



webMethods Process Performance Manager OPERATION GUIDE

Version 9.10

April 2016

This document applies to PPM Version 9.10 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 - 2016 [Software AG](#), Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. 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. Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Contents

1	Text conventions	1
2	General	2
3	IT service and application management.....	3
3.1	Early alert system in case of memory shortage	3
3.1.1	Memory monitoring	3
3.1.2	System messages	4
3.1.3	Configuration.....	5
3.2	Workload monitoring.....	6
3.3	Maintenance concept.....	7
3.3.1	System update	7
3.3.2	Data archiving	7
3.3.3	Restore.....	8
3.4	High availability systems	8
3.5	Support desk management concept.....	9
4	Third-party software.....	12
5	Performance aspects	13
6	Administration.....	15
6.1	Cloud Agent.....	15
6.2	System status.....	15
6.3	Configuration.....	16
6.4	System messages.....	18
7	PPM system messages	20
7.1	Overview.....	20
7.2	Log file directories	22
7.3	System messages (log output)	22
7.4	Audit messages (audit logging).....	27
7.5	Command line programs.....	32
8	Globalization	35
9	Job automation.....	36
10	Accessibility	37
11	SSO integration	38
11.1	Configure parameters.....	38
11.1.1	SAML2 connection	38
11.1.2	Kerberos.....	39
12	Security aspects	40
12.1	Safe communication channels	40
12.2	Load balancer	40
12.2.1	Integrate SSL certificate.....	40
12.2.2	Further security measures for load balancer	41
12.3	PPM server	41
12.3.1	Generate and sign an SSL keystore.....	42
12.3.1.1	Configure PPM for SSL	43
12.3.2	Security measures for the server system	43

13	Command line programs	44
13.1	General structure and function-specific structure.....	44
13.2	PPM command line programs.....	45
13.3	Common arguments	46
13.4	runinitdb	47
13.5	runppmanalytics.....	48
13.5.1	runppmanalytics arguments.....	49
13.6	runppmconfig.....	50
13.6.1	The runppmconfig argument	50
13.6.2	Export the system configuration	53
13.6.3	Import and export of user data	54
13.6.4	Import and export of favorites	55
13.7	runppmcpic.....	58
13.7.1	The runppmcpic argument	58
13.8	runppmimport.....	60
13.8.1	The runppmimport argument	60
13.8.2	Recalculation	63
13.9	runprocessimport	64
13.9.1	runprocessimport arguments	65
13.9.2	Import settings	65
13.9.3	Import logging.....	66
13.9.4	Import multiple data sources	67
13.10	runppmcompress.....	68
13.10.1	The runppmcompress argument	68
13.11	runppmdelete	70
13.11.1	The runppmdelete argument.....	71
13.12	runppmadmin	73
13.12.1	The runppmadmin argument.....	73
13.13	runppmsendmail	75
13.13.1	The runppmsendmail argument	75
13.14	runtableimport.....	76
14	Frequently asked questions	77
14.1	System integration	77
14.1.1	Database connection	82
14.1.2	SAP-2-PPM Process Extractor	84
14.1.2.1	Using Content Packages	85
14.2	System administration	85
14.2.1	PPM command line programs: Log output	90
14.2.2	Inconsistent data pools	92

1 Text conventions

Menu items, file names, etc. are indicated in texts as follows:

- Menu items, key combinations, dialogs, file names, entries, etc. are displayed in **bold**.
- User-defined entries are shown in **<bold and in angle brackets>**.
- Single-line example texts (e.g., a long directory path that covers several lines) are separated by ↵ at the end of the line.
- File extracts are shown in this font format:

This paragraph contains a file extract.

2 General

This guide provides information about integrating the PPM product components **webMethods 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.

3 IT service and application management

webMethods 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.

3.1 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).

3.1.1 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** or **runprocessimport** command line programs (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.

3.1.2 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 was 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** or **runprocessimport** command line programs. 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.

3.1.3 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", i.e., 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

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
```

3.2 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, e.g., CPU and memory utilization. Only the system load utilized by the Java process (e.g., PPM server) within the Java VM is analyzed. The program **Visual VM** is part of the Java SDK 1.7. 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 program **jvisualvm** 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.

3.3 Maintenance concept

3.3.1 System update

After installation of the initial release versions of PPM and MashZone by means of the Software AG Installer (see documentation **Installing webMethods and Intelligent Business Operations Products**), you can update the products only using the Software AG Update Manager. A product update is usually called **Official Fix** and contains important error fixes for the product. We urgently recommend that you apply official fixes using the Software AG Update Manager after you installed the release version.

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

3.3.2 Data archiving

All process instances created on the basis of imported source system data, process instance-independent measures, configurations imported by the **runppmconfig** configuration program (customizing), as well as configurations extended or edited by the user via the front-end (e.g. 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>\ppmmashzone\server\bin\work\data_ppm\`. Save these directories each time you change the configuration. Save the **patch** directory in `<installation directory>\ppmmashzone\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>\ppmmashzone\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. To identify the storage location of the Web server directory that you specified during installation, refer to the **ManagementViews.BaseDirectory** key in the client-specific **Templates_settings.properties** configuration file.

If you installed Performance Dashboard for http-based publication of your analysis results (dashboards and Management views) you can save the configurations of all clients by backing up the directory `<installation directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\config`.

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.

3.3.3 Restore

In case of a crash recovery, restore the PPM installation by reinstalling it with the same options you used for the original installation. Please note that the required third-party software (e.g., JRE for PPM or Tomcat for Performance Dashboard) must already be installed. Then you need to recreate all clients with the original settings using 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** (Page 7), just in reverse order, and start the PPM system.

3.4 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, i.e., 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 (e.g., 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 (e.g., application pings) or system resource monitoring, e.g., 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** (Page 7). In case of PPM system failure, proceed as described in the chapter **Restore** (Page 8) for a new system with comparable hardware properties.

3.5 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** (Page 20) 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** (Page 32) 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<\ppmmashzone\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>\ppmmashzone\server\bin\work\data_ppm\custom\umg_en\log\.

System messages of the PPM servers can be processed in different ways, e.g., by using individual appenders. Chapter **System messages (log output)** (Page 22) 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>\ppmmashzone\server\jre\bin\java -version** in a command prompt.

Example

