstances {yes|no}

You use **-errorinstances yes** to specify that all error instance are to be recalculated as part of the PPM import. Default value: **yes**.

-recoveryfile {yes|no}

With **-recoveryfile yes** (default setting) you specify that the analysis server recovery files relevant for process analysis are to be created under <installation directory>\ppm\server\bin\work\data_ppm\recovery\<client> after a successful PPM import.

14.8.2. Recalculation

Global recalculation

If you change the measure configuration or typifying rules and want to update the existing data, you need to do the following:

1. Perform a retypification (parameter **-typifier new**) or a recalculation (parameter **-keyindicator new**).
2. Reinitialize the analysis server (parameter **-reinitanalysisserver auto**), which rebuilds the memory structure of the analysis server completely.

You can combine both steps in one call, **runppmimport**, or perform them separately. The recalculated data is displayed in the analysis only after both steps have been performed successfully.

A retypification also implies a recalculation of measures because the change of association of a process instance with certain process types and process groups requires new process type-specific measures to be calculated.

Partial recalculation

You can limit the number of process instances to be recalculated by using the **-ps** parameter to specify a paramset that identifies the process instances to be recalculated. In this case, the analysis is updated automatically and a reinitialization is not required.

14.9. runppmcompress

Aggregation of process instances can be automated using the **runppmcompress** command line program. The process instances to be aggregated are determined by a paramset that is transferred to the program as an XML file when the program is run. You can set up this paramset conveniently using the analysis component of the PPM user interface and save it to an XML file.

Please note the following when creating the paramset:

- The XML file may contain only one paramset (**paramset** XML element).
- The paramset may only contain the query of the PNUM (number of processes) measure (**kiquery** XML element).
- It must not contain any filters or iterations on function dimensions.
- It must not contain any user-defined step widths and user-defined dimensions.
- It must either contain a time filter or the **-excluding** option must be specified.
- The **Process type** dimension must be included. The step width is automatically set to **Level 2 (detailed)**.
- Any number of additional filters can be defined to limit the set of process instances to be aggregated.

If one of the above conditions is not met, the program will abort with an error message.

14.9.1. runppmcompress parameters

-ps <file name>

This parameter is used to specify the XML file containing the paramset.

-mode <test|execute>

test: This parameter enables you to perform a test run using the output statistics to identify which process instances and how many would be aggregated. This does not make any changes to the database.

execute: The process instances are aggregated. This operation cannot be undone. A statistics is output.

Default value: **test**

-excluding <n>[y|q|m|w|d|h]

The process instances lying in the specified time period are not aggregated, even if they are included in the filter. The period is specified as a number of particular sections of time. The start time of the program is used as the reference time.

Abbreviation	Time section	Example
y	Year	**1y**: Process instances from January 1st of the current year to the start time of the command line program are not deleted.
q	Quarter	**2q**: Process instances for the current last completed quarter are not deleted.
m	Month	
w	Week	**4w**: Process instances for the current week and the last three weeks are not deleted.
d	Day	
h	Hour	**24h**: Process instances for the current hour and the last 23 hours are not deleted.

Example 30.

In the analysis component, the **Number of processes** measure has been displayed over time with the filter **1.1.2003** to **15.2.2003** with a weekly scale for the **Other orders** process type. The corresponding paramset was saved in the file **paramset.xml**:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE paramset SYSTEM "paramset.dtd">
<paramset>
<analysewizparam selectedrow="0"/>
<chartparam>
<chart-legend-visible> false</chart-legend-visible>
<chart-scale>1.0,10.00.10.0</chart-scale>
<chart-zoom>75</chart-zoom>
<chart-chartposition> 0.175,0.1</chart-chartposition>
<chart-colinfo/>
</chartparam>
<cpiparam/>
<epkparam/>
<favoritparam />
<kiquery>
<keyindicator>
<criterion name="PNUM">Number of processes</criterion>
</keyindicator>
<iteration>
<criterion name="TIME">Time</criterion>
<refinement name="BY_WEEK">By week</refinement>
</iteration>
<filter>
<criterion name="TIME">Time</criterion>
<filteritem operator="or">
<datarange operator="range">
1. Jan 03 - 15 Feb 03
<dataitem value="1.1.2003 0:00">
01.01.03 0:00
<datatype name="TIME">Time</datatype>
<scale name="MINUTESCALE" factor="60.0">
Minute(s)
</scale>
</dataitem>
<dataitem value="15.2.2003 23:59">
15.02.03 23:59
<datatype name="TIME">Time</datatype>
<scale name="MINUTESCALE" factor="60.0">
Minute(s)
</scale>
</dataitem>
</datarange>
</filteritem>
</filter>
<filter>
<criterion name="PROCESSTYPE">Process type</criterion>
<filteritem operator="or">
<dataitem>
Order processing\Other orders
<datatype name="TEXTPAIR">Text pair</datatype>
```

```
<scale name="LEVEL2SCALE" factor="2.0">
Detailed
</scale>
</dataitem>
</filteritem>
</filter>
</kiquery>
<navparam>
<nav-view>chart</nav-view>
</navparam>
<tableparam view="listtable">
<tbl-cross totalresult="yes"/>
<tbl-instance split="horizontal"/>
</tableparam>
</paramset>
```

Example 31.

Executing the command line

