



webMethods Process Performance Manager SYSTEMARCHITEKTUR

Version 9.12

Oktober 2016

Dieses Dokument gilt für PPM ab Version 9.12. Hierin enthaltene Beschreibungen unterliegen Änderungen und Ergänzungen, die in nachfolgenden Release Notes oder Neuauflagen bekanntgegeben werden.

Urheberrechtlich geschützt © 2000 - 2016 Software AG, Darmstadt, Deutschland und/oder Software AG USA Inc., Reston VA, USA und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.

Der Name Software AG und die Namen der Software AG Produkte sind Marken der Software AG und/oder Software AG USA Inc., einer ihrer Tochtergesellschaften oder ihrer Lizenzgeber. Namen anderer Gesellschaften oder Produkte können Marken ihrer jeweiligen Schutzrechtsinhaber sein. Genaue Informationen über die geschützten Marken und Patente der Software AG und ihrer Tochtergesellschaften sind veröffentlicht unter <http://softwareag.com/licenses>.

Die Nutzung dieser Software unterliegt den Lizenzbedingungen der Software AG. Diese Bedingungen sind Bestandteil der Produktdokumentation und befinden sich unter <http://softwareag.com/licenses> und/oder im Wurzelverzeichnis des lizenzierten Produkts.

Diese Software kann Teile von Software-Produkten Dritter enthalten. Urheberrechtshinweise, Lizenzbestimmungen sowie zusätzliche Rechte und Einschränkungen dieser Drittprodukte können dem Abschnitt „License Texts, Copyright Notices and Disclaimers of Third Party Products“ entnommen werden. Diese Dokumente enthalten den von den betreffenden Lizenzgebern oder den Lizenzen wörtlich vorgegebenen Wortlaut und werden daher in der jeweiligen Ursprungssprache wiedergegeben. Für einzelne, spezifische Lizenzbeschränkungen von Drittprodukten siehe PART E der Legal Notices, abrufbar unter dem Abschnitt „License Terms and Conditions for Use of Software AG Products / Copyrights and Trademark Notices of Software AG Products“. Diese Dokumente sind Teil der Produktdokumentation, die unter <http://softwareag.com/licenses> oder im Verzeichnis der lizenzierten Produkte zu finden ist.

Inhalt

1	Textkonventionen	1
2	Allgemeines	2
3	Übersicht	3
3.1	PPM-Mandanten-Konzept	3
3.2	Systemkomponenten	3
3.2.1	Infrastruktur	4
3.2.2	PPM-Komponenten	5
3.2.3	Datenbankserver (DBMS)	7
3.3	Grundlegendes zur In-Memory-Technologie	8
4	PPM-Netzwerk	9
4.1	Netzwerkprotokolle	9
4.2	Laden des PPM-Frontend	11
4.2.1	HTTPS-Protokoll	12
4.3	Datenkommunikation PPM-Frontend und PPM-Mandantenserver	12
4.4	Datenkommunikation PPM--Serverkomponenten	12
4.5	Betriebsarten der Datenübertragung	14
4.5.1	Komprimierte Datenübertragung (Standard)	14
4.5.2	Direkte RMI-Datenübertragung	14
4.5.3	Verschlüsselte Datenübertragung	15
5	Verteilte PPM-Server-Systeme	16
5.1	Zugriff auf ein verteiltes System	16
5.2	Datenimport in verteilten Systemen	17
5.3	Aufbau eines verteilten Systems	17
5.4	Skaliertes System initialisieren	18
5.5	Skaliertes System erweitern	19
5.6	Einschränkungen	19
5.6.1	Unterstützte Szenarien	19
5.6.2	Anfragearten	20
6	Konfiguration des PPM-Systems	21
6.1	PPM-Konfigurationsdateien	21
6.1.1	Aufbau	21
6.1.2	Globale Konfigurationsdateien	21
6.1.3	Mandantenspezifische Konfigurationsdateien	23

1 Textkonventionen

Im Text werden Menüelemente, Dateinamen usw. folgendermaßen kenntlich gemacht:

- Menüelemente, Tastenkombinationen, Dialoge, Dateinamen, Eingaben usw. werden **fett** dargestellt.
- Eingaben, über deren Inhalt Sie entscheiden, werden **<fett und in spitzen Klammern>** dargestellt.
- Einzeilige Beispieltex te werden am Zeilenende durch das Zeichen ↵ getrennt, z. B. ein langer Verzeichnispfad, der aus Platzgründen mehrere Zeilen umfasst.
- Dateiauszüge werden in folgendem Schriftformat dargestellt:

Dieser Absatz enthält einen Dateiauszug.

2 Allgemeines

Das Handbuch beschreibt die Architektur und den Betrieb von webMethods Process Performance Manager, kurz PPM. Es zeigt die Zusammenhänge und Datenflüsse zwischen den PPM-Systemkomponenten auf und stellt verschiedene Installationsarten vor. Darüber hinaus dokumentiert es die Architektur und die Funktionsweise der Software sowie die Installationseinstellungen und deren Auswirkungen.

Der technische Administrator des PPM-Systems erhält Grundwissen und Konfigurations-Know-how, das ihn bei der Inbetriebnahme und Anpassung an unterschiedliche Systemumgebungen und der Wartung unterstützt.

Das Handbuch ersetzt keine Anwender- oder Customizing-Schulung. Sondern es ist eine Referenz, die ergänzende Hinweise zu weiteren PPM-Handbüchern und der PPM-Online-Hilfe enthält.

3 Übersicht

webMethods Process Performance Manager ist ein Werkzeug zur Analyse von real existierenden Prozessen. Die Daten der Arbeitsschritte dieser Prozesse werden aus Anwendungssystemen ausgelesen und zu Prozessinstanzen zusammengefügt. Anschließend werden Kennzahlen für diese Prozessinstanzen berechnet. PPM ermöglicht zeitnah einen umfassenden Überblick über die Prozesse eines Unternehmens und unterstützt den Benutzer beim Erkennen von Schwachstellen. Als Repository verwendet PPM ein RDBMS, das alle Konfigurationen und Daten speichert. PPM ist als Client-Server-Applikation in Java entwickelt.

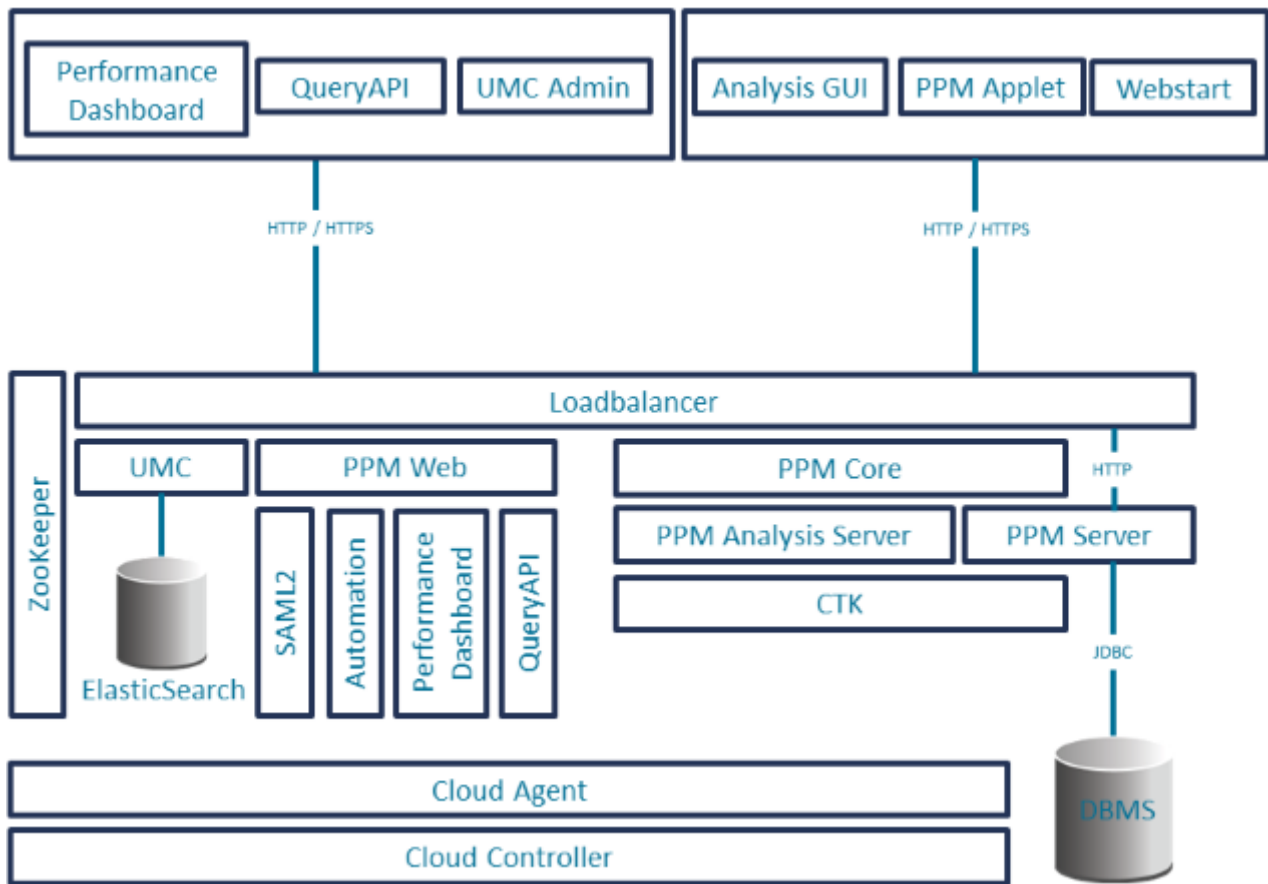
3.1 PPM-Mandanten-Konzept

Das PPM-Mandantenkonzept ermöglicht die unabhängige Verwaltung verschiedener Datenbestände innerhalb einer PPM-Installation. Jedem PPM-Mandanten ist ein unabhängiges Datenbankschema zugeordnet. Beim Anmelden am PPM-System gibt der Benutzer den gewünschten Mandanten an. Die Datenbankschemata können auf mehreren Datenbankservern verteilt sein.

PPM-Mandanten werden interaktiv mit Hilfe des PPM Customizing Toolkit (kurz CTK) angelegt. Detaillierte Informationen zu diesem Thema erhalten Sie im Handbuch **PPM Customizing Toolkit**.

3.2 Systemkomponenten

Kernkomponenten von PPM sind der PPM-Server und der PPM-Analyseserver. Deren Funktion wird sichergestellt durch weitere Infrastruktur-Komponenten. Der Zugriff auf die Funktionalitäten des PPM-Systems erfolgt mittels einer grafischen Benutzeroberfläche. Das nachfolgende Schaubild gibt einen Überblick über alle Komponenten des PPM Systems.



3.2.1 Infrastruktur

Die Infrastruktur besteht aus den Komponenten:

- Cloud Agent,
- Cloud Controller,
- Apache ZooKeeper,
- Apache Load Balancer,
- Elasticsearch
- Zentrale Benutzerverwaltung (User Management).

Bei der Installation wählt der Benutzer das Speichermodell, (S, M oder L), dass für die beinhalteten Dienste verwendet wird. Der Dienstname enthält das gewählte Speichermodell als Suffix.

3.2.1.1 Cloud Agent

Der Cloud Agent ist ein Service, der das Installieren, Konfigurieren, Starten, Stoppen und Überwachen von Produkt- und Infrastruktur-Komponenten erlaubt. Der Cloud Agent wird während der Installation eingerichtet und startet automatisch. Unter Windows wird der Cloud Agent als Dienst mit dem Namen **Software AG PPM/MashZone <version>** eingerichtet. Unter

Linux kann der Cloud Agent nach der Installation als Daemon-Prozess eingerichtet werden. Dazu sind jedoch root-Rechte erforderlich.

3.2.1.2 Cloud Controller

Der Cloud Controller ist ein Kommandozeilenprogramm, das zum Senden von Kommandos an einen aktiven Cloud Agent genutzt wird.

3.2.1.3 Apache Load Balancer

Der Load Balancer ist den Web-Applikationsservern vorgeschaltet und verteilt die eingehenden Anfragen auf die jeweiligen Web-Server und Applikationen.

Der Apache Load Balancer ist im Cloud Agent als Infrastruktur-Komponente unter dem Namen **loadbalancer_<memorymodel>** eingerichtet.

3.2.1.4 Apache ZooKeeper

Produkt- und allgemeine Infrastrukturkomponenten nutzen den Apache ZooKeeper als zentralen Registrierungsdienst. Der Apache ZooKeeper speichert die Konfigurationsinformation und die Namen der registrierten Komponenten zentral. Er erlaubt verteilte Synchronisation und schafft Gruppendienste.

Der Apache ZooKeeper ist im Cloud Agent als Infrastruktur-Komponente unter dem Namen **zoo_<memorymodel>** eingerichtet.

3.2.1.5 ElasticSearch

ElasticSearch ist eine Echtzeit-Such- und Analyse-Applikation. In der beschriebenen Architektur wird sie zum Speichern von Benutzerverwaltungs- und Revisionsdaten genutzt.

Elasticsearch ist im Cloud Agent als Infrastruktur-Komponente unter dem Namen **elastic_<memorymodel>** eingerichtet.

3.2.1.6 Zentrale Benutzerverwaltung (User Management)

Die webbasierte, zentrale Benutzerverwaltung **User Management** verwaltet die Benutzer, Benutzergruppen und Produktlizenzen. Die Benutzerdaten können in der Komponente **User Management** von Benutzern mit dem Funktionsrecht **Administrator** verwaltet werden. Um PPM zu registrieren, muss nach der Installation die Produktlizenz in die zentrale Benutzerverwaltung importiert werden.

Die Zentrale Benutzerverwaltung ist im Cloud Agent als Infrastruktur-Komponente unter dem Namen **umcadmin_<memorymodel>** eingerichtet.

3.2.2 PPM-Komponenten

3.2.2.1 PPM-Benutzeroberflächen

PPM Applet, Analysis GUI, Performance Dashboard und QueryAPI bilden das PPM-Frontend. Mit dem PPM-Frontend analysiert und visualisiert der Anwender die zuvor berechneten Prozesskenndaten.

3.2.2.2 PPM-Mandantenserver

Jeder PPM-Mandantenserver besteht aus einem PPM-Server und einem dazu gehörigen Analyseserver. Beide Server sind als eigenständige Dienste implementiert und können somit unabhängig voneinander gestartet und beendet werden. Der PPM-Server ist Schnittstelle für das PPM-Frontend und koordiniert den Zugriff auf den Analyse- und Datenbankserver.

Der Analyseserver hält alle Prozessdaten für eine performante Analyse in einer In-Memory-Struktur vor (siehe Grundlegendes zur In-Memory-Technologie (Seite 8)).

PPM-Server und Analyseserver sind im Cloud Agent als Produktkomponenten unter den Namen `<client>_cs` und `<client>_as` eingerichtet.

3.2.2.3 PPM Web

Bei PPM Web handelt es sich um einen Applikationsserver, der die folgenden Dienste anbietet:

- SAML2 Login
- Automation
- Performance Dashboard
- QueryAPI

Außerdem liefert PPM Web als Web-Server das PPM Applet.

PPM Web ist im Cloud Agent als Produktkomponente unter dem Namen **PPM_web** eingerichtet.

Der externe Zugriff auf die von PPM Web bereitgestellten Dienste erfolgt über den Load Balancer (siehe Apache Load Balancer (Seite 5))

3.2.2.4 PPM Core

PPM Core besteht aus RMI- und CORBA-Registry. Diese Registries sind Namensdienste, an denen die Server ihre Dienste anmelden.

Die PPM-Serverkomponenten (Analyseserver und Kommandozeilenprogramme) nutzen die RMI-Registry, um die Adressen der PPM-Server zu erfragen.

PPM Core ist im Cloud Agent als Produktkomponente unter dem Namen **PPM_core** eingerichtet.

3.2.2.5 Customizing Tool Kit (CTK)

Das CTK dient der Verwaltung der Mandanten und deren Konfigurationen. Neben der Systemkonfiguration, wie z. B. Spracheinstellungen, Speichereinstellungen, Datenquellen und interne Zugangsdaten, können eigene Auftragsautomatisierungen definiert werden, sowie die in

der Analyse zur Verfügung stehenden Prozesstypen, Kennzahlen, Dimensionen und Attribute bearbeitet werden.

3.2.3 Datenbankserver (DBMS)

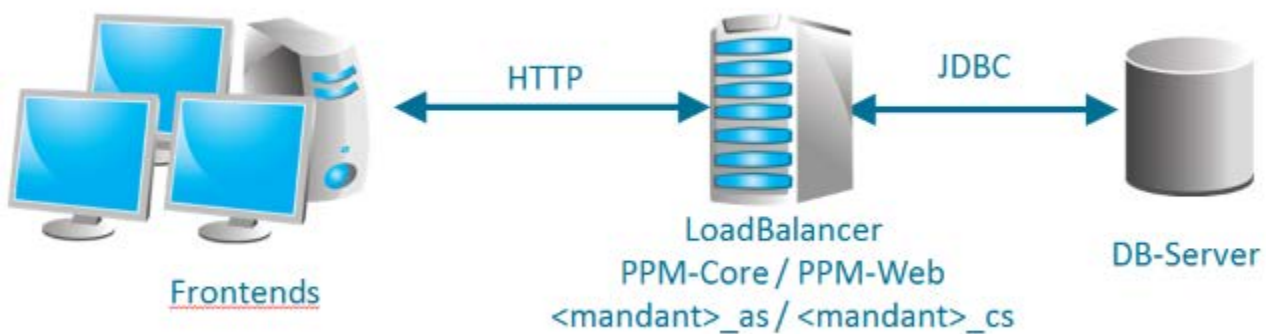
Der Datenbankserver (DBMS) dient dem PPM-Server und dem Datenimport als Persistenzschicht. Neben den Konfigurations-, Administrations- und Anwenderdaten (beispielsweise Favoriten) werden hier alle Prozessinstanzen sowie Data Analytics bezogene Daten abgelegt. Im Falle eines Systemausfalles können diese für die Wiederherstellung des Analyseservers verwendet werden.