```
C:\> SoftwareAG\ppmmashzone\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
SoftwareAG\ppmmashzone\server\bin\agentLocalRepo\.unpacked\ppm-client-run-pr
od-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).
```

4 Third-party software

PPM requires certain third-party software. PPM has been developed in the Java programming language as a client-server architecture. RMI and Corba 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 executed in the Java plug-in of the Web browser as an applet and requires a corresponding Java plug-in for the browser used to be installed. You can also use the PPM client as a Java application by selecting the PPM Analysis GUI component in the Software AG 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 (e.g., Apache license).

5 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 Software AG 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 that you reserve the main memory exclusively for the VM running PPM.

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

Hardware scenarios for PPM and MashZone

PPM and MashZone 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 9.10	YPI	up to 200 000 PI/DR	up to 5 million PI/DR	up to 25 million PI/DR
MashZone 9.10	YCZ	1 PPM client up to 2 users	1 PPM client up to u 10 users	1 PPM client up to 100 users
		2 CPU cores 4 GB RAM 10 GB free HD space	4 CPU cores 16 GB RAM 20 GB free HD space	8 CPU cores 64 GB RAM 50 GB free HD space

Legend

PI Process instances

Legend

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.

6 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.

6.1 Cloud Agent

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

You can start Cloud Controller in the Windows program group Start > All Programs > Software AG > Administration.

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

and enter the following command:

```
acc.bat -c "<PPM installation directory>\ppmmashzone\server\generated.apptypes.cfg" -h localhost -p 17005 -u Clous -pwd g3h31m.
```

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

6.2 System status

You can query the status of the individual system components installed in 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

Cloud Controller command	Description
list	Lists all components installed on the selected 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 Cloud Agent.
stopall	Stops all components in the reverse start sequence.

Cloud Controller command	Description
killall	<p>Closes all components regardless of the current state or existing connections.</p> <p>Use kill <instance id> to close a specific component.</p> <p>Use this command only if components no longer respond to other commands.</p>

6.3 Configuration

Using 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

Cloud Controller command	Description
show config	Displays the current configuration of Cloud Agent and Cloud Controller.
show instance <instance id>	<p>Displays the current configuration of a component.</p> <p>Example</p> <p>show instance ppm_core</p> <p>Returns all configuration parameters of the ppm_core component.</p>
enhance [driver]	<p>Use the enhance command to install required drivers.</p> <p>Install database drivers</p> <p>enhance <client> with dbDriver local file <path to DB driver></p> <p>Install SAP JCO drivers</p> <p>enhance <client> with dbDriver local file <path to SAP JCO driver></p> <p>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.</p> <p>Example</p> <p>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.</p> <p>enhance umg_en with dbDriver local file C:/Temp/ojdbc6.jar</p>

Cloud Controller command	Description
enhance [PPM components]	<p>To install PPM components, such as Content Packages or Extractors at a later time, use this command:</p> <p>CSV2PPM Extractor</p> <pre>enhance <client> with extractor com.aris.ppm.enhancements.csv2ppm <version> type zip</pre> <p>ppm4mm Content Package</p> <pre>enhance <client> with contentPackage com.aris.ppm.enhancements.ppm4mm <version> type zip</pre> <p>Please note that the selected component must already exist in the repository of your installation. The local repository of your installation for PPM extensions is located here: <installation directory>\ppmmashzone\server\bin\agentLocalRepo\com\aris\ppm\enhancements. For the version please refer to the name of the subdirectory of the relevant component. For example, if 95.1.0 is specified under <installation directory>\ppmmashzone\server\bin\agentLocalRepo\com\aris\ppm\enhancements\jdfs2ppm, the version is 95.1.0.</p>
enhance [user]	<p>The command enhance can also be used for creating a new user in central user management.</p> <pre>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"</pre> <p>Creates a new user user1 with the password user1. Please use the name of the component User management for the instance ID (default: umcadmin).</p>

Cloud Controller command	Description
reconfigure <instance id> <parameter>	<p>This command is used for reconfiguring components. You can change all parameters saved in the configuration file under <installation directory>\ppmmashzone\server\bin\work\work_<instance id>\runtimeinfo.properties.</p> <p>Changing these values in this file does not affect the instance running. Parameters can be changed only with 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.</p> <p>Replace a backslash \ used in the password with a double backslash \\.</p> <p>Example</p> <p>You want to change the RMI communication between PPM client and server to SSL encryption. The following commands need to be executed in Cloud Controller.</p> <ul style="list-style-type: none"> ▪ stop ppm_core ▪ reconfigure ppm_core ssl.enabled="true" ssl.keystorefile="<path to keystore file>" ssl.keystorepassword="<keystorepassword>" ▪ start ppm_core

6.4 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 (Page 20) are described in detail in the subsequent chapter.

Infrastructure component	Log file path
Cloud Agent	<installation directory>\ppmmashzone\server\logs\CloudAgent.log <installation directory>\ppmmashzone\server\bin\log\agent.log
Cloud Controller	<installation directory>\ppmmashzone\server\acc\log\acc.log

Infrastructure component	Log file path
Cloud search	<installation directory>\ppmmashzone\server\bin\work\work_cloudsearch_<memory model>\defaultLog\log.txt
Elasticsearch	<installation directory>\server\bin\work\work_elastic_<memory model>\elasticsearch\logs\elasticsearch.log
Load balancer	<installation directory>\ppmmashzone\server\bin\work\work_loadbalancer_<memory model>\httpd\logs*.log
Central user management	<installation directory>\ppmmashzone\server\bin\work\work_umcadmin_<memory model>\base\logs*.log
PostgreSQL DB	<installation directory>\ppmmashzone\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 this folder of your PPM installation.

```
<installation directory>\ppmmashzone\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.

