PPM CUSTOMIZING

ARIS Process Performance Manager 10

Oktober 2026

This document applies to ARIS Process Performance Manager Version 10 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2000-2026 SAG Aris GmbH, Saarbrücken, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name ARIS and all SAG Aris GmbH product names are either trademarks or registered trademarks of SAG Aris GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

The Software may contain or be distributed with certain commercial third party components (“Third Party Components”), which may be subject to special license terms and conditions (“Third Party Terms”) located at https://aris.com/terms.

Third Party Terms include important licensing and warranty information and disclaimers of third party licensors. For the avoidance of doubt, if Customer uses the Software in accordance with the Documentation, the Third Party Terms do not restrict the rights granted under this Agreement. In the event that a third party manufacturer no longer provides active support for any Third Party Component, Supplier will use commercially reasonable efforts to respond to any issues with the Software related to the Third Party Components.

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with SAG Aris GmbH and/or its subsidiaries and/or its affiliates and/or their licensors.

Inhaltsverzeichnis

1. Allgemein 6
2. Übersicht 7
2.1. Konfigurationskomponenten 7
2.2. Kommandozeilenprogramm 8
2.3. Methodisches Vorgehen 8
2.4. Hierarchie der Konfigurationskomponenten 9
3. Oberflächensprachen 11
3.1. Oberflächensprachen zur Benutzerführung 11
3.2. Oberflächensprache zur Anzeige von Konfigurationselementen 12
3.2.1. Verwendung von Multibyte-Zeichensätzen für Konfigurationselemente 13
4. Interne Namen 15
5. Attributtypen und Attributtypgruppen 16
5.1. Datentypen 16
5.1.1. Interne Datentypen 16
5.1.2. Benutzerdefinierte Datentypen 18
5.1.2.1. Benutzerdefinierte Datentypen in Multibyte-Zeichensätzen 19
5.2. Definition von Attributtypen und Attributtypgruppen 19
5.2.1. Definition von Attributtypen 20
5.2.2. Definition von Attributtypgruppen 20
5.2.3. Konfiguration von Attributtypen und Attributtypgruppen 20
5.2.3.1. Attributtyp- und Attributtypgruppendefinition in Multibyte-Zeichensätzen 22
6. Prozesszusammenführung 24
6.1. Prozesshierarchien 24
6.2. Schlüsselregeln 25
6.2.1. Prozessschlüsselregeln 26
6.2.2. Hierarchieschlüsselregeln 27
6.2.3. Shared-Fragment-Schlüsselregeln 28
6.2.4. Merge-Schlüsselregeln 29
6.2.4.1. Schlüsselbasierter Merge 31
6.2.4.2. Ordnungsbasierter Merge 32
6.2.4.3. Kombination von Merge-Verfahren 33
6.2.5. Objektschlüsselregeln 35
6.2.6. Ausgabeverhalten von Meldungen 35
6.2.7. Konfigurationsdatei 36
6.3. Merge der Prozessfragmente 38
6.3.1. Merge-Modus Ersetzen 38
6.3.2. Merge-Modus Zusammenführen 39
6.4. Zusammenführen von Ereignissen 41
6.4.1. Parallele Pfade mit mehrwertigen Schlüsseln 41
6.4.2. Merge-Modus 43
6.5. Attributkopierregeln 44
6.6. Anonymisieren 45
7. Prozesstypisierung 48
7.1. Typisierungsregeln erstellen 48
7.1.1. Kennzahlenkonfiguration 48
7.1.2. Prozessbaumkonfiguration 49
7.1.2.1. Priorisierung 50
7.1.3. Definition von Attributberechnungen 50
7.1.3.1. Berechnungsklassen 53
7.1.3.2. Operanden 72
7.1.3.3. Bedingter Attributtypzugriff 79
7.1.3.4. Operatoren 80
7.1.3.5. Verschachtelung von Operatoren 119

7.1.3.6. Berechnungsfunktionen ... 120
7.1.3.7. Wandlung des Attributtyps ... 122
7.1.3.8. Zusammenfassung ... 122
7.1.3.9. Beispiele Attributberechnungen ... 122
7.1.3.10. Besonderheiten der Attributberechnung ... 127
7.1.4. Typisierungsregeln im CTK ... 127
7.2. Typisierung durch Attributberechnung ... 128
8. Definition von Kennzahlen, Dimensionen, Attributberechnungen und Beziehungen ... 129
8.1. Begriffserläuterungen ... 129
8.1.1. Kennzahlen ... 129
8.1.1.1. Prozessinstanzabhängige Kennzahlen ... 130
8.1.1.2. Prozessinstanzunabhängige Kennzahlen (PIKIs) ... 130
8.1.2. Dimensionen ... 131
8.2. Definition von Kennzahlen ... 131
8.2.1. Definition von Standardkennzahlen ... 132
8.2.1.1. Formatierung von Kennzahlwerten ... 134
8.2.1.2. Definition von Prozesskostenkennzahlen ... 134
8.2.2. Kennzahldefinition in Multibyte-Zeichensätzen ... 136
8.2.3. Definition von Kardinalitätskennzahlen ... 136
8.2.4. Definition prozessinstanzunabhängiger Kennzahlen ... 137
8.2.4.1. Verwendung (Typ) einer Datenreihe ... 141
8.2.4.2. Dimensionsbezug ... 142
8.2.4.3. Definition prozessinstanzunabhängiger Kennzahlen in Multibyte-Zeichensätzen ... 143
8.2.4.4. Konfigurationsimport ... 144
8.2.4.5. Migration von Datenreihen ... 145
8.2.4.6. Zusätzliche Information: Benutzerdefinierte Kennzahlen auf Basis von prozessinstanzabhängigen Kennzahlen ... 146
8.2.5. Definition von Kennzahlengruppen ... 147
8.2.5.1. Sichtbare Kennzahlengruppen ... 150
8.2.5.2. Gruppe der unsichtbaren Kennzahlen ... 151
8.3. Definition von Dimensionen ... 152
8.3.1. Definition von Dimensionengruppen ... 152
8.3.2. Textdimensionen ... 154
8.3.2.1. Allgemeine XML-Struktur ... 154
8.3.2.2. Konfiguration ... 159
8.3.2.3. Import von Dimensionswerten ... 163
8.3.3. Fließkommadimensionen ... 163
8.3.4. Dimensionen Zeit ... 164
8.3.4.1. Zeitdimensionen für das Frühwarnsystem ... 165
8.3.4.2. Besonderheit der Berechnung kritischer Zeitattribute ... 166
8.3.5. Zeitraumdimensionen ... 168
8.3.6. Tageszeitdimensionen ... 170
8.3.7. Suchdimensionen ... 171
8.3.8. Variantendimension ... 172
8.3.8.1. Attributekonfiguration ... 172
8.3.8.2. Kennzahlenkonfiguration - Dimensionstyp ... 173
8.3.8.3. Prozessbaumkonfiguration ... 173
8.3.8.4. Verwendung von Variantenattributen beim Import ... 174
8.3.9. Shared Function Dimension ... 174
8.3.10. Verwenden von Organisationseinheiten als Dimensionen ... 176
8.4. Definition von Datenzugriffsdimensionen ... 176
8.4.1. Verwenden von Datenzugriffsdimensionen ... 178
8.5. Definition des Prozessbaums ... 179
8.5.1. Anmeldung von Kennzahlen und Dimensionen am PPM-System ... 181
8.5.1.1. Kennzahl anmelden ... 181

8.5.1.2. Dimension anmelden 184
8.5.2. Automatische Prozessbaumerweiterung 186
8.5.3. Manuelle Prozessbaumerweiterung 186
8.5.4. Definition des Prozessbaumes in Multibyte-Zeichensätzen 187
8.6. Beziehungen 189
8.6.1. Definition von Beziehungen 189
8.6.1.1. Bezugsdimensionen 190
8.6.2. Definition von Beziehungsberechnungen 191
8.6.3. Definition von Beziehungskennzahlen 195
8.6.4. Definition von Beziehungs- und Organisationsdimensionen 196
9. Verdichtungsverhalten ändern 198
9.1. Konfiguration des internen Verdichtungsattributes 198
9.2. Verdichtungswerte vergeben 199
10. Systemanbindungen 202
10.1. SAP ausführbare Einheiten 202
10.1.1. Software-Voraussetzungen 202
10.1.2. Berechtigungen im SAP-System 202
10.1.3. Transaktionsaufruf 202
10.1.4. Konfiguration 202
10.1.4.1. Konfigurationsbeispiele 204
10.1.4.2. Erläuterungen zur DTD 218
11. Support und rechtliche Hinweise 221
11.1. Dokumentationsumfang 221
11.2. Support 221
11.3. Legal notices 222

1. Allgemein

Dieses Handbuch beschreibt die Konfiguration von ARIS Process Performance Manager (PPM). Der Administrator des PPM-Systems erhält Grund- und Konfigurationswissen, das ihn bei der Konfiguration hinsichtlich unterschiedlicher Einsatzszenarien und Analyseaufgabenstellungen unterstützt.

Das Benutzerhandbuch richtet sich richtet sich an den PPM Customizing Toolkit-Anwender als Application Configuration Expert.

Als **Application Configuration Expert** sind Sie für das Customizing aller ETL-Prozesse (**E**xtraktion der Quellsystemdaten, **T**ransformation der Daten, **L**aden der Daten in die Zieldatenbank) verantwortlich, inklusive Prozessverkettung, Prozesstypisierung, Kennzahlen- und Dimensionsberechnung.

Bitte beachten Sie, dass dieses Handbuch keine Anwender- oder Konfigurationsschulung ersetzt. Es stellt eine Referenz dar, die ergänzende Hinweise zu den Handbüchern und der Online-Hilfe enthält.

2. Übersicht

Bevor Sie Daten in PPM einlesen können, müssen Sie die Prozesse des zu analysierenden Quellsystems zerlegen und darauf aufbauend eine Konfiguration für das PPM-System erstellen. Dieser Vorgang wird als Customizing bezeichnet. Nach Abschluss des Customizing steht ein Satz von spezifischen XML-Konfigurationsdateien zur Verfügung, mit denen das PPM-System initialisiert wird.

Die Importadapter zum Einlesen von Prozessinstanzfragmenten in die PPM-Datenbank sind im Handbuch **PPM Datenimport** beschrieben.

2.1. Konfigurationskomponenten

Es werden folgende Kategorien von Konfigurationskomponenten unterschieden:

Oberflächensprachen

PPM unterscheidet zwei Kategorien von Oberflächensprachen: die Sprache zur Benutzerführung und die Sprache zur Anzeige von Datenbankinhalten, die über die Konfigurationsdateien eingelesen wurden.

Datentypen

In PPM werden interne und benutzerdefinierte Datentypen unterschieden. Interne Datentypen können nicht verändert werden. Über eine XML-Datei können beliebig viele neue Datentypen ins PPM-System eingelesen werden.

Spezifische PPM-Attribute

Attribute sind die Informationsträger des PPM-Systems. Über Attributwerte gelangen Daten vom Quellsystem in das PPM-System. In Attributen werden auch Berechnungsergebnisse gespeichert. PPM wird mit einem umfassenden Satz an Standardattributen ausgeliefert, der sich durch benutzerdefinierte Attribute erweitern lässt.

Prozessschlüsselregeln

Prozessschlüsselregeln bestimmten, wie Prozessschlüssel berechnet werden.

Prozessschlüssel kennzeichnen zusammengehörige Prozessinstanzfragmente. Zusammengehörige Prozessinstanzfragmente werden anhand der Prozessschlüssel unverbunden in eine Prozessinstanz geschrieben. Prozessschlüssel werden während des Imports der Prozessinstanzfragmente erzeugt.

Merge-Schlüsselregeln

Merge-Schlüsselregeln bestimmen, welche Attribute PPM während des Einlesens der Prozessfragmentinstanzen zum Erzeugen von Merge-Schlüsseln verwendet. Bei der Verschmelzung von Ereignissen (Ereignis-Merge) werden die Prozessinstanzfragmente über die Merge-Ereignisse mit identischem Merge-Schlüssel verkettet.

Objektschlüsselregeln

Objektschlüsselregeln bestimmen, wie Objektschlüssel berechnet werden.

Objektschlüssel kennzeichnen identische Objekte. Damit wird ein Überschreiben dieser Objekte bei erneutem Datenimport gewährleistet und ein unbeabsichtigtes mehrfaches Vorkommen identischer Objekte innerhalb einer Prozessinstanz ausgeschlossen.

Attributkopierregeln

Attributkopierregeln legen fest, welche Objektattribute nach dem Merge-Vorgang an die zusammengeführte Prozessinstanz kopiert werden. Diese Attribute werden zur Berechnung von Prozesskennzahlen und Bildung von Dimensionen benötigt.

Typisierungsregeln

Mit Hilfe von Typisierungsregeln werden die eingelesenen Prozessinstanzen einem bestimmten Prozesstyp zugeordnet.

Kennzahlendefinition

Die Kennzahlendefinition setzt sich aus der Definition der Kennzahlen (Typ und Berechnungsvorschrift) und der Definition eines Prozessbaumes zusammen.

PPM enthält bereits die Berechnungsvorschrift für viele Standardkennzahlen und einen Standardprozessbaum. Wenn Sie weitere Kennzahlen benötigen, können Sie ergänzende Kennzahlen und deren Berechnung definieren.

2.2. Kommandozeilenprogramm

Nach Erstellen der Konfigurationsdateien werden die Quellsystemdaten folgendermaßen ins PPM-System eingelesen:

Vorgang	PPM-Kommando	Dokumentation
PPM-Datenbank initialisieren	**runinitdb -init -client <Mandant>**	**PPM Operation Guide**
Prozessinstanzfragmente importieren (XML-Import -adapter oder andere Importadapter)	runxmlimport -client <Mandant> -user ...	**PPM-Datenimport**
Prozessinstanzen aus importierten Prozessinstanzfragmenten bilden, Prozessinstanzen typisieren und Kennzahlen berechnen	runppmimport -client <Mandant> ...	**PPM Operation Guide**

Nach dem Importieren und Berechnen der Prozessinstanzen können diese im Frontend des PPM-Systems analysiert und ausgewertet werden.

2.3. Methodisches Vorgehen

Bevor Sie mit dem Erstellen der PPM-Konfigurationsdateien beginnen, müssen Sie definieren, welche Aktivitätenflüsse bzw. Prozesse des Quellsystems Sie analysieren und welche Messgrößen Sie hierfür verwenden möchten. Die zur Berechnung und Verarbeitung der Prozessinstanzen benötigten Informationen werden im Quellsystem identifiziert (Werte für Prozesszusammenführung, -typisierung und Kennzahlenberechnung).

Das Vorgehen zum Auslesen aus dem Quellsystem und Speichern der Prozessinstanzfragmente im PPM-System wird durch Konfiguration des jeweiligen PPM-Importadapters festgelegt. Der XML-Datenimport in das PPM-System ist ausführlich in der Technischen Referenz **PPM Datenimport** beschrieben.

Sind alle benötigten Quellsysteminformationen bekannt, wird eine Zuordnung der Informationsträger des Quellsystems zu PPM-Attributen erstellt. Abhängig von den gewünschten Analyseergebnissen werden Datentypen und Attribute für Kennzahlen und deren Berechnungsvorschriften sowie Attribute zur Bildung von Dimensionen bestimmt.

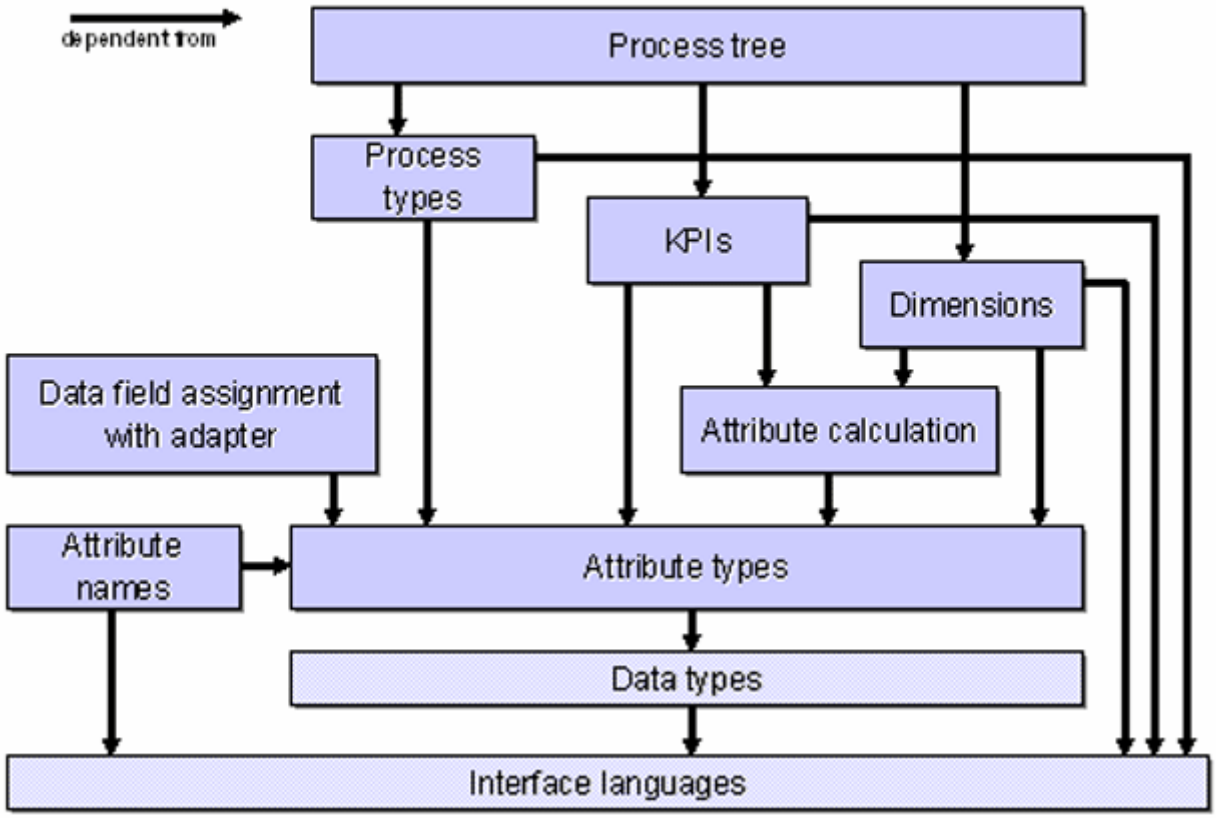
2.4. Hierarchie der Konfigurationskomponenten

Die folgende Tabelle zeigt die Zuordnung der XML-Konfigurationsdateien zu den PPM-Komponenten.