Pro Mandant wird jeweils ein im DBMS zu erstellendes Datenbankschema verwendet. Eine Speicherung mandantenübergreifender Daten ist nicht erforderlich. Somit können für unterschiedliche Mandanten auch unterschiedliche Datenbankinstanzen verwendet werden.

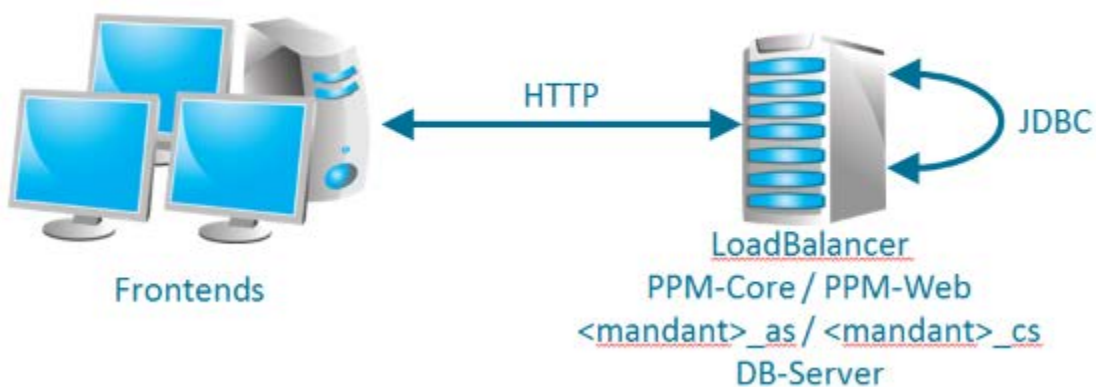
Je nach gewünschter Systemarchitektur kann das Datenbanksystem auf dem PPM-Server selbst (two-tier architecture) oder auf einem im Netzwerk verfügbaren Server installiert werden. Da insbesondere beim Datenimport große Datenmengen ausgetauscht werden, sollte bei der Verwendung einer externen Datenbank (three-tier architecture) eine schnelle Netzwerkverbindung zwischen den Servern bestehen.

Detaillierte Informationen erhalten Sie im Handbuch **PPM Datenbanksysteme**.

Three-Tier Architecture



Two-Tier Architecture



3.3 Grundlegendes zur In-Memory-Technologie

Durch die In-Memory-Technologie von PPM werden die Analyse-Daten in einer kompakten, eigens auf Analyseanfragen optimierten Hauptspeicherstruktur verwaltet. Dadurch können Anfragen um Größenordnungen schneller als mit herkömmlichen, auf Festplatten basierten Technologien beantwortet werden.

Die Hauptspeicherstrukturen werden durch den Analyseserver verwaltet. Um das Speichermanagement zu vereinfachen und ein separates Neustarten des PPM-Servers zu ermöglichen, wird dieser in einem eigenen Prozess gestartet. Für einen funktionsfähigen Mandanten müssen stets beide Prozesse aktiv sein.

Im Rahmen des Datenimports werden die speziellen Datenstrukturen des Analyseservers mit allen für die Analyse notwendigen Daten gefüllt. Werden neue Daten in das PPM-System importiert, werden die Analysestrukturen automatisch aktualisiert oder, falls notwendig, neue Strukturen erzeugt.

Um nach einem Beenden des Analyseservers (z. B. bei Neustart des Rechners oder Systemfehler) die Ausfallzeit möglichst kurz zu halten, legt der Analyseserver Wiederherstellungsdateien im Dateisystem ab. Aus diesen kann sich der Analyseserver nach einem Neustart vollständig wiederherstellen. Die initialen Wiederherstellungsdateien werden bei der Systeminitialisierung automatisch erstellt und konfigurierbar (Standard: aktiviert) bei jedem Datenimport neu generiert. Die Neugenerierung kann auch mittels des Administrations-Kommandozeilenprogramms **runppmadmin** angestoßen werden. Dadurch kann sichergestellt werden, dass gegebenenfalls gemachte Änderungen an der Konfiguration auch in den Wiederherstellungsdateien aktualisiert werden.

Veraltete oder fehlende Wiederherstellungsdateien werden automatisch vom System erkannt. In diesem Fall kann der Analyseserver sowie die zugehörigen Wiederherstellungsdateien aus der Datenbank neu generiert werden.

4 PPM-Netzwerk

PPM nutzt zum Datenaustausch Standardprotokolle des TCP/IP-Netzwerkes. Dieses Kapitel gibt einen Überblick über den Datenaustausch in einem Computernetzwerk, die Datenflüsse zwischen den PPM-Komponenten und mögliche Szenarien der PPM-Implementierung.

4.1 Netzwerkprotokolle

ALLGEMEINES

Der Informationsaustausch im Internet basiert auf dem Netzwerkprotokoll TCP/IP (Transmission Control Protocol / Internet Protocol) und erfolgt über direkte Socket-Verbindungen. Das TCP/IP-Protokoll ist ein fehlertolerantes Protokoll, das Übertragungsfehler erkennt und beseitigt. Ein Socket wird eindeutig durch die IP-Adresse des Rechners und eine Port-Nummer bestimmt. Das Serverprogramm erzeugt beim Start einen Socket und wartet auf Anfragen. Ein Client nimmt mit dem Server Verbindung auf, indem er zunächst einen Socket erzeugt und diesen mit dem Socket des Servers verbindet. Anschließend erfolgt der Informationsaustausch in dem vom Server-Socket vorgegebenen Protokoll. Die Verbindung kann durch Schließen des Sockets von einem der Partner beendet werden. In der Regel ist dies der Client, da der Server auf weitere Anfragen wartet.

Die Port-Nummern bis 1023 sind fest für bestimmte Übertragungsprotokolle vergeben. Sie werden "well-known ports" genannt und dürfen nur für die jeweiligen Protokolle verwendet werden. Im Internet wurden beispielsweise die folgende Zuordnungen für bestimmte Übertragungsprotokolle zu festen Port-Nummern des Servers festgelegt (in Klammern angegeben): HTTP (80), FTP (21), telnet (23), SMTP (25), POP3 (110) und HTTPS (443). Port-Nummern oberhalb 1023 werden "user ports" genannt.

Abweichend von den Standard-Port-Nummern kann die Port-Nummer für die meisten zusätzlich eingerichteten Dienste frei gewählt werden.

Eine Socket-Verbindung muss dabei innerhalb eines Netzwerkes eindeutig sein. Ein Port kann pro Netzwerkadresse nur einmal belegt werden.

NETZWERKSCHNITTSTELLEN

Jeder Server oder Desktop Rechner, auf dem PPM installiert werden kann, verfügt in der Regel über ein bis zwei Netzwerk-Ressourcen bzw. -Karten (auch NIC = Network Interface Card genannt). Jede NIC stellt im Netzwerk eine eindeutige Adresse (IP) bereit, unter der der Server erreichbar ist. Im Normalfall ist eine NIC für den Betrieb von PPM völlig ausreichend. Jeder Service, der von PPM bereitgestellt wird, läuft auf einer individuellen und für die NIC eindeutigen IP und Port Kombination.

Sollten mehrere NICs im Server verfügbar sein, die auch von einer Applikation genutzt werden können (Management Netzwerke sind dabei ausgeschlossen), kann PPM auch auf individuelle NIC-Adressen (IP + Port) konfiguriert werden. Die Nutzung mehrerer Netzwerkkarten wird besonders dann interessant, wenn nur eine eingeschränkte Anzahl an frei nutzbaren Ports zur Verfügung steht.

PPM-NETZWERKPROTOKOLLE

Die Kommunikation der PPM-Komponenten untereinander basiert auf TCP/IP. Andere Protokollfamilien (z. B. IPX, SPX) werden nicht unterstützt. PPM verwendet folgende TCP/IP-Protokolle:

Komponente	Protokoll	Standard-Port
Load Balancer	HTTP / HTTPS	4080/4443
CORBA Registry	Activation Port	17589
RMI Registry	RMI / SSL RMI / zipped RMI	17500
<Clientname>_cs	RMI / SSL RMI / zipped RMI	17501 + 2*n
<Clientname>_cs	CORBA (IIOP)	17590 + n
<Clientname>_cs	HTTP / HTTPS	17651 + n
<Clientname>_as	RMI / SSL RMI / zipped RMI	17502 + 2*n
PPM Web	HTTP	17101
PPM Web	AJP	17201
ElasticSearch	HTTP	17047
ZooKeeper	HTTP	17050
UMC	HTTP	17100
UMC	AJP	17200
Cloud Agent	HTTP / HTTPS	17004

RMI (REMOTE METHOD INVOCATION)

Die PPM-Serverkomponenten tauschen über dieses Protokoll Daten aus. Als Übertragungsstandard verwendet PPM die Protokollvariante **JRMP** (Java RMI Message Protocol). Das PPM-Frontend benötigt in der Regel zwei RMI-Kommunikationskanäle (Socket-Verbindungen), einen zum systemweiten RMI-Registry-Server und einen zum PPM-Mandantenserver, genauer gesagt zum PPM-Server selbst. Standardmäßig erfordert das RMI-Protokoll eine direkte Socket-Verbindung.

CORBA (COMMON OBJECT REQUEST BROKER ARCHITECTURE)

Die Web-Applikation **Performance Dashboard**, die Management Views sowie die Abfrageschnittstelle (Query-API) verwenden dieses Protokoll, um innerhalb einer Serverinstallation mit dem PPM-Mandantenserver zu kommunizieren.

JDBC (JAVA DATABASE CONNECTIVITY)

Über dieses standardisierte Protokoll greifen Java-Anwendungen auf ein RDBMS zu. Die hierfür benötigten JDBC-Treiber sind nicht Bestandteil der PPM-Installation und werden vom Datenbankhersteller zur Verfügung gestellt.

HTTP (HYPERTEXTTRANSFERPROTOCOL)

Der gesamte Datenaustausch zwischen PPM-Frontend und PPM-Server erfolgt mittels HTTP-Protokoll.

PPM verwendet dieses Protokoll zur Übertragung der HTML-Dokumentseiten, von textbasierten Konfigurationsdateien (*.properties) und von Java-Archivdateien, die zum Ausführen des PPM-Frontend im Browser erforderlich sind.

Auch der Web-Applikationsserver verwendet dieses Protokoll zum Publizieren von Management Views und zum Ausführen des Performance Dashboard.

Anstelle des HTTP-Protokolls können Sie auch die sichere Variante **HTTPS** einsetzen.

4.2 Laden des PPM-Frontend

Das PPM-Frontend kann in den folgenden drei unterschiedlichen Modi geladen und ausgeführt werden.

ALS JAVA-APPLIKATION IN EINER KOMMANDOZEILE AUF DEM SERVER

Sie können das PPM-Frontend direkt auf dem Server in der Installationsumgebung starten. Dazu öffnen Sie eine Kommandozeile auf Ihrem Rechner und wechseln in folgendes Verzeichnis:

```
<PPM-Installationsverzeichnis>/ppmmashzone/server/bin/agentLocalRepo/.unpacked/ppm-client-run-prod-<PPM Version Info>-runnable.zip/ppm/bin
```

und geben Sie dort für Windows-Betriebssysteme **runppmgui.bat** oder unter Linux **./runppmgui** zum Starten des PPM-Frontend ein.

ALS JAVA APPLLET IN EINEM WEB-BROWSER AUF DEM CLIENT

Abhängig von der Interaktion des Anwenders werden vom PPM-Web-Server JAR-Dateien zum PPM-Client übertragen und in der Java-Laufzeitumgebung des Browser-Plugin ausgeführt. Zum Starten des PPM-Frontend im Web-Browser geben Sie in der Adresszeile des Browsers die folgende URL ein:

```
http://<loadbalancer>:<loadbalancer-http-port>/ppm/html
```

Beispiel

```
http://localhost:4080/ppm/html
```

ALS JAVA WEBSTART-APPLIKATION AUF DEM CLIENT

Abhängig von der Interaktion des Anwenders wird vom PPM-Web-Server eine JNLP-Datei zum Starten der WebStart-Applikation herunter geladen, die dann in der Java-Laufzeitumgebung des Client ausgeführt wird. Das Starten des PPM-Frontend erfolgt in manchen Web-Browsern automatisch durch die Eingabe der folgenden URL in der Adresszeile des Browsers:

`http://<loadbalancer>:<loadbalancer-http-port>/ppm/html/ppm.jnlp`

Beispiel

`http://localhost:4080/ppm/html/ppm.jnlp`

Bei Benutzung des Chrome-Web-Browsers wird die Datei **ppm.jnlp** zunächst nur heruntergeladen und muss lokal ausgeführt werden. In beiden Fällen wird das PPM-Frontend jedoch als Applikation auf dem Client gestartet.

4.2.1 HTTPS-Protokoll

Wenn Sie anstelle des HTTP-Protokolls das sichere HTTPS-Protokoll verwenden möchten, müssen Sie lediglich die URL zum Starten des PPM-Frontend wie folgt ändern.

`https://<loadbalancer>:<loadbalancer-https-port>/ppm/html`

Beispiel

`https://localhost:4443/ppm/html`

Das HTTPS-Protokoll ist im Load Balancer bereits aktiviert und kann ohne weitere Änderungen der Konfiguration verwendet werden. Standardmäßig wird ein selbst erzeugtes und signiertes Zertifikat des Load Balancer für die HTTPS-Unterstützung verwendet. Dies wird in den meisten Browsern jedoch abgelehnt, da es kein gültiges Root-Zertifikat einer zugelassenen Zertifizierungsstelle enthält. Wie Sie ein eigenes gültiges Zertifikat für die HTTPS-Unterstützung des Load Balancer erstellen können, ist im Handbuch **PPM Operation Guide** beschrieben.

Durch Einsatz des HTTPS-Protokolls für den Load Balancer ist die gesamte Kommunikation zwischen PPM-Frontend und PPM-Server abgesichert. Wie Sie zusätzlich die RMI-Datenkommunikation der PPM-Serverkomponenten untereinander mittels SSL verschlüsseln, ist ebenfalls im Handbuch **PPM Operation Guide** beschrieben.

4.3 Datenkommunikation PPM-Frontend und PPM-Mandantenserver

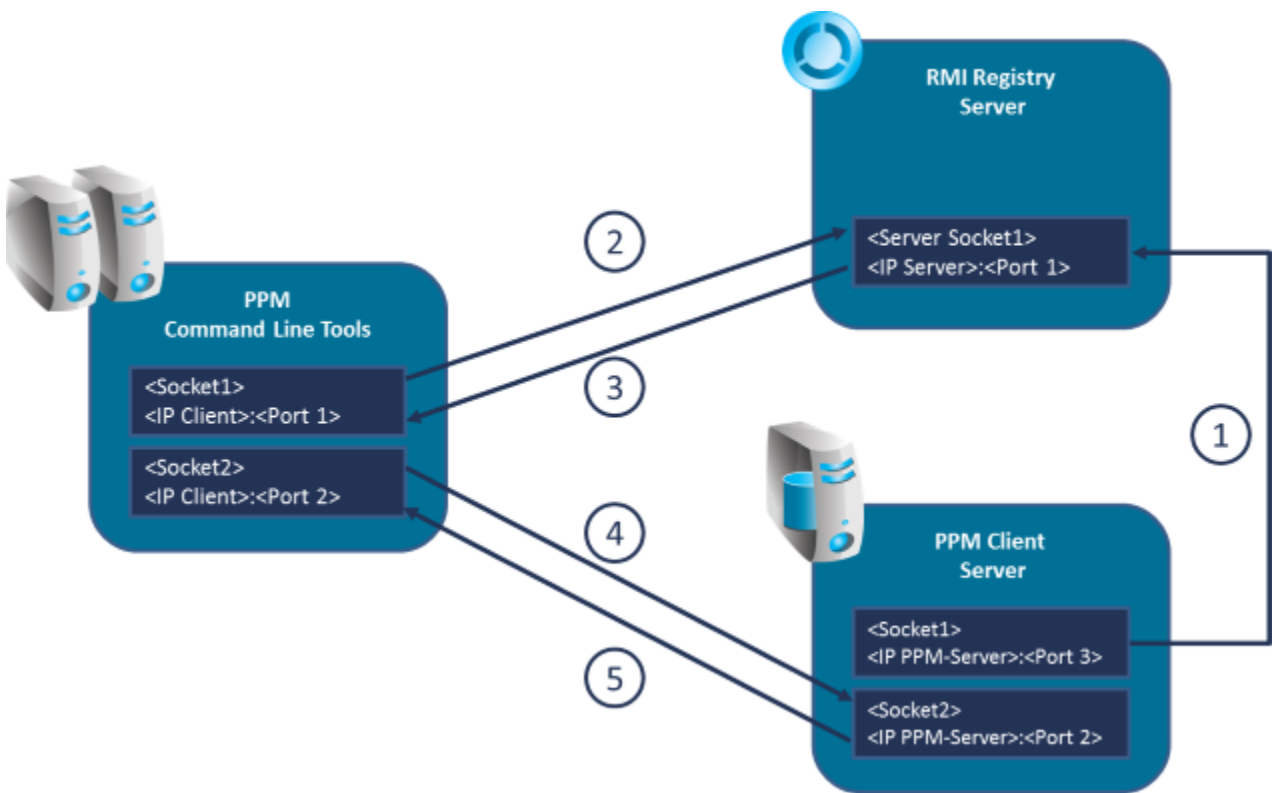
Der gesamte Datenaustausch zwischen PPM-Server und PPM-Frontend basiert auf SOAP-Webservices. Das PPM-Frontend ruft in seiner Umgebung Methoden auf, die real auf dem PPM-Serverrechner ausgeführt werden. Die Funktionalitäten des PPM-Servers sind in SOAP-Webservice-Klassen realisiert.