7 PPM system messages

7.1 Overview

To output system messages (log output), the PPM server uses the standard Java interface **log4J**. log4J 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 log4J 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, e.g., missing or wrong configurations
ERROR	Errors often leading to cancelation of the active component
FATAL	Errors leading to abortion of the application, e.g., lack of resources
OFF	No log output

The following line defines a logger in the log4J configuration file:

```
<logger name>=<log level>,<appender name 1> , ... , <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
log4j.rootLogger	INFO	none

Logger name	Assigned log level	Inherited log level
log4j.logger.LOG	DEBUG	INFO
log4j.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, e.g.:

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 line defines a logger in the log4J configuration file:

```
log4j.appender.<name>=<class name>
```

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, e.g., %d{HH:mm:ss,SSS}
%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

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

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

7.2 Log file directories

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

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

7.3 System messages (log output)

The PPM server uses the **log4J** interface to output system messages. Each PPM component (e.g., data import, merger, Measure calculator) uses own loggers. The PPM system modules (components) have the following superordinate loggers: **log**, **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 log4j.appender.logFile.File .
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 log4j.appender.errorFile.File . 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 log4j.appender.traceFile.File . 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 log4j.appender.auditFile.File .

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

Module name	Description
LOG.KIC	Calculation of measures and dimensions
LOG.PSC	Dashboard queries
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 (e.g. using runppmconfig)
LOG.ADM	Client administration (e.g. using runppmadmin)
LOG.KG	Key creation for imported process fragments (key generator)
LOG.PRF	Profiler (SA)
LOG.MV	Management views (applet-based)
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
LOG.LOG	General log output
LOG.EAL	Early alert system
LOG.ALY	Irregularity analysis (runppmanalytics)
LOG.MTAB	Minitab
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.CRC	Corba connections
LOG.MON	Online monitoring
LOG.PFS	Statistics of user-specific favorites

Module name	Description
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.PPI	Process import
LOG.EIP	Process import - Event import
LOG.CLC	Process import - Measure calculation
LOG.UPD	Process import - Update
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, i.e., if you set the log level to **WARN**, messages of the **ERROR** and **FATAL** levels are output, as well.

Example

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

```
log4j.logger.LOG.XML=INFO,console
```

- Log levels are not case-sensitive, e.g., **log4j.logger.LOG.XML=error** is the same as **log4j.logger.LOG.XML=ERROR**.
- Incorrect log levels, such as **log4j.logger.LOG.XML=error** are ignored. In this case, the log level of the superordinate logger is used, which is **log4j.logger.LOG=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 **log4j.logger.LOG=INFO**.

TRACE MESSAGE OUTPUT

In contrast to logger modules, trace modules do not have modules. 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**, **FRO**, **RET**, **RETDS**, **QPL**, **SELITY**, **DIM**, **IMP**, **RMI**, **MEM**, **PPI**, **EIP**, **CLC**, **UPD**, **DST**, **QUE**

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

Example

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

```
log4j.logger.TRACE.SRV=TRACE
```

```
log4j.logger.TRACE.ASRV=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 **%X{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 three languages supported by PPM:

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

Example

```
log4j.appender.error.File.layout.ConversionPattern=%p %X{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.

DISTRIBUTE LOG OUTPUT

You can use appenders to send the PPM server system messages to any log servers. These will process the messages further.

WINDOWS EVENT VIEWER

The following file extract shows the extended appender configuration for sending PPM server system messages to the local Windows log server.

```
...
log4j.appender.Logserver=org.apache.log4j.nt.NTEventLogAppender
log4j.appender.Logserver.layout=org.apache.log4j.EnhancedPatternLayout
log4j.appender.Logserver.layout.ConversionPattern=%d | %-5p | %-20c{1} | %m%n
...
```

System messages are displayed in Windows Event Viewer (Computer Management -> System -> Event Viewer).

REMOTE LOG SERVERS

To send PPM server system messages to a remote logging server (e.g., B. Lumbermill or Apache Chainsaw) in your network, you can use the extended appender configuration shown in the following file extract.

```
...
log4j.appender.AuditLogserver=org.apache.log4j.net.SocketAppender
log4j.appender.AuditLogserver.Port=4445
log4j.appender.AuditLogserver.RemoteHost=pcanywhere
log4j.appender.AuditLogserver.layout=org.apache.log4j.EnhancedPatternLayout
log4j.appender.AuditLogserver.layout.ConversionPattern=%X{ppmdate} [%c{1}] %m%n
...
```

7.4 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

Information	Description
Process	Process identifier and description
Component	Component code
User	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

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
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

Component ID	Description
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)
MMV	Management view
CHK	Checks, e.g., planned value deviations
CMP	Aggregation and deletion of instances
CVT	Database converter
INT	Internal application
SHR	Session management
ADT	Adapter
ADM	Admin tool
PFC	Performance Dashboard
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 (Page 20)).

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 measure configuration changes made with 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

Messages differ depending on whether the process tree configuration was changed using the command line program **runppmconfig ... -processtree**, or if it was automatically extended by the import of 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 Performance Dashboard, the query interface, or Management views are not output.

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

Due to the type of connection between Performance Dashboard and PPM, the PC address and user login ID cannot be uniquely determined if the password is changed in Performance Dashboard.

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.

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.

MANAGEMENT VIEWS

Creation, editing, and execution of Management views are not logged.

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 of process instance-independent measures were 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 note that certain processes are and logged only by the master server or one or multiple sub-servers.

7.5 Command line programs

The command line programs write their messages to the console or the log file you specified with the **-protocolfile** argument, see chapter **Common arguments** (Page 45)). 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** arguments) 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** arguments specified the last **no** parameter found is determined.
2. In the order of the **-information**, **-warning**, **-error** arguments 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 "no" 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 1

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
- $\text{MAX}(\text{INFO}, \text{ERROR})$ is ERROR
- Log level = $\text{MIN}(\text{ERROR}, \text{INFO})$ is INFO

The resulting log level is INFO.

Example 2

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
- $\text{MAX}(\text{WARN})$ is WARN
- Log level = $\text{MIN}(\text{WARN}, \text{INFO})$ is INFO

The resulting log level is INFO.

Example 3

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.
- $\text{MAX}(\text{ERROR}, \text{INFO})$ is ERROR
- Log level = $\text{MIN}(\text{ERROR})$ 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 arguments 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 arguments.

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