PPM-Komponente	XML-Datei (im xml-Verzeichnis der Standardmandantenvorlagen)
Prozessbaum	***_processstree.xml**
Kennzahlberechnung	*_keyindicator.xml
	*_kigroup.xml
Schlüsselerzeugung und Merge	*_keyrules.xml
	*_merger.xml
Fragmentdefinition und Mapping	Keine Standarddateien. Dateien müssen mittels PPM Customizing Toolkit erstellt werden.
Attributes	*_attributetypes.xml
	*_attributenames_<Sprache>.xml (für jede Sprache eine eigene Datei)
Datentypen	*_datatypes.xml
Oberflächensprachen	*_locales.xml

Die Standardmandantenvorlagen des PPM-Systems finden Sie unter <PPM-Installationsverzeichnis>\ppm\server\bin\agentLocalRepo\.unpacked\<Installationszeit>_ppm-client-run-prod-<Version>-runnable.zip\ppm\ctk\ctk\examples\custom\.

Die folgende Grafik veranschaulicht die Abhängigkeiten zwischen den verschiedenen PPM-Konfigurationskomponenten.



Achten Sie beim Initialisieren der PPM-Datenbank auf die richtige Reihenfolge der Konfigurationsschritte:

1. Sprachen
2. Datentypen
3. Attributes
4. Prozesszusammenführung (Merge) und ~typisierung
5. Kennzahlberechnung
6. Prozessbaum

3. Oberflächensprachen

PPM unterscheidet zwei Kategorien von Oberflächensprachen: die Sprache zur Benutzerführung, die für Menüelemente, Dialogfelder usw. verwendet wird, und die Sprache zur Anzeige von Konfigurationselementen (Namen von Kennzahlen, Dimensionen etc.).

Daten, die durch das Importieren der Prozessinstanzfragmente ins PPM-System eingelesen wurden, werden unabhängig von der gewählten Oberflächensprache immer in der Sprache des Quellsystems angezeigt.

3.1. Oberflächensprachen zur Benutzerführung

Die Sprache zur Benutzerführung legt der Anwender beim Anmelden an das System fest.

PPM unterstützt für Menüelemente, Dialogfelder usw. deutsche und englische Oberflächentexte. Die Oberflächentexte sind im Code der PPM-Software enthalten.

> **Warnung**
> Damit PPM mehrsprachig verwendet werden kann, muss eine internationale Version der Java-Laufzeitumgebung (JRE) installiert sein.

Welche Sprachen im Anmeldedialog ausgewählt werden können, kann für jeden PPM-Mandanten unabhängig in der Datei ***_locales.xml** eingestellt werden. Die Datei ist folgendermaßen aufgebaut:

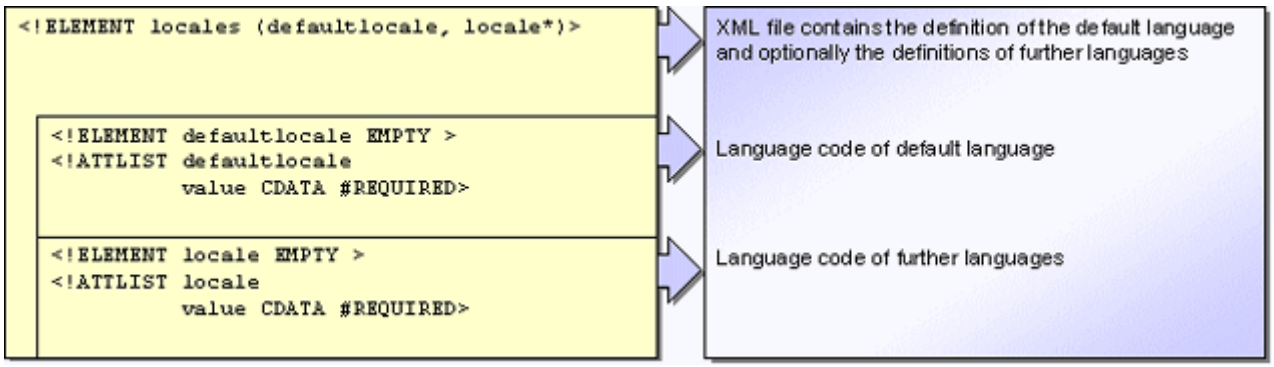
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE locales SYSTEM "Locales.dtd">
<locales>
<defaultlocale value="en" />
<locale value="de" />
<locale value="fr" />
</locales>
```

Wenn die Sprache des angegebenen Sprachkürzels nicht ermittelt werden kann, z. B. bei Angabe eines unbekannten Sprachkürzels in einer Anmelde-URL, wird die Standardsprache verwendet:

```
http://<Web server>/ppm/html/index.html?language=1234
```

Als Standardsprache können Sie ausschließlich eine der für die Benutzerführung verfügbaren Sprachen festlegen.

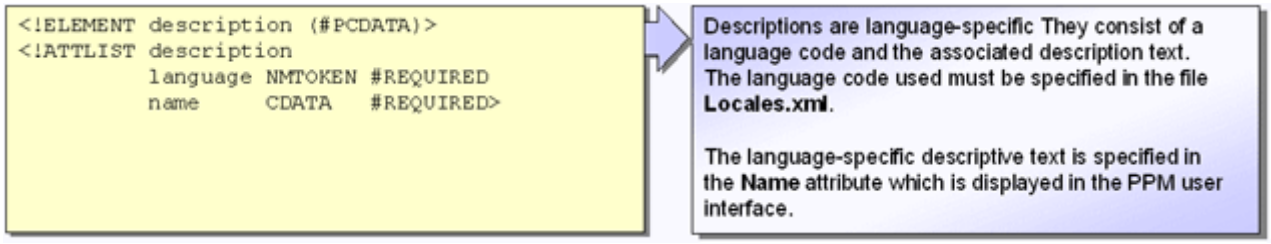
Die XML-Konfigurationsdatei ***_locales.xml** ist durch folgende DTD beschrieben:



3.2. Oberflächensprache zur Anzeige von Konfigurationselementen

Alle PPM-Oberflächenelemente, die auf Konfigurationsdateien basieren (Namen von Kennzahlen, Kennzahlengruppen und Dimensionen), können in beliebigen Sprachen angegeben werden. Die Benennung der Elemente in den verschiedenen Sprachen ist durch das XML-Element **description** definiert, das in allen XML-Konfigurationsdateien verwendet wird. Zusätzlich kann für jedes dieser Oberflächenelemente eine sprachabhängige Beschreibung gepflegt werden, die im Frontend als Tooltip angezeigt wird.

DTD des XML-Elements **description** (Datei **_description.dtd**):



Beispiel 1. ***[en-us]***

Auszug aus der XML-Kennzahlenkonfiguration:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<kidef name="PNUM" type="PROCESS" attrname="AT_KI_PNUM"
calculated="TRUE" distribution="FALSE"
standarddeviation="FALSE" retrievertype=
"NUM_KEYINDICATOR" sharedfunctionki="FALSE"
functionspanki="FALSE" dimreferring="LOOSE"
importmode="OPTIONAL">
<description language="de" name="Prozessanzahl">

  Anzahl der tatsächlich durchlaufenen Prozesse

  </description>
<description language="en" name="Number of processes">

  Number of processes actually passed through
```

```
  </description>
</kidef>
...
</keyindicatorconfig>
```

Der Inhalt des im Bereich **#PCDATA** des jeweiligen Elements **description** angegebenen Textes wird an der PPM-Oberfläche als Tooltip sprachabhängig angezeigt.

> **Warnung**
> Zum erfolgreichen Konfigurieren des PPM-Systems müssen Sie für jedes Konfigurationselement mindestens die Benennung in der Standardsprache angeben.

Das PPM-Frontend zeigt in den Dialogfeldern je nach verwendeter PPM-Anmeldesprache den sprachabhängigen Namen der Konfigurationselemente an, im Prozessattributdialog z. B. die Namen von Attributtypgruppen und Attributtypen. Die sprachabhängigen Namen werden in XML-Konfigurationsdateien definiert.

3.2.1. Verwendung von Multibyte-Zeichensätzen für Konfigurationselemente

PPM unterstützt die Darstellung von Quellsystemdaten und bestimmten Konfigurationselementen mit lokalen Zeichensätzen, die nicht im ANSI-Zeichensatz enthalten sind und mit einem Multibyte-Zeichensatz (Multibyte Character Set, MBCS) kodiert werden. Dazu gehören z. B. das japanische Kanji oder griechische Schriftzeichen.

Alle XML-Dateien, die in das PPM-System importiert werden und nicht auf dem Zeichensatz einer westeuropäischen Sprache beruhen, müssen als Encoding den Zeichensatz **UTF-8** referenzieren:

```
<?xml version="1.0" encoding="UTF-8"?>
<...>
...
</...>
```

Prinzipiell können Tooltips und alle sprachabhängigen Namen, die an der PPM-Oberfläche angezeigt werden, mit einem Multibyte-Zeichensatz dargestellt werden. Im Einzelnen gilt dies für folgende Elemente:

- Attributtypen
- Attributtypgruppen
- Kennzahlen
- Kennzahlengruppen
- Dimensionen
- Prozessinstanzunabhängige Kennzahlen (PIKIs)
- Prozesstypen
- Prozesstypgruppen

Des Weiteren können auch die sprachabhängigen Namen von benutzerdefinierten Datentypen mit einem Multibyte-Zeichensatz gepflegt werden.

Bei Attributtypen und Attributtypgruppen können auch die internen Namen (XML-Attribut **key**) mit einem Multibyte-Zeichensatz konfiguriert werden.

4. Interne Namen

In den PPM-Konfigurationsdateien werden Konfigurationselemente über einen eindeutigen internen Namen referenziert, der sprachunabhängig ist. Die Tabelle zeigt die XML-Attribute, die den internen Namen definieren.

PPM-Konfigurationselement	XML-Attribut
Attributes	**key**
Datentypen	name
Dimensionen	name
Kennzahlen	name
Berechnungsfunktionen (z. B. Typisierungsregeln)	name

Warnung
Interne Namen werden zum Referenzieren von Konfigurationselementen verwendet. Der interne Name eines Objektes einer Prozessinstanz wird im Objektattribut **AT_OBJNAME_INTERN** angegeben.

Interne Namen beginnen mit einem Buchstaben und setzen sich aus Großbuchstaben ohne Umlaute (A-Z), Ziffern (0- 9) und dem Zeichen **_** (Unterstrich) zusammen.

In der Praxis haben sich folgende Richtlinien zum Erstellen von internen Namen bewährt:

PPM-Konfigurationselement	Präfix
Attributnamen	**AT_**
Attributgruppennamen	AG_
Kennzahlengruppen	KI_GROUP_
Dimensionsgruppen	DIM_GROUP_
Kennzahlenattribute	AT_KI_
Kennzahlennamen	KI_
Dimensionsnamen	D_
Typisierungsregeln	TYP_

5. Attributtypen und Attributtypgruppen

Attribute sind die Datenspeicher des PPM-Systems. Für ausgelesene Instanzdaten, alle Kennzahlen und Dimensionen müssen Attribute mit entsprechendem Datentyp definiert werden. Attribute können in Attributgruppen zusammengefasst werden.

5.1. Datentypen

In PPM werden interne und benutzerdefinierte Datentypen unterschieden.

5.1.1. Interne Datentypen

Das PPM-System stellt die folgenden internen Datentypen zur Verfügung. Diese können nicht verändert werden.

Datentyp	Beispiel (Description)	Einheiten oder Skalierungsstufen	Einheiten (Beschreibung)
BOOLEAN	"true"	-	-
TEXT	"Beispieltext" Zeichenkette	-	-
TEXTPAIR	"Text 1\Text 2" 2 Zeichenketten, durch Backslash getrennt	**LEVEL1SCALE** **LEVEL2SCALE**	Grobe Stufe Feine Stufe
LONG	"-231456789" Ganzzahlen	-	-
DOUBLE	"3.1428" Fließkommazahlen, Dezimalstellen durch Punkt getrennt	-	-
DAY	"24.03.2003" Datum im Format dd.MM.yyyy	**DAYSCALE** **WEEKSCALE** **MONTHSCALE** **QUARTERSCALE** **YEARSCALE**	Keine Einheit, jedoch Genauigkeitsstufen: tagesgenau wochengenau monatsgenau quartalsgenau jahresgenau

Datentyp	Beispiel (Description)	Einheiten oder Skalierungsstufen	Einheiten (Beschreibung)
TIME	"01.01.2002 08:15:23" Datum und Uhrzeit im Format dd.MM.yyyy hh:mm:ss Datentyp ist identisch mit den Datentypen **DATE** und **TIMESTAMP**	**SECONDSCALE** **MINUTESCALE** **HOURSCALE** **DAYSCALE** **WEEKSCALE** **MONTHSCALE** **QUARTERSCALE** **YEARSCALE**	Keine Einheit, jedoch Genauigkeitsstufen: sekundengenau minutengenau stundengenau tagesgenau wochengenau monatsgenau quartalsgenau jahresgenau
TIMEOFDAY	"12:41:56" Uhrzeitangabe im Format hh:mm:ss	**SECOND_OF_DAY** **_SCALE** **MINUTE_OF_DAY** **_SCALE** **HOUR_OF_DAY** **_SCALE**	Keine Einheit, jedoch Genauigkeitsstufen: sekundengenau minutengenau stundengenau
TIMESPAN	"23 SECOND" Zeitspanne	**SECOND** **MINUTE** **HOUR** **DAY** **WEEK** **MONTH** **YEAR**	Sekunde Minute Stunde Tag Woche Monat Jahr
FACTORY TIMESPAN	"23 FACTORY_HOUR" Zeitspanne unter Berücksichtigung des Werkskalenders. Es wird nur die reine Arbeitszeit berücksichtigt.	**FACTORY_SECOND** **FACTORY_MINUTE** **FACTORY_HOUR** **FACTORY_DAY** **FACTORY_WEEK** **FACTORY_MONTH** **FACTORY_YEAR**	Personensekunde Personenminute Personenstunde Personentag Personenwoche Personenmonat Personenjahr
FREQUENCY	"86400 PER_DAY" Anzahl pro Zeiteinheit	**PER_SECOND** **PER_MINUTE** **PER_HOUR** **PER_DAY** **PER_WEEK** **PER_MONTH** **PER_YEAR**	pro Sekunde pro Minute pro Stunde pro Tag pro Woche pro Monat pro Jahr
PERCENTAGE	"63 PERCENT" Prozentangabe	**PERCENT** **VALUE_ONLY**	Prozent keine Einheit (Faktoranzeige)

Bei den zeitbasierten Datentypen ist die Basiseinheit jeweils Sekunde. Der Datentyp **PERCENTAGE** besitzt keine Basiseinheit.