Das PPM-Frontend meldet sich mittels der PPM-URL am Load Balancer an. Mit erfolgreichem Login wird eine Verbindung zum PPM-Mandantenserver hergestellt.

4.4 Datenkommunikation PPM--Serverkomponenten

Jeder PPM-Mandantenserver meldet sich beim Start mit einem eindeutigen Namen an der RMI-Registry an. Mittels diesen Namens können die PPM-Kommandozeilenprogramme und die anderen Serverkomponenten Verbindungsinformationen von der Registry erfragen und die Verbindung mit dem PPM-Mandantenserver aufbauen.

Der Datenaustausch über RMI ist für die Java-Anwendung vollkommen transparent. Alle RMI-Netzwerkverbindungen müssen für die gesamte Laufzeit der PPM-Software verfügbar sein. Die RMI-Objekte selbst sind für die direkte Netzwerkkommunikation ausgelegt. Die in den RMI-Objekten enthaltenen Adressinformationen (IP-Adresse und Portnummer) können bei der Übertragung über Netzwerkgrenzen hinweg von den Firewalls nicht ausgewertet werden. Die folgende Grafik und die Tabelle veranschaulichen den beschriebenen RMI-Datenfluss:



	Client	Server	Vorgang
1	PPM-Mandanten-server	RMI-Registry-Server	Während des Starts meldet sich der Mandantenserver beim Registry-Server mit seinem Namen an.
2	PPM-Kommandozeilenprogramme	RMI-Registry-Server	Anfrage, ob gewünschter Mandantenserver verfügbar ist (Angabe des Mandantennamens im Anmeldedialog)
3	PPM-Kommandozeilenprogramme	RMI-Registry-Server	RMI Registry antwortet mit RMI-Objekt-Referenz des gestarteten Mandantenservers.
4	PPM-Kommandozeilenprogramme	PPM-Mandanten-server	Das Frontend führt über die gelieferte RMI-Objekt-Referenz Methoden des Mandantenservers aus.

	Client	Server	Vorgang
5	PPM-Kommandozeilenprogramme	PPM-Mandantenserver	Mandantenserver schickt Ergebnisse zurück.

RMI-DATENÜBERTRAGUNGSMODI

Für die RMI-Kommunikation zwischen den PPM-Serverkomponenten können Sie unterschiedliche Betriebsarten einstellen, die in den folgenden Kapiteln beschrieben sind. Nach Anlegen eines Mandanten ist der in Kapitel Komprimierte Datenübertragung (Seite 14) beschriebene Modus **Komprimierte Datenübertragung** eingestellt.

4.5 Betriebsarten der Datenübertragung

In der Kommunikation zwischen PPM-Kommandozeilenprogrammen und PPM-Mandantenserver sind verschiedene Betriebsarten der Datenübertragung möglich:

- Direkt
- Komprimiert
- Verschlüsselt

Beim Starten eines Mandantenservers bestimmt die sogenannte RMI-SocketFactory die Art der Datenübertragung zwischen PPM-Server und -Kommandozeilenprogrammen. Eine Zeile der Form "... Der Server wurde mit der SocketFactory <Name der Factory> gestartet." der Log-Ausgaben des Serverstarts informiert über die verwendete Art der RMI-Datenübertragung.

In den folgenden Kapiteln werden diese Betriebsarten näher beschrieben.

Für die Einstellungen bzgl. der Datenübertragung siehe auch Kapitel RMI-Server (Seite 43).

4.5.1 Komprimierte Datenübertragung (Standard)

Zur Reduzierung der benötigten Netzwerkbandbreite verwendet PPM nach dem Anlegen eines Mandanten standardmäßig die komprimierte Datenübertragung. Diese Betriebsart basiert auf direkter Kommunikation zwischen den PPM-Kommandozeilenprogrammen. Diese Betriebsart ist nach dem Anlegen eines Mandanten voreingestellt, kann aber nachträglich geändert werden.

KONFIGURATION

Um den RMI-Datenübertragungsmodus mit Komprimierung zu aktivieren, weisen Sie dem Schlüssel **UseSSL** in der globalen Konfigurationsdatei **Registry_settings.properties** den Wert **false** und dem Schlüssel **RMI SocketFactory** in der Mandantenkonfigurationsdatei **RMI Server_settings.properties** den Wert **com.idsscheer.ppm.rmi.compress.ZCompressionSocketFactory** zu.

4.5.2 Direkte RMI-Datenübertragung

Die RMI-Datenübertragung erfolgt direkt, ohne Komprimierung und ohne Verschlüsselung. Diese Betriebsart basiert auf direkter Kommunikation zwischen den PPM-Serverkomponenten.

Bei dieser Art der Datenübertragung entfällt die zusätzliche Komprimierung. Sie erzeugt jedoch eine signifikant höhere Netzwerklast.

KONFIGURATION

Um den RMI-Standard-Datenübertragungsmodus zu aktivieren, weisen Sie dem Schlüssel **UseSSL** in der globalen Konfigurationsdatei **Registry_settings.properties** den Wert **false** und dem Schlüssel **RMI SocketFactory** in der Mandantenkonfigurationsdatei **RMI Server_settings.properties** den Wert **com.idsscheer.ppm.rmi.ZDefaultSocketFactory** zu.

4.5.3 Verschlüsselte Datenübertragung

Wenn der Datenaustausch zwischen PPM-Kommandozeilenprogrammen und -Server unter Verwendung der SSL-Technik verschlüsselt werden soll, müssen folgende Hinweise beachtet werden:

- Die eingesetzte SSL-Technik erfordert, dass die SSL-Verschlüsselung der Datenkommunikation immer für das gesamte PPM-System aktiviert wird. Es ist technisch nicht möglich, bestimmte PPM-Mandantenserver verschlüsselt, andere aber unverschlüsselt auf die gemeinsame RMI-Registry zugreifen zu lassen. Die Konfiguration der SSL-Verschlüsselung erfolgt daher PPM-systemweit.
- Bei Verwendung des SSL-Protokolls ist die zusätzliche Komprimierung des Datenaustauschs aus technischen Gründen nicht möglich.
- Die verschlüsselte Datenkommunikation erhöht geringfügig den benötigten Rechenaufwand.
- Die Verschlüsselung der Datenkommunikation ist nur für die Betriebsart Standard des PPM-Server empfohlen. Vom Einsatz der SSL-Technik im skalierten (verteilten) PPM-System raten wir aus Performanz-technischen Gründen ab.

KONFIGURATION

Die Betriebsart aktivieren Sie, indem Sie in der globalen Konfigurationsdatei **Registry_settings.properties** dem Schlüssels **UseSSL** den Wert **true** zuweisen. Die Konfiguration der SSL-Verschlüsselung ist in den Kapiteln Registry (Seite 21) und SSL (Seite 22) beschrieben. Weitere Informationen erhalten Sie im Handbuch **PPM Operation Guide** (Kapitel **Sicherheitsaspekte/PPM-Server**).

Bei aktivierter, verschlüsselter Datenübertragung wird der Parameter **RMI SocketFactory** in der Mandantenkonfigurationsdatei **RMI Server_settings.properties** ignoriert und durch eine eigene **SSL SocketFactory** ersetzt. Die Verwendung dieser Betriebsart ist lediglich in der Log-Datei des PPM-Mandantenservers zu verifizieren.

Bei der verschlüsselten Datenübertragung ist zu beachten, dass ein eigenes Schlüsselpaar für die SSL-Verschlüsselung erzeugt werden muss. Dazu sind in der Datei **SSL_settings.properties** die Parameter **PATH_TO_KEYSTORE** und **KEYSTORE_PASSWORD** zwingend anzugeben.

5 Verteilte PPM-Server-Systeme

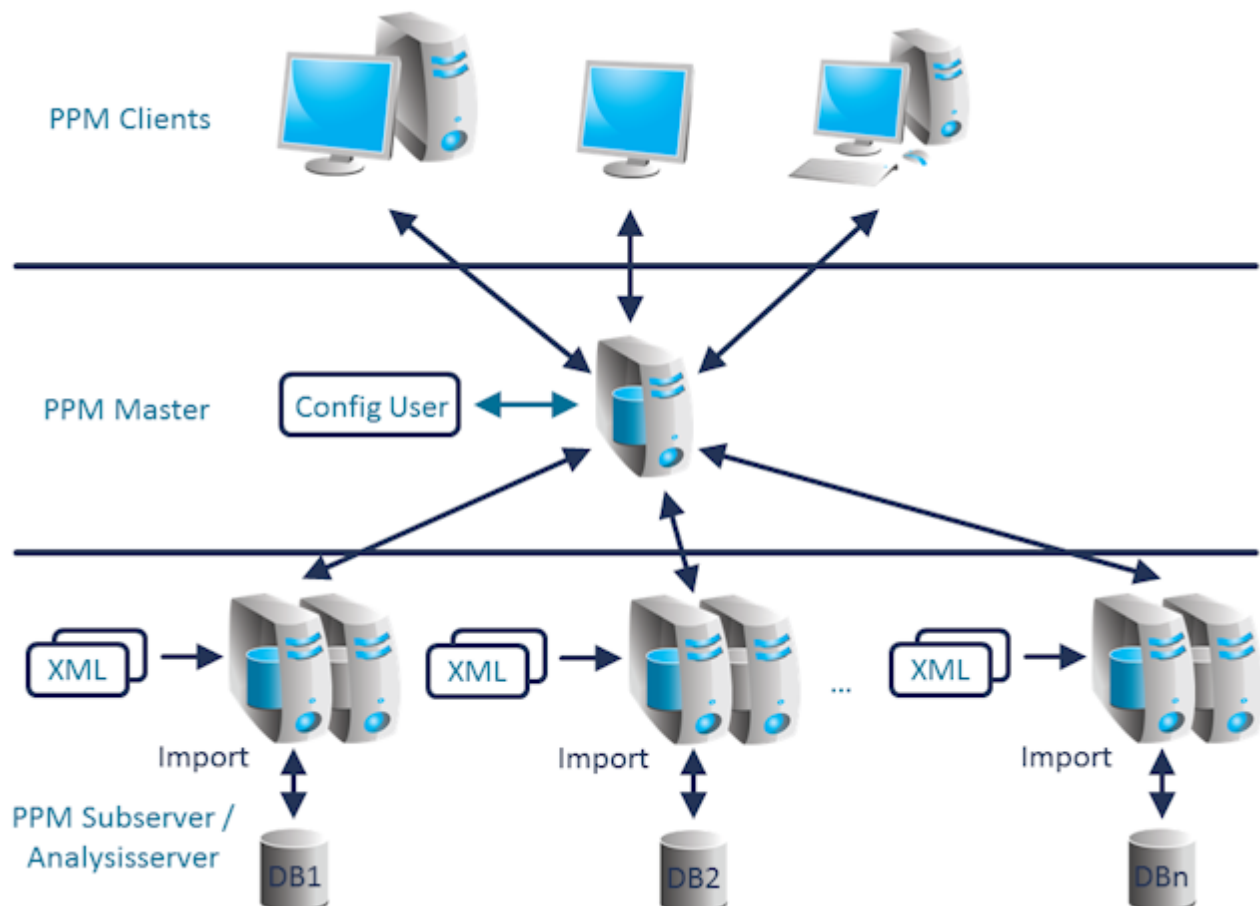
Zur Steigerung der Import- und Analyse-Performanz können mehrere PPM-Server parallel betrieben werden. Die parallel geschalteten Server werden von einem zentralen Master-Server verwaltet.

5.1 Zugriff auf ein verteiltes System

Das PPM-Frontend kommuniziert ausschließlich mit dem Master-Server. Anfragen werden vom Master-Server an die angeschlossenen Sub-Server weitergegeben. Die Berechnung der Analyseteilergebnisse erfolgt zeitgleich auf den Sub-Servern. Der Master-Server aggregiert die Ergebnisse der Sub-Server und schickt das Gesamtergebnis zum Frontend. Um einen optimalen Parallelisierungseffekt zu erreichen, sollten bei Anfragen alle Sub-Server möglichst gleichmäßig ausgelastet sein. Die Gesamt-Performanz des Systems wird vom langsamsten Sub-Server bestimmt.

Systemkonfigurationen werden am Master-Server eingelesen. Dieser gibt sie anschließend an die Sub-Server weiter. Nach Abschluss des Konfigurationsvorgangs haben alle PPM-Server dieselbe Konfiguration.

Beispiel: Datenfluss zwischen Frontend und skaliertem Server-System



5.2 Datenimport in verteilten Systemen

Um ein verteiltes System aufzubauen, müssen die Quellsystemdaten zunächst mittels eines geeigneten Kriteriums in verschiedene Datenlieferungen aufgeteilt werden. Dabei entspricht jedem Sub-Server eine Datenlieferung. Es ist darauf zu achten, dass zusammengehörende Prozessinstanzdaten stets dem gleichen Sub-Server zugeordnet werden. Werden Prozesshierarchien eingesetzt, ist darauf zu achten, dass die Sub-Server jeweils vollständig abgeschlossene Teilbäume der Hierarchie enthalten.

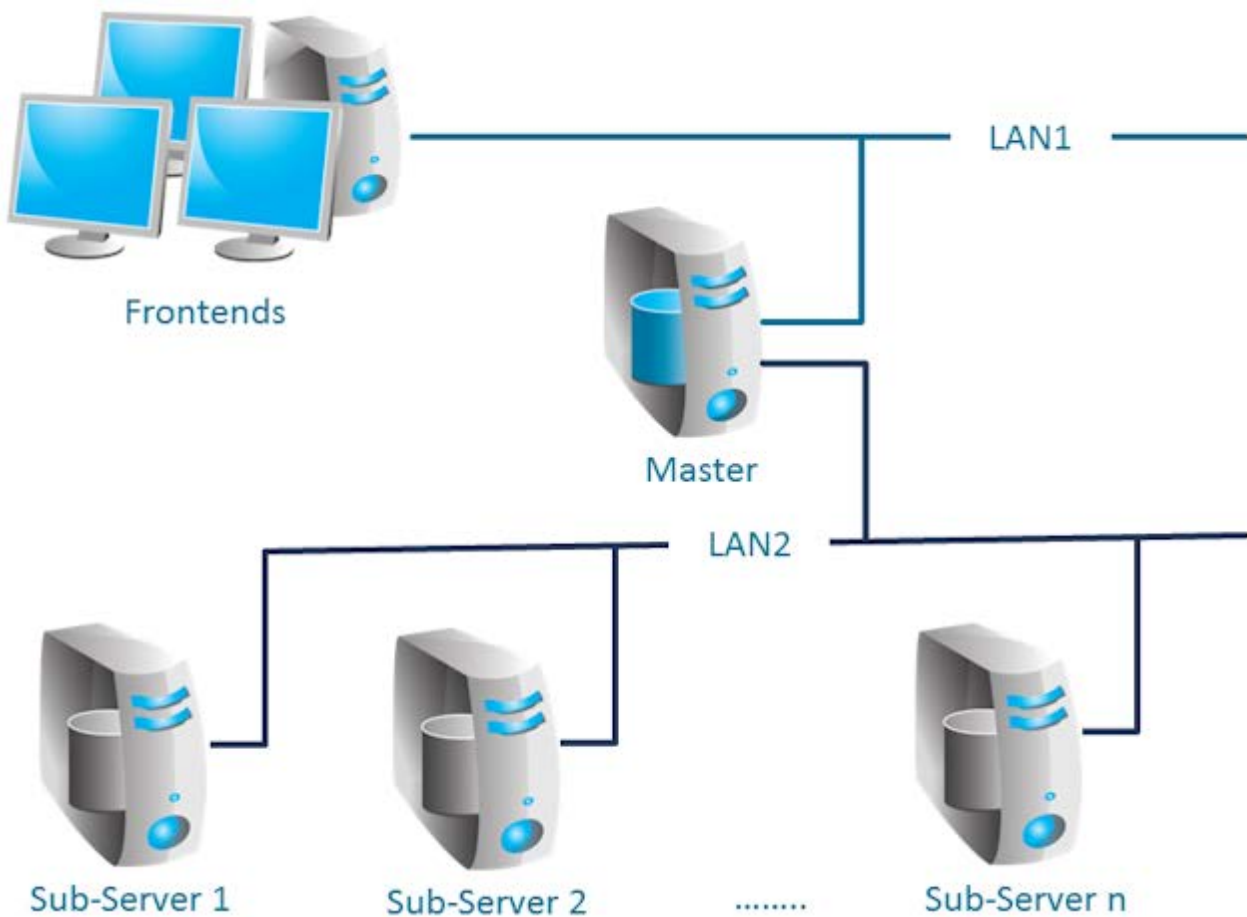
Die Auswahl des Kriteriums ist abhängig vom Anwendungsfall. Als Verfahren zum Aufteilen der Datenpakete können zum Beispiel Dimensionen wie etwa der Standort dienen: Sub-Server 1 erhält die Daten von Standort A, Sub-Server 2 die Daten von Standort B usw. Eher ungeeignet ist das Kriterium **Zeitpunkt**, da dieses in der Regel die Parallelität der Analyseanfragen und der Importe limitiert.