```
...
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
...
```

8 Globalization

LOCALIZATION

APPLICATION

webMethods Process Performance Manager is available for the following languages, for example:

- German
- English
- French
- Hungarian
- 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** and **English**.

CALENDAR

Currently, PPM supports the Gregorian calendar only.

BIDIRECTIONAL TEXT DISPLAY

The bidirectional display of RTL-based interface texts is supported only with limitations, i.e., interface texts (e.g., 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.

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 SSO integration

With the connection of PPM to SAML2, and Kerberos you can integrate PPM in a single sign-on scenario.

SAML (Security Assertion Markup Language) and Kerberos enable single sign-on (SSO) of a user for multiple applications in parallel. This means after login to one application, the user is automatically authorized to use additional applications.

From version 9.0, PPM supports single sign-on via SAML 2 and Kerberos.

11.1 Configure parameters

To integrate PPM in a single sign-on scenario using SAML or 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

(<https://<host>/umc/help/en/handling/index.htm#52345.htm>).

11.1.1 SAML2 connection

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.

SAML 2 PARAMETERS

Use the following entries in central user administration 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, e.g. '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, e.g. '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).

11.1.2 Kerberos

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

KERBEROS PARAMETERS

Use the following entries in central user administration 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`

12 Security aspects

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

12.1 Safe communication channels

The PPM client communicates directly with load balancer and the PPM server communicates via a TCP/IP network. The following describes the individual steps to secure communication with Load Balancer and the PPM server in more detail.

12.2 Load balancer

By default, the load balancer enables unencrypted access via HTTP (port 4080) and encrypted access via HTTPS/SSL (port 4443). Additional configuration for activating the SSL encryption is not required. By default, a certificate generated by the load balancer is used for HTTPS support. Most browsers do not support it, however, because it does not contain any 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 SSL certificate

This section explains how to create a valid certificate for HTTPS support of the load balancer. The 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 SSL 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 PPM server (Page 41).

Procedure

1. Add both files to a ZIP file.
2. Copy the ZIP file to a location where Cloud Controller can access it.
3. Start Cloud Controller.
4. Stop the load balancer in Cloud Controller.
5. In 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, e.g.: **"c:/temp/lbcert.zip"**.

6. Restart the load balancer.

The SSL 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\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 SSL certificate of the server in the following directory.
<installation directory>\server\bin\work\work_<loadbalancerInstanceId>\httpd\conf
 - Restart load balancer to apply the changes.

12.3 PPM server

In this context, PPM server is the PPM client server receiving the queries for a specific client from any client. The communication between client and server is not secure after installation. The following sections briefly describes how to set up the SSL encryption between client and server.

12.3.1 Generate and sign an SSL keystore

To secure RMI communication in PPM through SSL, you need to create a keystore. Use the Java keytool to set up and manage your own keystore. Detailed information on this topic is available in the documentation **Keytool - Key and Certificate Management Tool** (<http://docs.oracle.com/javase/7/docs/technotes/tools/windows/keytool.html>).

Warning

The following describes the generation of the keystore and the Certificate Signing Requests using simple examples and demo data. The values of the individual parameters used in the examples need to be adapted to their local environment (particularly the information on the "-dname" parameter). The meaning of the individual parameters is described in the documentation **Keytool - Key and Certificate Management Tool**.

Procedure

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

```
keytool
```

```
-keystore ppm_keystore  
-alias ppm  
-keypass changeit  
-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 Certificate Signing Request with the following parameters.

```
keytool
```

```
-certreq  
-alias ppm  
-file ppm_keystore.csr  
-keystore ppm_keystore
```

The program generates two files: a key (extension: **.key**) and the actual certificate request (extension: **.csr**).

3. Submit the content of the file with the **.csr** extension to a certification authority.

It will sign the certificate with a valid root certificate that is usually acknowledged by all browsers and the default Java Runtime Environments.

4. Use the keytool to import the signed certificate into your keystore.

```
keytool
```

```
-import  
-trustcacerts  
-alias ppm
```

```
-file <FileName>  
-keystore ppm_keystore
```

The file **ppm_keystore** can now be used for SSL encryption in PPM.

12.3.1.1 Configure PPM for SSL

You can now use the newly created keystore to switch RMI communication between PPM client and PPM server to SSL encryption. To do so, the following commands need to be executed in Cloud Controller.

- stop ppm_core
- reconfigure ppm_core ssl.enabled="true" ssl.keystorefile="<path to keystore file>"
ssl.keystorepassword="<keystorepassword>"
- start ppm_core

12.3.2 Security measures for the server system

To protect your system from unauthorized access you should grant direct system access (e.g., remote 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.

13 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.

13.1 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 argument character string of all transferred command line arguments
3. Creating a local class path
4. Running the Java Virtual Machine with specification of the class path and the argument 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>\ppmmashzone\server\bin\agentLocalRepo\unpacked\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>\ppmmashzone\server\bin\agentLocalRepo\unpacked\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>\ppmmashzone\server\bin\work\data_ppm\ custom\<client>\bat.**
- Unix: The shell scripts are stored in the directory **<installation directory>\ppmmashzone\server\bin\work\data_ppm\ custom\<client>\bin.**

The batch files (both Windows and Unix) are also referenced in 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>\ppmmashzone\server\bin\work\data_ppm\bin**
- **<installation directory>\ppmmashzone\server**

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

13.2 PPM command line programs

The following command line programs are available:

- `runinitdb` (Page 47)
Initializes the database schema of a PPM client
- `runppmanalytics` (Page 48)
Checks process instances for anomalies during Easy mining.
- `runppmconfig` (Page 50)
Imports configurations into the database schema of a PPM client
- `runppmcpi` (Page 58)
Automatically deletes multiple messages simultaneously during Easy mining.
- `runppmimport` (Page 60)
Controls the subsequent processing of the process instance fragments (merging, typification, measure calculation) imported using **runxmlimport**.
- `runprocessimport` (Page 64)
Imports process-specific data into PPM
- `runppmcompress` (Page 68)
Aggregates imported process instances
- `runppmdelete` (Page 70)
Deletes process instances either for testing or for real
- `runppmadmin` (Page 73)
Executes various administrative functions
- `runppmsendmail` (Page 74)
Sends simple text messages
- `runtableimport` (Page 76)
Extracts data from a file in **XML** or **ZIP** (XML Event Format) format

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

For generally applicable specifications (e. g. PPM user account and client) the same arguments are used in all command line programs. Special arguments 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:

13.3 Common arguments

-VERSION

The version of the PPM software and the database schema are output on the console. Other arguments 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, runprocessimport and other source system adapters	Import process instance fragments
Data calculation	runppmimport and runprocessimport	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 argument **-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** (Page 32).

13.4 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 argument 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

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>\ppmmashzone\server\bin\work\data_ppm\custom\<client>\xml**.

13.5 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>\ppmmashzone\server\bin\agentLocalRepo\unpacked\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**.

13.5.1 runppmanalytics arguments

You can run runppmanalytics with the following specific arguments.

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

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 (Page 48)).