```
runppmcompress -user system -password manager -ps paramset.xml -mode execute
```

Example 32.

will aggregate the process instances in the period from 1.1.2003 to 15.2.2003 by week.

-recoveryfile {yes|no}

With **-recoveryfile yes** (default setting) you specify that the analysis server recovery files relevant for process analysis are to be created after successful aggregation under <installation directory>\ppm\server\bin\work\data_ppm\recovery\<client>.

14.10. runppmdelete

This command line program enables you test and execute the deletion of process instances. The process instances to be deleted are determined by a paramset that is transferred to the program as an XML file when the program is run. You can set up this paramset using the analysis component of the PPM user interface and save it to an XML file.

Please note the following when creating the paramset:

- The XML file may contain only one paramset (**paramset** XML element).
- The paramset may only contain the query of the **Number of processes** (PNUM) measure (**kiquery** XML element).
- It must not contain any filters or iterations on function dimensions.
- It must not contain any user-defined step widths.
- It must either contain a time filter or the **-excluding** option must be specified.
- It must contain a process type filter. If the iteration on the **Process type** is not specified, it is added automatically.

- Any number of additional filters can be defined to limit the set of process instances to be deleted.

If one of the above conditions is not met, the program will abort with an error message.

14.10.1. runppmdelete parameters

-ps <file name>

This parameter is used to specify the XML file containing the paramset.

-mode <test|execute>

test: This parameter enables you to perform a test run using the output statistics to identify which process instances and how many would be deleted. This does not make any changes to the database.

execute: The process instances are deleted. This operation cannot be undone. A statistics is output.

Default value: **test**

-excluding <n>[y|q|m|w|d|h]

The process instances lying in the specified time period are not deleted, even if they are included in the filter. The period is specified as a number of particular sections of time. The starting time of the command line program is used as the reference time.

Abbreviation	Time section	Example
y	Year	**1y**: Process instances from January 1st of the current year to the start time of the command line program are not deleted.
q	Quarter	**2q**: Process instances for the current last completed quarter are not deleted.
m	Month	-
w	Week	**4w**: Process instances for the current week and the last three weeks are not deleted.
d	Day	-
h	Hour	**24h**: Process instances for the current hour and the last 23 hours are not deleted.

Example 33.

In the analysis component, the **Number of processes** measure has been displayed over time with a filter for the **Customer/Austria** dimension for the **Other orders** process type. The corresponding paramset was saved in the file **paramset.xml**:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE paramset SYSTEM "paramset.dtd">
<paramset>
<analysewizparam selectedrow="0"/>
```

```
<chartparam>
<chart-chartposition> 0.175,0.1</chart-chartposition>
<chart-legend-visible> false</chart-legend-visible>
<chart-zoom>75</chart-zoom>
<chart-scale>1.0,10.00.10.0</chart-scale>
<chart-colinfo/>
</chartparam>
<cpiparam/>
<epkparam/>
<favoritparam usecache="yes"/>
<kiquery usecache="yes" showzero="auto">
<keyindicator>
<criterion name="PNUM">Number of processes</criterion>
</keyindicator>
<iteration>
<criterion name="TIME">Time</criterion>
<refinement name="BY_MONTH">By month</refinement>
</iteration>
<filter>
<criterion name="PROCESSTYPE">Process type</criterion>
<filteritem operator="or">
<datalist usewildcard="no">
Order processing
<dataitem>
Order processing\Other orders
<datatype name="TEXTPAIR">Text pair</datatype>
<scale name="LEVEL1SCALE">Rough</scale>
</dataitem>
</datalist>
</filteritem>
</filter>
<filter>
<criterion name="PRINCIPAL">Sold-to party</criterion>
<filteritem operator="or">
<datalist usewildcard="no">
AUSTRIA (A)
<dataitem>
A{AUSTRIA}
<datatype name="TEXTPAIR">Text pair</datatype>
<scale name="LEVEL2SCALE" factor="2.0">
Detailed
</scale>
</dataitem>
</datalist>
</filteritem>
</filter>
</kiquery>
<navparam>
<nav-view>chart</nav-view>
</navparam>
<tableparam view="listtable">
<tbl-cross totalresult="yes"/>
<tbl-instance split="horizontal"/>
```

```
</tableparam>
</paramset>
```

Example 34.

Executing the following command line will delete all process instances with the **Sold-to party Austria**:

```
runppmdelete -user system -password manager -ps paramset.xml -mode execute
```

-recoveryfile {yes|no}

With **-recoveryfile yes** (default setting) you specify that the analysis server recovery files relevant for process analysis are to be created after successful deletion under <installation directory<\ppm\server\bin\work\data_ppm\recovery\<client>.

14.11. runppmexport

The **runppmexport** command line program is used to create exports by means of batch scrips. The parameters required to perform the export have to be stored in the requested export definition file or the parameters have to added to the command line as follow.

runppmexport [-client <name>] -user <user> -password <password>

-Export <Exportdefinitionfile> -destination <outputfilename> [-style <style>]

[-format <outputformat>] [-language <ISO-code>] [-convert]

[-collisionhandling [<IGNORE|IGNORE_WITH_WARNING|OMIT]]

[[-globalfilter <filter>] | [-globalfilterfavorite <favorit> [-favoriteserver FAVORITE_PRIVATE | FAVORITE_SHARED] [-ignoreproctypefilter]]]

[protocoloptions]

14.11.1. runppmexport parameters

-client

Client for which the export is performed. If no client is specified the default client is used.

-user

User performing the export.

-password

Password of theperforming user.

-Export

XML file containing the export definitions.