Die Datenpakete werden direkt auf den jeweiligen Sub-Servern eingespielt. Der Importzeitpunkt kann unabhängig auf allen Sub-Servern frei gewählt werden. Insbesondere kann dieser auf mehreren Servern gleichzeitig erfolgen, um maximale Parallelität zu erreichen.

5.3 Aufbau eines verteilten Systems

Für eine optimale Performanz sollten alle Sub-Server eigene Systemressourcen (CPU, RAM, Festplatte) verwenden. Darüber hinaus sollte der Master-Server mit den Sub-Servern über ein Hochgeschwindigkeitsnetzwerk verbunden sein. Bei Szenarien mit hohem Analysedurchsatz, sollte das Netzwerk, über das der Master-Server mit den Sub-Servern kommuniziert, vom übrigen Netzwerk entkoppelt sein, um eine störungsfreie Kommunikation zwischen den PPM-Servern zu gewährleisten.

Beispiel: Deziertes Hochgeschwindigkeitsnetzwerk



Eine Installation mehrerer Sub-Server auf einem Multiprozessorsystem ist prinzipiell möglich. Beim Auslegen des Skalierungskonzeptes sollte man darauf achten, dass der Skalierungseffekt nicht durch gemeinsam genutzte Ressourcen vermindert wird. Wird eine gemeinsame Datenbankinstanz für mehrere Sub-Server verwendet, sollten jedem Sub-Server eigene Datendateien auf physikalisch unterschiedlichen Datenträgern zugeordnet werden.

5.4 Skaliertes System initialisieren

Zunächst werden alle Sub-Server und dann der Master-Server installiert. Im Mandanten-Setup des Master-Servers werden alle Sub-Server angegeben (Mandanten-Setup-Dialog **Servereinstellungen**, Abschnitt **Betriebsart**).

Vorgehen

1. Zunächst initialisiert man alle Sub-Server. Zur Initialisierung der PPM-Sprachen verwenden alle Sub-Server und der Master-Server die gleiche Konfigurationsdatei. Diese wird in der Datei **Initdb_settings.properties** (siehe Kapitel Initdb (Seite 37)) jedes Servers angegeben. Die restlichen Konfigurationsdateien des Gesamtsystems werden in der Datei **InitSystem_settings.properties** (siehe Kapitel InitSystem (Seite 37)) des Master-Servers angegeben. Die Konfiguration der Sub-Server erfolgt durch den Master in Schritt 1. Für alle

Sub-Server wird diese Datei vom Mandanten-Setup deshalb automatisch ohne Einträge vorgegeben.

2. Anschließend initialisiert man den Master-Server. Der Master-Server liest die in der Datei **InitSystem_settings.properties** angegebenen Systemkonfigurationen ein und gibt diese auf die Sub-Server weiter.

5.5 Skaliertes System erweitern

Ein skaliertes System lässt sich jederzeit durch Hinzufügen von weiteren Sub-Servern erweitern. Der Master-Server kann maximal 64 Sub-Server verwalten.

Vorgehen

1. Mittels CTK kann ein Sub-Server zu einem bestehenden System hinzugefügt werden. Dazu wechselt man im Mandanten-Setup des Master-Servers zum Setup-Dialog **Servereinstellungen** und öffnet im Abschnitt **Betriebsart** den Dialog **Betriebsart ändern**. Hier kann man über die Schaltfläche **Hinzufügen** im Dialog **Sub-Server-Spezifikation** weitere Sub-Server ergänzen. Anschließen richtet man die neuen Sub-Server ein wie im Folgenden beschrieben.
2. Zunächst exportiert man mit dem PPM-Systemkommando **runppmconfig** die gesamte Konfiguration (inklusive dazugehöriger Datei **InitSystem_settings.properties**) eines bereits bestehenden Sub-Servers.

Beispiel

```
runppmconfig -user system -client sub1 -mode export -system <installation directory>\ppmmashzone\server\bin\work\data_ppm\custom\sub2\xml
```

3. Danach kopiert man die durch **runppmconfig** erstellte Datei **InitSystem_settings.properties** in das config-Verzeichnis des neu erstellten Sub-Servers, sodass die exportierte Konfiguration verwendet wird.
4. Anschließend wird der neue Sub-Server mit den zuvor exportierten Konfigurationsdateien initialisiert (PPM-Systemkommando **runinitdb**). Dadurch wird sichergestellt, dass der neue Sub-Server dieselbe Konfiguration wie der Master-Server hat.
5. Danach führt man einen Neustart des Master-Servers aus.

Die vorhandenen Sub-Server brauchen nicht erneut gestartet zu werden. Die in Schritt 2 kopierte Datei **InitSystem_settings.properties** sollte nach dem Initialisieren des neu hinzugefügten Sub-Servers in dessen config-Verzeichnis wieder geleert werden. Die Datei darf keinen Inhalt mehr enthalten.

5.6 Einschränkungen

5.6.1 Unterstützte Szenarien

Das skalierte System wurde speziell für prozessbasierte Szenarien entwickelt. Eine Verwendung in Kombination mit Data-Analytics ist generell nicht möglich.

Die Verwendung von prozessinstanzunabhängigen Kennzahlen („PIKIs“) ist prinzipiell möglich. Diese können aber nicht über mehrere Sub-Server verteilt werden. Bei der Konfiguration des

Master-Servers muss der Sub-Server, der für die PIKIs verwendet werden soll, eingetragen werden.

5.6.2 Anfragearten

Die meisten Kennzahltypen und Vergleichswerte von PPM können unverändert im Master-Sub-Serversystem verwendet werden. Statistische Berechnungen (Standardabweichung, Perzentile) sowie mengenbasierte Dimensionsanfragen (Kardinalitätskennzahlen) werden nicht unterstützt oder liefern nur unter bestimmten Rahmenbedingungen korrekte Ergebnisse.

Des Weiteren können in der interaktiven Filterkomponente keine Häufigkeiten angezeigt werden.

6 Konfiguration des PPM-Systems

6.1 PPM-Konfigurationsdateien

Alle PPM-Konfigurationsdateien sind Textdateien, die in einem beliebigen Texteditor bearbeitet werden können. Die Konfigurationsdateien haben das Namensmuster **<PPM-Komponente>_settings.properties**. Basisverzeichnis für alle Konfigurationsdateien ist das Verzeichnis **config** der PPM-Installation. Direkt im Verzeichnis **config** befinden sich die globalen Konfigurationsdateien **Registry_settings.properties** und **SSL_settings.properties**. In den darunter liegenden Mandantenverzeichnissen befinden sich die mandantenabhängigen Konfigurationsdateien.

6.1.1 Aufbau

Alle Konfigurationsdateien sind gleich aufgebaut, wobei die Konfigurationseinträge zeilenorientiert sind. Sie haben das Format **SCHLÜSSEL = WERT**. Dem Konfigurationsmerkmal **SCHLÜSSEL** wird der Wert **WERT** zugewiesen. **SCHLÜSSEL** kann durch Punkte (.) in unterschiedliche Konfigurationsklassen aufgeteilt sein, z. B.

<PPM-Komponente>.<Konfigurationsmerkmal>. **WERT** kann eine konkrete Wertangabe als Zahl oder Dateiname, ein komponentenspezifisches Schlüsselwort oder ein Schalter mit den Werten **true** oder **false** sein. Zeilen, die mit dem Zeichen **#** beginnen, sind Kommentarzeilen. Leerzeilen sind erlaubt.

Es gibt verschiedene Typen von Konfigurationsdateien:

- Globale Konfigurationsdateien wirken auf das gesamte PPM-System.
- Mandantenspezifische Konfigurationsdateien wirken auf den jeweiligen Mandanten.

Bei Änderungen an den globalen Konfigurationsdateien müssen RMI- und Corba-Registry-Server (PPM Core) sowie alle PPM-Mandantenserver neu gestartet werden.

Beachten Sie bei der Schreibweise der Schlüsselwörter die Groß- und Kleinschreibung.

6.1.2 Globale Konfigurationsdateien

Die in den globalen Konfigurationsdateien gespeicherten Angaben werden während der Installation interaktiv vom Anwender eingegeben und gelten für alle Mandanten des PPM-Systems.

6.1.2.1 Registry

Die beiden Einträge **RMILeaseValue** und **RMICheckInterval** konfigurieren die regelmäßige Überprüfung der RMI-Verbindung in Millisekunden und sollten nur in Ausnahmefällen verändert werden.

Der Eintrag **RMIHandshakeTimeout** gibt die Dauer in Millisekunden an, die ein RMI-Client auf die Antwort von einem RMI-Server wartet, bevor er bei einem fehlgeschlagenen

Verbindungsaufbau eine Exception wirft und den Verbindungsversuch abbricht. Die Einstellung gilt global für die RMI-Kommunikation zwischen allen PPM-Komponenten aller Mandanten.

Durch Entfernen des Eintrags aus der Datei oder durch Setzen auf einen Wert **kleiner 0** kann das Standardverhalten von Java wiederhergestellt werden. Wird der Wert auf **0** gesetzt, gibt es überhaupt keinen Timeout. Es wird ein Wert von **15 Minuten** (=900000ms) für die zeitliche Beschränkung fehlschlagender Verbindungsversuche empfohlen.

Der Schlüssel **UseSSL** bestimmt, ob zur RMI-Datenübertragung des PPM-Systems ein unverschlüsseltes Protokoll, Wert **false**, oder das Protokoll **Secure Socket Layer (SSL)**, Wert **true**, verwendet wird. Weitere Einstellungen der SSL-Verschlüsselung werden in der Datei **SSL_settings.properties** angegeben.

Der Schlüssel **CorbaServerActivationPort** gibt den Port an, über den Laufzeitinformationen und Steueranweisungen für Corba-Objekte übertragen werden können.

Sowohl der RMI- und der Corba-Registry-Server starten ihre Dienste unter der IP-Adresse, die vom DNS-System dem angegebenen Rechnernamen zugeordnet wurden.

Die Ports werden bei der Installation eingestellt und lauten **17500** für die RMI-Registry und **17590** für die CORBA-Registry. Die Werte für den Host-Namen und die Ports können mittels Cloud Controller geändert werden. Details hierzu erhalten Sie in der Dokumentation **PPM Operation Guide** im Kapitel **Administration/Konfiguration**.

Die gestarteten PPM-Mandantenserver erzeugen jeweils RMI- und ggf. Corba-Objekte und melden diese mit der in den mandantenspezifischen Settings-Dateien **RMI Server_settings.properties** und **Corba_settings.properties** angegebenen Port-Nummern an den Registry-Servern an.

Wenn in dem PPM-Serverrechner mehrere Netzwerkkarten installiert sind, gibt man anstelle des Rechnernamens die IP-Adresse der gewünschten Netzwerkkarte an, in deren Netzwerk das PPM-System verfügbar sein soll. Die Einstellung können Sie mittels Cloud Controller vornehmen. Details hierzu erhalten Sie in der Dokumentation **PPM Operation Guide** im Kapitel **Administration/Konfiguration**.

Die URLs des Corba- und RMI-Registry-Servers können Sie mit Hilfe des Cloud Controller ändern. Details hierzu erhalten Sie in der Dokumentation **PPM Operation Guide** im Kapitel **Administration/Konfiguration**.

6.1.2.2 SSL

Wenn man in der Datei **Registry_settings.properties** die verschlüsselte RMI-Datenübertragung aktiviert hat, konfiguriert man in der Datei **SSL_settings.properties** weitere Einstellungen der SSL-Verschlüsselung.

Der Schlüssel **PATH_TO_KEYSTORE** gibt den zu verwendenden Keystore an. Ein Keystore ist eine Datenbank in Form einer Datei, in der öffentliche und private Schlüsselpaare in bereits verschlüsselter Form gespeichert sind.

Der Schlüssel **KEYSTORE_PASSWORD** spezifiziert das Kennwort, mit dem der Keystore verschlüsselt wurde.

In dem Keystore gibt man immer nur ein einziges Public-Private-Schlüsselpaar an, dass für die

SSL-Verbindung verwendet werden soll. Dadurch wird sichergestellt, dass immer das passende Zertifikat zum Aufbau der SSL-Verbindung verfügbar ist.

6.1.3 Mandantenspezifische Konfigurationsdateien

Die mandantenspezifische Konfiguration ist jeweils in Konfigurationsdateien in einem Unterverzeichnis, dessen Namen dem Mandantennamen entspricht, gespeichert.

6.1.3.1 AdapterConfig

Für den XML-Import spezifizieren die Schlüssel **XML_DATEFORMAT** und **XML_DATEFORMAT_ALTERNATIVE** das in den Protokolldateien erwartete Zeitpunktformat (Angabe von Datum und Uhrzeit). Die Schlüssel **XML_TIMEOFDAYFORMAT** und **XML_TIMEOFDAYFORMAT_ALTERNATIVE** spezifizieren das in den Protokolldateien erwartete Uhrzeitformat. Die Schlüssel **XML_DAYFORMAT** und **XML_DAYFORMAT_ALTERNATIVE** spezifizieren das in den Protokolldateien erwartete Format eines Kalendertags.

Die Formatspezifikationen werden bei der Zuweisung eines Quellsystemattributwertes zu einem PPM-Attribut mit den Datentypen **TIME (DATE)** bzw. **DAY** verwendet.

Die Formateinstellungen in der Datei **AdapterConfig_settings.properties** beziehen sich ausschließlich auf den XML-Import im Graph-Format sowie den Data-Analytics-Import.

6.1.3.2 AnalysisServer

Mit den Angaben in dieser Datei werden Einstellungen des Analyseservers vorgenommen. Die folgende Tabelle listet die wichtigsten Einstellungsmöglichkeiten auf:

Schlüssel	Beispielwert	Beschreibung
RECOVERY_FOLDER	C:\SoftwareAG\ppmmashzone\server\bin\work\data_ppm\recovery\umg_en\	Pfad für die Recovery-Dateien des Analyseservers (Standard: <installationsverzeichnis>\ppmmashzone\ server\bin\work\data_ppm\recovery\ <mandant>)< td=""> </mandant>)<>
INDEX.PROCESS.<internaldimname>.USE	INDEX.PROCESS.MATERIAL.USE=true	Optionale Konfiguration eines zusätzlichen Index. Mit true wird ein Index auf der mit ihrem internen Namen (<internaldimname>) angegebenen Dimension erstellt. Es sind nur die folgenden Prozessdimensionstypen erlaubt: Textdimension (ein-, zwei- bzw. n-stufig) Zeitdimension (timedim) Uhrzeitdimension (hourdim)
INDEX.PROCESS.<internaldimname>.	INDEX.PROCESS.MATERIAL.	Zusätzliche, optionale Angabe zum erstellten Index einer Dimension (nur bei mehrstufigen

Schlüssel	Beispielwert	Beschreibung
REFINEMENT	REFINEMENT= BY_LEVEL2_5	<p>Textdimensionen).</p> <p>Verfeinerung (Refinement), mit der der Index gepflegt werden soll. Vorgabe ist der größte Wert (BY_LEVEL_1, BY_YEAR, BY_HOUR_OF_DAY).</p> <p>Mögliche Werte:</p> <p>Zweistufige Textdimension BY_LEVEL1, BY_LEVEL2</p> <p>N-stufige Textdimension BY_LEVELX_Y, wobei X die gewählte Verfeinerungsstufe und Y die maximale Stufenanzahl der Dimension angibt (mit dem Beispielwert BY_LEVEL2_5 wird ein Index auf der zweiten Stufe der fünfstufigen Textdimension MATERIAL erstellt)</p> <p>Zeitdimension, erlaubte Werte: BY_DAY, BY_MONTH, BY_QUARTER, BY_YEAR</p> <p>Uhrzeitdimension, erlaubte Werte: BY_MINUTE_OF_DAY, BY_HOUR_OF_DAY</p> <p>Pro Dimension ist nur eine Indexangabe erlaubt. Sind bzgl. ein- und derselben Dimension mehrere Indizes an- bzw. ausgeschaltet bzw. Verfeinerungsangaben konfiguriert, wird stets die Letzte der Angaben verwendet.</p> <p>Falsche Dimensionstypen werden ohne Fehlermeldung ignoriert.</p> <p>Wurde ein Index erfolgreich erstellt, wird dazu eine Information ausgegeben.</p>
RETRIEVER_MAX_INSTANCES	50000 (Standardwert)	<p>Max. Anzahl von Prozessinstanzen bzw. Funktions-/Relationsinstanzen, die vom Analyseserver angefragt werden können. Wird der angegebene Wert in einer Prozessinstanzenanfrage überschritten, bricht diese mit einer Fehlermeldung ab. Größere Werte führen zu erhöhtem Hauptspeicherbedarf in Servern und Benutzeroberflächen.</p>
RETRIEVER_MAX_ITERATION_STEPS	5000 (Standardwert)	<p>Maximale Anzahl von Datenzeilen, die vom Analyseserver angefragt werden können. Wird</p>

Schlüssel	Beispielwert	Beschreibung
		der angegebene Wert im Ergebnis einer Anfrage überschritten, bricht diese mit einer Fehlermeldung ab. Größere Werte führen zu erhöhtem Hauptspeicherbedarf in Servern und Benutzeroberflächen.
RETRIEVER_MAX_FILTER_ITEMS_FOR_XA_TO_PI	10000 (Standardwert)	Maximale Anzahl von Werten im Filter bei der Übernahme einer Anfrage aus einem Data-Analytics-Analyseraum in die Prozessanalyse. Würden mehr Werte entstehen, wird die Anfrage abgebrochen. Größere Werte führen zu erhöhtem Hauptspeicherbedarf in Servern und Benutzeroberflächen
MemoryLoadGuard.Enabled	true	Aktivierung des Speicherbelegungs-Warnsystems
MemoryLoadGuard.Warn.PercentageOfMemoryUsed	90	Speicherbelegung, ab der gewarnt wird.
MemoryLoadGuard.PreventImport.PercentageOfMemoryUsed	95	Speicherbelegung, ab der keine neuen Datenimporte in den Analyseserver mehr zugelassen werden.
MemoryLoadGuard.BackgroundCheck.TimeWindow.Start	22:00	Start-Tageszeit, ab der die Hintergrundauslastungskontrolle stattfindet
MemoryLoadGuard.BackgroundCheck.TimeWindow.End	05:00	End-Tageszeit, ab der die Hintergrundauslastungskontrolle endet.
LANGUAGE	de (ISO-Code nach ISO 639-1)	Sprache, mit der der Analyseserver betrieben werden soll (nach ISO 639-1; wenn leer wird die Betriebssystem-Standardsprache verwendet; wenn letztere oder die angegebene Sprache vom Mandanten nicht unterstützt werden, so wird die Standardsprache des Mandanten verwendet).