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 SSL-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.

13.6 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.

13.6.1 The runppmconfig argument

-MODE <MODE>

This is where you specify whether the configuration of the PPM component specified by the **<command>** argument 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

command	PPM component
attributenames	Names of the PPM attributes. To export, use the -language <ISO code> argument to specify the relevant language for the PPM attribute names. If this argument 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

command	PPM component
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 (Page 53).)
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

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, e.g., 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 SSL-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.

13.6.2 Export the system configuration

The **-system** parameter in the **runppmconfig** command line program enables almost the entire configuration (except report automation) 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

Execution of the command line

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

extracts the configuration of the **default** client and saves it under **<installation directory>\ppmmashzone\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
```

13.6.3 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.

13.6.4 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, i.e., content of the favorites and access privileges of the shared favorites folders. Specification: Optional

Parameter	Description
[-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

Examples

- 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

13.7 runppmcpu

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 **runppmcpu** 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 **runppmcpu**, you need the **CPI administration** function privilege.

Tip

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

13.7.1 The runppmcpu argument

-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 1

```
runppmcpu ... -deletecpimessages --excluding 5w --mode mark_deleted
```

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

Example 2

```
runppmcpu ... -deletecpimessages --excluding 5w --mode delete
```

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

```
runppmcpu ... -deletecpimessages --excluding 5w --mode delete
```

All CPI messages older than 5 weeks are deleted.

-MESSAGE_TYPE <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 1

```
runppmcpu ... -deletecpimessages -messagetype comment --mode delete
```

All comments will be deleted.

Example 2

```
runppmcpu ... -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

```
runppmcpu ... -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 SSL-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 `runppmcpu`, the number of deleted CPI messages is output.
- After calling `runppmcpu`, the owner of the CPI message (`CURRENT_OWNER`) is set to the user specified in the `-user` argument.
- The `-excluding` parameter is exact to the day at the most (e.g., `-excluding 1d`). It is impossible to specify hours (e.g., `-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 `runppmcpu`, master and sub-servers are considered stand-alone servers. No deletion will take place.

13.8 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.

13.8.1 The runppmimport argument

-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** (Page 63)).

-REINITANALYSISSEVER {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** argument, 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** (Page 63)). 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.

-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**.

Example

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>
<analyzewizparam 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>
<filter>
  <criteria name="MATERIAL">Material</criteria>
  <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>

```

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
```

-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>\ppmmashzone\server\bin\work\data_ppm\recovery\<client> after a successful PPM import.

13.8.2 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 (argument **-typifier new**) or a recalculation (argument **-keyindicator new**).
2. Reinitialize the analysis server (argument **-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** argument 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.

13.9 runprocessimport

Use the **runprocessimport** command line program to start the process import. The process import enables prompt analysis of imported processes in PPM.

The program is stored in the directory **<PPM installation directory>\server\bin\agentLocalRepo\unpacked\ppm-client-run-prod-<version>-runnable.zip\ppm\bin**.

Run the program with the following parameters in the command line. Parameters in brackets [] are optional.

```
runprocessimport -user <user name> -password <password>
    (-datasource <file>) | (-datasourcelist <file>) | (-recoverimport)
    [-client <name>]
    [-language <ISO code>]
    [-recoveryfile {yes|no}]
    [-version]
    [protocoloptions]
```

Process import is another method of importing process-dependent data into PPM in addition to the standard XML import (runxmlimport) and PPM imports (runppmimport (Page 60)). Process import enables the import of almost all process-related data sources, except for data in the **GRAPH** format. You can import data in the formats **EVENT**, **JDBC**, **CSV**, **SAP**, and **Universal Messaging**. Process import includes most functions of the PPM import and XML import, e.g., event import, key generation, merge, as well as attribute, measure, and dimension calculation for process hierarchies, etc.

The following functions of the PPM import and XML import are not supported by process import.

- Graph import
- Import of entire EPCs
- Shared fragments import
- Anonymization (such as the **anonymizing** mode of the PPM import)
- Retypification (such as the **typifier new** mode of the PPM import)
- Recalculation (such as the **keyindicator new** mode of the PPM import)
- Reinitialization of the analysis server (such as the **reinitanalysisserver** mode of the PPM import)
- Auto-attributes for runtime data (new attributes must be created at the time of configuration)

The command line program starts with proprietary VM settings specified in the file `_vmparam.bat`. The file is stored under `<installation directory>\ppmmashzone\server\bin\work\data_ppm\config\<client>\bin`. You configure the values using CTK.

Default values are:

- `LOCAL_PPM5_VM_MIN_MEM_RT_PROCESSIMPORT=1024`
- `LOCAL_PPM5_VM_MAX_MEM_RT_PROCESSIMPORT=4096`

Example

```
runprocessimport -client e2ppm -user system -password manager -datasource  
C:\SoftwareAG\ppmmashzone\server\bin\work\data_ppm\custom\e2ppm\xml\datasource\dat  
asource.xml
```

13.9.1 runprocessimport arguments

-DATASOURCE <FILE> | -RECOVERIMPORT | -DATASOURCELIST <FILE>

The parameters **datasource**, **datasourcelist**, and **recoverimport** are alternative import modes. **datasource** specifies the configuration file of the data source for process import. **datasource** starts a new process import. **recoverimport** starts the process import in recovery mode and completes a previously aborted import process.

Multiple data sources can be imported simultaneously by means of the **-datasourcelist** argument. The import corresponds to multiple importing using the **-datasource** argument.

See chapter Import multiple data sources (Page 67).

-RECOVERYFILE {YES|NO}

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

13.9.2 Import settings

By means of various settings you can control process import behavior.

You enter the settings in the file **ProcessImport_settings.properties** under `<installation directory>\ppmmashzone\server\bin\work\data_ppm\config\<client>`.

The file contains settings for the configuration of PPM server, analysis server, and the data store (memory and hard drive) used for saving data objects (EPCs, keys, IDs, etc.) during the import process.

The various process import components use multiple data stores for data exchange. They are cache instances with continuous lifetime of data objects and the ability to swap elements out to the hard drive if main memory does not suffice. Users can specify the size of available memory

(MAX_BYTES_LOCAL_HEAP) and hard drive memory (MAX_BYTES_LOCAL_DISK). They can also specify the location (DISK_STORE_PATH) where temporary data is to be saved.

The individual variables are described in the following table.

Key	Example	Description
MAX_BYTES_LOCAL_HEAP	3072	Maximum size of available memory in MB
MAX_BYTES_LOCAL_DISK	32768	Maximum size of available hard drive memory in MB
DISK_STORE_PATH	<installation directory>\ppmmashzone\server\bin\work\data_ppm\diskstore	Path to temporary data store

13.9.3 Import logging

Four log modules are available for process import, one general process import module, and three for other components.

The log levels of all modules are configured in the file **Server_Log_settings**. The log output of the modules is written to the corresponding import log file.

Log modules