-destination

Path and file name of the export file.

-style

Name of the style file stored on the server. If no style file is specified for the export and the parameter is not set, a error message is displayed.

-format

Output format of the export. Permitted values are HTML, PDF, XML, and CSS.

-language

Language in witch the export is performed.

-convert

Conversion of previous export definitions.

-collisionhandling

Procedure in case of global filter collisions.

IGNORE = filter is ignored

IGNORE_WITH_WORNING = filter is ignored and a warning is displayed.

OMIT = The not compatible part of the export is not be exported.

Default is IGNORE_WITH_WORNING.

-globalfilter

Settings of a global filter.

-globalfilterfavorite

Favorite containing a filter used to perform the export.

The parameter is used in the following form.

-globalfilterfavorite <path><favorite name>

<Path> starts with **** followed by any folder name. <favorite name> is the name of the stored favorite file.

Example 35.

- **-globalfilterfavorite \Details\Sales\CycleTime_1** describes the **CycleTime_1** favorite in the private **\Details\Sales** directory.
- **-globalfilterfavorite \Charts\Chart_1 -favoriteserver FAVORITES_SHARED** describes the **Chart_1** favorite in the shared **\Charts** directory.

-favoriteserver

Specifies the favorite used as private or shared. Values are FAVORITES_PRIVATE or FAVORITES_SHARED.

Parameter is optional.

Default is FAVORITES_PRIVATE.

-ignoreproctypefilter

Ignores the process type set in the favorite specified. If the parameter is not set the process type filter is applied.

Example 36.

-globalfilterfavorite \visualization\table\crosstable -favoriteserver FAVORITES_PRIVATE -ignoreproctypefilter describes the **crosstable** favorite in the private directory **\visualization\table**. The filter for the process type is not applied during export.

[protocoloptions]

Protocol settings

The **-globalfilter** and **-globalfilterfavorite** parameters can not be applied simultaneously. If both parameters are used the first occurring parameter is applied.

If the **-globalfilter** parameter is not used, the **-favoriteserver** and **-ignoreproctypefilter** parameters are ignored without error message.

14.12. runppmadmin

All functions except Help can only be accessed by the system administrator (PPM user **system**) and are thus not available to all users without the corresponding privileges.

The system administrator is authenticated immediately after the command line program is called up. After successful authentication, the remaining parameters specified are analyzed. The individual actions are completed with a status message on the console.

14.12.1. runppmadmin parameters

-killsession <Session ID>

You use this parameter to log off the session of the current user. The parameter logs off only the user interface sessions. You cannot log off sessions that were started using command line programs (runppmconfig, runppmimport, for example).

-showusers

Outputs all sessions for the users currently logged into the system to the console in tabular form.

-clearcache

Clears the caches used in the system.

-fillcache

Triggers the calculation of the caches used in the system. Only entries that are pending recalculation will be updated. If you want to make sure all favorites are recalculated, call **runppmadmin -clearcache** before.

If you have deactivated automatic filling of the cache in the client-specific file **Keyindicator_settings.properties** with the setting **AUTO_FILL_CACHE=false**, each time data relevant to the cache changes, you can trigger re-filling of the cache using the command line with the **–clearcache** and **-fillcache** options.

-clearpchistory

The entire Performance Dashboard history cache will be deleted.

-savepchistory

The contents of the memory-based Performance Dashboard history cache are saved in the database. Already existing cache content will be lost.

-addpchistory

The contents of the memory-based Performance Dashboard history cache is extended by the newly added entries. Already existing cache entries will be retained.

-memoryinfo <file>

The following information is output and written to the file specified:

- General information on PPM and the analysis server
- Information on the runtime environment
- Information on cube and dimension data on the analysis server
- Information on configured indices
- Information on data collected for data analytics in the analysis engine

This information is intended for experiences users to help identify optimization potential of the analysis server.

When executing this command, you may experience wait times in the analysis because the analysis server first runs an internal memory streamlining process (garbage collection) before collecting the required data. During this time, the server does not respond to analysis queries.

Selectivity of configured indices

For each index the average selectivity regarding the queries is displayed.

The selectivity is calculated individually for each request in which the dimension appears as a filter. The selectivity is calculated as follows:

Expected number of process cube rows (process instances) to be visited if the specified filter values of this dimension are used, divided by the number of total rows if only the process type filter is used.

In the index statistics, the average of the selectivity of the individual requests per dimension is specified. In addition, the selectivity values of the dimension **< 25%**, **<50%**, **<75%**, and **<100%** are displayed. A small selectivity close to 0 (or 0%) is good, because this means that by using the index only very few data rows/

process instances had to be checked. Without index, considerably more data rows would have had to be checked.

-recoveryfile force

With the **-recoveryfile force** option, you force the recreation of recovery files of the PPM analysis server, regardless of server status. For each client, all recovery files are created.

By default, the recovery files of the client are saved under **<installation directory>\ppm\server\bin\work\data_ppm\recovery\<client>**.

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

-statisticinfo <file>

Returns statistical information as output and writes the output to the specified file. In scaled systems, the statistical information of each analysis (sub-) server is displayed separately.

The following information is returned as output:

- General information on PPM and the analysis server
- Information on cube and dimension data on the analysis server
- Information on configured indices
- Information on data collected for data analytics in the analysis engine
- Statistical Information on queries to the analysis engine

The output for the first four points is identical to the output of the **-memoryinfo** parameter. This parameter differs from the **-memoryinfo** parameter because it does not instruct the server to run garbage collection and the output does not include the information about the runtime environment. The information on the configured indices also includes statistical information on the indices.