6.1.3.3 AnalysisServer_Log

Mit den Angaben in dieser Datei werden die Einstellungen für die Systemmeldungen des Analyseservers vorgenommen.

Detaillierte Informationen erhalten Sie im Handbuch **PPM Operation Guide**.

6.1.3.4 Chart

Angaben in dieser Datei beeinflussen das Erscheinungsbild der vom PPM-Server erzeugten Diagramme. Es können z. B. die Farben und Zeichensätze der Beschriftung bestimmt werden. Die Datei enthält ausführliche Kommentare, die die Auswirkung der Einträge beschreiben.

6.1.3.5 CNet (Communication Net)

Die Angaben in der Datei geben die Vorgabewerte der einzelnen Schlüssel wieder, die das Aussehen eines im Modul Organisationsanalyse erzeugten Kommunikationsnetzes beeinflussen. Alle Parameter sind optional. Die Datei enthält ausführliche Kommentare, welche die Auswirkung der Einträge beschreiben.

Schlüssel	Beschreibung
DEFAULT_NODE_MODE	Modus der Knotendarstellung. Mögliche Werte: COMPLETE_MODE (OE in ARIS-Notation) oder SIMPLE_NODE (Knoten als farbig gefüllter Kreis). Vorgabewert: COMPLETE_MODE
DEFAULT_NODE_SIZE	Standardmäßiger Kreisdurchmesser (nur im Modus SIMPLE_NODE)
DEFAULT_EDGE_THICKNESS_MODE	Modus der Kantenstärkendarstellung in Abhängigkeit von: - dem proz. Anteil an ausgehenden Kanten (OUTGOING_MODE) - dem proz. Anteil an eingehenden Kanten (INCOMING_MODE) - dem absoluten Kennzahlwert (VALUE_MODE) Darstellung einer einheitlichen Kantenstärke: NONE_MODE
DEFAULT_EDGE_VALUE_MODE	Darstellung der Kantenwerte in Abhängigkeit von: - dem proz. Anteil an ausgehenden Kanten (OUTGOING_MODE) - dem proz. Anteil an eingehenden Kanten (INCOMING_MODE) - dem absoluten Kennzahlwert (VALUE_MODE) Keine Darstellung von Kantenwerten: NONE_MODE
DEFAULT_EDGE_THICKNESS	Kantenstärke bei einheitlicher Darstellung (nur im Modus NONE_MODE)
DEFAULT_EDGE_COLOR	Kantenfarbe
DEFAULT_NODE_COLOR	Knotenfarbe
DEFAULT_BACKGROUND_COLOR	Hintergrundfarbe

Schlüssel	Beschreibung
DEFAULT_SELFEDGE_MODE	Visualisierung der internen Kommunikation eines Knotens (Kennzahlwert) anhand des Kreisdurchmessers (nur im Modus SIMPLE_NODE). Mögliche Werte: - true (anzeigen) - false (nicht anzeigen)
DEFAULT_EDGE_PIXEL_RANGE	Wertebereich (Fließkommazahl) für Kantenstärke in Pixeln (min.,max.) bei abhängiger Darstellung (alle Modi außer NONE_MODE)
DEFAULT_NODE_PIXEL_RANGE	Wertebereich für Kreisdurchmesser in Pixeln (min.,max.) bei vom Kennzahlwert abhängiger Darstellung (DEFAULT_SELFEDGE_MODE=true)
UNDER_WARNING_EDGE_COLOR	Kantenfarbe in im RGB-Format für Kennzahlwerte, die besser als der gepflegte Warnwert (= Planwert 1 der Kennzahl) sind.
WARNING_EDGE_COLOR	Kantenfarbe im RGB-Format für Kennzahlwerte, die zwischen dem Warnwert (= Planwert 1) und dem Alarmwert (= Planwert 2) liegen.
ALARM_EDGE_COLOR	Kantenfarbe im RGB-Format für Kennzahlwerte, die schlechter sind als der Alarmwert (= Planwert 2).
ONLY_FOR_RELATIONS	Analyseart des Kommunikationsnetzes. Mögliche Werte: - true (Vorgabewert: nur Beziehungen zwischen Organisationsdimensionen) - false (zwischen zwei beliebigen Dimensionen)

6.1.3.6 CorbaServer

Die Angaben in dieser Datei konfigurieren den Datenaustausch über das Corba-Protokoll. Die PPM-Web-Applikationen **Performance Dashboard**, **Management View** sowie **Abfrageschnittstelle** verwenden das Corba-Protokoll zur Kommunikation mit dem PPM-Mandantenserver.

Der Schlüssel **CorbaObjectsPort** bestimmt die vom PPM-Mandantenserver verwendete Port-Nummer. Vom Mandantenserver erzeugte Corba-Objekte werden mit dieser Portnummer an der Corba-Registry angemeldet.

6.1.3.7 Database

Die Vorgaben in dieser Datei konfigurieren datenbankabhängige Basismechanismen des PPM-Systems und beschreiben die Schnittstelle zwischen PPM-Server und Datenbankserver.

ALLGEMEINER TEIL

Schlüssel	Beschreibung
DATABASE_TYPE	Spezifiziert den Typ des zu verwendenden RDBMS. Erlaubte Werte sind ORACLE 11 , ORACLE_12 , DB2_9 , DB2_10 , SQLSERVER_2008 , SQLSERVER_2008_UNICODE , SQLSERVER_2012 , SQLSERVER2012_UNICODE , SQLSERVER_2014 und SQLSERVER_2014_UNICODE
URL	URL der PPM-Datenbanken in JDBC-Notation. Abhängig vom verwendeten Datenbanksystem und DATABASE_TYPE.
USER	Datenbankbenutzer der PPM-Datenbanken
PASSWD	Kennwort der PPM-Datenbanken. Wird das Kennwort in den Mandanteneinstellungen des CTK eingegeben oder geändert, wird diese automatisch verschlüsselt. Das manuelle Setzen eines unverschlüsselten Kennwortes direkt in den Database-Settings, zum Beispiel für Testzwecke, wird ebenfalls unterstützt.
TRUNCATE_DIMENSION_STRINGS/ TRUNCATE_DIMENSION_MARK	Verhalten bei textbasierten Dimensionswerten, welche die konfigurierte Maximallänge der Dimension überschreiten. Mögliche Werte sind TRUE , FALSE (Standard) oder MARK . FALSE, Werte und Beschreibungen werden auf die konfigurierte Maximallänge der Dimension gekürzt. Es erfolgt eine Fehlermeldung durch den Kennzahlberechner. TRUE, analog zu FALSE. Es erfolgt aber keine Fehlermeldung. MARK, Werte und Beschreibungen von Dimensionen, welche die Maximallänge überschreiten, werden durch den Text in TRUNCATE_DIMENSION_MARK ersetzt. Es erfolgt eine Warnung durch den Kennzahlberechner.

Für einige Sonderfälle, die im Rahmen des Datenimportes auftreten können, müssen PPM bei der Initialisierung bereits Standardwerte vorgegeben werden.

Warnung

Bitte beachten Sie, dass diese Werte keinesfalls in einem bereits initialisierten System geändert werden dürfen.

Schlüssel	Bedeutung
PPM_NULL	Wert, der für Dimensionen verwendet wird, die nicht gepflegt sind. (UTF-kodierte Zeichenfolge. Der Wert muss vor der Initialisierung der Datenbank geändert werden).
ERROR_NODE.STR	Name, der Prozesstypgruppe im Prozessbaum, die alle fehlerhaft berechneten/typisierten Instanzen enthält (UTF-kodierte Zeichenfolge. Der Wert muss vor der Initialisierung der Datenbank geändert werden).
NOT_TYPIFIED.STR	Name des Prozesstyps im Prozessbaum, der nicht typisierbare Instanzen enthält. (UTF-kodierte Zeichenfolge. Der Wert muss vor der Initialisierung der Datenbank geändert werden)
WRONG_TYPIFIED.STR	Name des Prozesstyps im Prozessbaum, der falsch typisierte Instanzen enthält. (UTF-kodierte Zeichenfolge. Der Wert muss vor der Initialisierung der Datenbank geändert werden)

VERWENDETE TABLESPACES/DATENDATEIEN

Aus Performanzgründen empfiehlt es sich, unterschiedliche Datenarten im Datenbanksystem separat zu speichern. Die Definition ist abhängig vom verwendeten RDBMS sowie der konkreten Installation des Datenbanksystems. Eine genaue Beschreibung entnehmen Sie bitte dem Handbuch **PPM Datenbanksysteme**.

Schlüssel	Bedeutung
<DATABASE_TYPE>_TBLCONF_STDTABLE/ <DATABASE_TYPE>_TBLCONF_STDINDEX/ <DATABASE_TYPE>_TBLCONF_STDBLOB	Diese Parameter legen fest, in welchen Tablespaces beziehungsweise Datendateien bestimmte Arten von Daten abgelegt werden.

INTERNER TEIL

Die sonstigen in den Database-Settings enthalten Parameter dienen der Konfiguration von Datenbank- und PPM-Basismechanismen. Sie ermöglichen spezifische Anpassungen und Feineinstellungen in speziellen Konstellationen (zum Beispiel bei unterschiedlichem Verhalten des RDBMS innerhalb einer Hauptversion oder bei besonderen Importszenarien). Die Voreinstellungen decken in der Regel alle Einsatzszenarien optimal ab und sollten daher nur im Ausnahmefall geändert werden.

Warnung

Änderungen an den internen Einstellungen können sich in der gesamten Datenbank auswirken und die Korrektheit, Performanz und Konsistenz des PPM-Systems nachhaltig und irreparabel beeinträchtigen. Bitte ändern Sie diese nur auf explizite Aufforderung durch den PPM Produktsupport.

6.1.3.8 EPK

GRÖßENBEGRENZUNG DER PROZESSINSTANZTABELLE

```
#Limit for instances in instancelist (0 = No question at all)
CRITICAL_INSTANCE_COUNT_TO_VIEW = 15000
```

Maximale Anzahl der in einer Prozessinstanztafel angezeigten Prozessinstanzen. Wird diese Anzahl bei einer Anfrage überschritten, erfolgt eine Rückfrage des Systems, ob diese wirklich durchgeführt werden soll. Standardwert: 15000.

6.1.3.8.1 Anpassung der EPK-Darstellung

VOREINSTELLUNG DER EPK-ANSICHT

```
#Configuration of the default view
#possible values: epk, attributes, functions, gantt
Default_Epk_View = epk
```

Ansicht, die beim Öffnen einer EPK angezeigt sein soll. Wird kein Wert angegeben oder fehlt der Eintrag, wird per Standard die Ansicht **epk** ausgewählt.

FARBINTENSITÄT VON FUNKTIONEN

```
#Configuration of color intensities:
FUNCTION_INTENSITY_MAX.0 = 0.2
FUNCTION_INTENSITY_MAX.1 = 0.4
FUNCTION_INTENSITY_MAX.2 = 0.6
FUNCTION_INTENSITY_MAX.3 = 0.8
```

Bei der Anzeige verdichteter EPKs konfigurieren diese Einstellungen die Farbintensität in Abhängigkeit von der Anzahl der Bearbeitungen der jeweiligen Funktion. Für jede der vier Stufen wird der Schwellenwert als Faktor angegeben.

KANTENKONFIGURATION

```
#Configuration of connection categories:
EDGE.0 = EDGE1
EDGE.1 = EDGE2
EDGE.2 = EDGE3
EDGE.DEFAULT_WITH = 2

EDGE1.max = 0.3
EDGE1.width = 1

EDGE2.max = 0.7
EDGE2.width = 3

EDGE3.max = 1
EDGE3.width = 5
```

Bei der Anzeige verdichteter EPKs konfigurieren diese Einstellungen die Dicke von Flusskanten in Abhängigkeit von den Ausführungen der Kante. Für jede der drei Stufen wird der Schwellenwert als Faktor und die Kantendicke in Pixeln angegeben.

LAYOUT-KONFIGURATION

Warnung

Die folgenden Einstellungen stammen aus dem Layout-Algorithmus von ARIS Business Architect und sollten nicht geändert werden.

Einstellung	Wert	Beschreibung
LAYOUT_ALGO_NO	20	Nummer des Layout-Algorithmus'
LAYOUT_EPK_ALGO	1	Layout-Verfahren für die unverdichtete EPK-Ansicht
LAYOUT_EPK_ALGO_COMPRESSED	1	Layout-Verfahren für die verdichtete EPK-Ansicht
LAYOUT_LONGEST_PATH_POSITION	0	Ausrichtung des längsten Pfades: zentriert, mittig, rechts, links
LAYOUT_BREAK_SPACE	true	Leerzeichen durch Zeilenumbruch ersetzen
LAYOUT_MAKE_SPACE	true	Platz schaffen bei Teil-Layout
LAYOUT_ROOT_DOWN	1	Lage der Wurzel zum Unterbaum
LAYOUT_ROOT_POSITION	0	Lage der Wurzelknoten
LAYOUT_CHANGE_SON_ARRANGEMENT	2	Übergang zu vertikalem Layout
LAYOUT_ARRANGE_SATELLITES	false	Satelliten anordnen
LAYOUT_OBJECT_SCALING	false	Objekte skalieren
LAYOUT_SHOW_TRIMMED	false	EPK-Darstellung ungetrimmt anzeigen
LAYOUT_SHOW_ATTRIB_TITLE	false	Titel der Attribute anzeigen
LAYOUT_XSPACING	30	Horizontaler Objektabstand
LAYOUT_YSPACING	50	Vertikaler Objektabstand
LAYOUT_FONTSIZE	10	Schriftgröße in EPK-Objekten
LAYOUT_HORIZONTAL	false	EPK wird nicht horizontal dargestellt

PLATZIERUNG VON OBJEKTATTRIBUTEN

```
#Configuration of default attribute placements:
EPK_NODE_ATTRIBUTE_PLACE_CENTER      = AT_OBJNAME
EPK_NODE_ATTRIBUTE_PLACE_RIGHT_TOP   = AT_AV_PROC_TIME
EPK_NODE_ATTRIBUTE_PLACE_RIGHT_BOTTOM =
EPK_NODE_ATTRIBUTE_PLACE_LEFT_BOTTOM = AT_END_TIME
EPK_NODE_ATTRIBUTE_PLACE_LEFT_TOP    = AT_START_TIME
```

Enthält Angaben, an welcher Position die einzelnen Attribute von Objekten in der EPK-Ansicht angezeigt werden. Die Angabe erfolgt in folgender Form:

```
EPK_NODE_ATTRIBUTE_PLACE_<Position> = <Identifizierer des Attributes>
```

ANZEIGE VON OBJEKTATTRIBUTEN IN TOOLTIPS

```
#Configuration of default tooltip display
SHOW_TOOLTIP.0           = AT_PROCTYPEGROUP
SHOW_TOOLTIP.1           = AT_PROCTYPE
```

Bestimmt die Attributwerte, die angezeigt werden, wenn sich der Mauszeiger über einem Objekt befindet. Um weitere Attributwerte anzuzeigen, kann die Liste durch weitere Schlüsselwertpaare ergänzt werden. Der numerische Teil des Schlüsselfeldes muss dazu entsprechend fortgezählt werden.

Es werden nur durchgängig nummerierte Listeneinträge berücksichtigt

6.1.3.8.1.1 Funktionsspezifische Anpassungen

Neben der Konfiguration der Anzeige allgemeiner Objektattribute ist es möglich, diese auch funktionspezifisch anzupassen.

Dazu müssen die betroffenen Funktionen zunächst dem System bekannt gemacht werden. Dies erfolgt durch das Schlüsselwort:

```
FUNCTION.0 = <Funktionsname>
```

Durch weitere Schlüsselwertpaare können weitere Funktionen spezifiziert werden. Der numerische Teil des Schlüsselfeldes muss dazu entsprechend fortgezählt werden.

Es werden nur durchgängig nummerierte Listeneinträge berücksichtigt.

Für die so definierten Funktionen können nun Tooltips und platzierte Objektattribute definiert werden.

Im Folgenden wird dies beispielhaft für die Funktion **SAP.AUFT** beschrieben:

```
FUNCTION.0           = SAP.AUFT
```

ANZEIGE VON OBJEKTATTRIBUTEN IN TOOLTIPS (FUNKTIONSSPEZIFISCH)

```
#Configuration of default tooltip display
SAP.AUFT.SHOW_TOOLTIP.0           = AT_PROCTYPEGROUP
SAP.AUFT.SHOW_TOOLTIP.1           = AT_PROCTYPE
SAP.AUFT.SHOW_TOOLTIP.2           = AT_KI_FEDFREQ
```

Bestimmt die Attributwerte, die angezeigt werden, wenn sich der Mauszeiger über der entsprechenden Funktion befindet. Um weitere Attributwerte anzuzeigen, kann die Liste durch weitere Schlüsselwertpaare ergänzt werden. Der numerische Teil des Schlüsselfeldes muss dazu entsprechend fortgezählt werden.