```
#MODULE_PPM_PROCESS_IMPORT
log4j.logger.LOG.PPI=INFO
#MODULE_PPM_PROCESS_IMPORT_EVENT_IMPORT
log4j.logger.LOG.EIP=INFO
#MODULE_PPM_PROCESS_IMPORT_CALCULATION
log4j.logger.LOG.CLC=INFO
#MODULE_PPM_PROCESS_IMPORT_UPDATE
log4j.logger.LOG.UPD=INFO
```

Trace modules

```
#TRACEMODULE_PPM_PROCESS_IMPORT
log4j.logger.TRACE.PPI=OFF
#TRACEMODULE_PPM_PROCESS_IMPORT_EVENT_IMPORT
log4j.logger.TRACE.EIP=OFF
#TRACEMODULE_PPM_PROCESS_IMPORT_CALCULATION
log4j.logger.TRACE.CLC=OFF
#TRACEMODULE_PPM_PROCESS_IMPORT_UPDATE
log4j.logger.TRACE.UPD=OFF
#TRACEMODULE_PPM_PROCESS_IMPORT_DATASTORE
log4j.logger.TRACE.DST=OFF
#TRACEMODULE_PPM_PROCESS_IMPORT_QUEUE
log4j.logger.TRACE.QUE=OFF
```

13.9.4 Import multiple data sources

Multiple data sources can be imported simultaneously by means of the **-datasourcelist** **<datasource>** argument (see chapter `runprocessimport` argument (Page 65)). To import multiple data sources a configuration file is available in which you can specify a list of data sources. During a process import, the data of the data sources specified in the configuration file are imported consecutively, just as if the process import was called consecutively multiple times using the **-datasource <datasource>** argument. The sequence of the data source import is specified in the configuration file.

The reading/importing of data sources of the **GRAPH** type is not supported for process imports using the **-datasourcelist** argument.

The configuration file must match the **datasourcelist.dtd** DTD, which looks as follows.

```
<!ELEMENT datasourcelist (datasource*)>
<!ELEMENT datasource (#PCDATA)>
<!ATTLIST datasource
  name ID #REQUIRED
  type (EVENT | GRAPH | MYSAP | JDBC | CSV | NIRVANA ) #REQUIRED
>
```

You need to specify an ID for each data source, i.e., the name of the data source also used in CTK, the data source type, and the path to the data source file.

An XML file can look as follows.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE datasourcelist SYSTEM "datasourcelist.dtd">
<datasourcelist>
  <datasource name="CLEARING"
type="EVENT">M:/SoftwareAG/ppmmashzone/server/bin/work/data_ppm/custom/umg_en/xml/CLEARING/CLEARING.xml</datasource>
  <datasource name="ACCOUNTING" type="GRAPH">
M:/SoftwareAG/ppmmashzone/server/bin/work/data_ppm/custom/umg_en/xml/ACCOUNTING/ACCOUNTING.xml</datasource>
  <datasource name="BILLING" type="MYSAP">
M:/SoftwareAG/ppmmashzone/server/bin/work/data_ppm/custom/umg_en/xml/BILLING/BILLING.xml</datasource>
  <datasource name="SHIPMENT" type="JDBC">
M:/SoftwareAG/ppmmashzone/server/bin/work/data_ppm/custom/umg_en/xml/SHIPMENT/SHIPMENT.xml</datasource>
  <datasource name="MATERIAL_DOCUMENT"
type="CSV">M:/SoftwareAG/ppmmashzone/server/bin/work/data_ppm/custom/umg_en/xml/MATERIAL_DOCUMENT/MATERIAL_DOCUMENT.xml</datasource>
  <datasource name="PURCHASE_PROCESS" type="NIRVANA">
M:/SoftwareAG/ppmmashzone/server/bin/work/data_ppm/custom/umg_en/xml/PURCHASE_PROCESS/PURCHASE_PROCESS.xml</datasource>
</datasourcelist>
```

ERROR BEHAVIOR

If the import is called via a valid configuration file that does not contain any data sources the import ends without outputting an error message.

If the import is called with a configuration file containing multiple data sources and if an error occurs during the import of a data source the import will be completely canceled.

If an error occurs during the import of at least one data source from a configuration file, which has so far lead to an exit error status (i.e., "-1") during the import of individual data sources, the import using that configuration file will also return this exit error state.

13.10 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.

13.10.1 The runppmcompress argument

-PS <FILE NAME>

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

-MODE <TEST|EXECUTE>

test: This argument 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

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>
  <analyzewizparam 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>
        </datarange>
      </filteritem>
    </filter>
  </kiquery>
</paramset>
```

```

    </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>
  <criteria name="PROCESSTYPE">Process type</criteria>
  <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>

```

Executing the command line

`runppmcompress -user system -password manager -ps paramset.xml -mode execute`
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>\ppmmashzone\server\bin\work\data_ppm\recovery\<client>`.

13.11 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.

13.11.1 The `runppmdelete` argument

-PS <FILE NAME>

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

-MODE <TEST|EXECUTE>

test: This argument 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

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>
```

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>\ppmmashzone\server\bin\work\data_ppm\recovery\<client>.

13.12 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.

13.12.1 The runppmadmin argument

-KILLSESSION <SESSION ID>

You use this argument to log off the session of a user who is logged in. Only user interface sessions can be logged off. 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 recalculation of the caches used in the system. Any existing data in the caches is overwritten.

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 **-fillcache** option.

-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

This information is intended for experienced 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.

-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>\ppmmashzone\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**.

13.13 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 arguments, supports only the **-protocoloptions** parameter.

13.13.1 The runppmsendmail argument

-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, e.g., in a ZIP archive.

13.14 **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 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. Software AG will not provide any support.