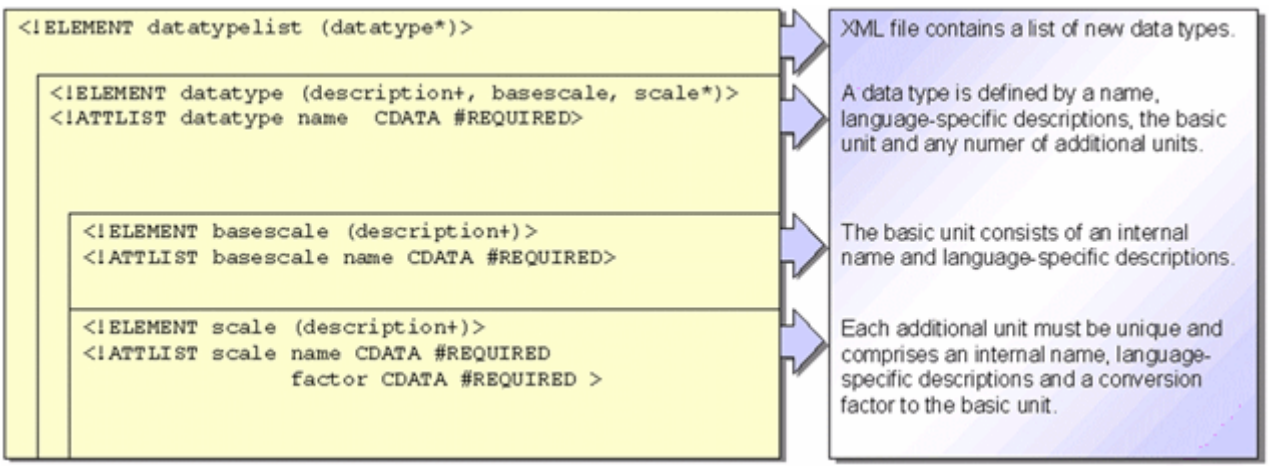
Warnung
Der Datentyp **TEXTPAIR** wird von PPM intern verwendet, um binäre Anfrageergebnisse zu verarbeiten (z.B. Prozesstypgruppe\Prozesstyp). Dieser Datentyp ist nicht zum direkten Datenaustausch geeignet, da das Trennzeichen **Rückwärtsschrägstrich** (****) beim Einlesen einer XML-Datei durch das Zeichen **Vorwärtsschrägstrich** (/) ersetzt wird.

5.1.2. Benutzerdefinierte Datentypen

Über eine XML-Datei können beliebig viele neue Datentypen ins PPM-System eingelesen werden. Die folgende Beispieldatei erzeugt einen neuen Datentyp **Kosten** (interner Name **COST**) in der Basiseinheit **Euro (EUR)** und der weiteren Einheit **Dollar (USD)** mit entsprechendem Umrechnungsfaktor.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE datatypelist SYSTEM 'userdefdatatypes.dtd'>
<datatypelist>
...
<!-- Data type: Costs -->
<datatype name="COST">
<description language="de" name ="Kosten"/>
<description language="en" name ="Costs"/>
<basescale name="EUR">
<description language="de" name="EUR"/>
<description language="en" name="EUR"/>
</basescale>
<scale name="USD" factor="0.9">
<description language="de" name="US Dollar"/>
<description language="en" name="US Dollars"/>
</scale>
</datatype>
...
</datatypelist>
```

Dokumenttypdefinition der XML-Datei zur Definition neuer PPM-Datentypen (**userdefdatatypes.dtd**):



Import und Export benutzerdefinierter Datentypen

Der Import und Export benutzerdefinierter Datentypen wird durch Ausführen des Kommandos **runppmconfig** mit der Option **-datatypes** auf dem PPM-Server-Rechner durchgeführt (siehe Benutzerhandbuch **PPM Operation Guide**). Beim Import wird der interne Name (XML-Tag **datatype name**) des zu importierenden Datentyps überprüft. Existiert im PPM-System bereits ein Datentyp mit diesem Namen, wird dieser Datentyp nicht eingelesen und eine entsprechende Meldung ausgegeben.

Einmal importierte benutzerdefinierte Datentypen können aus dem PPM-System nicht wieder gelöscht werden. Ihre Definition kann lediglich durch Angabe der Option **-overwrite** des Kommandozeilenprogramms **runppmconfig** überschrieben werden.

5.1.2.1. Benutzerdefinierte Datentypen in Multibyte-Zeichensätzen

Der folgende Auszug aus der XML-Konfigurationsdatei für Datentypen zeigt beispielhaft die Definitionsmöglichkeiten eines benutzerdefinierten Datentyps unter Verwendung eines Multibyte-Zeichensatzes:

Beispiel mit Basisskalierung und einer weiteren Skalierung:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE datatypelist SYSTEM "userdefdatatypes.dtd">
<datatypelist>
...
<datatype name="COST">
<description language="en" name="Costs"/>
<description language="en" name="Costs" />
<description language="el" name="έξοδα"/>
<basescale name="EUR">
<description language="de" name="EUR"/>
<description language="en" name="EUR"/>
<description language="el" name="ΕΥΡ"/>
</basescale>
<scale factor="0.001" name="TEUR">
<description language="de" name="TEUR" />
<description language="en" name="EUR Thousands" />
<description language="el" name="Χ.ΕΥΡ"/>
</scale>
</datatype>
...
</datatypelist>
```

Die benutzerspezifische Konfiguration der Datei **DataTypes.xml** können Sie auch mit Hilfe von PPM Customizing Toolkit vornehmen.

5.2. Definition von Attributtypen und Attributtypgruppen

Alle dem PPM-System bekannten Attributtypen und Attributtypgruppen werden in den XML-Konfigurationsdateien ***_AttributeNames_<Sprache>.xml** und ***_AttributeTypes.xml** > definiert.

Verwenden Sie die Angabe **id="auto"** in der Attributtypdefinition, wenn Sie die Identifizierer von Attributtypen bzw. Attributtypgruppen automatisch während des Imports erzeugen lassen wollen.

5.2.1. Definition von Attributtypen

Attribute werden durch Angabe eines eindeutigen Identifizierers (XML-Attribut **id**), eines eindeutigen internen Namens (XML-Attribut **key**) und eines Datentyps (XML-Attribut **type**) definiert. Optional können Attribute einer Attributgruppe zugeordnet werden (XML-Attribut **group**).

Für Standardattribute sind die Identifizierer bis **500** intern fest vergeben. Diese dürfen nicht für die Konfiguration verwendet werden.

5.2.2. Definition von Attributtypgruppen

Attributgruppen werden durch Angabe eines eindeutigen Identifizierers (XML-Attribut **id**) und eines eindeutigen internen Namens (XML- Attribut **key**) definiert.

Durch optionale Angabe des internen Namens der übergeordneten Attributtypgruppe (XML-Attribut **group**) lassen sich Attributtypgruppen in einer Baumstruktur anordnen.

Der interne Name (XML-Attribut **key**) wird im Folgenden **Bezeichner** genannt.

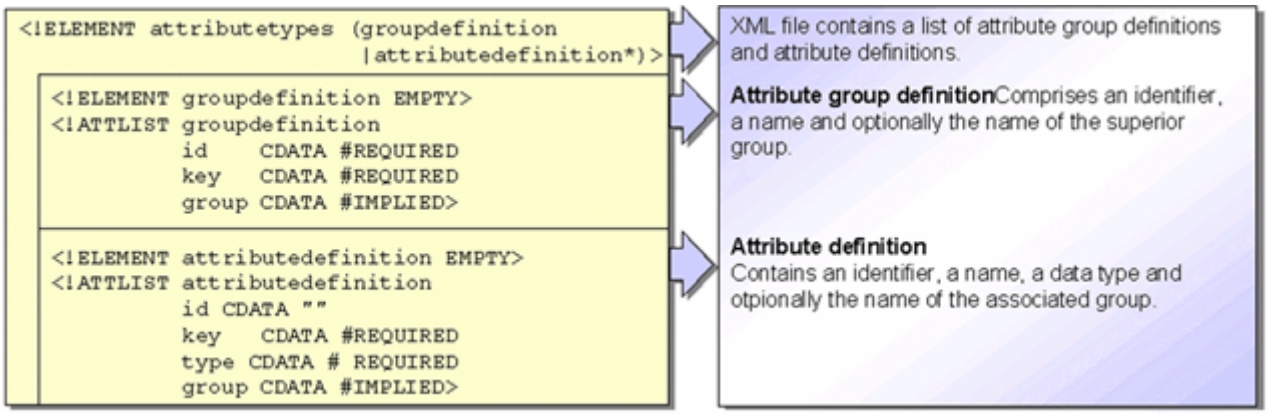
Verwenden Sie die vom System vorbelegte Attributgruppe **AG_INVISIBLE** für Attribute, die nicht im Dialogfeld **Objektattribute** bzw. **Prozessattribute** der EPK-Ansicht an der PPM-Oberfläche angezeigt werden sollen.

Standardmäßig ist diese Attributgruppe nicht definiert.

5.2.3. Konfiguration von Attributtypen und Attributtypgruppen

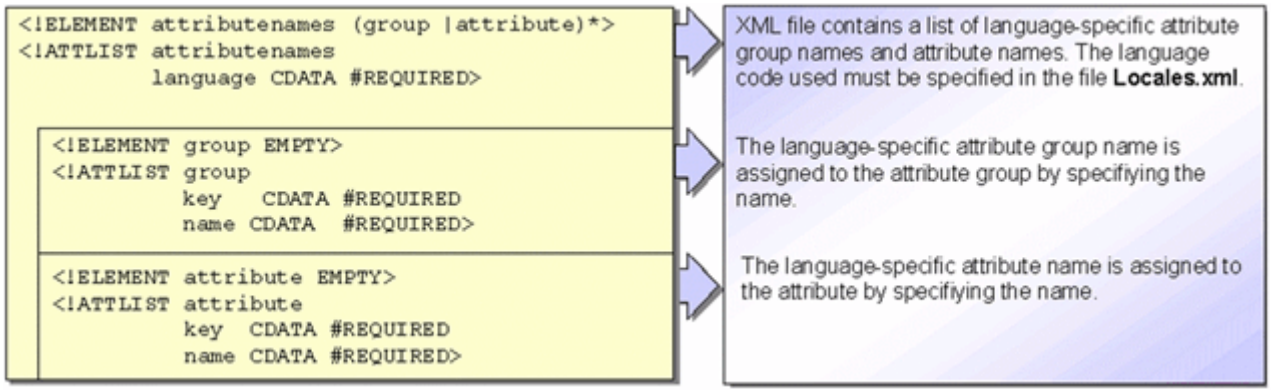
Die XML-Konfigurationsdateien ***_AttributeNames_<Sprache>.xml** und ***_AttributeTypes.xml** sind durch folgende Dokumenttypdefinitionen beschrieben:

attributetypes.dtd



attributenames.dtd

Die Zuordnung der sprachabhängigen Attributnamen zur Attributdefinition erfolgt über die Bezeichner der Attribute (XML-Attribut **key)**.



Zur Definition eines PPM-Attributs und einer Attributgruppe werden die Dateien ***_attributetypes.xml** und ***_attributenames.xml** verwendet.

XML-Datei *_attributetypes.xml

In der Datei werden die folgenden Angaben gemacht:

Attributtypgruppe:

- Eindeutiger Attributtypgruppenidentifizierer (optional)
- Eindeutiger Attributtypgruppenbezeichner (optional)
- Attributtypgruppenbezeichner der übergeordneten Gruppe (optional)

Attributtyp:

- Eindeutiger Identifizierer (Zahl ab 501)
- Eindeutiger Bezeichner
- Datentyp
- Attributtypgruppe (optional)

Der folgende Dateiauszug zeigt die Definition eines Standardattributtyps und eines benutzerdefinierten Attributtyps:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE attributetypes SYSTEM "attributetypes.dtd">
<attributetypes>
...
<groupdefinition id="2" key="AG_MERGER" group="AG_INTERNAL"/>
...
<groupdefinition id="5" key="AG_COSTING"/>
...
<attributedefinition key="AT_EPK_KEY" type="TEXT" group="AG_MERGER" />
...
<attributedefinition id="1000" key="AT_LS" type="TIMESPAN" group="AG_COSTING"/>
...
</attributetypes>
```

XML-Datei *_attributenames.xml

In der Datei werden die folgenden Angaben gemacht:

- Sprachabhängige Attributtypnamen
- Sprachabhängige Attributtypgruppennamen

Für jede gewünschte PPM-Oberflächensprache müssen Sie eine eigene Konfigurationsdatei der Attributtypnamen und -gruppen anlegen.

Der folgende Auszug aus der Datei ***_attributenames_de.xml** enthält den Attributtypnamen in Deutsch:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE attributenames SYSTEM "attributenames.dtd">
<attributenames language="de">
...
<group key="AG_MERGER" name="Merger"/>
...
<attribute key="AT_EPK_KEY" name="EPK-Schlüssel"/>
...
</attributenames>
```

Der folgende Auszug aus der Datei ***_attributenames_en.xml** enthält den Attributtypnamen in Englisch:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE attributenames SYSTEM "attributenames.dtd">
<attributenames language="en">
...
<group key="AG_MERGER" name="Merge"/>
...
<attribute key="AT_EPK_KEY" name="EPC key"/>
...
</attributenames>
```

5.2.3.1. Attributtyp- und Attributtypgruppendefinition in Multibyte-Zeichensätzen

Die folgenden Auszüge aus den XML-Konfigurationsdateien für Attributtypdefinitionen zeigen beispielhaft die Definitionsmöglichkeiten benutzerdefinierter Attributtypen und Attributtypgruppen unter Verwendung eines Multibyte-Zeichensatzes.

Einträge in der Datei AttributeTypes.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE attributetypes SYSTEM "attributetypes.dtd">
<attributetypes>
...
<attributedefinition id="5013" key="ΙΔ_ΧΡ_ΕΠΕΞ" type="TIMESPAN"
group="ΣΥΝ_ΙΔ_ΔΕΙΚΤ_ΧΡΟΝ"/>
...
</attributetypes>
```

Korrespondierende Einträge in der Datei AttributeNames_el.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE attributenames SYSTEM "attributenames.dtd">
```