The statistical information on the queries is also included in the output for the analysis engine for all process dimensions for which an index can be created:

- Number of all queries to the analysis server.

- Number of all queries to the analysis server with a filter on a process dimension,for which the server can create an index.

- Table of queries on process dimensions (the first four columns together form the primary key of the table):

- **Dimension**: Dimension name (an asterisk in brackets after the name indicates that an index is already defined on the dimension).
- **Process dimensions in query**: Number of filter values in query: 1, 2, 3, > 4. A value of 1 indicates that no other dimension was filtered except this dimension.
- **Index available**: The query was filtered by at least one dimension that had an index.
- **Index used**: Indicates if an index was used when the request was executed.
- **Usages in queries**: Number of queries

- **Selectivity**: **Average|<25%|<50%|<75%|<200%**: Average selectivity of the dimension and for how many requests the selectivity was <25%, <50%, <75% and <=100%.

  To calculate the selectivity, PPM divides the number of process cube lines that correspond to the specified process type filter by the number of viewed/visited process cube lines. A high selectivity, that is close to the value of 1 (or 100%) is good, because it indicates that only very few of the visited rows were not part of the result.
- **Filter values**: **Average|Minimum|Maximum**: Number of filter values for queries on this dimension. Displays the average, minimum and maximum of the number of filter values.

Note that only queries that are sent to the analysis engine are included in the statistical information. Queries that use data from the cache are not included in the statistics. The statistical information in not stored and all statistics are deleted after a restart of the analysis server.

-clearstatistic

The statistics information of the analysis server is deleted. In scaled systems, the statistics information of all analysis (sub-) servers is deleted.

14.13. runppmsendmail

Use the **runppmsendmail** command line program to send simple text messages. The command line program does not require a direct reference to a PPM client and, of the known general parameters, supports only the **-protocoloptions** parameter.

14.13.1. runppmsendmail parameters

-mailhost

IP address of the SMTP server. If the connection to the specified SMTP server fails, a corresponding message is output.

-from

E-mail address of the sender. This address is used when replying to the e-mail.

Some SMTP servers forward e-mails only if the sender's address is a valid e-mail address.

-to

E-mail address of the recipient. Use **;** to separate e-mail addresses of multiple recipients. If an e-mail address is invalid or if an e-mail cannot be sent to this address, a corresponding message is output.

It is mandatory to specify the parameters **-mailhost**, **-from**, and **-to**. Otherwise, a corresponding message is output.

-subject

Subject line of the e-mail. If this parameter is not specified, the subject line will read **PPM test mail**.

-salutation

Title and name of the recipient. If this parameter is not specified, the e-mail starts with the content specified by the **-message** parameter.

-fullname

Name of the sender of the e-mail. The name is displayed as the e-mail's author. If this parameter is not specified, the text **PPM server** is inserted.

-message

Text contents of the e-mail. Line breaks are inserted by keying in **
**. If this parameter is not specified, the following text is used for contents:

```
This message was sent by PPM in order to verify proper functionality.
Please do not reply to this message.
```

-language <ISO code>

Language to be used. The default texts of the parameters **-subject**, **-fullname**, and **-message** are also output in the specified language if the corresponding parameter was not indicated.

-attachment

Path to an attachment. Only one file can be sent as an attachment. To attach multiple files, you need to combine them in an archive, for example, in a ZIP archive.

14.14. runtableimport

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

Detailed information on runtableimport is available in the document **PPM Data Analytics**.

14.15. rundimdata

The **rundimdata** program imports process-independent dimension values in XML and CSV format.

Detailed information on rundimdata is available in the document **Data Import**.

Since Java 21, the encoding parameter is mandatory to import data from a CSV file.

14.16. runpikidata

The **runpikidata** program imports or exports data of process-independent data series in CML or CSV format.
Detailed information on runpikidata is available in the document **Data Import**.

Since Java 21, the encoding parameter is mandatory to import or export data from or to a CSV file.

15. Frequently asked questions

This chapter contains frequently asked questions about various topics and the corresponding answers.

Disclaimer/dissociation

The solutions described in this chapter represent tips and suggestions. Particularly with descriptions of characteristics and configurations of third-party software, this information is used at your own risk. SAG ARIS GmbH will not provide any support.

15.1. System integration

Q1.01	**If you install PPM under Windows Vista, Windows Server 2008, or Windows 7 the installation is incomplete. ARIS Cloud Agent cannot be set up as a Windows system service. Why?**
	The above Windows operating systems have stricter security policies. Run the installation program as an administrator (pop-up menu item **Run as administrator**). Other user accounts do not work, even if they were assigned administrator privileges.
Q1.02	**Which network resources does a PPM system require?**
	For detailed information, refer to the **PPM system architecture** documentation (**PPM network/Network protocols** chapter).
Q1.03	**How can the PPM installation be protected against unauthorized access?**
	To protect your PPM installation against unauthorized access, you can use the operating system's file access privileges. Note that the operating system user whose ID is used to start system services (the Windows default is the **SYSTEM** user) has full access privileges for the directory structure.
Q1.04	**Does PPM support NAT (NetworkAddressTranslation)?**
	The http or https protocol used by the PPM user interface supports NAT. Therefore, running PPM in a public cloud environment is possible provided the system is only to be accessed externally using the PPM interface. The Java RMI protocol used by PPM for the command line programs does not support NAT. Therefore, running PPM in a public cloud environment is impossible if the system is to be accessed externally using the command line programs.
Q1.05	**Can https be used to access PPM?**
	Yes. The load balancer supports https since PPM 9.0 by default. See also chapter https support [55]
Q1.06	**How can data transfer between PPM server and user interface be encrypted (TLS encryption)?**
	See Safe communication channels [55] chapter.
Q1.07	**What is the "server" mode on a PPM server?**
	The **server** option is configured during installation of a client. If the option is enabled, the PPM server uses the Java VM version optimized for server applications. To use this option, you must use a Java SDK version for the PPM server.
Q1.08	**How do I change the Java Runtime Environment used by the PPM server?**
	It is impossible to change the Java Runtime Environment. From PPM 9, only the one supplied can be used.
Q1.09	**When using PPM with MS Internet Information Server (IIS), I cannot access PPM. Why?**
	From PPM 9.0, IIS is no longer supported.