14.1 System integration

Q1.01	If you install PPM under Windows Vista, Windows Server 2008, or Windows 7 the installation is incomplete. 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?
	<p>The only network protocol supported by PPM is TCP/IP. A standard PPM system with http access (e.g., the demo scenario supplied with the product) is made up of the following services and requires the following resources:</p> <ul style="list-style-type: none"> ▪ PPM RMI registry (default port 17500) ▪ PPM RMI server (per client, default port from 17501 upwards) ▪ PPM analysis server (per client, default port from 17502 upwards) ▪ http server for PPM <p>If you also want to use Performance Dashboard, the following additional system resources are required:</p> <ul style="list-style-type: none"> ▪ PPM Corba registry (default port 17590) ▪ PPM Corba server (per client, default port from 17591 upwards) ▪ Web application server (WAS) <p>The default ports specified in brackets are automatically entered by the installation program.</p>
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)?
	No, the Java RMI protocol used by PPM does not support NAT. Therefore, running PPM in a public cloud environment is impossible if the system is to be accessed from an external location.
Q1.05	Can https be used to access PPM?
	Yes. The load balancer supports https since PPM 9.0 by default. However, you need to note that a self-signed certificate generated during installation is used. This leads to error messages in most browsers. See also chapter https support (Page 40)
Q1.06	How can data transfer between PPM server and user interface be encrypted (SSL encryption)?
	You can use Cloud Controller to configure the SSL encryption (reconfigure ppm_core ssl.enabled="true" ssl.keystorefile="<path to keystore file>" ssl.keystorepassword="<keystorepassword>"). 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 \\ If you want to use SSL encryption between PPM server and client, you need to use a separate keystore. Use the Java keytool to set up and manage your own keystore. For detailed information on this topic, please refer to the keytool documentation keytool - Key and Certificate Management Tool (http://docs.oracle.com/javase/7/docs/technotes/tools/windows/keytool.html).
Q1.07	What is the server mode of 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?
	<p>Possible causes:</p> <ul style="list-style-type: none"> ▪ The client server port is blocked by a firewall. ▪ 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.
Q1.11	The connection to the PPM server fails. Example message in log output:
	<pre>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</pre>
	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, e.g., 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?
	<p>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.</p> <p>Detailed information on using central user management is available in the online</p>

	help.
Q1.13	<p>What is important about environment variables?</p> <p>Since PPM 9.0, environment variables have no longer been supported.</p>
Q1.14	<p>What is a client-specific configuration file?</p> <p>For each PPM client installed, a directory of the same name exists in the subdirectory <installation directory>\ppmmashzone\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.</p>
Q1.15	<p>What is a system-wide configuration file?</p> <p>The system-wide configuration files specify global settings for the entire PPM system and are saved under <installation directory>\ppmmashzone\server\bin\work\data_ppm\config. The system-wide configuration files Registry_settings.properties and Help_settings.properties in the config directory of the PPM installation must have the same content as those in the clientjars directory on the Web server.</p>
Q1.16	<p>How do I change the TCP/IP ports of the PPM services?</p> <p>You change the TCP/IP ports using Cloud Controller.</p> <p>Further information on the individual commands is available in the help of the Cloud Controller command line. You can view the help by entering the command help in the command line.</p> <ul style="list-style-type: none"> ▪ PPM registry reconfigure ppm_core ▪ PPM analysis server reconfigure <client name>_as ▪ PPM server reconfigure <client name>_cs ▪ PPM Corba registry reconfigure ppm_core ▪ PPM Corba server reconfigure <client name>_cs

	If you change the Corba registry port, you must repeat this change in all login modules of the file jaas.config (corbaurl property) and in all _conf files of your Performance Dashboard and the query interface (corbaurl property in each case) (Cloud Controller: reconfigure ppm_web with the respective client name).
Q1.17	What does the key CorbaServerActivationPort mean in the configuration file Registry_settings.properties?
	The Corba registry server uses this socket for control at runtime. This feature cannot be disabled.
Q1.18	Since the update to Chrome Web browser version 45.x and higher, the PPM front-end cannot be loaded as an applet any more. What is the reason for this, and is there a solution?
	<p>There are two possible causes.</p> <ul style="list-style-type: none"> ▪ 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.
Q1.19	When installing PPM, errors occur when the required Visual C++ Redistributable packages are installed. What can be done?
	<p>There are two possible causes.</p> <ul style="list-style-type: none"> ▪ 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.
Q1.20	I use http proxy tunneling with an existing client firewall. Suddenly, this does not work any more. What is the cause, and is there a solution for it?

In PPM version 9.10 and higher, a new Java version **JDK 1.8.0.51** is used, and the new version no longer supports this functionality in the default configuration (see also http://bugs.java.com/bugdatabase/view_bug.do?bug_id=8023862). However, the functionality can still be reactivated until further notice. To do this, you simply need to complete the following setting in the PPM specific **PPM_PROXY** environment variable:

```
-Djava.rmi.server.disableHttp=false
```

Example

```
PPM_PROXY=-Dhttp.proxyHost=proxy.intranet.ourcompany.com -  
Dhttp.proxyPort=8080 -Djava.rmi.server.disableHttp=false
```

However, since the functionality has been discontinued, it must be assumed that it will not be available in the next version.

14.1.1 Database connection

Q2.01	What settings are important for the DB schema used?
	<p>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.</p> <p>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.</p>
Q2.02	How do I change database access data?
	<p>Edit the client-specific configuration file <installation directory>\ppmmashzone\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.</p> <p>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.</p>
Q2.03	When starting the PPM server, the database error <code>java.lang.ClassNotFoundException</code> is displayed. What is the reason for

	this?
	<p>The required database drivers must be copied manually after PPM installation and before client installation to the directory <installation directory>\ppmmashzone\server\bin\work\data_ppm\drivers.</p> <p>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>).</p> <p>For further information about the database connection, please refer to the PPM Database Systems user guide.</p>
Q2.04	<p>When starting the PPM server, the message DB error: Connection not established. (1006) [17002/null] (1034) is displayed. Why?</p>
	<p>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.</p>
Q2.05	<p>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?</p>
	<p>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.</p>
Q2.06	<p>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?</p>
	<p>The database and JDBC driver version for the IBM DB2 database do not match.</p>
Q2.07	<p>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?</p>