```
<attributenames language="el">
...
<attribute key="ΙΔ_ΧΡ_ΕΠΕΞ" name="χρόνος επεξεργασίας"/>
...
<group key="ΣΥΝ_ΙΔ_ΔΕΙΚΤ_ΧΡΟΝ" name="δείκτης χρόνου"/>
...
</attributenames>
```

6. Prozesszusammenführung

Damit die importierten Prozessinstanzfragmente zu vollständigen Prozessinstanzen zusammengefügt werden können, benötigen Sie Informationen aus dem Quellsystem, anhand derer PPM die zu einer Prozessinstanz gehörenden Fragmente identifizieren und die zeitliche Reihenfolge (Ablauflogik) der Fragmente rekonstruieren kann.

Die Prozesszusammenführung läuft zweistufig ab.

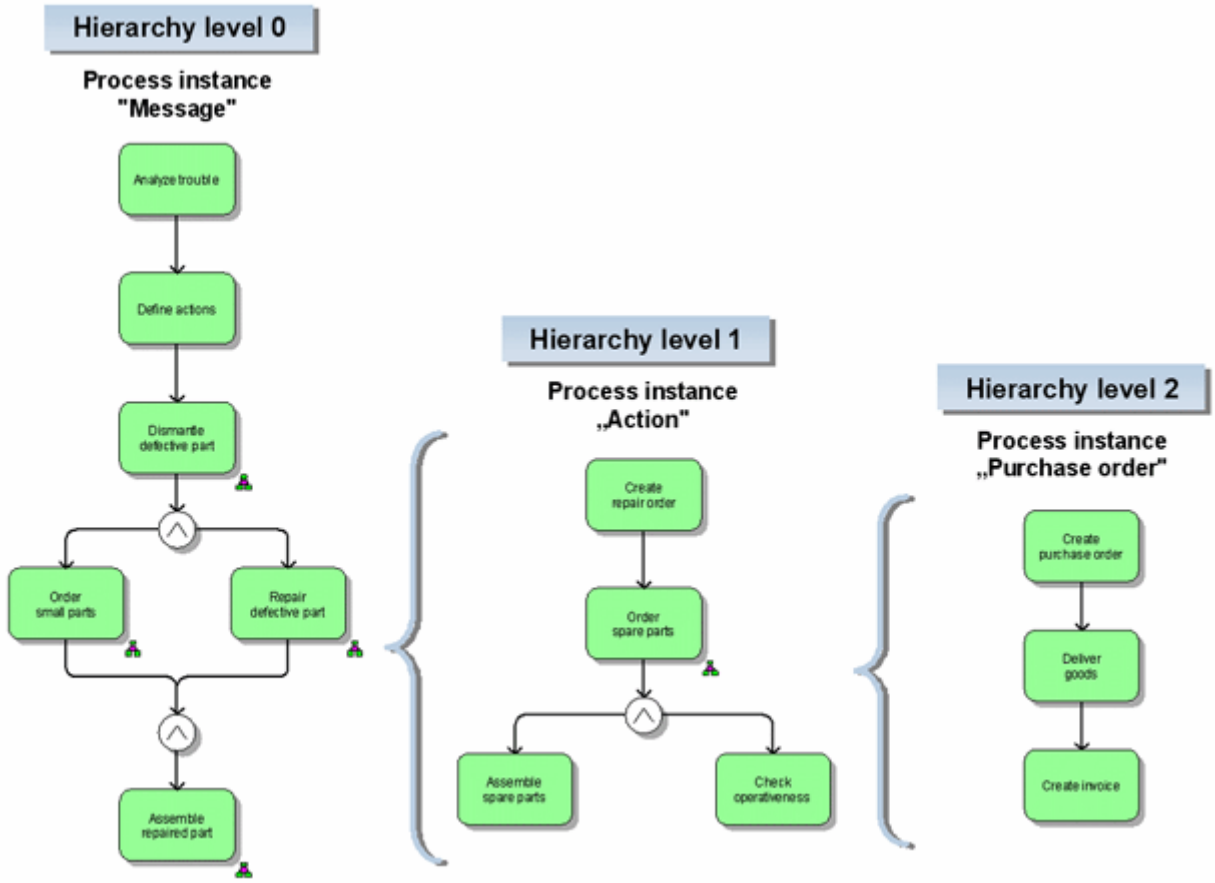
Vorgehen

1. Alle zu einer Prozessinstanz gehörenden Fragmente werden identifiziert und in einer Prozessinstanz gesammelt (Prozess-Merge).
2. Die unverbundenen Fragmente einer Prozessinstanz werden miteinander verkettet (Ereignis-Merge). Der Ereignis-Merge kann entweder schlüsselbasiert oder ordnungsbasiert erfolgen.

6.1. Prozesshierarchien

Ähnlich den Hinterlegungen in ARIS können in PPM an Funktionen untergeordnete Prozessinstanzen hinterlegt werden. In der EPK-Ansicht erhält eine solche Funktion das aus ARIS bekannte Hinterlegungssymbol. Die hinterlegte Prozessinstanz kann über den Kontextmenüeintrag **Hinterlegung öffnen** der Funktion angezeigt werden.

Die folgende Darstellung verdeutlicht die hierarchische Verfeinerung einer Prozessinstanz:



Jede hinterlegte Prozessinstanz ist eine eigenständige Prozessinstanz. Die Prozesshierarchie stellt lediglich eine Zuordnung zwischen Prozessinstanzen dar. Die Verfeinerungstiefe ist beliebig. Alle Prozessinstanzen innerhalb einer Verfeinerung bilden eine Hierarchiestruktur. Jede an dieser Hierarchiestruktur beteiligte Prozessinstanz liegt auf einer anderen Hierarchieebene.

Eine Prozessinstanz darf innerhalb einer Hierarchiestruktur nicht mehreren Funktionen hinterlegt sein, da die Mehrfachberücksichtigung von Attributwerten zu fehlerhaften Ergebnissen bei der Kennzahlenberechnung führt.

Warnung
Achten Sie beim Aufbau einer Hierarchiestruktur Ihrer Prozesse darauf, dass jede Prozessinstanz innerhalb der Hierarchiestruktur jeweils einer anderen Prozesstypgruppe zugeordnet ist.

Jede Funktion, der eine Prozessinstanz hinterlegt ist, steht stellvertretend für die untergeordnete Prozessinstanz. Das Funktionsattribut AT_INTERNAL_HIER_REF verweist eindeutig auf die untergeordnete Prozessinstanz. Der Wert des Hierarchieschlüssels der untergeordneten Prozessinstanz stimmt mit dem Wert des Funktionsattributs AT_INTERNAL_HIER_REF überein. Der Wert des Funktionsattributs wird vom Quellsystemadapter ausgelesen.

Warnung
Das Funktionsattribut **AT_INTERNAL_HIER_REF** darf nicht nachträglich geändert werden. Für den Aufbau von Prozesshierarchien ist der Attributwert zum Importzeitpunkt ausschlaggebend.

Die Prozessattribute der hinterlegten Prozessinstanz werden im Rahmen der Kennzahlenberechnung zusätzlich zu den bereits vorhandenen Funktionsattributen an die Funktion der übergeordneten Prozessinstanz kopiert. Bereits vorhandene Attribute der Funktion werden durch gleichnamige Attribute der hinterlegten Prozessinstanz überschrieben. Wenn bei der Verarbeitung der eingelesenen Prozessinstanzfragmente (runppmimport) für die hinterlegte Prozessinstanz durch die Zuordnung zu einem anderen Prozesstypen andere Kennzahlen berechnet werden, werden die bereits an die Funktion der übergeordneten Prozessinstanz kopierten Attribute nicht gelöscht und bei Analysen weiterhin berücksichtigt. Die kopierten Attribute bleiben auch dann an der Funktion erhalten, wenn die hinterlegte Prozessinstanz gelöscht wird.

Um die kopierten Attribute an Funktionen mit hinterlegter Prozessinstanz zu löschen, müssen Sie die Prozessinstanzfragmente der übergeordneten Prozessinstanz erneut importieren. Durch geeignete Objektschlüsselregeln wird sichergestellt, dass die Funktionen beim erneuten Import überschrieben werden.

6.2. Schlüsselregeln

Entsprechend ihrem Verwendungszweck werden die Schlüsselregeln in fünf Kategorien eingeteilt:

Kategorie	XML-Element	Beschreibung
Prozessschlüssel Regeln	processkeyrule	Sammeln von zusammengehörigen Fragmenten in einer Prozessinstanz
Hierarchie-schlüsselregeln	hierarchykeyrule	Bildung von Prozesshierarchien
Shared-Fragment-Schlüsselregeln	sharedfragmentkeyrule	Kopieren von Shared Fragments in Prozessinstanzen

Kategorie	XML-Element	Beschreibung
Merge-Schlüssel-Regeln	mergekeyrule	Verschmelzen von Merge-Ereignissen innerhalb einer Prozessinstanz
Objektschlüssel-Regeln	internalobjectkeyrule	Kennzeichnen identischer Objekte

Warnung
Verwenden Sie keine vorangestellten oder nachstehenden Leerzeichen (z.B. Leertaste oder Tab) in von Schlüsselregeln erstellten Schlüsseln, in an Schlüsseln beteiligten Attributen oder in Attributen, die einen der folgenden Schlüssel referenzieren (z.B.: AT_INTERNAL_HIER_REF).

6.2.1. Prozessschlüsselregeln

Prozessschlüssel ordnen Prozessinstanzfragmente eindeutig einer Prozessinstanz zu. Prozessinstanzfragmente mit identischem Prozessschlüssel werden unverbunden in eine Prozessinstanz geschrieben.

Prozessschlüssel lassen sich effizient durch Wahl prozessinstanzspezifischer Attributwerte (z.B. **Auftragsnummer** oder **Bearbeitungsnummer**) erzeugen.

Für jedes Prozessfragment muss beim Import mindestens ein Prozessschlüssel erzeugt werden.

Beispiel 2.

Der Dateiauszug definiert eine Prozessschlüsselregel, die zur Erzeugung des Prozessschlüssels den Attributtyp **AT_AUFTRAGSNUMMER** der Ereignisse **EVT_START** und **EVT_END** verwendet.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyrules SYSTEM "keyrules.dtd">
<keyrules>
...
<processkeyrule>
<refobjects>
<refobject objecttype="OT_EVT">
<objectname name="EVT_START"/>
<objectname name="EVT_END"/>
</refobject>
</refobjects>
<keyparts>
<keypart attributetype="AT_AUFTRAGSNUMMER"/>
</keyparts>
</processkeyrule>
...
</keyrules>
```

Warnung
Prozessfragmentinstanzen, für die keine Prozessschlüssel berechnet werden können, werden nicht importiert, da sie keiner Prozessinstanz zugeordnet werden können. Es wird eine Warnung ausgegeben.

Erhalt aller Prozessattribute beim Merge

Standardmäßig bleiben beim Zusammenführen von zwei Prozessinstanzfragmenten nur die Prozessattribute des jüngeren (später importierten) Fragments im resultierenden Fragment erhalten.

Wenn beim Zusammenführen von zwei Prozessinstanzfragmenten die Vereinigungsmenge der Prozessattribute beider Fragmente in das zusammengeführte Fragment übernommen werden soll, müssen Sie das Standard-Verhalten durch Angabe der Klasse **ZRetainingProcessAttributesPMAlgo** überschreiben. Folgender Dateiauszug veranschaulicht die Merge-Konfiguration:

```
...
<mergerconfig>
<mergehandling>
<processmerge>
<algorithm
classname="com.idsscheer.ppm.server.merger.merger.impl.ZRetainingProcessAttribut
esPMAlgo"/>
</processmerge>
...
</mergehandling>
</mergerconfig>
...
```

Die verwendete Java-Klasse ist Bestandteil der PPM-Standardinstallation.

6.2.2. Hierarchieschlüsselregeln

Hierarchieschlüssel ordnen Prozessinstanzen einer übergeordneten Funktion zu und werden zur Erzeugung von Prozesshierarchien verwendet (Kapitel Prozesshierarchien [24]). Sie können als Prozesshierarchieschlüssel in der Detailansicht der Prozessinstanz eingeblendet werden.

Die Hierarchieschlüsselregeln werden auf alle importierten Prozessinstanzen angewendet.

Beispiel 3. *[en-us]*

Für die Funktionen mit den Identifizierern **FCT_ANGEBOT_ERSTELLEN**, **FCT_AUFTR_ANLEGEN** und **FCT_RECHNG_ERSTELLEN** wird eine Hierarchieschlüsselregel erzeugt, die einen Hierarchieschlüssel aus den Werten der Attribute **AT_AUFTRAGSNUMMER**, **AT_RECHNUNGSNUMMER** und **AT_ANGEBOTSSNUMMER** zusammensetzt.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyrules SYSTEM "keyrules.dtd">
<keyrules>
...
<hierarchykeyrule>
<refobjects>
<refobject objecttype="OT_FUNC">
<objectname name="FCT_ANGEBOT_ERSTELLEN"/>
<objectname name="FCT_AUFTR_ANLEGEN"/>
<objectname name="FCT_RECHNG_ERSTELLEN"/>
</refobject>
</refobjects>
<keyparts>
<keypart attributetype="AT_AUFTRAGSNUMMER"/>
```

```
<keypart attributetype="AT_RECHNUNGSNUMMER"/>
<keypart attributetype="AT_ANGEBOTSSNUMMER"/>
</keyparts>
</hierarchykeyrule>
...
</keyrules>
```

6.2.3. Shared-Fragment-Schlüsselregeln

Shared-Fragment-Schlüssel ordnen Prozessinstanzfragmenten Shared Fragments zu. Shared Fragments sind spezielle Prozessfragmente, die ausschließlich an mehreren Prozessinstanzen beteiligte Funktionen enthalten. Da diese Funktionen im Quellsystem nur einmal ausgeführt werden, aber in mehreren Prozessinstanzen auftreten, heißen sie Shared Functions.

Shared Fragments werden über den XML-Import im Graphformat eingelesen. Die Definition des Graphen eines Shared Fragment enthält das Prozessinstanzattribut **AT_IS_SHARED_FRAGMENT** mit dem Wert **TRUE**. Alle Funktionen eines Shared Fragment müssen durch den Wert **TRUE** des Funktionsattributs **AT_IS_SHARED_FUNCTION** als Shared Function gekennzeichnet sein. Während des Imports wird für jedes importierte Shared Fragment mindestens ein Shared-Fragment-Schlüssel berechnet. Shared Fragments, für die kein Schlüssel berechnet werden kann, werden nicht importiert.

Die Shared-Fragment-Schlüsselregeln werden auf alle importierten Prozessinstanzfragmente angewendet. Bei der Weiterverarbeitung (runppmimport) werden alle Fragmente (Shared Fragments und Prozessinstanzfragmente), für die identische Shared-Fragment-Schlüssel berechnet wurden, in eine Prozessinstanz geschrieben und anschließend anhand der Merge-Regeln verkettet.

Der Shared-Fragment-Schlüssel bewirkt das Kopieren der Shared Fragments in eine Prozessinstanz. Zum Verketten von Shared Fragments untereinander oder mit normalen Prozessinstanzfragmenten müssen geeignete Regeln angegeben werden, die vom verwendeten Merge-Verfahren abhängig sind.

Zum Kopieren eines Prozessfragments in eine Prozessinstanz kann ein Shared-Fragment-Schlüssel nur einmal verwendet werden. Nach der ersten Verwendung wird der Schlüssel aus der Prozessinstanz entfernt. Daher werden Kopien von Shared Fragments in einer Prozessinstanz nicht aktualisiert, wenn sich ein Shared Fragment ändert.

Beispiel 4. *[en-us]*

Für die Ereignisse mit den Identifizierern **EVT_ACE** und **EVT_GIK** wird eine Shared-Fragment-Schlüsselregel erzeugt, die einen Shared-Fragment-Schlüssel aus dem Wert des Attributs **AT_XYZ** bildet.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyrules SYSTEM "keyrules.dtd">
<keyrules>
...
<sharedfragmentkeyrule>
<refobjects>
<refobject objecttype="OT_EVT">
<objectname name="EVT_ACE"/>
<objectname name="EVT_GIK"/>
</refobject>
</refobjects>
<keyparts>
<keypart attributetype="AT_XYZ"/>
```

```
</keyparts>
</sharedfragmentkeyrule>
...
</keyrules>
```

6.2.4. Merge-Schlüsselregeln

Merge-Schlüsselregeln dienen dem Verschmelzen von Merge-Ereignissen innerhalb einer Prozessinstanz. Auf diese Weise werden die durch Prozessschlüssel zugeordneten Prozessinstanzfragmente zu einer Prozessinstanz verkettet.

Merge-Schlüssel werden aus bestimmten Objektattributen des Prozessinstanzfragments berechnet. Mit ihrer Hilfe wird die Ablauflogik der Prozessinstanz rekonstruiert und die unverbundenen Fragmente werden entsprechend verkettet.

PPM unterscheidet zwei Merge-Verfahren:

- Schlüsselbasierter Merge
- Ordnungsbasierter Merge

Das gewünschte Merge-Verfahren wird in der XML-Konfigurationsdatei ***_merger.xml** angegeben. Der Aufbau dieser Datei ist durch die DTD **mergerconfig.dtd** vorgegeben.

Datei mergerconfig.dtd (1. Teil)

```
<!ELEMENT mergerconfig ( mergehandling, connectorhandling? )>
<!ELEMENT mergehandling ( sharedfragmentmerge?, processmerge?, eventmerge+ )>
<!ATTLIST mergehandling eventmode (startevent | endevent | importtime)
"importtime">
<!ELEMENT sharedfragmentmerge ( algorithm )>
<!ELEMENT processmerge ( algorithm?, mergeattributes?)>
<!ATTLIST processmerge
mode (replace|update) 'replace'>
<!ELEMENT eventmerge ( mode, condition?, algorithm? )>
<!ATTLIST eventmerge
key ID #IMPLIED
priority CDATA #IMPLIED>
<!ELEMENT condition EMPTY>
<!ATTLIST condition
classname NMTOKEN #REQUIRED
value (TRUE|FALSE) 'TRUE'
comment CDATA #IMPLIED>
<!ELEMENT algorithm EMPTY>
<!ATTLIST algorithm
classname NMTOKEN #REQUIRED
comment CDATA #IMPLIED>
<!ELEMENT mergeattributes (attribute+)>
<!ELEMENT attribute EMPTY>
<!ATTLIST attribute
key CDATA #REQUIRED>
<!ELEMENT mode ( keymerge | sortmerge )>
<!ELEMENT keymerge EMPTY >
```

```
<!ELEMENT sortmerge ( criterion*, algorithm? )>
<!ELEMENT criterion EMPTY>
<!ATTLIST criterion
name NMTOKEN #REQUIRED>
```

XML-Tag	Beschreibung
mergerconfig	**Gruppierung der Merge-Konfiguration**
mergehandling	Merge-Art, die konfiguriert wird. Es muss mindestens das Element **eventmerge** (Ereignis-Merge) angegeben werden.
sharedfragmentmerge (optional)	Vom Standardalgorithmus abweichender Algorithmus zum Zusammenführen der Shared Fragments mit Prozessinstanzfragmenten
processmerge (optional)	Vom Standardalgorithmus abweichender Algorithmus zum Zusammenführen der Prozessinstanzfragmente. Wählbar sind die Modi **Ersetzen (replace)** oder **Zusammenführen (update)**. Standardwert ist **replace**.
eventmerge	Vom Standardalgorithmus abweichender Algorithmus zum Verschmelzen der Merge-Ereignisse
key (optional)	ID, über die das Element eventmerge referenziert werden kann
priorität (optional)	Priorisierung des eventmerge-Elements - je niedriger der ganzzahlige Wert, desto höher priorisiert.
bedingung (optional)	Bedingung für das Verschmelzen der Merge-Ereignisse
classname	Name der JAVA-Klasse, die die angegebene Bedingung überprüft
value (optional)	Bedingung ist erfüllt, wenn die Überprüfung durch die JAVA-Klasse den angegebenen Wert (**TRUE** oder **FALSE**) zurückliefert. Vorgabewert ist **TRUE**.
algorithmus (optional)	Berechnungsvorschrift (JAVA-Klasse)
classname	JAVA-Klasse, die eine bestimmte Berechnungsvorschrift implementiert
mode	Merge-Verfahren zur Ereignisverschmelzung
keymerge	Schlüsselbasierter Ereignis-Merge auf der Basis von definierten Merge-Schlüsseln
sortmerge	Ordnungsbasierter Ereignis-Merge auf der Basis der Sortierung von Funktionen
criterion	Kriterium (Funktionsattribut), auf dessen Basis die Fragmente geordnet werden sollen. Es können mehrere Kriterien angegeben werden.
name	Bezeichner des Funktionsattributs

Datei mergerconfig.dtd (2. Teil)

```
<!ELEMENT connectorhandling ( multiindegreehandling?, multioutdegreehandling?,
andhandling?, orhandling?, xorhandling?)>
<!ELEMENT multiindegreehandling (algorithm) >
<!ELEMENT multioutdegreehandling (algorithm) >
<!ELEMENT andhandling (algorithm) >
<!ELEMENT orhandling (algorithm) >
<!ELEMENT xorhandling (algorithm) >
```

XML-Tag	Beschreibung
connectorhandling (optional)	**Behandlung von Konnektoren in Prozessgraphen über die Angabe einer JAVA-Klasse (algorithm)**
multiindegreehandling (optional)	Algorithmus, der beim Objekt-Merge das Einfügen von Konnektoren mit mehreren eingehenden Kanten steuert
multioutdegreehandling (optional)	Algorithmus, der beim Objekt-Merge das Einfügen von Konnektoren mit mehreren ausgehenden Kanten steuert
andhandling (optional)	Algorithmus zur Behandlung von AND-Konnektoren
orhandling (optional)	Algorithmus zur Behandlung von OR-Konnektoren
xorhandling (optional)	Algorithmus zur Behandlung von XOR-Konnektoren

Beispiel 5.