Q1.10 **It is impossible to connect to a PPM client server. The extended information in the login dialog states that the computer name of the PPM client server is unknown. What is the reason for this?**

Possible causes:

- The client server does not have the status **STARTED**.
- The DNS system detects a different computer name for the client server, depending on whether you address it using the name or IP address. Make sure that the **nslookup <computer name>** and **nslookup <IP address>** refer to the same computer.
- The DNS system in your network cannot resolve the computer name of the PPM client server. If PPM server and user interface are being operated in different subnets or multiple DNS servers exist in the network, the computer name may be specified in the file **Registry_settings.properties** located in the **clientjars** directory on the http server. Avoid inconsistent name resolutions using etc/hosts files and the DNS system.
- Forwarding of the request from the load balancer to the client server is not configured or is incorrectly configured. Check whether the target addresses are entered in the files **httpd-proxy.conf** and **httpd-proxy-ssl.conf** in the directory **<Installation directory>\ppm\server\bin\work\work_loadbalancer_<Memory model>\httpdconf\extra** and are accessible.

Q1.11 **The connection to the PPM server fails. Example message in log output**:

```
Client "umg_en" does not exist.
Error connecting to server.
Trying to connect PPM server //pcppm:16300/ppm_rmi_server_umg_en
ppm_rmi_server_umg_en
java.rmi.NotBoundException: ppm_rmi_server_umg_en
```

The required client server may not have been started or the specified client name is not correct. If the client server is running, it is also possible that the PPM RMI registry has been restarted, for example, due to an unexpected system incident. In this case, restart the client server.

Q1.12 **Does PPM support the use of a cross-company directory service?**

The LDAP interface integrated into central User Management allows particular users to be transferred. The users transferred from the LDAP system are authenticated synchronously against the corresponding LDAP system.

Detailed information on using central User Management is available in the online help.

Q1.13 **What is important about environment variables?**

Since PPM 9.0, environment variables have no longer been supported.

Q1.14 **What is a client-specific configuration file?**

For each PPM client installed, a directory of the same name exists in the subdirectory **<installation directory>\ppm\server\bin\work\data_ppm\config**. This directory contains the component-specific configuration files for the client server and analysis server. These text-based configuration files have the extension **_settings.properties**.

Q1.15 **What is a system-wide configuration file?**

The system-wide configuration files specify global settings for the entire PPM system and are saved under **<installation directory>\ppm\server\bin\work\data_ppm\config**.

Q1.16 **How do I change the TCP/IP ports of the PPM services?**

You change the TCP/IP ports using ARIS PPM Cloud Controller.

Further information on the individual commands is available in the help of the ARIS PPM Cloud Controller command line. You can view the help by entering the command **help** in the command line.

- **PPM registry**
  reconfigure ppm_core
- **PPM analysis server**
  using PPM CTK
- **PPM server**
  using PPM CTK

Q1.17	**When installing PPM, errors occur when the required Visual C++ Redistributable packages are installed. What can be done?**
	There are two possible causes. ▪ The first may be that the packages cannot be installed because the computer has to be rebooted by an earlier update process. In this case, restart your computer. Then you should be able to carry out the PPM installation successfully. ▪ Secondly, it can happen that the packages have already been installed once, and now they can no longer be overwritten. In this case, uninstall the existing packages. They will then be re-installed by the PPM installation and the PPM installation can be completed successfully.

15.1.1. Database connection

Q2.01	**What settings are important for the DB schema used?**
	The data for a PPM client is saved in a separate DB schema of the database type used. For detailed information about how to configure the DB schema and the connection data, please refer to the **PPM Database Systems** user guide. If the connection to the database cannot be established when starting the client server, corresponding error messages indicate the possible causes. Use PPM Customizing Toolkit (**Edit client**, **Database settings** dialog, **Test database connection** button) to check and adapt the correct database connection of the client.
Q2.02	**How do I change database access data?**
	Edit the client-specific configuration file **<installation directory>\ppm\server\bin\work\data_ppm\config\<client>\Database_settings.properties** and adjust the key values **URL**, **USER**, and **PASSWD** accordingly. Adjustment of the database type specified in the **DATABASE_TYPE** key is also supported. Use the comments in the file for brief information. If you want to specify the database password in encrypted form, use the PPM command line program **runpassencryptor** with the **-mode db** and **-password <new password>** parameters.
Q2.03	**When starting the PPM server, the database error java.lang.ClassNotFoundException is displayed. What is the reason for this?**
	The required database drivers must be copied manually after PPM installation and before client installation to the directory **<installation directory>\ppm\server\bin\work\data_ppm\drivers**. If the client already exists you can install a database driver via ACC (enhance <client name> with dbDriver local file <path to driver jar file>). For further information about the database connection, please refer to the **PPM Database Systems** user guide.
Q2.04	**When starting the PPM server, the message** `DB error: Connection not established. (1006) [17002/null] (1034)` **is displayed. Why?**
	The Oracle database listener is not available. Check the Host and Port entries for the JDBC connect string (URL=, line 83) in the client-specific configuration file **Database_settings.properties**.