Es werden nur durchgängig nummerierte Listeneinträge berücksichtigt

PLATZIERUNG VON OBJEKTATTRIBUTEN (FUNKTIONSSPEZIFISCH)

```
#Configuration of attribute placements for function 'Create customer order'
SAP.AUFT.EPK_NODE_ATTRIBUTE_PLACE_CENTER           = AT_OBJNAME
SAP.AUFT.EPK_NODE_ATTRIBUTE_PLACE_RIGHT_TOP       = AT_AV_PROC_TIME
SAP.AUFT.EPK_NODE_ATTRIBUTE_PLACE_RIGHT_BOTTOM    =
SAP.AUFT.EPK_NODE_ATTRIBUTE_PLACE_LEFT_BOTTOM     = AT_END_TIME
SAP.AUFT.EPK_NODE_ATTRIBUTE_PLACE_LEFT_TOP        = AT_START_TIME
```

Bestimmt die Attributwerte, die an einem EPK-Knoten angezeigt werden sollen. Der Schlüssel setzt sich aus dem Schlüssel der Funktion (hier SAP.AUFT) und der Position **EPK_NODE_ATTRIBUTE_PLACE_<Position>** zusammen.

6.1.3.8.2 Einstellungen des EPK-Verdichters

ART DER VERDICHTUNG VON FUNKTIONEN

```
#Should orgunit attributes at functions be used for compression?  
EPK_USE_ORGUNIT_FOR_COMPRESS = false
```

Ist der Wert dieser Einstellung **true**, werden beim Verdichten von Prozessinstanzen Organisationseinheiten berücksichtigt. Gleiche Funktionen die verschiedenen Organisationseinheiten zugewiesen sind, werden dabei wie unterschiedliche Funktionen behandelt. Standardwert: **false**

BERECHNUNGSGRANULAT DES PERSISTENTEN VERDICHTERS

```
# Number of Instances marked to delete before a KI-Calculation is initiated  
COMPRESSOR_KICALCULATION_THRESHOLD=100000
```

Der persistente Verdichter arbeitet in Paketen die jeweils zwei Phasen umfassen. In der ersten Phase werden die zu verdichteten Prozessinstanzen zusammengefasst und als neue EPK gespeichert. Anschließend berechnet der Kennzahlberechner die neuen Instanzen und konsolidiert den Datenbestand. Dieser Parameter bestimmt die Größe der Pakete, also nach wie vielen Quellinstanzen jeweils in die Phase **Kennzahlberechnung** gewechselt wird. Standardwert: **100000**

LÖSCHEN VON HIERARCHISCHEN PROZESSINSTANZEN

```
#Default behavior when deleting hierarchical EPCs  
DELETE_REFERENCED_EPCS_WHEN_DELETETED = false
```

Ist der Wert dieser Einstellung **true**, werden beim Löschen von Prozessinstanzen deren hinterlegte Prozessinstanzen aller Hierarchiestufen gelöscht. Standardwert: **false**

```
#Default behavior when compressing hierarchical EPCs  
DELETE_REFERENCED_EPCS_WHEN_COMPRESSED = false
```

Ist der Wert dieser Einstellung **true**, werden beim persistenten Verdichten von Prozessinstanzen deren hinterlegte Prozessinstanzen aller Hierarchiestufen gelöscht. Standardwert: **false**

Warnung

Beim Verdichten bzw. Löschen von Prozessinstanzen werden hinterlegte Prozessinstanzen ohne Berücksichtigung von Prozesszugriffsrechten gelöscht. Ein PPM-Benutzer mit dem Funktionsrecht **Prozessinstanzverdichtung** kann auch Prozessinstanzen löschen, für die er keine Zugriffsrechte hat.

VERDICHTEN VON ZEITRAUMDIMENSIONEN

```
#Delete rangedimension values for compression  
DELETE_RANGEDIMENSION_VALUES_FOR_COMPRESSION = false
```

Dieser Wert bestimmt das Verhalten bei der persistenten Verdichtung von Prozessen in Kombination mit Zeitraumdimensionen. Eine persistente Verdichtung von Prozessinstanzen unter Beibehaltung der Zeitraumdimensionen ist nicht möglich. Sind für betroffene Prozessinstanzen Zeitraumdimensionen definiert, wird der Versuch einer persistenten Verdichtung deshalb standardmäßig (Standardwert: **false**) mit Fehlermeldung abgebrochen. Wird der Wert dieses Parameters auf **true** gesetzt, wird auch bei Vorhandensein von Zeitraumdimensionen eine Verdichtung durchgeführt. Die Zeitraumdimensionen werden bei der Verdichtung nicht berücksichtigt, so dass deren Werte an der verdichteten Instanz nicht mehr verfügbar sind.

6.1.3.9 EPK-Import

Diese Parameter steuern Mechanismen, die von PPM im Rahmen des Datenimportes verwendet werden. Soweit nicht anders gekennzeichnet, werden diese Parameter nicht für den Prozessimport (runprocessimport) verwendet.

Schlüssel	Beschreibung
READ_RATE_EPC	<p>Um den notwendigen Hauptspeicherbedarf für Strukturinformationen zu beschränken, paketierte der PPM-Import die Fragmentdaten. Die so entstehenden Pakete werden anschließend in Einzeldurchläufen der Importkomponente abgearbeitet. Der angegebene Wert entspricht dabei der zugrundeliegenden Paketgröße.</p> <p>Durch eine Vergrößerung dieses Wertes kann der Overhead, der durch die Einzelbearbeitung der Komponenten (wie zum Beispiel die Mehrfachberechnung einzelner Instanzen) entsteht, reduziert werden. Allerdings muss darauf geachtet werden, dass für den PPM-Import ausreichend Hauptspeicher zur Ablage der größeren Menge von Strukturinformationen zur Verfügung steht. Die JVM-Parameter des PPM-Imports sollten gegebenenfalls für diese Mandanten im CTK angepasst werden.</p> <p>Standardwert: 100000</p>
XML_IMPORT_COMMIT_RATE	<p>Anzahl der Fragmente, die durch den XML-Import in einer Transaktion in die Datenbank geschrieben werden.</p> <p>Standardwert: 50000</p>
EPC_IMPORT_COMMIT_RATE	<p>Anzahl der Fragmente, die in der Phase „EPC-Import“ im Rahmen des PPM-Imports in einer Transaktion aus der Datenbank bearbeitet werden.</p> <p>Standardwert: 25000</p>

Schlüssel	Beschreibung
XML_IMPORT_WRITE_BUFFER	<p>Beim XML-Import werden die Fragmente durch einen gleichzeitig laufenden Hintergrundprozess in der Datenbank persistiert. Um Geschwindigkeitsschwankungen des Imports oder des RDBMS auszugleichen wird ein Puffer verwendet.</p> <p>Der Parameter steuert die Anzahl der Fragmente, die maximal in diesem Puffer zwischengespeichert werden. Wird die Puffergröße auf 0 reduziert, wird der Hintergrundprozess deaktiviert.</p> <p>Standardwert: 75</p>
KI_EPC_FUNCTION_COUNT_THRESHOLD	<p>Dieser Parameter wird im Rahmen der Behandlung großer EPKS verwendet. Er legt fest bis zu welcher maximalen Anzahl von Funktionen eine Prozessinstanz als „normal groß“ gilt.</p> <p>Details zu diesem Mechanismus entnehmen Sie bitte dem Handbuch PPM Datenimport.</p> <p>Dieser Parameter wird auch für den Prozessimport (runprocessimport) verwendet.</p> <p>Standardwert: 500</p>

PPM PRE-MERGER MECHANISMUS

Die Pre-Merger (XML und EPC) fassen eingelesene Fragmente zunächst in einem laufzeitbasierten Cache zusammen, bevor diese in der Datenbank persistiert werden. So kann die Anzahl der Datenbankobjekte deutlich reduziert und der Durchsatz des Imports und darauf aufbauender Phasen deutlich erhöht werden.

Die Effektivität des Pre-Mergers wird wesentlich durch die Reihenfolge der System-Events innerhalb der Eingabedateien beeinflusst. Falls möglich, sollten zusammengehörende Events, zum Beispiel solche mit identischem Prozessschlüssel, in den Eingabedateien möglichst dicht aufeinander folgen.

Als Standard verwendet PPM lediglich den XML-Pre-Merger. Dieser wird im Rahmen des XML-Importes verwendet. Er bearbeitet die Fragmente somit zum frühestmöglichen Zeitpunkt. Der EPC-Pre-Merger ist standardmäßig deaktiviert. Er kann verwendet werden, falls beim XML-Import sehr viele kleine Einzelimports erfolgen, so dass ein Zusammenfassen der Fragmente durch den XML-Pre-Merger nicht möglich ist.

Eine Vergrößerung der Pre-Merger-Caches kann genutzt werden, um den Bereich in dem zusammengehörende Fragmente erkannt werden zu vergrößern. Es ist darauf zu achten, dass dies zu einem erhöhten Speicherbedarf des XML- beziehungsweise PPM-Importes führt. Die JVM-Parameter dieser Komponenten sollten gegebenenfalls für diese Mandanten im CTK angepasst werden.

Da der Laufzeit-Cache beim Beenden einer Datenbanktransaktion geleert wird, sollte bei einer deutlichen Vergrößerung der Pre-Merger-Caches auch das Transaktionsgranulat `<XML/EPC>_IMPORT_COMMIT_RATE` (siehe oben) angepasst werden.

Schlüssel	Beschreibung
XML_IMPORT_PREMERGER_CACHE_SIZE	Größe des XML-Pre-Merger Cache. Standardwert: 5000
EPC_IMPORT_PREMERGER_CACHE_SIZE	Größe des EPC-Pre-Merger Cache. Standardwert: 0 (deaktiviert)

Wenn Sie für die Cache-Größe den Wert **0** angeben, wird der entsprechende Cache abgeschaltet. Die Einstellungen zum Pre-Merger betreffen den XML-Import (runxmlimport) bzw. PPM-Import (runppmimport) und sind nicht relevant für den Prozessimport (runprocessimport).

6.1.3.10 Help

In dieser Datei können optionale benutzerdefinierte Menüeinträge für den Client konfiguriert werden, welche weitere Webseiten anzeigen. Für unterschiedliche Sprachen können jeweils eigene URLs konfiguriert werden. Die URLs können absolut oder relativ bezogen auf die URL des Clients angezeigt werden (http(s)://host:port/ppm/html/). Bei Verwendung von PPM in der Cloud müssen bei Angabe absoluter URLs die aus dem Internet erreichbaren Host-Adressen eingestellt werden. Für nicht definierte Sprachen werden Rückfalleinträge konfiguriert.

In der URL kann ein Platzhalter **{0}** für den Sprachcode eingesetzt werden, der dann beim Aufruf aufgelöst wird.

So wird z. B. für die URL **http://host:port/help/{0}/help.html** für die Anzeige in Englisch der Platzhalter **{0}** durch **en** ersetzt und für die Anzeige in Deutsch mit **de**. Falls für alle Einträge die gleiche URL, bzw. eine URL mit Platzhalter verwendet wird, muss nicht für jeden Eintrag eine URL konfiguriert werden.

Falls alle sprachbezogenen URLs den gleichen sprachunabhängigen Menüeintrag haben, muss auch nur der Rückfalleintrag konfiguriert werden.

Schlüssel	Bedeutung
HELP.x.URL	URL des Rückfalleintrags. „x“ gibt einen Index beginnend mit „0“ für jeden Menüeintrag an.
HELP.x.MNI	Menüname des Rückfalleintrags.
HELP.x.<sprache>.URL	URL eines Menüeintrags für eine beliebige Sprache (für „<sprache>“ gelten die zweistelligen ISO-codes, z. B. „de“ für Deutsch.)
HELP.x.<sprache>.MNI	Menüname des Eintrags für eine beliebige Sprache.

Beispiele

Verwendung einer Platzhalter-URL und mehreren sprachabhängigen Einträgen:

HELP.0.URL = http://host:port/html/help/{0}/custom/KI_HELP.htm

HELP.0.MNI = KPI-Help

HELP.0.de.MNI = Kennzahlenhilfe

HELP.0.fr.MNI = L'aide de l'indicateur de performance

Verwendung von einem Eintrag und eigenen URLs für unterschiedliche Sprachen:

HELP.1.URL = <http://www.softwareag.com>

HELP.1.MNI = Software AG

HELP.1.de.URL = <http://www.softwareag.de>

HELP.1.fr.URL = <http://www.softwareag.fr>

Relative URLs:

HELP.2.URL = </docs/customizing/index.html>

HELP.2.MNI = Customizing Overview

6.1.3.11 Initdb

Steuert die erste Phase der Datenbankinitialisierung. Die in der referenzierten XML-Datei angegebenen Sprachschlüssel bestimmen die Standardsprache und die möglichen Alternativsprachen von PPM.

6.1.3.12 InitSystem

Steuert die zweite Phase der Datenbankinitialisierung, in der die Customizing-Dateien importiert werden. Die Datei ist abschnittsweise aufgebaut. Die Angaben eines Abschnitts werden als Argumente für intern durchgeführte Aufrufe des Konfigurationsprogramms **runppmconfig** verwendet. **XXX** entspricht einer fortlaufenden Nummerierung.

```
INIT_MODULE_XXX =  
INIT_MODULE_XXX_NAME = <Name>
```

Name der Konfigurationskomponente. Entspricht dem Argument **-command** des Tools **runppmconfig**.

```
INIT_MODULE_XXX_FILE = <Dateiname>
```

Zu verwendende XML-Datei.

6.1.3.13 Keyindicator

Steuert den Kennzahlenberechner. Nachfolgende Tabelle listet die Beschreibungen einer Auswahl der wichtigsten Schlüssel der Datei auf.

Schlüssel	Bedeutung
MAX_STEP_COUNT	Max. Anzahl von Iterationsschritten der Diagrammanzeige
MAX_TIME_STEP_COUNT	Max. Anzahl von Iterationsschritten der Dimension Zeit

Schlüssel	Bedeutung
MAX_DATASET_EXTENDER_SIZE	Wenn bei Kennzahlen des Ermittlertyps NUM_KEYINDICATOR und FREQ_KEYINDICATOR der Wert nicht ermittelt werden kann, wird die Ergebnismenge der Kennzahlenanfrage mit 0 angegeben. Ergebnismengen bis zu der hier angegebenen Größe werden berücksichtigt.
USE_KI_CACHE	true aktiviert den Zwischenspeicher der vorberechneten Favoriten.
AUTO_FILL_CACHE	true stößt automatisches Füllen des Cache nach dem Löschen an (nur bei USE_KI_CACHE=true)
FILL_CACHE_DELAY_TIME	Verzögerung in Millisekunden bis zum Beginn der Vorbereitung der Favoriten nach Beenden eines PPM-Datenimports
KI_LRU_CACHE_SIZE	Anzahl der im Hauptspeicherbasierten LRU-Kennzahlen-Cache (Least-Recently-Used) gespeicherten Anfragen. Empfohlener Wert: 50 . Der Wert 0 schaltet diese Option aus.
EPK_LRU_CACHE_SIZE	Anzahl der im Hauptspeicherbasierten LRU-EPK-Cache gespeicherten EPK-Anfragen. Empfohlener Wert: 50 . Der Wert 0 schaltet diese Option aus.
DEFAULT_LIMITPERC	Vorgegebener Schwellenwert (in Proz.) für die Anzeige von Analyseergebnissen im Process-Mining-Assistenten, ab welchem ungünstige Abweichungen vom Kennzahlwert bzw. auffällige Schwankungsbreiten von Dimensionswerten angezeigt werden sollen. Standardwert: 10.0%
DEFAULT_RELEVANCEPERC	Vorgegebener Schwellenwert (in Proz.) hinsichtlich der Relevanz von Dimensionswerten für die Anzeige von Analyseergebnissen im Process-Mining-Assistenten, ab welchem Dimensionswerte angezeigt werden sollen. Default-Wert: 10.0%
DEFAULT_NUMBEROFLINES	Max. Anzahl der Ergebniszeilen pro Prozesstyp im Process-Mining-Assistenten. Standardwert: 15
DEFAULT_DIMENSION_LEVEL_DELIMITER_IN_URL	Trenner zwischen den Stufenwerten einer Dimension
OUTLIER_SIGMA_VALUE	Sigma-Wert (Faktor) zur Berechnung des Grenzwertes in einer Standardberechnung
ORGUNITS_MAX_STRING_LENGTH	Maximale Länge des Namens einer Organisationseinheit (Standardwert: 100); wird in der Administration-Komponente der Organisationseinheiten verwendet.

Schlüssel	Bedeutung
FAVORITE_CACHE_LOGFILE	Name der Log-Datei für die Favoritenberechnung; leer bedeutet dass das Log abgeschaltet ist (Standard).
FAVORITE_CACHE_LOGFILE_THRESHOLD	Minimale Ausführungszeit (in Sekunden) die ein Log bewirkt (Standard: 0 , alle Berechnungen werden geloggt). Ist FAVORITE_CACHE_LOGFILE nicht spezifiziert wird dieser Wert ignoriert.
FAVORITE_CACHE_LOGFILE_PARAMETER_THRESHOLD	Minimale Ausführungszeit (in Sekunden) die ein zusätzliches Logging des zugrunde liegenden Paramsets bewirkt (Standard: -1 , kein zusätzliches Logging). Ist FAVORITE_CACHE_LOGFILE nicht spezifiziert wird dieser Wert ignoriert.
FAVORITE_CACHE_LOGFILE_LOG_CALCULATION_START	Wert TRUE oder FALSE . Bei TRUE wird jedesmal ein Logeintrag geschrieben wenn ein Favoriten-Thread eine neue Berechnung startet.