Für das Verketten der Prozessinstanzfragmente wird der schlüsselbasierte Ereignis-Merge verwendet.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mergerconfig SYSTEM "mergerconfig.dtd">
<mergerconfig>
...
<mergehandling>
<eventmerge>
<mode>
<keymerge/>
</mode>
</eventmerge>
</mergehandling>
...
</mergerconfig>
```

Während der Ereignisverschmelzung wird die Vereinigungsmenge der Attribute beider Merge-Ereignisse an das verbleibende Ereignis kopiert. Bereits vorhandene Objektattribute werden nicht überschrieben. Das zuerst importierte Merge-Ereignis wird gelöscht.

6.2.4.1. Schlüsselbasierter Merge

Der schlüsselbasierte Merge verschmilzt Ereignisse mit identischen Merge-Schlüsseln. Dabei wird das zuerst eingelesene Merge-Ereignis gelöscht und die Anzahl identischer Merge-Schlüssel verringert. Der Merge-Vorgang wird so oft wiederholt, bis keine identischen Merge-Schlüssel innerhalb der aktuellen Prozessinstanz mehr gefunden werden.

Merge-Schlüsselregeln werden in der XML-Datei **KeyRules.xml** definiert. Dazu werden die an der Bildung des Merge-Schlüssels beteiligten Attribute eines Fragmentereignisses angegeben. Der Merge-Schlüssel selbst wird durch Aneinanderreihen der angegebenen Attributwerte erzeugt.

Beispiel

Folgender Dateiauszug definiert eine Merge-Schlüsselregel, die zur Erzeugung des Merge-Schlüssels für das Start- und Endereignis eines Prozessinstanzfragmentes jeweils den internen Objektnamen des Ereignisses

(Attributtyp **AT_OBJNAME_INTERN**) verwendet. Da in einer Prozessinstanz mehrere Fragmente derselben Fragmentdefinition vorkommen können, wird der Merge-Schlüssel um den Wert des Attributs **AT_END_TIME** erweitert.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyrules SYSTEM "keyrules.dtd">
<keyrules>
...
<processkeyrule>
...
</processkeyrule>
...
<mergekeyrule>
<refobjects>
<refobject objecttype="OT_EVT">
<objectname name="EVT_START"/>
<objectname name="EVT_END"/>
</refobject>
</refobjects>
<keyparts>
<keypart attributetype="AT_OBJNAME_INTERN"/>
<keypart attributetype="AT_END_TIME"/>
</keyparts>
</mergekeyrule>
...
</keyrules>
```

6.2.4.2. Ordnungsbasierter Merge

Der ordnungsbasierte Merge verschmilzt Ereignisse anhand von Sortierkriterien. Es können beliebig viele Sortierkriterien in Form von Funktionsattributtypen angegeben werden. Das Nachfolger-Ereignis einer Funktion wird mit dem Vorgänger-Ereignis der folgenden Funktion verschmolzen.

Standardmäßig sind in PPM alphanumerisches und chronologisches Sortierverfahren implementiert. Welches Verfahren verwendet wird, entscheidet der Datentyp des spezifizierten Sortierkriteriums.

Ein Sortierkriterium könnte beispielsweise das Funktionsattribut **AT_END_TIME** vom Datentyp **TIME** sein.

Beim ordnungsbasierten Merge dürfen die importierten Prozessinstanzfragmente keine Regeln enthalten. Prozessinstanzfragmente mit aufeinanderfolgenden Funktionen werden in Minimal-EPKs (Ereignis-Funktion-Ereignis) aufgespalten.

> **Warnung**
> Achten Sie darauf, dass das von Ihnen definierte Sortierkriterium an jeder Funktion der zusammenzuführenden Instanzen mit entsprechend gepflegten Werten vorhanden ist.

Ab PPM 4.0 können Sie in einer Mandantenkonfiguration beide Merge-Verfahren anwenden

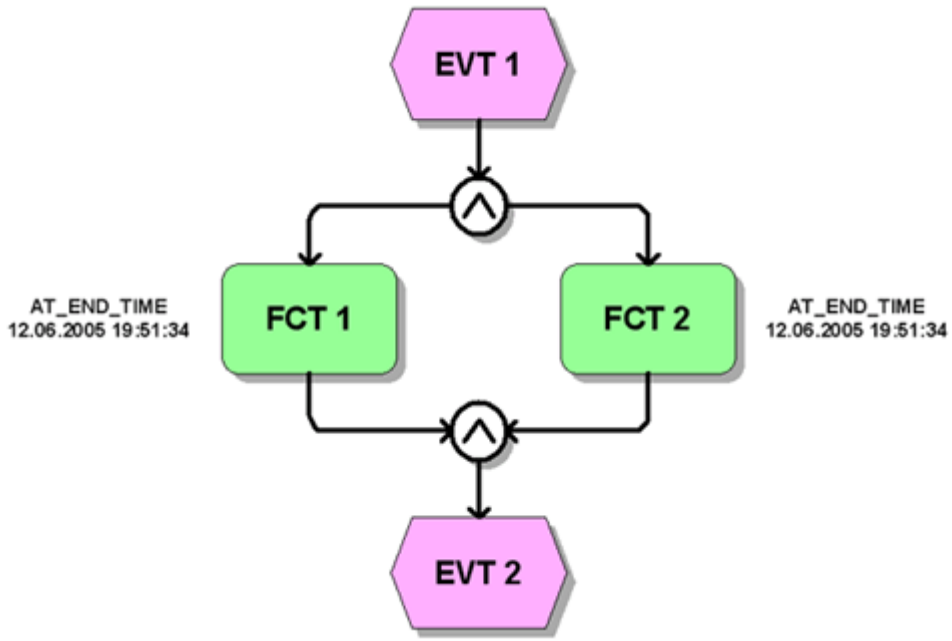
Beispiel 6. Beispiel

Für den ordnungsbasierten Merge wird das Funktionsattribut **AT_END_TIME** verwendet. Der Zeitstempel muss an jeder Funktion der Instanz gepflegt sein.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mergerconfig SYSTEM "mergerconfig.dtd">
<mergerconfig>
...
<mergehandling>
<eventmerge>
<mode>
<sortmerge>
<criterion name = "AT_END_TIME" />
</sortmerge>
</mode>
</eventmerge>
</mergehandling>
...
</mergerconfig>
```

Wenn das Sortierattribut an mehreren Funktionen den gleichen Wert hat, werden über UND-Regeln in der zusammengeführten Prozessinstanz parallele Pfade erzeugt.

Beispiel 7. Beispiel



Da das verwendete Sortierkriterium **AT_END_TIME** für die beiden Funktionen **FCT 1** und **FCT 2** einen identischen Wert liefert, wird in der zusammengeführten Prozessinstanz über UND-Reglen ein paralleler Pfad erzeugt.

6.2.4.3. Kombination von Merge-Verfahren

Um Fragmente unter Verwendung verschiedener Verfahren zusammenführen zu können, müssen Sie mehrere Merge-Verfahren definieren. Hierfür wird die Merger-Konfiguration (XML-Element **eventmerge**) um die Attribute **key** und **priority** erweitert.

Der Wert des Attributs **key** bestimmt den Namen des Merge-Verfahrens und wird vom Graphattribut **AT_INTERNAL_EVENT_MERGE_MODE** zu importierender Fragmente referenziert. Werden während eines Importvorgangs Fragmente mit verschiedenen Merge-Verfahren zusammengeführt, wird das Verfahren mit der niedrigeren Priorität (XML-Element **priority**) verwendet.

Für Fragmente, die das Attribute **AT_INTERNAL_EVENT_MERGE_MODE** nicht gepflegt haben, wird das Merge-Standardverfahren verwendet. Das Standardverfahren ist dasjenige, das in der Merge-Konfiguration ohne Angabe eines Schlüssels definiert ist.

Warnung
Geben Sie für alle Merge-Verfahren unterschiedliche Prioritäten an.

Um das für verschiedene System-Event-Typen anzuwendende Merge-Verfahren zu spezifizieren, geben Sie an den Fragmentdefinitions-EPKs im Prozessattribut **AT_INTERNAL_EVENT_MERGE_MODE** den Schlüssel des gewünschten Merge-Verfahrens an. Alle mit dieser Fragmentdefinition importierten System-Events werden dann automatisch mit dem angegebenen Verfahren zusammengeführt.

Warnung
Wenn das im zu importierenden Fragment angegebene Merge-Verfahren nicht existiert, wird das Fragment nicht importiert und eine Fehlermeldung ausgegeben. Die Fehlermeldung wird auch an das entsprechende Fragment im Attribut **AT_MERGE_ERROR_MESSAGE** gespeichert. Zusätzlich wird das Attribut **AT_MERGE_ERROR_FLAG** mit Wert **true** geschrieben.

Beispiel 8.

Für den Datenimport wird folgende Merge-Konfiguration verwendet, die 3 Verfahren definiert:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mergerconfig SYSTEM "mergerconfig.dtd">
<mergerconfig>
<mergehandling>
<eventmerge priority="10">
<mode>
<keymerge/>
</mode>
</eventmerge>
<eventmerge key="SORTMERGE_ID" priority="3">
<mode>
<sortmerge>
<criterion name = "AT_ID" />
</sortmerge>
</mode>
</eventmerge>
<eventmerge key="SORTMERGE_DATE" priority="4">
<mode>
<sortmerge>
<criterion name = "AT_START_TIME" />
</sortmerge>
</mode>
</eventmerge>
</mergehandling>
</mergerconfig>
```

Fragmente ohne Attribut **AT_INTERNAL_EVENT_MERGE_MODE** werden unter Verwendung des schlüsselbasierten Merge zusammengeführt. Fragmente, die das Attribut mit Wert **SORTMERGE_ID** gepflegt haben, werden ordnugsbasiert nach dem Funktionsattribut **AT_ID** zusammengeführt. Fragmente mit Attributwert **SORTMEGRE_DATE** werden ordnugsbasiert nach dem Funktionsattribut **AT_START_TIME** zusammengeführt.

6.2.5. Objektschlüsselregeln

Objektschlüssel werden beim Re-Import von Daten dazu verwendet, gleiche Objekte zu kennzeichnen und zu überschreiben. Zwei Ereignis- bzw. Funktionsobjekte sind gleich, wenn für sie identische Objektschlüssel berechnet wurden. Werden Objekte als gleich erkannt, überschreibt das zuletzt importierte Objekt das zuvor importierte Objekt. Die Ablauflogik der Prozessinstanz wird entsprechend angepasst. Der berechnete Objektschlüssel wird als Attributtyp **AT_INTERNAL_OBJECT_KEY** an das entsprechende Objekt geschrieben.