Q2.05	**When starting the PPM server, the message**
	`DB error: Connection not established. (1006) [-4499/null] (1034)`
	`Connection not established. (1006) [-4499/null]`
	`[ibm][db2][jcc][t4][2043][11550] Exception java.net.ConnectException: Error opening socket to server ...`
	is displayed. Why?

	The DB2 database is not available. Check the **Host** and **Port** for the JDBC connect string (URL=, line 83) in the client-specific configuration file **Database_settings.properties**.

Q2.06 **When starting the PPM server, the message**

```
DB error: Error reading Isolation-Level (1500) [-99999/null] (1034)

        Error reading Isolation-Level (1500) [-99999/null]

        DatabaseMetaData information is not known for server DB2/ ...
```

is displayed. Why?

The database and JDBC driver version for the IBM DB2 database do not match.

Q2.07 **When starting the PPM server, the message**

```
DB error: Connection not established. (1006) [0/08S01] (1034)

        Connection not established. (1006) [0/08S01]

        Unable to establish TCP/IP connection with host. java.net.ConnectException:
Connection refused

        DB error: Connection not established. (1006) [17002/null] (1034) ...
```

is displayed. Why?

The Oracle database listener is not available. Check the **Host** and **Port** for the JDBC connect string (URL=, line 83) in the client-specific configuration file **Database_settings.properties**.

Q2.08 **How do I identify the database and JDBC driver version used by the PPM server?**

When starting, the PPM server outputs messages stating which database and JDBC driver versions are used. Examples:

- **Oracle**

```
I: ...: [SRV] Database version used: Oracle9i Enterprise Edition Release 9.2.0.7.0 -
Production
I: ...: [SRV] JDBC driver used: Oracle JDBC driver (10.2.0.4.0).
```

- **IBM DB2**

```
I: ...: [SRV] Database version used: SQL09013.
I: ...: [SRV] JDBC driver used: IBM DB2 JDBC Universal Driver Architecture (3.4.65).
```

- **MS SQL Server**

```
I: ...: [SRV] Database version used: 8.00.2039.
I: ...: [SRV] JDBC driver used: Microsoft SQL Server 2005 JDBC Driver (1.1.1501.101).
```

15.1.2. SAP-2-PPM Process Extractor

Q3.01 **What to observe when using the SAP Java connector**

When using a 64-bit variant of the SAP Java connector **sapjco.jar** you must use a 64-bit Java version. Otherwise, the connector will not recognize the required Java libraries.

Q3.02 **How to display information on the SAP Java connector used**

	Open a DOS prompt on the local directory in which you saved the SAP Java connector and execute the following command: `java -showversion -jar sapjco.jar` If you double-click the connector file, the information is displayed graphically.
Q3.03	**Which class paths must contain the SAP Java connector?**
	Ensure that the path to the file **sapjco.jar** is contained in the class path of all PPM product components that are to use the SAP Java connector (SAP-2-PPM Process Extractor, PPM Customizing Toolkit, and PPM if you want SAP transaction calls to be possible from the process instance table). The file must be stored under **<installation directory>\ppm\server\bin\work\data_ppm\drivers**.

15.1.2.1. Using Content Packages

Q4.01	**Where to find the client configuration templates PPM Content Package for SAP MM, or PPM Content Package for SAP SD after the installation**
	After installation, the client configuration templates you can use in combination with **PPM Process Extractor SAP-2-PPM** are located under **<installation directory>\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-<version>-runnable.zip\ppm\ctk\ctk\examples\custom\...**
Q4.02	**Which Content Package language versions are available?**
	The client configuration templates for extracting data from the SAP modules HR, MM and SD are available in German and English. The English version of the client configuration templates has the suffix **_en**. For example, if you want to use the **PPM Content Package for SAP SD** Content Package in English, use **PPM Customizing Toolkit** and the corresponding client template to create the client **ppm4sd_en**.

15.2. System administration

Q5.01	**A PPM user has forgotten the password. How do I change the password for an external PPM user**?
	Use the SYSTEM administration access to assign the user a new password in user management. In the Administrator Password input box, enter the password for the SYSTEM user.
Q5.02	**How do I block particular users from the system?**
	Withdraw all of the user's function privileges (at least login, data calculation and data import) or change the user's password to a value that the user does not know.
Q5.03	**How do I determine which users are currently logged on to the system?**

	In a command prompt for the relevant client, run the **runppmadmin** command with the **-showusers** option.
	Example:
	`C:\>pushd ARISSoftwareAG\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-95.1.0-RC8-trunk-20130404.122823-4-runnable.zip\ppm\bin`
	`D:\ARIS\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-95.1.0-RC8-trunk-20130404.122823-4-runnable.zip\ppm>runppmadmin -user system -password manager -showusers`
	The program displays a table showing you the sessions of all users currently logged into the PPM system on the default client.