	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: <ul style="list-style-type: none"> ▪ Oracle <pre>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).</pre> <ul style="list-style-type: none"> ▪ IBM DB2 <pre>I: ...: [SRV] Database version used: SQL09013. I: ...: [SRV] JDBC driver used: IBM DB2 JDBC Universal Driver Architecture (3.4.65).</pre> <ul style="list-style-type: none"> ▪ MS SQL Server <pre>I: ...: [SRV] Database version used: 8.00.2039. I: ...: [SRV] JDBC driver used: Microsoft SQL Server 2005 JDBC Driver (1.1.1501.101).</pre>

14.1.2 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: <pre>java -showversion -jar sapjco.jar</pre> 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

	<p>Extractor, PPM Customizing Toolkit, and PPM if you want SAP transaction calls to be possible from the process instance table).</p> <p>The file must be stored under <installation directory>\ppmmashzone\server\bin\work\data_ppm\drivers.</p>

14.1.2.1 Using Content Packages

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

14.2 System administration

Q5.01	<p>A PPM user has forgotten the password. How do I change the password for an external PPM user?</p>
	<p>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.</p>
Q5.02	<p>How do I block particular users from the system?</p>
	<p>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.</p>

Q5.03	<p>How do I determine which users are currently logged on to the system?</p>
	<p>In a command prompt for the relevant client, run the runppmadmin command with the -showusers option.</p> <p>Example:</p> <pre>C:\>pushd SoftwareAG\ppmmashzone\server\bin\agentLocalRepo\.unpacked\ppm-client-run-prod-95.1.0-RC8-trunk-20130404.122823-4-runnable.zip\ppm\bin D:\SoftwareAG\ppmmashzone\server\bin\agentLocalRepo\.unpacked\ppm-client-run-prod-95.1.0-RC8-trunk-20130404.122823-4-runnable.zip\ppm>runppmadmin -user system -password manager -showusers</pre> <p>The program displays a table showing you the sessions of all users currently logged into the PPM system on the default client.</p>
Q5.04	<p>Why does system initialization fail using runinitdb?</p> <p>Log output includes:</p> <pre>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]</pre>
	<p>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 /.</p>
Q5.05	<p>What emergency user exists in the PPM system?</p>
	<p>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.</p>
Q5.06	<p>What authentications are performed when a PPM user runs a PPM command line program (e.g., runppmconfig)?</p>
	<p>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</p>

	<p>based on the following components:</p> <ul style="list-style-type: none"> ▪ Function privileges ▪ Data access privileges ▪ Process access privileges ▪ Data visibility privilege (via user group association) <p>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:</p> <pre>C:\SoftwareAG\ppmmashzone\server\bin\agentLocalRepo\.unpacked\ppm-client-run-prod-<version>-runnable.zip\ppm\bin>runppmconfig -user vogel -password vogel -mode export -users user.xml -client umg_en</pre> <p>As the user PARKER does not have the Configuration import function privilege, the following message appears in the command line:</p> <pre>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</pre> <p>No data exported. User PARKER must first obtain the required function privilege from the PPM system administrator.</p>
Q5.07	<p>When running PPM command line programs, does the log indicate restricted data access privileges of a PPM user (e.g., runppmcompress)?</p>
	<p>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.</p> <p>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:</p> <pre>C:\SoftwareAG\ppmmashzone\server\bin\agentLocalRepo\.unpacked\ppm-client-run-prod-<version>-runnable.zip\ppm\bin>runppmcompress -user vogel -password vogel -client umg_en -mode execute -ps ps.xml</pre> <p>Of the 68 process instances contained in the aggregation paramset (ps.xml), the user PARKER only has data access privileges for the</p>

	<p>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:</p> <pre>[...] 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).</pre>
Q5.08	<p>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?</p>
	<p>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.</p> <p>Example:</p> <pre>C:\SoftwareAG\ppmmashzone\server\bin\agentLocalRepo\.unpacked\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. [...]</pre>
Q5.09	<p>Why does a PPM user not see any data in the PPM user interface despite appropriate process access and function privileges?</p>
	<p>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).</p>
Q5.10	<p>How can restricted process access privileges affect actions carried out using the command line?</p>

	<p>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:</p> <pre>C:\SoftwareAG\ppmmashzone\server\bin\agentLocalRepo\.unpacked\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.</pre> <p>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.</p>
Q5.11	<p>When importing data, I encounter OutOfMemory messages. What can I do?</p>
	<p>Allocate more memory to the respective program. You can change the memory data in PPM Customizing Toolkit via Edit client (calling the client installation) on the Server settings page under JVM parameters of server programs > Configure JVM parameters.</p> <p>Adapt the maximum system memory for runxmlimport, e.g., 1024 MB for a maximum of 1 GB system memory.</p> <p>Subsequently, you must restart the system service of the PPM server (if set up) or the PPM server.</p>
Q5.12	<p>How do I increase the system memory for the PPM server?</p>

	<p>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.</p> <p>Subsequently, you must restart the system service of the PPM server (if set up) or the PPM server.</p>
Q5,121	<p>How do I adjust system memory for the analysis server?</p> <p>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.</p> <p>You must restart the analysis server afterwards.</p>
Q5.13	<p>Which periodical tasks exist for a PPM system?</p> <p>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:</p> <ul style="list-style-type: none"> ▪ Back up the client's database schema ▪ Back up the client configuration (web, config, and custom directories under <installation directory>\ppmmashzone\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 <p>For further information, please refer to the documentations PPM Database Systems and PPM Data Import.</p>

14.2.1 PPM command line programs: Log output

Q6.01	<p>How can log output be configured?</p> <p>The log options information warning error in the command line programs can be configured in any combination (see chapter Command line programs (Page 32)). 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).</p>
-------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Q6.02	Can log output be routed to a file?
	The instruction -logfile <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, e.g., -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>\ppmmashzone\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 (Page 20)).
Q6.06	Are changes to the PPM configuration using the interface logged?
	<p>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).</p> <p>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):</p> <pre>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.</pre>

14.2.2 Inconsistent data pools

Q7.01	How to restore consistency in the data pools of analysis server and PPM server
	<p>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.</p> <p>Open a DOS command prompt, switch to the directory <PPM installation directory>\server\bin\agentLocalRepo\unpacked\ppm-client-run-prod-<version>-runnable.zip\ppm\bin, and execute the following command line:</p> <pre>runppmimport -user <ppm user> -password <password> -client <ppm client> -reinitanalysisserver auto</pre> <p>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.</p>