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

Für alle Funktionen wird der Objektschlüssel aus den Werten der Attribute **AT_OBJNAME_INTERN** und **AT_END_TIME** zusammengesetzt.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyrules SYSTEM "keyrules.dtd">
<keyrules>
...
<processkeyrule>
...
</processkeyrule>
...
<internalobjectkeyrule>
<refobjects>
<refobject objecttype = "OT_FUNC" />
</refobjects>
<keyparts>
<keypart attributetype="AT_OBJNAME_INTERN"/>
<keypart attributetype="AT_END_TIME"/>
</keyparts>
</internalobjectkeyrule>
...
</keyrules>
```

Warnung
Achten Sie beim Erstellen der Objektschlüsselregeln darauf, dass für unterschiedliche Objekttypen (Funktion bzw. Ereignis) verschiedene Objektschlüssel berechnet werden. Das Überschreiben von Objekten unterschiedlichen Typs führt zu nicht definierten Ergebnissen.

6.2.6. Ausgabeverhalten von Meldungen

Sie können für die Schlüsselregelwerke **processkeyrule**, **hierarchykeyrule**, **mergekeyrule** und **sharedfragmentkeyrule** das Ausgabeverhalten von Systemmeldungen durch das XML-Attribut **onmissingkeypart** beeinflussen, wenn der im XML-Element **keyparts** angegebene Teilschlüssel nicht

berechnet werden kann. Das Attribut kann einen der Werte **info**, **warning** oder **ignore** annehmen, Vorgabewert ist **warning**.

Attributwert	Beschreibung
info	Die Meldung wird als Information ausgegeben.
warning	Die Meldung wird als Warnung ausgegeben.
ignore	Es wird keine Meldung ausgegeben.

Beispiel 10.

Wenn Sie als Merge-Verfahren den Vorgänger-Merge verwenden, fehlt i.d.R.die Information zur Berechnung des Schlüssels zum Vorgängerfragment des ersten Fragments in der Prozessinstanz. Wenn Sie die erwarteten Meldungen unterdrücken möchten, geben Sie im XML-Attribut **onmissingkeypart** den Wert **ignore** an.

6.2.7. Konfigurationsdatei

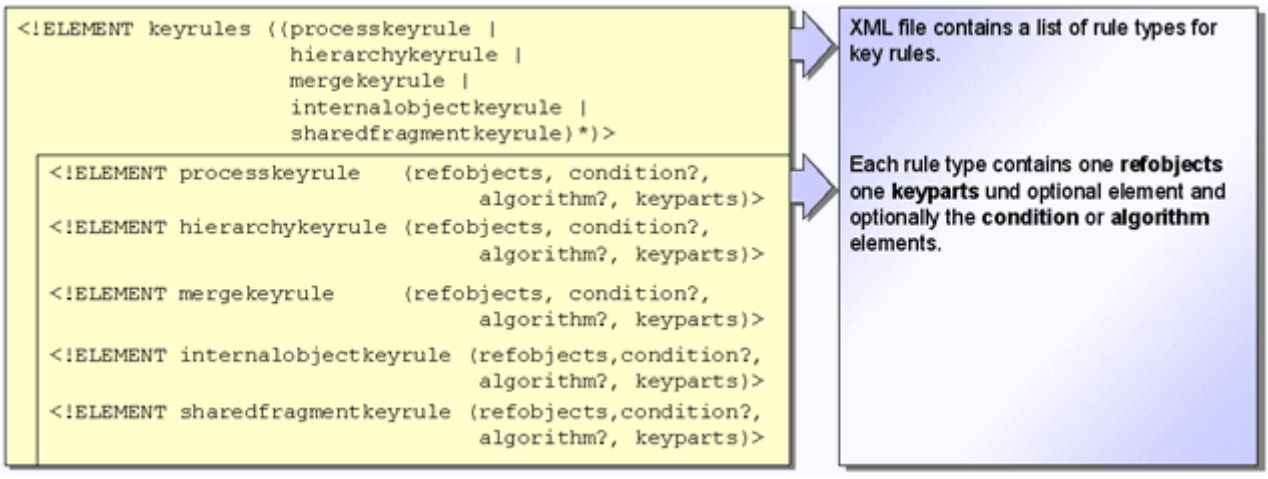
Die XML-Konfigurationsdatei enthält alle Schlüsselregeln. Eine Regel besteht aus einer Liste von Objektattributbezeichnern, deren Werte zur Schlüsselerzeugung verwendet werden. Ein Schlüssel wird standardmäßig durch Aneinanderreihen der Werte der angegebenen Attributtypen gebildet. Eine alternative Verarbeitungsart kann durch Verwendung einer anderen Java-Klasse eingestellt werden (s. u.).

Prozess- und Hierarchieschlüssel werden in der Datenbank gespeichert.

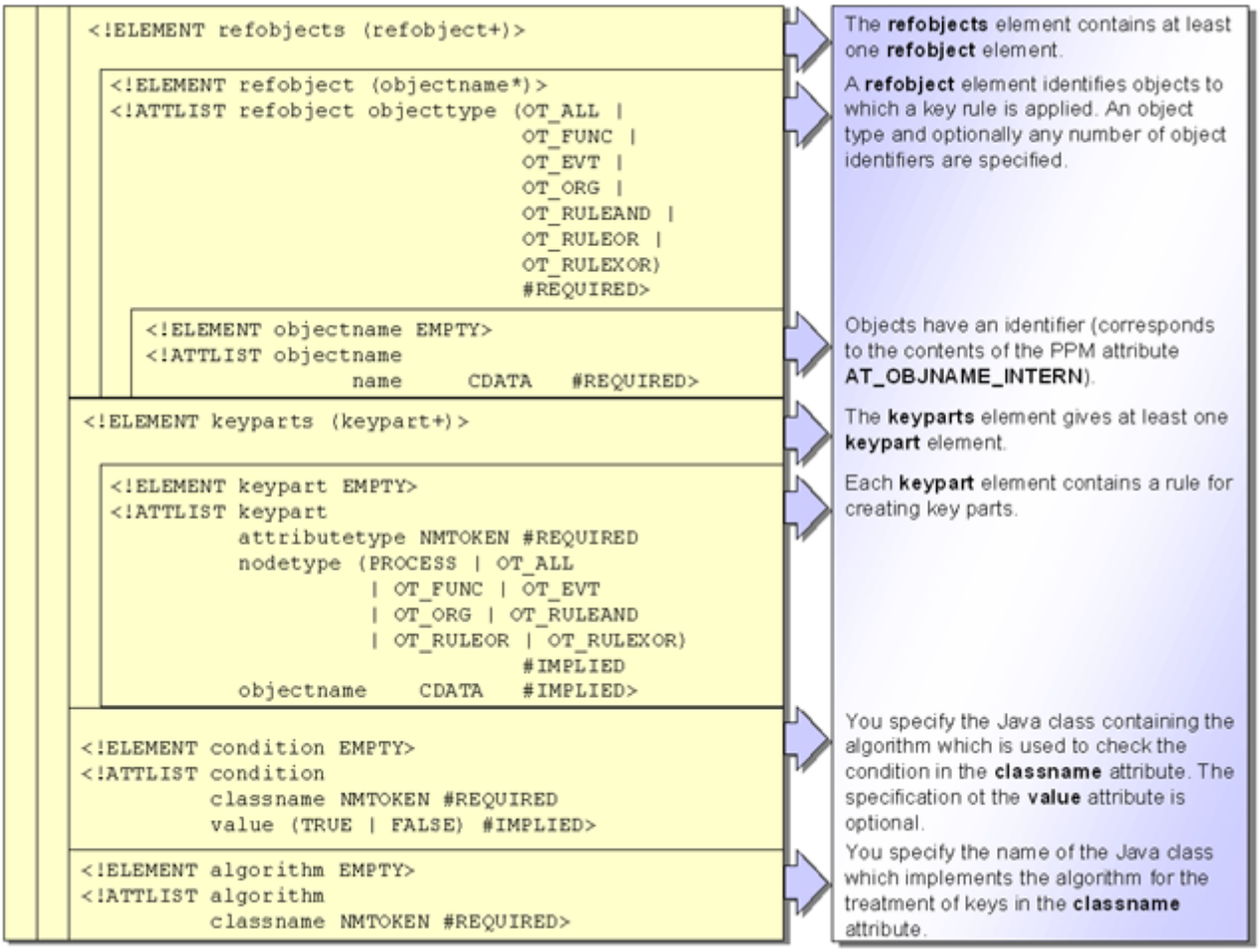
Shared-Fragment-Schlüssel stehen als Attributtyp an den Prozessinstanzen und werden zusätzlich in der Datenbank gespeichert.

Merge- und Objektschlüssel werden als Objektattribut an das entsprechende Objekt geschrieben. Merge-Schlüssel werden in den Attributtypen **AT_MERGE_KEY_1** bis **AT_MERGE_KEY_10**, der Objektschlüssel wird im Attributtyp **AT_INTERNAL_OBJECTKEY** des referenzierten Objektes gespeichert.

Datei keyrules.dtd (1.Teil):



Datei keyrules.dtd (2.Teil):



Alle in der Konfigurationsdatei angegebenen Regeln referenzieren das Objektattribut **AT_OBJNAME_INTERN**.

Im XML-Element **refobjects** wird eine Objektliste angegeben, auf die sich die betreffende Regel bezieht. Im XML-Element **refobject** wird im XML-Attribut **objecttype** der Objekttyp und im XML-Element **objectname** ein Objektidentifizierer angegeben. Es können mehrere XML-Elemente **objectname** angegeben werden.

Als Schlüsselbestandteile (XML-Element **keyparts**) spezifizieren Sie die Bezeichner der Attribute, die für die Schlüsselgenerierung verwendet werden.

Wenn Sie eine andere Verarbeitungsart als das Aneinanderreihen verwenden möchten, geben Sie im XML-Element **algorithm** den Namen der Java-Klasse (XML-Attribut **classname**) an, die einen anderen Algorithmus zur Verarbeitung der Schlüsselattributwerte implementiert.

Warnung
Die fixen Attribute (**AT_MERGE_KEY_1** bis **AT_MERGE_KEY_10**) sind zur Speicherung berechneter Merge-Schlüssel vorgesehen. Sie dürfen weder geändert noch vom XML-Import mit Werten belegt werden.

Schlüssel werden beim Einlesen der Prozessinstanzfragmente berechnet. Wenn Sie die Schlüsselregeln für einen existierenden Prozesstypen ändern, müssen Sie alle Prozessinstanzfragmente dieses Prozesstyps erneut einlesen. Ansonsten führt das spätere Zusammenführen der Prozessinstanzen zu unerwünschten Ergebnissen. Änderungen von Schlüsselregeln für Prozessinstanzfragmente eines Prozesses, für den bereits Prozessinstanzen im PPM-System existieren, sind deshalb mit äußerster Aufmerksamkeit durchzuführen.

Die Erweiterung des Regelwerks bei der Aufnahme eines neuen Prozesstyps in das PPM-System ist dagegen unkritisch, wenn dabei die bereits vorhandenen Prozessinstanzen unbeeinflusst bleiben.

6.3. Merge der Prozessfragmente

Beim Prozess-Merge können Sie zwischen den Modi **Ersetzen** und **Zusammenführen** wählen.

In PPM Customizing Toolkit stehen Ihnen in der Komponente **Merge** des Moduls **Prozess-Merge** entsprechend zwei Merge-Varianten **Attribute/Objekte ersetzen (Standard)** und **Attribute/Objekte zusammenführen** zur Verfügung. Standardmäßig ist für den Prozess-Merge der Merge-Modus **Attribute/Objekte ersetzen (Standard)** aktiviert.

Die Modi **Ersetzen** und **Zusammenführen** haben nur Auswirkungen auf den Prozess-Merge. Der Event-Merge funktioniert in beiden Fällen gleich.

Sie können für jeden PPM-Mandanten in der Merger-Konfiguration [29] (mergerconfig.dtd) einstellen, ob beim Merge Attribute zusammengeführt oder ersetzt werden.

6.3.1. Merge-Modus Ersetzen

Beim Merge-Modus **Ersetzen** werden die Prozessattribute der neueren (zuletzt importierten) Prozessinstanz (mit Ausnahme der Merge-Attribute) in das Ergebnisfragment übernommen. Bei gleichen Funktionen und Events (selber interner Objektschlüssel) ersetzt immer das zuletzt importierte (neuere) das ältere Objekt vollständig (mit allen Attributen). Es bleiben somit nur die Objektattribute und Organisationseinheiten des neueren (zuletzt importierten) Objektes erhalten. Welches Objekt neuer ist, wird mittels des Attributs **AT_ORIG_EPK_ID** entschieden.

Sie können optional eine Liste mit Prozessattributen angeben, die beim Zusammenführen von Fragmenten aus der älteren Prozessinstanz in das Ergebnisfragment übernommen werden. Das Prozessattribut des zuvor importierten Fragments oder der bereits existierenden Prozessinstanz wird durch das Prozessattribut des später importierten Fragments überschrieben.

Im XML-Element **mergeattributes** der Merge-Konfiguration geben Sie eine Liste von Prozessattributen an, die beim Zusammenführen von Fragmenten übernommen werden. Alle anderen Prozessattribute bleiben unberücksichtigt.

Beispiel 11. ***[en-us]***

Der folgende Dateiauszug der Merge-Konfiguration bewirkt, dass die beiden Prozessattribute **AT_SAPSYSTEM** und **AT_SAPCLIENT** beim Zusammenführen von Fragmentinstanzen in das Ergebnisfragment oder eine bereits existierende Prozessinstanz übernommen werden.

```
<mergerconfig>
<mergehandling>
<processmerge>
<mergeattributes>
<attribute key = "AT_SAPSYSTEM"/>
<attribute key = "AT_SAPCLIENT"/>
</mergeattributes>
</processmerge>
...
</mergehandling>
</mergerconfig>
```

Tipp
Durch Übernahme von Prozessattributen können Sie Dimensionswerte auf Basis von Prozessattributen direkt überschreiben, indem Sie ein Fragment importieren, das ausschließlich das Prozessattribut mit neuem Wert enthält, auf dem die Dimension erstellt wurde.

Bitte beachten Sie, dass bereits existierende Prozessattribute durch späteres Kopieren von Objektattributen (siehe Kapitel Attributkopierregeln) überschrieben werden.

Wenn das Prozessattribut an mehreren Fragmenten mit gleichem Prozessschlüssel gepflegt ist und der Import aller Fragmente über mehrere Importvorgänge verteilt ist, kann nicht sicher gestellt werden, dass der Attributwert des zuletzt importierten Fragments in das Ergebnisfragment übernommen wird.

6.3.2. Merge-Modus Zusammenführen

Beim Merge-Modus **Zusammenführen** werden die Prozessattribute der neueren (zuletzt importierten) Prozessinstanz mit den Prozessattribute der älteren Prozessinstanz zusammengeführt. Das Gleiche gilt für Funktionen mit Funktionsattributen und zugehörigen Organisationseinheiten.

Importzeitpunkt auf Attributebene

Anhand des Importzeitpunkts wird beim Merge-Modus **Zusammenführen** entschieden, welches Objekt neuer ist. Der Importzeitpunkt wird für Prozessinstanzen, Funktionen, Ereignisse, Organisationseinheiten (jeweils als AT_ORIG_EPK_ID) und auf Attributebene gemerkt. Nach dem Import ist für jedes Attribut bekannt, wann es importiert wurde. Der Importzeitpunkt wird während des XML-Imports oder Process Imports an die importierte EPK geschrieben.

Ist für ein Attribut kein Importzeitpunkt bekannt (z. B. bei Bestandsdaten, die im Merge-Modus **Ersetzen** importiert wurden), wird statt dessen der Importzeitpunkt des Objektes (Funktion, Prozess, …) verwendet, dem das Attribut zugeordnet ist.

Additiver Merge auf Funktionsebene

Wenn während der Zusammenführung zwei Funktionen mit dem gleichem internen Objektschlüssel vorhanden sind, werden diese wie folgt zusammengeführt.

1. Die neuere Funktion (zuletzt importiert) wird mit ihren Attributen und Organisationseinheiten in die zusammengeführte Prozessinstanz übernommen.
2. Alle Attribute der alten Funktion werden an die neue Funktion kopiert. Falls ein Attribut an beiden Funktionen vorhanden ist, wird das Attribut der alten Funktion kopiert falls dies neuer ist.
3. Alle Organisationseinheiten der alten Funktion und der neuen Funktion werden zusammengeführt. Folgendes Kapitel beschreibt das Zusammenführen der Organisationseinheiten.

Organisationseinheiten

Beim Zusammenführen zweier Funktionen werden alle Organisationseinheiten der alten Funktion an die neue Funktion kopiert. Falls dieselbe Organisationseinheit an beiden Funktionen vorhanden ist, bleibt die neuere Organisationseinheit mit ihrer zugehörigen Kante und ihren Attributen erhalten. In diesem Fall werden Attribute der Kante, die der älteren Organisationseinheiten zugeordnet ist und Attribute der älteren Organisationseinheit, nicht übernommen.

Gleichheit von Organisationseinheiten

1. Beim Zusammenführen wird die Gleichheit zweier Organisationseinheiten, die nicht anonymisiert sind, ausschließlich anhand des gleichen Attributs **AT_OBJNAME** festgestellt.

   Damit eine Organisationseinheit, die anonymisiert wurde (und somit der ursprüngliche Objektname geändert wurde) bei einem erneuten Reimport noch als gleich erkannt wird, muss zusätzlich ein interner Objektschlüssel **AT_INTERNAL_OBJECT_KEY** an der Organisationseinheit vorhanden sein. Der Objektschlüssel kann über Objektschlüsselregeln definiert werden.
2. Eine nicht anonymisierte und eine anonymisierte Organisationseinheit sind gleich, wenn der Objektschlüssel **AT_INTERNAL_OBJECT_KEY** übereinstimmt.
3. Zwei anonymisierte Organisationseinheiten sind gleich, wenn die Objektschlüssel **AT_OBJNAME** und **AT_INTERNAL_OBJECT_KEY** übereinstimmen.

Importzeitpunkt der Organisationseinheit

Ist für eine Organisationseinheit kein Schlüssel **AT_ORIG_EPK_ID** definiert, wird der Schlüssel **AT_ORIG_EPK_ID** der verbundenen Funktion verwendet, um den Importzeitpunkt der Organisationseinheit zu bestimmen.

Additiver Merge auf Prozessebene

Bei der Zusammenführung werden alle Attribute der neueren und älteren Prozessinstanz an das Ergebnisfragment kopiert. Ist ein Attribute an beiden Prozessinstanzen vorhanden, wird das neuere Attribut genommen.

Additiver Merge auf Eventebene

Wenn während des Zusammenführens zwei Events mit dem gleichen internen Objektschlüssel vorhanden sind, werden diese behandelt wie im Modus **Ersetzen**. Das heißt, das neuere Event ersetzt das ältere Event, ohne dass vom alten Event Attribute zum Neueren übernommen werden.

Konfiguration

Die DTD mergerconfig.dtd [29] enthält zur Konfiguration die Modi **Ersetzen (replace)** und **Zusammenführen (update)**.

```
<!ELEMENT processmerge ( algorithm?, mergeattributes?)>
<!ATTLIST processmerge
mode (replace|update) 'replace'
>
```

Das Attribut **mode** ist optional und falls es fehlt, wird standardmäßig der Modus **Ersetzen** genommen.

Im Modus **Zusammenführen** werden die Merge-Attribute (mergeattributes) nicht ausgewertet. Wird eine Konfiguration mit Attribut **mode update** importiert, die auch (nicht-leere) Merge-Attribute enthält, so wird eine entsprechende Meldung ausgegeben.

Ändern des Merge-Modus

Ein Umstellen des Merge-Modus ist über die Merge-Konfiguration beliebig oft möglich. Somit können bestehende Mandaten im Merge-Modus **Ersetzen** auf den Modus **Zusammenführen** umgestellt werden.

Im Merge-Modus **Zusammenführen** wird die Liste der Merge-Attribute nicht verwendet.

Sharedfragment

Für den Merge zweier Sharedfragments im Modus **Zusammenführen** wird bei Verwendung des Standardalgorithmus (ZDefaultSharedFragmentMergeAlgorithm) automatisch der gleiche Merge-Modus verwendet, der auch für den Merge zweier normaler Fragmente in der Merge-Konfiguration angegeben ist.

Spezielle Attribute

Spezielle Attribute (wie z. B. interne PPM-Attribute oder Attribute wie AT_IS_SHARED_FUNCTION) werden im Merge-Modus **Zusammenführen** wie alle anderen Attribute behandelt.

Prozesstypisierung

Um im Merge-Modus **Zusammenführen** Prozesstypinformationen direkt aus dem Quellsystem zu übernehmen (ohne Typsierungsregeln) müssen die Attribute **AT_INTERNAL_PROCTYPE** und **AT_INTERNAL_PROCTYPEGROUP** mit den Typisierungsinformationen (Prozesstyp und Prozesstypgruppe) gepflegt sein.

Berechnete Attribute

Berechnete Attribute am Prozess oder an Funktionen, werden ebenfalls durch den Merge-Modus **Zusammenführen** kopiert.

Wenn beim Zusammenführen zweier Funktionen ein berechnetes Attribut der alten Funktion an die neue Funktion kopiert wird und dieses später nicht mehr berechnet werden kann, dann wäre der alte, berechnete Wert an der zusammengeführten Funktion vorhanden.

Wenn Sie dieses Verhalten nicht wünschen, müssen Sie an der Berechnungsvorschrift den Parameter **calcattr delete=yes** setzen. Ist der Parameter gesetzt, werden Attribute, die nicht berechnet werden können, später wieder gelöscht.

6.4. Zusammenführen von Ereignissen

6.4.1. Parallele Pfade mit mehrwertigen Schlüsseln

Wenn Sie beim Verwenden von schlüsselbasiertem Merge parallele Prozesspfade wieder zusammenführen möchten, müssen Sie für das Startereignis des zusammenführenden Fragments mehrere Merge-Schlüssel für die Endereignisse der Vorgängerfragmente berechnen. Im XML-Element **multikey** geben Sie hierfür ein mehrwertiges Attribut und ein Trennzeichen an. Mehrwertig bedeutet, dass der Wert des angegebenen Attributes mit Hilfe des Trennzeichens in mehrere Teile aufgespalten wird. Für jeden Teil wird ein Merge-Schlüssel berechnet.

Beispiel 12.

Die Fragmente einer Prozessinstanz sind über die Merge-Attribute **THIS_KEY** und **PREV_KEY** miteinander verknüpft. Der Schlüssel zum Vorgänger wird im Attribut **PREV_KEY** gespeichert. Wenn ein System-Event mehrere Vorgänger hat, wird jeder Schlüssel der Vorgänger im Attribut **PREV_KEY** mehrfach in die Protokolldatei geschrieben. Das verwendete Attribut-Mapping ist so konfiguriert, dass das Attribut **PREV_KEY** mehrwertig als **AT_KEY** an den Merge-Ereignissen instanziiert wird.

Die Extraktion aus Ihrem Quellsystem enthält folgendes System-Event:

```
...
<attribute type="EVENTTYP">Change customer order</attribute>
<attribute type="PROC_ID">123456</attribute>
<attribute type="THIS_KEY">3</attribute>
<attribute type="PREV_KEY">1</attribute>
<attribute type="PREV_KEY">2</attribute>
<attribute type="USER">Team A</attribute>
...
```

Beispiel 13.

Die verwendete Mapping-Datei enthält folgendes Attribut-Mapping für Start- und Endereignisse:

```
...
<!-- mapping startevents -->
<attribute ppmattributetype="AT_KEY">
<multieventattributetype delimiter=";">PREV_KEY</multieventattributetype>
</attribute>
<!-- mapping endevents -->
<attribute ppmattributetype="AT_KEY">
<eventattributetype>THIS_KEY</eventattributetype>
</attribute>
...
```

Beispiel 14.

Die aktive Merge-Konfiguration enthält folgende Regel:

```
...
<mergekeyrule>
<refobjects>
<refobject objecttype="OT_EVT"></refobject>
</refobjects>
<keyparts>
<multikey attributetype="AT_KEY" delimiter=";" />
</keyparts>
</mergekeyrule>
...
```

Beispiel 15.

Im Fragment, dessen Startereignis ein Attribut **THIS_KEY** mit dem Wert **3** hat, werden die Vorgängerfragmente zusammengeführt, deren Endereignisse ein Attribut **THIS_KEY** mit dem Wert **1** bzw. **2** haben.

Nach dem Zusammenführen von Fragmenten werden die Attribute unverändert übernommen, d. h. die entstandene Gesamtmenge der Teilschlüssel wird nicht konsolidiert an das Merge-Attribut des verbleibenden Ereignisses zurück geschrieben.

Beispiel 16.

Wenn Sie beispielsweise ein Regelwerk verwenden, das aus dem Multivalue-Attribut **AT_KEY** einen Merge-Schlüssel berechnet, und **AT_KEY** am System-Event **A** den Wert **x;y** und an System-Event **B** den Wert **y;z** hat, hat das Attribut **AT_KEY** den Wert des Attributes von System-Event **B**, unter der Annahme, dass System-Event **B** das später importierte ist. Das Zusammenführen von Fragmenten bleibt davon unberührt, da Fragmente über bereits beim Import berechnete Merge-Schlüssel zusammengeführt werden.

6.4.2. Merge-Modus

Beim Zusammenführen der Merge-Ereignisse können Sie im Attribut **eventmode** des XML-Elementes **mergehandling** der Merger-Konfiguration optional angeben, welches der Merge-Ereignisse in das Ergebnisfragment übernommen werden soll. Zulässige Werte sind **STARTEVENT**, **ENDEVENT** und **IMPORTTIME**, Vorgabewert ist **IMPORTTIME**.

Es gibt folgende Ereignistypen:

Typ	Beschreibung
Startereignis	Ein Startereignis hat keinen Vorgänger (nur Ausgangskante).
Koppelereignis	Ein Koppelereignis hat sowohl Vorgänger als auch Nachfolger (Ein- und Ausgangskante).
Endereignis	Ein Endereignis hat keine Nachfolger (nur Eingangskante).

Schlüsselbasierter Merge

Bei Verwendung des schlüsselbasierten Merge verhält sich das Zusammenführen von Merge-Ereignissen wie folgt:

eventmode	Beschreibung
IMPORTTIME	Das später importierte Ereignis wird unabhängig vom Ereignistyp übernommen. Standardwert
STARTEVENT	Es wird das später importierte Ereignis übernommen, das kein Endereignis ist. Ein Endereignis wird nur dann übernommen, wenn zwei Endereignisse zusammengeführt werden.
ENDEVENT	Es wird das später importierte Ereignis übernommen, das kein Startereignis ist. Ein Startereignis wird nur dann übernommen, wenn zwei Startereignis zusammengeführt werden.

Ordnungsbasierter Merge

Bei Verwendung des Ordnungsbasierten Merge werden die gewählten Ereignistypen direkt beim Zusammenführen bevorzugt, da die Prozessinstanz vor dem Merge immer in Einzelfragmente der Form Ereignis-Funktion-Ereignis zerlegt wird. Werden zwei Merge-Ereignisse gleichen Typs zusammengeführt, wird das später importierte übernommen.

6.5. Attributkopierregeln

Zur Berechnung der instanzbezogenen Kennzahlen und Bildung von Dimensionen werden Attribute der Prozessinstanz benötigt. Beim Datenimport im PPM-Ereignisformat können Attribute der Prozessinstanzfragmente nicht direkt eingelesen werden, da das Instanzfragment dynamisch aus einer Fragmentdefinition erzeugt wird. Daher werden Objektattribute des instanziierten Prozessfragments an die Prozessinstanz kopiert.

Die Regeln zum Kopieren von Objektattributen an die Prozessinstanz setzen sich aus folgenden Teilen zusammen:

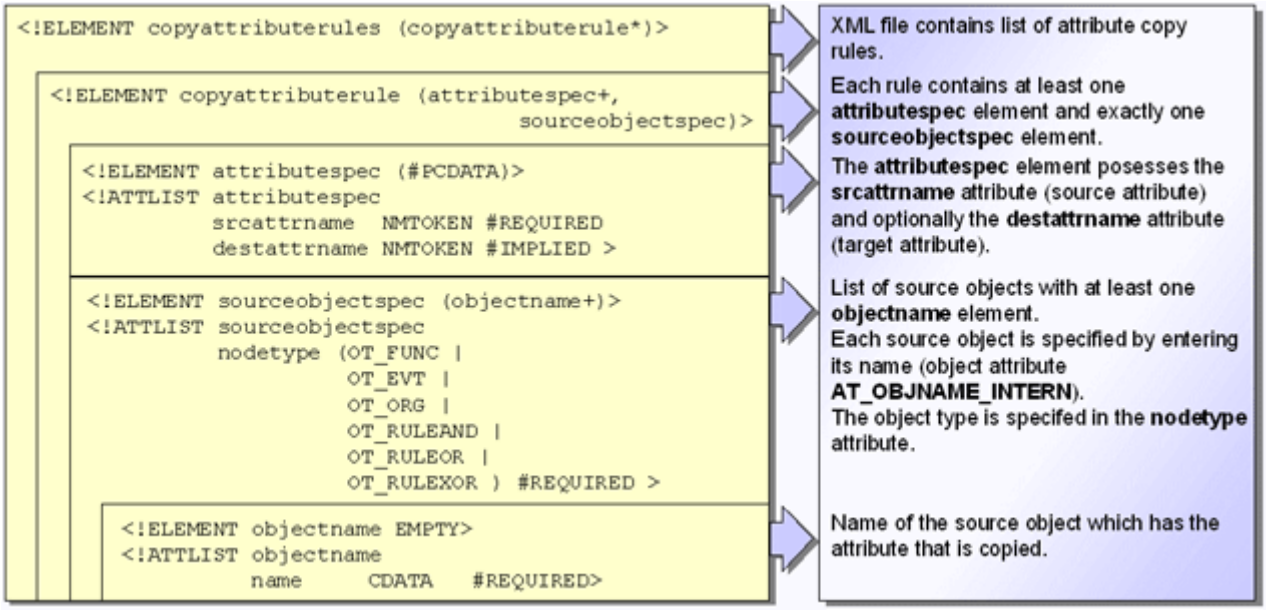
- Liste der zu kopierenden Attribute. Die angegebene Attributkopierregel wird für jeden Attributtyp der Liste angewendet.
- Quellobjekttyp der Objekte, deren Attributtyp kopiert werden soll
- Priorisierte Liste der Objekte (Objektattribut AT_OBJNAME_INTERN), deren Attributtyp kopiert werden soll. Die Liste der Objekte wird von oben nach unten abgearbeitet. Sobald der Attributtyp kopiert werden konnte, wird die nächste Kopierregel bearbeitet.

Falls der Attributtyp an keinem der angegebenen Objekte und auch nicht an der Prozessinstanz gepflegt ist, wird der Attributtyp mit dem Vorgabewert erzeugt, der im Bereich **#PCDATA** des XML-Elements **attributspec** angegeben ist.

Das folgende Beispiel kopiert das Attribut **AT_ID** der Funktion **FCT_Auftrag_anlegen** an die Prozessinstanz. Falls auf das Attribut nicht zugegriffen werden kann, da es nicht gepflegt ist oder die Funktion nicht existiert, wird das Attribut des danach angegebenen Objekts **FCT_Rechnung_erstellen** kopiert.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE copyattributerules SYSTEM "copyattributerules.dtd">
<copyattributerules>
...
<copyattributerule>
<attributespec srcattrname="AT_ID"/>
...
<sourceobjectspec nodetype="OT_FUNC">
<objectname name="FCT_Create_order"/>
<objectname name="FCT_Create_invoice"/>
...
</sourceobjectspec>
</copyattributerule>
...
</copyattributerules>
```

Die DTD **CopyAttributeRules.dtd** beschreibt den Aufbau der XML-Datei für Attributkopierregeln:



Alle in der Konfigurationsdatei angegebenen Quellobjektnamen referenzieren das Attribut **AT_OBJNAME_INTERN**.

Durch das Kopieren von Objektattributen ist es auch bei Verwendung des PPM-Ereignisformats möglich, Prozesstypinformationen direkt aus dem Quellsystem zu übernehmen. Die dem Prozesstypen und der Prozesstypgruppe entsprechenden Attribute werden über das Mapping als Attribute **AT_PROCTYPE** und **AT_PROCTYPEGROUP** an die Prozessinstanzobjekte geschrieben, die in jeder Prozessinstanz vorkommen, und über Attributkopierregeln an die Prozessinstanz kopiert.

6.6. Anonymisieren

Es kann sinnvoll sein, die Namen der Bearbeiter, die an der Ausführung einer Funktion beteiligt sind, nicht anzuzeigen, z. B. aus datenschutzrechtlichen Erwägungen. Nach dem Initialisieren der Datenbank des PPM-Mandanten können Sie im PPM-Frontend mit Hilfe der Administrationskomponente **Organisationseinheiten** bestimmen, wie die Namen der in den Instanzdaten vorkommenden Bearbeiter durch den Namen von Organisationseinheiten ersetzt (anonymisiert) werden.

Beim Verdichten von Prozessinstanzen gehen die Informationen über die Bearbeiter von Funktionen verloren. Um Informationen über die Bearbeiter in eine verdichtete EPK zu übernehmen, müssen diese anonymisiert werden.

Hierfür erzeugen Sie PPM-Organisationseinheiten und ordnen alle gewünschten Bearbeiter Organisationseinheiten zu. Beim Datenimport werden die Namen der Bearbeiter durch den Namen der entsprechenden Organisationseinheit ersetzt.

Eine Organisationseinheit wird im PPM-System durch folgende Eigenschaften beschrieben:

Oberflächenelement	Beschreibung
Name	Name der Organisationseinheit (frei wählbar)
Bearbeiter	Liste der Bearbeiter, die der gewählten Organisationseinheit zugewiesen sind
Kostensatz	Kosten eines Mitarbeiters einer Organisationseinheit pro Zeiteinheit. Der Kostensatz fließt in die Berechnung bestimmter Kennzahlen der Prozesskostenrechnung ein (siehe Kapitel Definition von Prozesskostenkennzahlen [134]).

Oberflächenelement	Beschreibung
Bei Kennzahlenberechnung ignorieren	Die ausgewählte Organisationseinheit wird bei der Kennzahlenberechnung nicht berücksichtigt. Dies wird z. B. angegeben, wenn Bearbeiter Funktionen im Rahmen einer Stapelverarbeitung ausführen (sogenannte Batch User).
Alle nicht zugeordneten Bearbeiter in diese Organisationseinheit	Bearbeiter, die keiner Organisationseinheit zugeordnet sind, werden mit dem Namen dieser Organisationseinheit anonymisiert. Sie müssen eine solche Standard- Organisationseinheit definieren, um die Konfiguration speichern zu können. Die Organisationseinheit muss mindestens einen Bearbeiter haben.

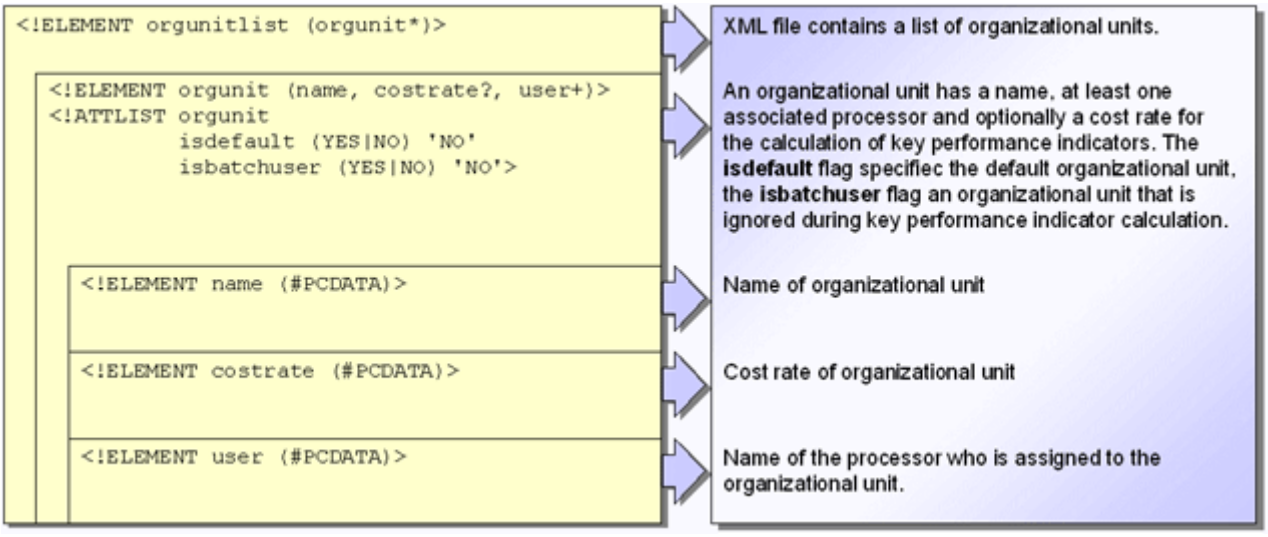
Organisationseinheiten können mit dem Kommandozeilenprogramm **runppmconfig** in eine XML-Datei, die Sie auch von Hand erweitern können, exportiert werden. Nähere Informationen zum Kommandozeilenprogramm finden Sie im Benutzerhandbuch **PPM Operation Guide**.

Beispiel 17.