Q5.04 **Why does system initialization fail using runinitdb?**

Log output includes:

```
E: 21.10.08 14:50:22: [STD] Error opening XML file "customsysmonxmlsysmon_locales.xml".
customsysmonxmlsysmon_locales.xml (The system cannot find the specified file)
[java.io.FileNotFoundException]
```

The directory path **<PPM installation directory>...\custom\sysmon\xml\sysmon_locales.xml** in the configuration file **Initdb_settings.properties** is probably specified in MS DOS format. The Java Runtime Environment interprets the standard MS DOS **** (backslash) character as an Escape character. Replace the character with a double backslash or a single forward slash **/**.

Q5.05 **What emergency user exists in the PPM system?**

The PPM user **SYSTEM** is provided as an emergency user. It is always available – neither this user nor this user's unrestricted system privileges can be deleted.

Q5.06 **What authentications are performed when a PPM user runs a PPM command line program (for example, runppmconfig)?**

If a PPM user runs a PPM command line program, the system verifies that this user has the appropriate privileges to perform the requested action(s). The authorization concept of the PPM system is based on the following components:

- Function privileges
- Data access privileges
- Process access privileges
- Data visibility privilege (via user group association)

Example (missing function privilege): The user **PARKER** wants to export the user management for the PPM system in which he is registered to an XML file using the command line. The call for the example client **umg_en** looks like this:

```
C:\ARIS\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-
<version>-runnable.zip\ppm\bin>runppmconfig -user vogel -password vogel -mode export
-users user.xml -client umg_en
```

As the user **PARKER** does not have the **Configuration import** function privilege, the following message appears in the command line:

```
I: 23.10.08 10:15:35: [IMP] Exporting configuration of component "users"...
E: 23.10.08 10:15:35: [IMP] You need the "Configuration import" function privilege.
S: 23.10.08 10:15:35: [IMP] Messages of component "IMP":
S: 23.10.08 10:15:35: [IMP] 1 error
```

No data exported. User **PARKER** must first obtain the required function privilege from the PPM system administrator.

Q5.07 **When running PPM command line programs, does the log indicate restricted data access privileges of a PPM user (for example, runppmcompress)?**

When running the command line aggregation, as well as function and process access privileges, the data access privileges of the relevant PPM user are also checked. Only those process instances for which the user has visibility privileges (data access privilege) will be aggregated. Data access privileges are defined for user groups in PPM user management. They are then applicable for all users assigned to that group.

Example: The user **PARKER** wants to aggregate all process instances of the **Standard order** process type for the period March 2008 using the command line. The call looks like this:

```
C:\ARIS\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-
<version>-runnable.zip\ppm\bin>runppmcompress -user vogel -password vogel -client umg_en
-mode execute -ps ps.xml
```

Of the 68 process instances contained in the aggregation paramset (**ps.xml**), the user **PARKER** only has data access privileges for the 20 process instances with the value **3000 (Frankfurt)** for the data access dimension **Sales organization**. Therefore, the user **PARKER** can only aggregate these 20 process instances. The following messages are displayed in the command line:

[...]

```
S: 23.10.08 11:05:48: [IMP] The following process sets will be aggregated:
S: 23.10.08 11:05:48: [IMP] 20 processes Order processing - Standard order; Sales
organization=3000(Frankfurt).
I: 23.10.08 11:05:48: [IMP] The program was terminated without aggregating the processes.
I: 23.10.08 11:05:48: [SRV] Establishing connection between user PPMDB and
jdbc:oracle:thin:@ppmdbsrv1:1521:PPM...
I: 23.10.08 11:05:48: [SRV] Database version used: Oracle9i Enterprise Edition Release
9.2.0.8.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.8.0 - Production.
I: 23.10.08 11:05:48: [SRV] JDBC driver used: Oracle JDBC driver (9.2.0.8.0).
```

Q5.08 **What happens if a PPM user with no user group association and thus no data access privileges wants to carry out editing actions using the command line?**

PPM users can only be assigned data access privileges through their association with a group. If a user with no user group association and thus no data access privileges runs a command line program, a message indicates that the user does not have any data access privileges.

Example:

```
C:\ARIS\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-
prod-95.1.0-RC8-trunk-20130404.122823-4-runnable.zip\ppm>runppmcompress -user vogel
-password vogel -client umg_en -mode test -ps ps.xml
[...]
I: 23.10.08 11:36:22: [IMP] No data delivered by the query.
[...]
```

Q5.09 **Why does a PPM user not see any data in the PPM user interface despite appropriate process access and function privileges?**

As a PPM administrator, check whether the user is assigned to a user group and whether this group has data access privileges (**all** or **restricted** data access privileges).