6.1.3.14 Mail

Der im Schlüssel **EMAIL_SERVER** angegebene Rechner (TCP/IP-Netzwerkname oder IP-Adresse) empfängt über SMTP-Protokoll die vom PPM-Server verschickten E-Mails.

Die im Schlüssel **EMAIL_FROM** angegebene E-Mail-Adresse wird als Absenderadresse für folgende Arten von E-Mails verwendet:

1. Als Absenderadresse für E-Mails, welche von der Automation erzeugt werden
2. Als Absenderadresse für E-Mails, welche von der Reportautomatisierung erzeugt werden, wenn für den ausführenden Benutzer keine E-Mail Adresse gepflegt ist oder aus anderen Gründen keine E-Mail Adresse ermittelt werden kann
3. Als Absenderadresse für E-Mails, welche aus dem Modul **Maßnahmen** erzeugt werden, wenn für den ausführenden Benutzer keine E-Mail Adresse gepflegt ist oder aus anderen Gründen keine E-Mail Adresse ermittelt werden kann
4. Als Absenderadresse für E-Mails, welche von dem Programm **runppmanalytics** erzeugt werden, wenn für den ausführenden Benutzer keine E-Mail Adresse gepflegt ist oder aus anderen Gründen keine E-Mail Adresse ermittelt werden kann.

Die Schlüssel **FORMAT** und **STYLE** spezifizieren die in der Aktivitätentabelle und der Nachrichtentabelle zu verwendenden Formate. Diese müssen in der Datei

Mail_settings.properties definiert sein.

```
REPORT_DETAIL_ACTIVITY_STYLE = null
REPORT_DETAIL_ACTIVITY_FORMAT = cpi_detail_html
```

```
REPORT_DETAIL_COMMENT_STYLE = null
REPORT_DETAIL_COMMENT_FORMAT = cpi_detail_html
```

```
REPORT_EMAIL_ACTIVITY_STYLE = null
REPORT_EMAIL_ACTIVITY_FORMAT = cpi_detail_plaintext
```

```
REPORT_EMAIL_COMMENT_STYLE = null  
REPORT_EMAIL_COMMENT_FORMAT = cpi_detail_plaintext
```

Die beiden Schlüssel **REPORT_EMAIL_ACTIVITY_TYPE** und **REPORT_EMAIL_COMMENT_TYPE** geben das Format der erzeugten E-Mail an. Mögliche Werte sind **text/plain** für E-Mails in einfacher Textform und **text/html** für E-Mails im html-Format.

```
REPORT_EMAIL_ACTIVITY_TYPE = text/html  
REPORT_EMAIL_COMMENT_TYPE = text/html
```

Im Schlüssel **EMAIL_ATTACH_RESULT** gibt man an, ob und wie die mit einer Nachricht verknüpfte Analyse hinzugefügt wird. Mögliche Werte:

Schlüsselwert	Beschreibung
false	Die verknüpfte Analyse wird nicht an die Nachricht angehängt.
inline	Die verknüpfte Analyse wird in die Nachricht eingebettet. Der Typ der Nachricht wird automatisch auf text/html gesetzt.
pdf	Die verknüpfte Analyse wird der Nachricht als pdf-Datei angehängt.

Die einer Nachricht hinzugefügte und verknüpfte Analyse wird über die jeweiligen, in den Schlüsseln **EMAIL_INLINE_STYLE** bzw. **EMAIL_PDF_STYLE** angegebenen Styles erzeugt.

Im Schlüssel **SEND_ANALYTICS_EMAILS** gibt man an, ob bei ungünstigen Abweichungen zusätzlich zum Erstellen einer CPI-Nachricht standardmäßig eine E-Mail versendet werden soll (Wert = **true**) oder nicht (Wert = **false**).

```
SEND_ANALYTICS_EMAILS = false
```

Für jeden Nachrichtentyp des Easy Mining (Frühwarn-, Planwert-, Alarmwert-, Auffälligkeits- und Ausreißeranalyse) kann man durch entsprechende Werte der folgenden Schlüssel festlegen, ob die Benachrichtigung per E-Mail (Schlüsselwert **EMAIL**), durch eine Nachricht im Modul **Verbesserungen** (Schlüsselwert **CPI**) oder beides (Schlüsselwert **CPI,EMAIL**) erfolgen soll.

```
TARGETVALUE_BEHAVIOUR=CPI  
PROCESSMINING_BEHAVIOUR=CPI  
ALARMVALUE_BEHAVIOUR=CPI  
EARLYALERT_BEHAVIOUR=CPI  
OUTLIER_BEHAVIOUR=CPI
```

Wenn man keinen Schlüsselwert angibt, wird eine Nachricht im Modul **Verbesserungen** erstellt und abhängig vom Wert des Schlüssels **SEND_ANALYTICS_EMAILS** zusätzlich eine E-Mail verschickt (Wert = **true**) oder nicht (Wert = **false**). Wenn man einen Schlüsselwert angibt, wird die Angabe im Schlüssel **SEND_ANALYTICS_EMAILS** ignoriert.

Warnung

Die Angaben der Werte **EMAIL** und **CPI** berücksichtigen die Groß-/Kleinschreibung. Falsche Angaben führen dazu, dass die entsprechende Aktion nicht ausgeführt wird.

6.1.3.15 ProcessImport

In dieser Datei werden die Einstellungen des Prozessimports (runppmprocessimport) vorgenommen.

Detaillierte Informationen erhalten Sie im Handbuch **PPM Operation Guide**.

6.1.3.16 RE (Relation Explorer)

Die Angaben in der Datei geben die Vorgabewerte der einzelnen Schlüssel wieder, die das Aussehen eines Relation Explorer-Diagrammes beeinflussen. Alle Parameter sind optional. Die Datei enthält ausführliche Kommentare, welche die Auswirkung der Einträge beschreiben.

Schlüssel	Beschreibung
DEFAULT_NODE_MODE	Knotendarstellungsmodus. Mögliche Werte: COMPLETE_MODE (OE in ARIS-Notation) oder SIMPLE_NODE (Knoten als farbig gefüllter Kreis). Vorgabewert: COMPLETE_MODE
DEFAULT_NODE_SIZE	Standardmäßiger Kreisdurchmesser (nur im Modus SIMPLE_NODE) bei kennzahlunabhängiger Darstellung (DEFAULT_SELFEDGE_MODE=false)
DEFAULT_EDGE_THICKNESS_MODE	Modus der Kantenstärkendarstellung in Abhängigkeit von: - dem prozentualen Anteil ausgehender Kanten (OUTGOING_MODE) - dem prozentualen Anteil eingehender Kanten (INCOMING_MODE) - dem absoluten Kennzahlwert (VALUE_MODE) Darstellung einer einheitlichen Kantenstärke: NONE_MODE
DEFAULT_EDGE_VALUE_MODE	Darstellung der Kantenwerte in Abhängigkeit von: - dem prozentualen Anteil ausgehender Kanten (OUTGOING_MODE) - dem prozentualen Anteil eingehender Kanten (INCOMING_MODE) - dem absoluten Kennzahlwert (VALUE_MODE) Keine Darstellung von Kantenwerten: NONE_MODE
DEFAULT_EDGE_THICKNESS	Kantenstärke bei einheitlicher Darstellung (nur im Modus NONE_MODE)
DEFAULT_EDGE_COLOR	Kantenfarbe im RGB-Format
DEFAULT_NODE_COLOR	Knotenfarbe im RGB-Format
DEFAULT_BACKGROUND_COLOR	Hintergrundfarbe im RGB-Format

Schlüssel	Beschreibung
DEFAULT_SELFEDGE_MODE	Visualisierung der internen Kommunikation eines Knotens (abhängig vom jeweiligen Kennzahlwert) anhand des Kreisdurchmessers (nur im Modus SIMPLE_NODE). Mögliche Werte: - true (anzeigen) - false (nicht anzeigen)
DEFAULT_EDGE_PIXEL_RANGE	Wertebereich (Fließkommazahlen) für Kantenstärke in Pixeln (min.,max.) bei abhängiger Darstellung (alle Modi außer NONE_MODE), z. B. DEFAULT_EDGE_PIXEL_RANGE=1,3.5
DEFAULT_NODE_PIXEL_RANGE	Wertebereich für Kreisdurchmesser in Pixeln (min., max.) bei vom Kennzahlwert abhängiger Darstellung (DEFAULT_SELFEDGE_MODE=true)
UNDER_WARNING_EDGE_COLOR	Kantenfarbe im RGB-Format für Kennzahlwerte, die besser als der gepflegte Warnwert (= Planwert 1 der Kennzahl) sind.
WARNING_EDGE_COLOR	Kantenfarbe im RGB-Format für Kennzahlwerte, die zwischen dem Warnwert (= Planwert 1) und dem Alarmwert (= Planwert 2) liegen.
ALARM_EDGE_COLOR	Kantenfarbe im RGB-Format für Kennzahlwerte, die schlechter sind als der Alarmwert (= Planwert 2).

FARBANGABE IM RGB-FORMAT

Die Farbwerte werden als Helligkeitsstufen der drei Grundfarben **Rot**, **Grün** und **Blau** angegeben (Farb-Triple). Die Angabe erfolgt durch drei ganze Zahlen zwischen 0 und 255 (entspricht 256 Stufen). Der erste Wert gibt die Abstufung des Rotanteils, der zweite die Abstufung des Grünanteils und der dritte Wert die Abstufung des Blauanteils an. 0 entspricht der geringsten und 255 der höchsten Helligkeitsstufe eines Grundfarbwertes. Grautöne entstehen durch gleichwertige Abstufungsangaben, z. B. 240,240,240.

Beispiele

Farbe	RGB-Format
Schwarz	0,0,0
Dunkelgrau	64,64,64
Sattes rot	255,0,0
Sattes grün	0,255,0
Sattes blau	0,0,255
Braun	165,44,42
Gold	255,215,0
Hellgrau	240,240,240

Farbe	RGB-Format
Weiß	255,255,255

6.1.3.17 RMI-Server

Die Angaben in dieser Datei konfigurieren den Datenaustausch über das RMI-Protokoll. Der Schlüssel **RMIObjectsPort** bestimmt die vom PPM-Mandantenserver verwendete Port-Nummer. Vom PPM-Mandantenserver erzeugte RMI-Objekte werden mit dieser Port-Nummer an der RMI-Registry angemeldet.

Mit dem Schlüssel **RMIObjectsPortAnalysisServer** wird die Port-Nummer angegeben, über die der Analyseserver kommuniziert.

Der Schlüssel **RMIConnectionFactory** bestimmt die Art des RMI-Datenaustausches.

Wert	Beschreibung
com.idsscheer.ppm.rmi.compress.ZCompressionSocketFactory	Komprimierter RMI-Datenaustausch (PPM-Standard)
com.idsscheer.ppm.rmi.ZDefaultSocketFactory	RMI-Datenaustausch im Klartext
keine Angabe	Native RMI-Kommunikation default

Wenn die SSL-Verschlüsselung eingeschaltet wurde (Schlüssel **UseSSL=true** in der Datei **Registry_settings.properties**), wird der Schlüssel **RMIConnectionFactory** ignoriert und als RMI-ConnectionFactory automatisch die **ZSSLConnectionFactory** verwendet.

6.1.3.18 Report

Die mehrstufigen Schlüssel unterscheiden die unterschiedlichen Ausgabeformate der PPM-Report-Komponente. Auf Basis des Verzeichnisses **report** im Mandantenkonfigurationsverzeichnis sind für jedes Ausgabeformat spezifische Definitionsdateien referenziert. Die Schlüssel sind in der Datei mit Kommentaren versehen.

Schlüssel	Wert	Beschreibung
editor.showInfoMessages	Boolean	Standardwert ist FALSE Zeigt nach dem Öffnen und dem Speichern einer Exportdefinition einen Hinweisdiallog an, wenn der Wert auf TRUE gesetzt wurde.
editor.execute.autosave		Standardwert ist TRUE Wenn der Wert auf TRUE gesetzt wurde, wird die Exportdefinition vor dem Ausführen automatisch gespeichert.

Schlüssel	Wert	Beschreibung
mnemonic_date_timezone		Standardwert ist GMT+0:00 Mit dieser Einstellung kann die Zeitzone geändert werden für das Datenfeld Datum . Diese Einstellung gilt nur für den Export.

EXPORTFORMATE DEFINIEREN

Die in PPM wählbaren Exportformate werden in dem Schlüssel **exportformats** definiert. Standardmäßig stehen für Exportdefinitionen die Formate **XML**, **CSV** und **CSVF** zur Verfügung und sind im Dialog **Exporteigenschaften** wählbar.

Zu jedem Format existiert ein Satz von Einstellungen, die für ein neues Format kopiert und entsprechend angepasst werden müssen.

Die Werte des Schlüssels müssen in Großbuchstaben angegeben sein.

AUSGABEVERZEICHNIS DER REPORTAUTOMATISIERUNG

Der Schlüssel **reportautomation.result_directory** gibt ein Verzeichnis an (standardmäßig <installationsverzeichnis>/ppmmashzone/server/bin/work/data_ppm/reportautomation), in dem die Ergebnisse der Reportautomatisierung abgelegt werden. Für jeden Mandanten wird jeweils ein eigenes Unterverzeichnis stellt.

LOKALISIERUNG VON E-MAIL-TEXTEN

Der Schlüssel **reports.resource_directory** gibt ein Verzeichnis an (standardmäßig <installationsverzeichnis>\ppmmashzone\server\bin\work\data_ppm\config\report_resource_<Sprachkürzel>.xml, z. B. **report_resource_en.xml**.

In verschiedenen kontextspezifischen XML-Strukturelementen sind die Texte im Feld **PCDATA** der jeweiligen XML-Elemente **resourceelement** angegeben. Im Attribut **name** des XML-Elementes **resourceelement** ist der Kontext des entsprechenden Textes angegeben.

E-Mail-Texte für noch nicht unterstützte Sprachen können effizient erstellt werden, indem die vorhandene Datei **report_resource_en.xml** kopiert und entsprechend dem Sprachkürzel des gewünschten Landes umbenannt wird (z. B. **report_resource_es.xml** für spanisch). Alle im Feld **PCDATA** angegebenen Texte dieser neu erzeugten Ressource-Datei können anschließend durch Bearbeiten in einem Texteditor in die gewünschte Landessprache übersetzt werden.

LOKALISIERUNG VON CPI-TEXTEN

Oberflächentexte und Texte in einer erzeugten E-Mail, die Ausgaben bestimmter Werte enthalten, die aktuell von PPM übergeben werden (z. B. Plan- und Kennzahlenwerte), kann man anpassen. Dazu gibt man bestimmte XML-Kindelemente im XML-Element **cpiresource** der Datei **report_resource.xml** an. Im XML-Attribut **name** dieser Kindelemente wird die interne Resource-ID angegeben, die dann durch den im Feld **PCDATA** angegebenen Text überschrieben wird:

```
<resourceelement name="<Resource-ID>"><Text></resourceelement>
```

Beispiel (Auszug report_resource.xml)

```
<resource>
...
  <cpiresource>
...
    <resourceelement name="state">Status</resourceelement>
...

```

FRÜHWARNÜBERPRÜFUNG

TOPIC: BETREFF

Resource-ID	ZEarlyAlertChecker.subject.STR
Variable {0}	Frühwarndimension
Variable {1}	Prozesstyp
Text	Kritische Prozessinstanzen bezüglich "{0}" ("{1}")

Resource-ID	ZEarlyAlertChecker.message.exceeding_non.non_tv.war n_not_set.STR
Variable {0}	Prozesstypgruppe
Variable {1}	Frühwarndimension
Text	Unter der Prozesstypgruppe "{0}" wurden für die Frühwarndimension(en) "{1}" kritische Prozessinstanzen ermittelt.

TOPIC: NACHRICHTENTEXTE

Resource-ID	ZEarlyAlertChecker.message.clickHere.STR
Text	Klicken Sie hier:

Resource-ID	ZEarlyAlertChecker.message.Information.STR
Text	Hinweis: Es traten Probleme bei der Verbindung zum System auf. Dadurch wurden wahrscheinlich nicht alle Prozessinstanzen bei der Überprüfung berücksichtigt.

Resource-ID	ZEarlyAlertChecker.message.MessageDetails.STR
Text	Details entnehmen Sie bitte folgender Meldung bzw. setzen Sie sich mit Ihrem Systemadministrator in Verbindung:

PLANWERTÜBERPRÜFUNG

TOPIC: BETREFF

Resource-ID	ZPlannedValueAnalyticManager.subject.under_tv.STR
Variable {0}	Kennzahl
Variable {1}	Prozesstyp
Text	Planwertunterschreitung für Kennzahl "{0}" ("{1}")

Resource-ID	ZPlannedValueAnalyticManager.subject.over_tv.STR
Variable {0}	Kennzahl
Variable {1}	Prozesstyp
Text	Planwertüberschreitung für Kennzahl "{0}" ("{1}")

TOPIC: NACHRICHTENTEXTE

Die nachfolgend beschriebenen Resource-ID / Textzuordnungen verwenden jeweils folgende Variablen:

Variable {0}	Istwert
Variable {1}	Alarmwert
Variable {2}	Warnwert

Resource-ID	ZPlannedValueAnalyticManager.message.exceeding_warn1.over_tv.STR
Text	Planwert 1 ({2}) wurde überschritten. Der Istwert beträgt {0}. Planwert 2 beträgt {1}.