Q5.10 **How can restricted process access privileges affect actions carried out using the command line?**

If a PPM user only has access privileges for particular process type groups / process types, as is the case for the user **PARKER** in the following example for only the **Order processing\Cash sale** process type, it is possible that no data will be available for certain actions:

```
C:\ARIS\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-
<version>-runnable.zip\ppm\bin>runppmcompress -user vogel -password vogel -client umg_en
-ps ps.xml -mode execute
I: 23.10.08 13:25:30: [IMP] Aggregating in "execute" mode...
I: 23.10.08 13:25:30: [SRV] Establishing connection between user PPMDB and
jdbc:oracle:thin:@ppmdbsrv1:1521:PPM...
I: 23.10.08 13:25:30: [SRV] Database version used: Oracle9i Enterprise Edition
Release 9.2.0.8.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.8.0 - Production.
I: 23.10.08 13:25:30: [SRV] JDBC driver used: Oracle JDBC driver (9.2.0.8.0).
I: 23.10.08 13:25:31: [IMP] The component(s) "EPC: Keyindicator, KeyindicatorCalculation,
KeyindicatorCoreConfig, UserAdmin" has/have been locked by user PARKER
I: 23.10.08 13:25:32: [SRV] The caches have been cleared.
E: 23.10.08 13:25:32: [IMP] Unknown process type: Order processing - Standard order.
I: 23.10.08 13:25:32: [IMP] The component(s) "EPC: Keyindicator, KeyindicatorCalculation,
KeyindicatorCoreConfig, UserAdmin" have/has been unlocked by user PARKER.
```

The PPM user **PARKER** can only aggregate the process instances of the **Standard order** process type contained in the paramset (**ps.xml**) if he has been granted access privileges for this process type by the administrator.

Q5.11 **When importing data, I encounter OutOfMemory messages. What can I do?**

Allocate more memory to the corresponding program. You can change the memory data in PPM Customizing Toolkit using **Edit client** (calling the client installation) on the **Server settings** page under **JVM parameters of server programs > Configure JVM parameters**.

Adapt the maximum system memory for runxmlimport, for example, **1024 MB** for a maximum of 1 GB system memory.

Subsequently, you must restart the system service of the PPM server (if set up) or the PPM server.

Q5.12 **How do I increase the system memory for the PPM server?**

You can easily change the memory data in PPM Customizing Toolkit via **Edit client** (calling the client installation) on the **Server settings** page under **Client server settings**.

Subsequently, you must restart the system service of the PPM server (if set up) or the PPM server.

Q5,121 **How do I adjust system memory for the analysis server?**

You can easily edit the data for minimum or maximum memory requirements in PPM Customizing Toolkit via **Edit client** (calling the client installation) on the **Server settings** page under **Analysis server settings**.

You must restart the analysis server afterwards.

Q5.13 **Which periodical tasks exist for a PPM system?**

The regular maintenance tasks of a PPM system are project-specific and coordinated with the customer. We recommend that you regularly perform the following actions:

- Back up the client's database schema
- Back up the client configuration (**web**, **config**, and **custom** directories under <installation directory>\ppm\server\bin\work\data_ppm)
- Archive the imported source system data
- Archive the log files of the analysis server and PPM server and command line programs

For further information, please refer to the documentations **PPM Database Systems** and **PPM Data Import**.

15.2.1. PPM command line programs: Log output

Q6.01	**How can log output be configured?**
	The log options **information**|**warning** |**error** in the command line programs can be configured in any combination (see chapter Command line programs [36]). If you do not specify log options, only module-specific log levels from the **Server_Log_settings.properties** will be used (recommended for XML import and PPM import).
Q6.02	**Can log output be routed to a file?**
	The instruction **-protocolfile <File name>** can be used to write log output from the PPM command line programs to the specified file. You can combine this instruction with the log options described above as required.
Q6.03	**Can log output be set to a specific language?**
	The instruction **-language <ISO code>** can be used to set the language, provided it is supported by the PPM system, for example, **-language en**.
Q6.04	**Where are the log files of PPM server and analysis server located?**
	By default, the log files of PPM server and analysis server are located in the client-specific directory **<installation directory>\ppm\server\bin\work\data_ppm\log\<client>**. The file name is made up of the client name with the extension **.log**. After restarting the PPM server and analysis server, the log files are updated and not overwritten.
Q6.05	**Can I change the location where the log files are saved?**
	You can save the location of log files of the PPM server or analysis server later by adjusting the relevant appenders in the files **Server_log_settings** and **AnalyseServer_log_settings** (see chapter PPM system messages [26]).
Q6.06	**Are changes to the PPM configuration using the interface logged?**

	Changes to the PPM configuration via the GUI that are saved appear as messages in the log on the console of the PPM server or analysis server. In addition, output is saved in the trace files of the PPM server or analysis server if you activated the corresponding trace module in the settings for the PPM analysis server (see chapter **System messages**).
	
	For example, if a PPM user has created and saved a new user-defined measure, the processing components output corresponding messages (in this case: Measure calculator and server):
	
	`I: 22.10.08 16:30:29: [KZB] Checking measure configuration...`
	`I: 22.10.08 16:30:29: [KZB] The measure configuration has been successfully checked.`
	`I: 22.10.08 16:30:29: [SRV] Executing consistency check for user-defined measures...`
	`I: 22.10.08 16:30:29: [SRV] Consistency check for user-defined measures successful.`

15.2.2. Inconsistent data pools

Q7.01	**How to restore consistency in the data pools of analysis server and PPM server**

Changes to the configuration of the PPM systems or repeat data imports can lead to inconsistent data pools in the analysis server and the PPM server. Corresponding error messages of the PPM interface will notify the user of this.

Open a DOS command prompt, switch to the directory **<PPM installation directory>\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-<version>-runnable.zip\ppm\bin**, and execute the following command line:

```
runppmimport -user <ppm user> -password <password> -client <ppm client>
-reinitanalysisserver auto
```

The analysis server is then reinitialized and the changed data is loaded into its data structures. Now the data pools of both servers are consistent again.

16. Support and legal information

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

16.1. 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.

16.2. 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.

16.3. 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.