Resource-ID	ZPlannedValueAnalyticManager.message.exceeding_warn1.under_tv.STR
Text	Planwert 1 ({2}) wurde unterschritten. Der Istwert beträgt {0}. Planwert 2 beträgt {1}.

Resource-ID	ZPlannedValueAnalyticManager.message.exceeding_warn2.over_tv.warn_set.STR
Text	Planwert 2 ({1}) wurde überschritten. Der Istwert beträgt {0}. Planwert 1 beträgt {2}.

Resource-ID	ZPlannedValueAnalyticManager.message.exceeding_warn2.under_tv.warn_set.STR
Text	Planwert 2 ({1}) wurde unterschritten. Der Istwert beträgt {0}. Planwert 1 beträgt {2}.

Resource-ID	ZPlannedValueAnalyticManager.message.exceeding_warn2.over_tv.warn_not_set.STR
Text	Planwert 2 ({1}) wurde überschritten. Der Istwert beträgt {0}. Planwert 1 wurde nicht definiert.

Resource-ID	ZPlannedValueAnalyticManager.message.exceeding_warn2.under_tv.warn_not_set.STR
Text	Planwert 2 ({1}) wurde unterschritten. Der Istwert beträgt {0}. Planwert 1 wurde nicht definiert.

TOPIC: TEXT FÜR DEN FILTER DER PLANWERTDEFINITION

Resource-ID	ZPlannedValueAnalyticManager.filter.STR
Text	Die Planwertdefinition bezieht sich auf folgende Filter:

ALARMWERTÜBERPRÜFUNG

TOPIC: BETREFF

Resource-ID	ZAlarmValueAnalyticManager.subject.under_tv.STR
Variable {0}	Frühwarndimension
Variable {1}	Prozesstyp
Text	Alarmwertunterschreitung für Kennzahl "{0}" (" {1}")

Resource-ID	ZAlarmValueAnalyticManager.subject.over_tv.STR
Variable {0}	Frühwarndimension
Variable {1}	Prozesstyp
Text	Alarmwertüberschreitung für Kennzahl "{0}" (" {1}")

TOPIC: NACHRICHTENTEXTE

Resource-ID	ZAlarmValueAnalyticManager.message.exceeding_alarm.under_tv.warn_not_set.STR
Variable {0}	Alarmwert

Text	Der Alarmwert "{0}" wurde unterschritten.
Resource-ID	ZAlarmValueAnalyticManager.message.exceeding_alarm .over_tv.warn_not_set.STR
Variable {0}	Alarmwert
Text	Der Alarmwert "{0}" wurde überschritten.

PROCESSMINING

TOPIC: BETREFF

Resource-ID	ZProcessMiningAnalyticManager.subject.STR
Text	Auffälligkeiten von Process Mining ermittelt

TOPIC: NACHRICHTENTEXTE

Resource-ID	ZProcessMiningAnalyticManager.message.exceeding_no n.non_tv.warn_not_set.STR
Text	Das Process Mining hat Auffälligkeiten ermittelt.

AUSREIßERANALYSE

TOPIC: BETREFF

Resource-ID	ZOutlierAnalyticManager.subject.STR
Variable {0}	Kennzahl
Variable {1}	Prozesstyp
Text	Ausreißer wurden gefunden für Kennzahl "{0}" (" {1}")

TOPIC: NACHRICHTENTEXTE

Resource-ID	ZOutlierAnalyticManager.message.exceeding_non.over_ tv.warn_not_set.STR
Variable {0}	Grenzwert
Text	Die Ausreißeranalyse hat Ausreißer für den Grenzwert größer "{0}" gefunden.

Resource-ID	ZOutlierAnalyticManager.message.exceeding_non.under_ _tv.warn_not_set.STR
Variable {0}	Grenzwert
Text	Die Ausreißeranalyse hat Ausreißer für den Grenzwert kleiner "{0}" gefunden.

6.1.3.19 Server

Mit den Angaben in dieser Datei werden Einstellungen des PPM-Servers vorgenommen. Die folgende Tabelle listet die wichtigsten Einstellungsmöglichkeiten auf:

Schlüssel	Wert	Beschreibung
PRINT_STACKTRACE_ON_EXCEPTION	Schalter	Wird verwendet, um im Fehlerfall zusätzliche interne Details zu Fehlermeldungen zu erhalten. TRUE aktiviert die erweiterte Ausgabe von Fehlermeldungen („Stacktraces“). Standardwert: FALSE
ANALYSIS_SERVER_CONNECT_RETRY_INTERVAL	Zahl	Anzahl der Millisekunden, die der PPM-Server bei fehlgeschlagenem Verbindungsaufbau zum Analyseserver wartet, bevor er einen erneuten Verbindungsversuch startet. Standardwert: 1500
ANALYSIS_SERVER_MAX_CONNECT_RETRIES	Zahl	Maximale Anzahl der Verbindungsversuche zwischen PPM und Analyseserver, bevor ein bis dato fehlgeschlagener Verbindungsaufbau generell abgebrochen wird. Standardwert: 120
ENCODING	Text	Legt die Zeichensatzkodierung exportierter XML-Dateien fest. Mögliche Werte: ISO-8859-1, UTF-8 Dieser Wert wird durch das Mandanten-Setup gesetzt und darf nachträglich nicht mehr geändert werden.
PREV_PERIOD.TOLERANCE_IN_PERCENT	Zahl	Prozentuale Toleranz, innerhalb derer Vergleichswerte als gleich dem Kennzahlwert gewertet werden. Wird in der EPK-Ansicht zur Darstellung von Trends verwendet. Standardwert: 1
SUBSERVER_RETRY_WAIT_TIME	Zahl	Anzahl der Millisekunden, die der Master bei fehlgeschlagenem Verbindungsaufbau zum Sub-Server wartet, bevor er einen erneuten Verbindungsversuch startet. Standardwert: 10000
SUBSERVER_MAX_CONNECT_RETRIES	Zahl	Maximale Anzahl der Verbindungsversuche von Master zu Sub-Server, bevor ein bis dato fehlgeschlagener Verbindungsaufbau generell abgebrochen wird. Standardwert: 10
SUBSERVER_RETRY_WAIT_TIME	Zahl	Wartezeit in Millisekunden, nach der ein erneuter Verbindungsversuch zum Sub-Server gestartet wird. Standardwert: 10000

Schlüssel	Wert	Beschreibung
SERVER_MODE	Text	Bestimmt die Betriebsart des Servers. Mögliche Werte: STANDALONE, SUBSERVER, MASTER . Dieser Wert wird durch das Mandanten-Setup gesetzt.
HTTPS_DISABLE_CERTIFICATE_AUTHORITY_VERIFICATION	Schalter	True schaltet die Prüfung der Zertifizierungsstelle Standardwert: false . Wenn der Schlüssel fehlt oder einen beliebig anderen Wert besitzt, gilt der Standardfall und die Prüfung erfolgt strikt. Ist die Prüfung deaktiviert, dann wird bei sämtlichen https-Verbindungen zwischen dem PPM-Server und anderen Hosts auf die Prüfung der Zertifizierungsstelle verzichtet. Dies gilt nicht nur für Verbindungen zu Zentrale Benutzerverwaltung. Die Verbindung zur Ermittlung der verfügbaren Dashboards und Management Views ist davon ausgenommen. Diese erfolgt immer ohne Prüfung der Zertifizierungsstelle. Die RMI-Verbindung wird von diesen Einstellungen nicht beeinflusst. Weitere Informationen zur HTTPS-Unterstützung finden sich im Handbuch PPM Operation Guide .
HTTPS_DISABLE_CERTIFICATE_HOSTNAME_VERIFICATION	Schalter	True schaltet die Prüfung des Host-Names ab, wenn bei der SSL-Kommunikation die Hostname-Validierung des Zertifikats entfallen soll. Standardwert: false . Wenn der Schlüssel fehlt oder einen beliebig anderen Wert besitzt, gilt der Standardfall und die Prüfung erfolgt strikt. Dieser Eintrag gilt für alle https-Verbindungen. Die RMI-Verbindung wird von diesen Einstellungen nicht beeinflusst. Weitere Informationen zur HTTPS-Unterstützung finden sich im Handbuch PPM Operation Guide .
LANGUAGE	Text	Sprache mit der der PPM-Server betrieben werden soll (nach ISO 639-1. Wenn die angegebene Sprache vom Mandanten nicht unterstützt wird, so wird die Standardsprache des Mandanten verwendet) Standwert: „" (nicht gesetzt)
QUERY_CONNECTION_POOL_SIZE	Zahl	Größe des Query Connection Pool für Anfragen (> 0). Standardwert: 16

Schlüssel	Wert	Beschreibung
COCKPIT_THREAD_COUNT	Zahl	Anzahl der Threads die für die Berechnung von Cockpit Anfragen verwendet werden. (-1 für unbegrenzte Thread-Anzahl) Standardwert: 8
USER_ADMIN_LIST_LIMIT	Zahl	Anzahl der maximal angezeigten Benutzer in der Benutzerverwaltung bei Verwendung von LDAP (Legacy oder UMC) Standardwert: 100
PRIORITY_FREQUENCY_QUERY	Zahl	Priorität der Häufigkeitsanfragen (Ganzzahl ≥ 0). Kleinere Werte haben höhere Priorität. Standardwert: 8
REPORT_TOKEN_TIMEOUT	Zahl	Maximale Laufzeit (in Sekunden) für Report-Sessions (UMC). Standardwert: 604800 (60*60*24*7)
IMPORT_TOKEN_TIMEOUT	Zahl	Maximale Laufzeit (in Sekunden) für Import- und Interne-Sessions (UMC). Standardwert: 604800 (60*60*24*7)
UMC_SESSION_REFRESH	Zahl	Erweiterung des Ablaufzeitpunkts (Timeout) der UMC-Session. Wird mit dem in der UMC festgelegten Wert (Standard: 60 Sekunden) multipliziert. Standardwert: 15 (somit 15 * 60 Sekunden = 15 Minuten)
COMPARE_MASTER_SUBSERVER_CONFIGS	Schalter	Gibt an ob die Konfiguration zwischen dem Master und den Sub-Servern verglichen wird. Der Parameter ist nur relevant in der Betriebsart MASTER . Mögliche Werte TRUE oder FALSE . Die Standardeinstellung TRUE bewirkt einen Vergleich der Konfiguration des Masters mit der Konfiguration der Sub-Server. Bei unterschiedlicher Konfiguration wird der jeweilige Sub-Server nicht mit dem Master verbunden. Die Einstellung FALSE kann zu fehlerhaftem und unvorhersehbarem Verhalten führen, und sollte ausschließlich nach Anweisung durch die Software AG gesetzt werden!
HTTP_PORT	Zahl	Gibt die Port-Nummer an, mittels der der Load Balancer mit dem PPM-Server kommuniziert.
WEB_SERVICE_SESSION_TIMEOUT	Zahl	Gültigkeitsdauer in Millisekunden bis zum Ablauf einer Verbindung. Wenn während dieser Zeitspanne keine Kommunikation mit dem Server stattfindet, wird der Benutzer abgemeldet.

Schlüssel	Wert	Beschreibung
WEB_SERVICE_SESSION_TIMEOUT_RENEWAL	Zahl	Zeitspanne in Millisekunden, in der ein angemeldeter Client dem Server zum Erhalt der Session ein Ping-Signal sendet.
WEB_SERVICE_MIN_THREADS	Zahl	Minimale Anzahl der Threads für Webservice-Anfragen
WEB_SERVICE_THREAD_IDLE_TIMEOUT	Zahl	Maximale Dauer eines inaktiven Threads. Nach Ablauf der Dauer ohne Aktivität wird der Thread beendet.
WEB_SERVICE_REQUEST_TIMEOUT	Zahl	Maximale Wartezeit auf eine Antwort des Servers. Wenn eine Anfrage an den Server länger dauert, als die in diesem Wert angegebene Dauer, schickt der Server einen Timeout an den Client, damit der Client das Ergebnis erneut anfragen kann. Dieser Wert muss kleiner sein, als die Abbruchzeit zwischen den vorgeschalteten Load Balancern und dem Mandantenserver.

Während des Mandanten-Setup des Master-Servers werden alle Sub-Server angegeben, die der Master verwenden soll. Die Angaben werden in den folgenden Einträgen gespeichert (X ist Platzhalter für die Nummer eines Sub-Servers, Wertebereich 1-64):

Schlüssel	Wert	Beschreibung
SUBSERVER.X.URL	URL	Registry-URL, über die der Sub-Server angesprochen wird
SUBSERVER.X.CLIENT	Text	Name des Mandanten
SUBSERVER.X.USER	Text	Name des PPM-Benutzers, unter dessen Kennung sich der Master-Server mit dem Sub-Server verbindet
SUBSERVER.X.PASSWORD	Text	Verschlüsseltes Kennwort des PPM-Benutzers
PIKI_SUBSERVER	Zahl	Bestimmt den Sub-Server (Wert entspricht der Nummer des Sub-Servers), der die Kennzahlenreihen für die Dateneingabe prozessinstanzunabhängiger Kennzahlen im PPM-Frontend enthält.

Zur Kommunikation des Master-Servers mit den Sub-Servern soll ein Systemadministrator-Benutzer verwendet werden. Dadurch ist sichergestellt, dass der Master-Server mit allen Systemberechtigungen auf die Sub-Server zugreifen kann.

Die Daten für prozessinstanzunabhängige Kennzahlenreihen sollten immer auf dem unter PIKI_SUBSERVER angegebenen Sub-Server importiert werden.

Warnung

Das Verteilen von Daten einer prozessinstanzunabhängigen Kennzahlenreihe auf mehrere Sub-Server kann zu falschen Ergebnissen führen.

6.1.3.20 Server_Log

Mit den Angaben in dieser Datei werden die Einstellungen für die Systemmeldungen des PPM-Servers vorgenommen.

Detaillierte Informationen erhalten Sie im Handbuch **PPM Operation Guide**.

6.1.3.21 Sysmon

Für den Systemmonitormandanten werden folgende Einstellungen gesetzt. Diese Parameter sind nicht relevant für den für Prozessimport (runprocessimport):

Schlüssel	Wert	Beschreibung
GENERATE_EVENT_XML	Schalter	Wird verwendet, um während des PPM-Imports Event-Dateien für den Systemmonitormandanten zu erzeugen. TRUE aktiviert die Erzeugung. Standardwert: FALSE
PPM_EVENT_DIR	Text	Speicherort für die generierten Event-Dateien. Standardwert: <installationsverzeichnis>/ppmmashzone/server/bin/work/data_ppm/custom/sysmon/data/SysmonData
CHECK_DB_STATUS	Schalter	Abfragen des DB-Status bei bestimmten Messpunkten. TRUE kann die Performance negativ beeinflussen. Standardwert: FALSE
ATTR_AT_PPM_CUSTOMER	Text	Benutzerdefinierte Attribute können mit dem Präfix ATTR_ spezifiziert werden. Diese werden an jedes Event kopiert. Diese Attribute müssen im Customizing gepflegt werden und es muss ein Mapping dafür erstellt werden.

6.1.3.22 Templates

Für Management Views, Reporte sowie Performance Cockpits und Performance Dashboards werden folgende Einstellungen gesetzt:

Teilschlüssel	Bedeutung
BaseDirectory	Verzeichnis mit den jeweiligen Definitionsdateien (Zugriff über das Dateisystem). Der Systembenutzer, der die PPM-Software ausführt, benötigt in diesem Verzeichnis die Zugriffsart Vollzugriff .
BaseURL	Für Management Views wird der URL-Zugriff über HTTP, für Reporte über FILE auf das unter BaseDirectory angegebene Verzeichnis angegeben.

Teilschlüssel	Bedeutung
Filter	Dateierweiterung der Management Views bzw. Reportdefinitionsdateien.
PerformanceDashboard.Context	Name des Dashboard-Context für den Mandanten
QueryAPI.Context	Name des QueryAPI-Context für den Mandanten
QueryApi.PPMUIMode	Sprungziel für vom QueryAPI erzeugte PPM-URLs. Mögliche Werte: APPLET, WEBSTART. APPLET gibt die index.html und WEBSTART gibt die ppm_<client>.jnlp als Zieladresse für Sprünge zur PPM-UI aus dem Performance Dashboard, aus Management Views und aus MashZone an. Default-Wert: WEBSTART
MashZone.UseSSO	Single-Sign-On für MashZone. Mögliche Werte: OTP, SAML2. Default-Wert: OTP
Optimize.UseSSO	Single-Sign-On für Optimize. Einzig möglicher Wert: SAML2

Das Base-Directory und die Base-URL werden von den Templates, also von den angemeldeten Reporten und ManagementViews benötigt, wobei das BaseDirectory der alten MVs nur noch zum Abspeichern von dem alten MV-Editor benötigt wurde.

Die BaseURL der ManagmentViews wird über die URL aus dem Apache ZooKeeper geladen und wird deshalb nicht mehr angegeben. Die BaseURL für die Reporte muss nur der Vollständigkeit halber angegeben werden.

6.1.3.23 MT_Export

Für den Export von Minitab-Grafiken werden folgende Einstellungen gesetzt:

Teilschlüssel	Bedeutung
MT_graphic_heigth	Gibt die Höhe einer Grafik an. Der Standardwert ist 480.
MT_graphic_width	Gibt die Breite einer Grafik an. Der Standardwert ist 760.