Herr Maier und **Frau Schmidt** arbeiten in der Abteilung Auftragsannahme, **Herr Müller** in der Abteilung Rechnungswesen. Beim Datenimport werden die Bearbeiter **Herr Maier** und **Frau Schmidt** durch den Namen der Organisationseinheit Auftragsannahme ersetzt, **Herr Müller** wird durch den Namen Rechnungswesen ersetzt, alle anderen Bearbeiter durch den Namen der Standardorganisationseinheit **Nicht spezifiziert**. Die angegebenen Kostensätze werden vom Kennzahlenberechner zur Prozesskostenanalyse verwendet.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE orgunitlist SYSTEM "orgunits.dtd">
<orgunitlist>
<orgunit isdefault="NO" isbatchuser="NO">
<name>ORDER ACCEPTANCE</name>
<costrate>15.75 EUR_PER_HOUR</costrate>
<user>Mrs Smith</user>
<user>Mr Brown</user>
</orgunit>
<orgunit isdefault="NO" isbatchuser="NO">
<name>ACCOUNTING</name>
<costrate>25.3 EUR_PER_HOUR</costrate>
<user>Mr Miller</user>
</orgunit>
<orgunit isdefault="YES" isbatchuser="NO">
<name>Not specified</name>
<costrate>24 EUR_PER_HOUR</costrate>
<user>DUMMY</user>
</orgunit>
</orgunitlist>
```

Die Dokumenttypdefinition **orgunits.dtd** dieser XML-Datei sieht folgendermaßen aus:



Die erstellte XML-Datei können Sie beim Initialisieren der Datenbank durch Angabe des Dateinamens einlesen, sodass sofort Anonymisierungsregeln zur Verfügung stehen.

7. Prozesstypisierung

Die automatische Zuordnung von Prozessinstanzen zu einem bestimmten Prozesstyp erfolgt über eine Typisierungsregel, die mit ihrem eindeutigen, internen Namen in der Prozessbaumkonfiguration am entsprechenden Prozesstyp angemeldet ist.

Die Typisierungsregel selbst wird als Berechnungsfunktion in der Kennzahlenkonfiguration definiert. Pro Prozesstyp kann maximal eine Typisierungsregel definiert werden. Für Details siehe Typisierungsregeln erstellen [48].

Alternativ kann die Prozesstypisierung durch den Import von Werten in spezifische Attribute erfolgen, die sogenannte „Vortypisierung“. Für Details siehe Typisierung durch Attributberechnung [128].

7.1. Typisierungsregeln erstellen

In den folgenden Kapiteln werden die beiden Schritte zur Erstellung einer Typisierungsregel beschrieben:

- Definition einer Typisierungsregel in der Kennzahlenkonfiguration
- Verwendung dieser Typisierungsregel in der Prozessbaumkonfiguration

Verwenden Sie PPM Customizing Toolkit zur Erstellung von Typisierungsregeln. Sie können damit auf einfache Weise Regeln erstellen und vermeiden Fehlerquellen speziell bei komplexeren Berechnungsvorschriften von Typisierungsregeln (Kapitel **Typisierungsregeln im CTK [127]**). Die Änderungen importieren Sie ins PPM-System, indem Sie die Konfiguration aktivieren.

7.1.1. Kennzahlenkonfiguration

Die Definition der Typisierungsregeln erfolgt auf der Basis von Attributberechnungen in der Kennzahlenkonfiguration. Die Berechnungsvorschrift wird im XML-Element **function** angegeben.

XML-Tag	Beschreibung
function name	Name der Typisierungsregel, wird vom XML-Attribut **typifierrule function** der Prozessbaumkonfiguration referenziert.
resulttype	Ergebniswert. Muss vom Typ BOOLEAN sein.
datatype	Datentyp. Muss vom Typ BOOLEAN sein.

Eine Typisierungsregel zu obigem Beispiel könnte folgendermaßen aussehen:

Beispiel 18. ***[en-us]***

Auszug aus der Datei **Keyindicator.xml**

```
...
<function name="typifierrule_OrderProcessing_StandardOrder"
resulttype="BOOLEAN" datatype="BOOLEAN">
<in>
<constant>
<dataitem>
```

```
C
<datatype name="TEXT">Text</datatype>
</dataitem>
</constant>
<attribute name="AT_SAP_VBTYP" nodetype="OT_FUNC">
<in>
<constant>
<dataitem>
SO
<datatype name="TEXT">Text</datatype>
</dataitem>
</constant>
<attribute name="AT_SAP_VKBELEGART" nodetype="OT_FUNC" objectname="this"/>
</in>
</attribute>
</in>
</function>
...
```

Durch den Aufruf der Funktion (Typisierungsregel) **typifierrule_OrderProcessing_StandardOrder** wird überprüft, ob die zu typisierende Prozessinstanz als **Order processing\Standard order** typisiert wird (Rückgabewert=**true**) oder nicht (Rückgabewert=**false**).

Die Prozessinstanz wird unter folgender Bedingung dem Prozesstyp **Standard order** zugeordnet: Es muss mindestens eine Funktion in der Prozessinstanz geben, die die Attributtypkombination **AT_SAP_VBTYP** mit dem Wert **C** und **AT_SAP_VKBELEGART** mit dem Wert **SO** trägt.

7.1.2. Prozessbaumkonfiguration

Die zuvor in der Datei **Keyindicator.xml** definierten Typisierungsregeln müssen Sie nun den einzelnen Prozesstypen in der Datei **Processtree.xml** zuweisen. Dabei kann pro Prozesstyp jeweils nur eine Typisierungsregel angegeben werden. Das XML-Element **typifierrule** ist optional. Ist für einen Prozesstyp keine Typisierungsregel angegeben, wird dieser bei der Typisierung ignoriert.

Zu den Attributen **function** und **priority** müssen Sie entsprechende Angaben machen. Mit dem Attribut **function** geben Sie den Namen der Typisierungsregel, die Sie für diesen Prozesstyp verwenden möchten, aus der Kennzahlenkonfiguration an.

```
<processtree name="Processes">
<processtypegroup name="OrderProcessing">
...
<processtype name="StandardOrder" autovisible="TRUE">
<typifierrule function="typifierrule_OrderProcessing_StandardOrder"
priority="0"/>
<processparamset>
...
</processparamset>
...
<functionparamset>
...
</functionparamset>
...
```

```
<useki name="..." assessment="..."/>
...
<usedim name="..."/>
...
</processtype>
...
</processtypegroup>
...
</processtree>
```

7.1.2.1. Priorisierung

Eine Prozessinstanz wird stets genau einem Prozesstyp zugeordnet. Treffen auf eine Prozessinstanz mehrere Typisierungsregeln zu, wird mit Hilfe des ganzzahligen Attributs **priority** festgelegt, welchem Prozesstyp die Prozessinstanz letztlich zugeordnet wird. Eine Regel mit der Priorität **0** hat dabei die höchste Priorität und wird vom Typisierer vorrangig abgearbeitet.

Treffen bspw. auf eine Prozessinstanz drei Typisierungsregeln mit den Prioritäten **2**, **3** und **6** zu, wird diese gemäß der Typisierungsregel mit der Priorität **2** typisiert. Die Typisierungsregeln **3** und **6** werden für diese Prozessinstanz ignoriert. Der Typisierer prüft zunächst, ob eine Typisierungsregel der Priorität **0** auf die zu typisierende Prozessinstanz zutrifft. Ist dies nicht der Fall, wird eine Regel der nächst niedrigeren Prioritätsstufe auf Zutreffen überprüft usw.

Sobald eine Regel zutrifft, wird die Abarbeitung der Typisierungsregeln abgebrochen und die Prozessinstanz dem entsprechenden Prozesstyp zugeordnet.

Die Priorisierung von Typisierungsregeln legen Sie im PPM CTK-Modul **Prozesse** über über die Schaltfläche **Prozesstypisierungsregeln priorisieren** auf der gewählten Typisierungsregel fest.

7.1.3. Definition von Attributberechnungen

Zur Berechnung einer Kennzahl oder Bildung einer Dimension wird entweder der Wert eines bereits existierenden Attributes verwendet oder der Algorithmus zur Berechnung des Attributes in der XML-Konfigurationsdatei angegeben (XML-Element **calcattr**). Der angegebene Attributname muss in der importierten Attributdefinition des PPM-Systems enthalten sein (Dateien **AttributeTypes.xml** und **AttributeNames.xml**). Durch die Attributdefinition sind Datentyp und Basiseinheit festgelegt. Die Berechnung eines Attributes wird immer in der Basiseinheit des Attributtyps durchgeführt. Auch das Ergebnis wird als Wert in der Basiseinheit gespeichert.

Attribute sind nur innerhalb einer Prozessinstanz gültig. Es ist nicht möglich, Attribute anderer Prozessinstanzen zu berechnen oder in die Berechnung mit einzubeziehen.

Die Berechnung eines Attributes kann von anderen Attributen abhängig gemacht werden, die ihrerseits ebenfalls berechnete Attribute sein können. Alle mit dem XML-Element **depends** angegebenen Attribute werden berechnet, bevor die Berechnung des aktuellen Attributes ausgeführt wird. Zyklische Abhängigkeiten werden während des Imports der Kennzahlenkonfiguration erkannt und mit einer Fehlermeldung quittiert.

Optional kann ein Standardwert angegeben werden (XML-Element **defaultvalue**), der dem Attributwert zugewiesen wird, falls die Berechnung nicht erfolgreich durchgeführt werden konnte. Der Standardwert muss immer mit einer Einheit angegeben werden, die für den Datentyp des Attributes zulässig ist. Nur auf diese Weise kann der Wert in der Basiseinheit korrekt bestimmt werden.

Wird eine Berechnungsvorschrift zur Berechnung angewendet (XML-Element **calculation**), wird immer der Standardwert verwendet, falls die Berechnung nicht erfolgreich durchgeführt werden konnte. Wird allerdings ein Algorithmus zur Berechnung verwendet (XML-Element **calcattr**),wird der Standardwert nur zusammen mit den folgenden Klassen verwendet:

- ZAttributeCalculatorPKSR
- ZAttributeCalculatorPKSS
- ZAttributeCalculatorTransformUniversalMappingByParam

Alle anderen Berechnungsklassen ignorieren den Standardwert.

Wenn ein Attribut nicht berechnet werden kann und kein Standardwert angegeben ist, wird das Attribut an der Prozessinstanz bzw. den Prozessinstanzobjekten nicht erzeugt und eine entsprechende Meldung ausgegeben.

Die XML-Struktur zur Definition einer Attributberechnung sieht folgendermaßen aus:

```
...
<calcattr name="..." type="..." objectname="..." scale="..." delete="...">
<depends attrname="..." type="..."/>
<defaultvalue>"..."</defaultvalue>
```

entweder

```
<calculation> ... </calculation>

  </calcattr>
...
```

or

```
<calcclass name="..."/>
</calcattr>
...
```

XML-Tag	Beschreibung
name	Interner Name des zu berechnenden Attributes. Das Attribut wird an allen durch **type** angegebenen Objekttypen der gerade bearbeiteten Prozessinstanz erzeugt. Ein bereits existierendes Attribut wird überschrieben.
type	Objekttyp, an den das Attribut geschrieben wird **PROCESS**: Berechnetes Attribut wird an die Prozessinstanz geschrieben. **OT_FUNC**: Berechnetes Attribut wird an alle Funktionen der Prozessinstanz geschrieben. **OT_ORG**: Berechnetes Attribut wird an alle Organisationseinheiten der Prozessinstanz geschrieben. **OT_EVT**: Berechnetes Attribut wird an alle Ereignisse der Prozessinstanz geschrieben. **RELATION**: Berechnetes Attribut wird an die Beziehung, die mittels **relname** angegeben ist, in der Prozessinstanz geschrieben.
relname (optional)	Nur bei **type="RELATION"**. Angabe der Beziehung, auf die sich die Attributberechnung beziehen soll. Ist anstelle von dependsrel zu verwenden.

XML-Tag	Beschreibung
objectname (optional)	Interner Name der Funktion (Objektattribut **AT_OBJNAME_INTERN**), an die das Attribut geschrieben wird. Diese Option darf nur für die Berechnung von Funktionsattributen verwendet werden. Mehrerer Objektnamen werden durch Komma getrennt angegeben, Platzhalter ***** und **?** können im Objektnamen beliebige verwendet werden.
scale (optional)	Das Ergebnis einer Berechnungsvorschrift wird in der angegebenen Skalierung an das Attribut geschrieben. Der Skalierungswert hängt vom Datentyp des Attributs ab, das der Berechnung zugrunde liegt. Ohne Angabe wird das Ergebnis in der Basiseinheit des Attributdatentyps ausgegeben.
delete	Mit yes wird ein früher berechneter Attributwert vor der erneuten Berechnung gelöscht (vorgeschrieben z.B. für die Definition kritischer Zeitattribute, die vom Frühwarnsystem verwendet werden, Kapitel Zeitdimensionen für das Frühwarnsystem [165]). Standardwert:**nein**

Wenn Sie eine Berechnungsvorschrift gezielt für eine oder mehrere bestimmte Funktionen angeben möchten, spezifizieren Sie die entsprechenden Funktionsnamen im XML-Attribut **objectname**.

Beispiel 19.

Die Berechnungsvorschrift wird nur für die Funktionen, deren interne Namen den angegebenen Mustern entsprechen, ausgeführt.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="AT_END_TIME" type="OT_FUNC" objectname="FCT_AUFT??_*,
FCT_END_*_?, *_AUFTRAG,FCT_AUFTR_START">
<calculation>
...
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Beispiel 20.

Die Definition einer Berechnungsvorschrift wird durch Angabe folgender XML-Elemente vervollständigt:

XML-Element	Beschreibung
depends (optional)	Name und Typ eines Attributs (**PROCESS**, **OT_FUNC**, **OT_EVT**, **OT_ORG** oder **RELATION**), das existieren muss, damit die Berechnung ausgeführt wird. Falls es sich bei dem angegebenen Attribut um ein zu berechnendes Attribut handelt, wird dieses zunächst berechnet. Im Attribut **relname** wird die Beziehung angegeben, von der eine Abhängigkeit besteht (nur bei **type="RELATION"**). Es können gleichzeitig mehrere Elemente **depends** angegeben werden. Nicht zusammen mit **dependsrel** zu verwenden.
dependsrel (optional)	Name der Beziehung, von der eine Abhängigkeit besteht (nur bei **type="RELATION"**). Anstelle von **relname** zu verwenden. Nicht zusammen mit **depends** zu verwenden.
defaultvalue (optional)	Vorgabewert des Attributes, falls das Attribut aus einem beliebigen Grund nicht berechnet werden kann.
calcclass	Eindeutige Angabe der Berechnungsvorschrift durch eines der beiden XML-Elemente.
calculation	Mit **calcclass** wird der Algorithmus durch Angabe eines vollständigen Java-Klassenpfades spezifiziert. Durch **calculation** wird eine Berechnungsformel direkt in XML-Notation angegeben.

XML-Element	Beschreibung
calcparam (optional)	Nur für **calcclass**. Übergabe beliebig vieler Parameter (calcparam) beim Aufruf einer Berechnungsklasse. Mit **key** wird der eindeutige, interne Namen des Parameters und mit **value** der entsprechende Wert des Parameters definiert.

Beispiel 21.

Zur Berechnung eines Attributes durch das XML-Element **calculation** steht ein umfangreiches Regelwerk zur Definition von Berechnungsvorschriften zur Verfügung.

Das folgende Beispiel zeigt die XML-Definition der Berechnung der Bearbeitungszeit einer Prozessinstanz. Die Bearbeitungszeit ist als Differenz des frühesten Startzeitpunkts und spätesten Endzeitpunkts aller Funktionen einer Prozessinstanz definiert. Zur Speicherung des Kennzahlenwerts wird das Attribut **AT_KI_PROCESSTIME** (Typ: Zeitspanne) gewählt. Die Berechnung soll nur dann durchgeführt werden, wenn die beiden Attribute **AT_START_TIME** und **AT_END_TIME** an wenigstens einer Funktion der Prozessinstanz gepflegt sind. Dies muss nicht notwendigerweise dieselbe Funktion sein. Schlägt die Berechnung aus einem beliebigen Grund fehl, wird dem Ergebnisattribut der Vorgabewert **0 SECOND** zugewiesen.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="AT_KI_PROCESSTIME" type="PROCESS">
<depends attrname="AT_END_TIME" nodetype="OT_FUNC">
<depends attrname="AT_START_TIME" nodetype="OT_FUNC">
<defaultvalue>0 SECOND</defaultvalue>
<calculation>
<timespan>
<filteredattribute name="AT_END_TIME" nodetype="OT_FUNC" filter="LATEST"/>
<filteredattribute name="AT_START_TIME" nodetype="OT_FUNC" filter="EARLY"/>
</timespan>
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Ein berechnetes Attribut enthält immer den Ergebniswert und die Ergebniseinheit. Die Ergebniseinheit wird immer in der dem Datentyp des Ergebnisattributes entsprechenden Basiseinheit angegeben.

7.1.3.1. Berechnungsklassen

Dieses Kapitel beschreibt alle in PPM enthaltenen Berechnungsvorschriften, die im XML-Element **calcclass** als Klassenname zur Berechnung eines Attributes angegeben werden können. Der Klassenname muss mit Java-Paketstrukturpfad angegeben werden.

Beispiel

```
<calcattr name="AT_KI_FDLZWK" type="OT_FUNC">
<calcclass name="com.idsscheer.ppm.server.keyindicator.
attributecalculator.ZAttributeCalculatorFDLZWK"/>
</calcattr>
```

Im Folgenden wird der feststehende Teil des Klassennamens ausgelassen. Anstelle von z. B.

com.idsscheer.ppm.server.keyindicator.attributecalculator.ZAttributeCalculatorFDLZWK,

wird **ZAttributeCalculatorFDLZWK** angegeben.

7.1.3.1.1. Log-Ausgaben der Berechnungsklassen

Sie können für jede Berechnungsklasse optional im XML-Attribut **loglevel** angeben, welche Meldungen ausgegeben werden sollen. Geben Sie einen der Werte **SILENT**, **DEFAULT** oder **VERBOSE** an.

Die folgende Tabelle zeigt die standardmäßige Verknüpfung des im XML-Attribut **loglevel** angegebenen Wertes mit der Art der Meldungen, die von der Berechnungsklasse ausgegeben werden. Die Art der Meldungen wird hierbei über das zugeordnete Logger-Modul in der mandantenspezifischen Konfigurationsdatei **Server_Log_settings.properties** bestimmt.

Log-Level	Beschreibung
SILENT	Alle Log-Ausgaben werden unterdrückt.
DEFAULT	Warnungen und Fehlermeldungen werden ausgegeben.
VERBOSE	Informationen, Warnungen und Fehlermeldungen werden ausgegeben.

Dateiauszug **Server_Log_settings.properties**:

```
...
#MODULE_CALCCLASS_SILENT
log4j.logger.LOG.CCS=FATAL
#MODULE_CALCCLASS_DEFAULT
log4j.logger.LOG.CCD=WARN
#MODULE_CALCCLASS_VERBOSE
log4j.logger.LOG.CCV=INFO
...
```

Beispiel 22.

Der folgende Dateiauszug definiert für die Berechnung der Funktionsdurchlaufzeit, dass weder Informationen, Warnungen noch Fehlermeldungen ausgegeben werden sollen.

```
...
<calcattr name="AT_KI_FDLZ" type="OT_FUNC" loglevel="SILENT">
<calcclass
name="com.idsscheer.ppm.server.keyindicator.attributecalculator.ZAttributeCalcul
atorFDLZ"/>
</calcattr>
...
```

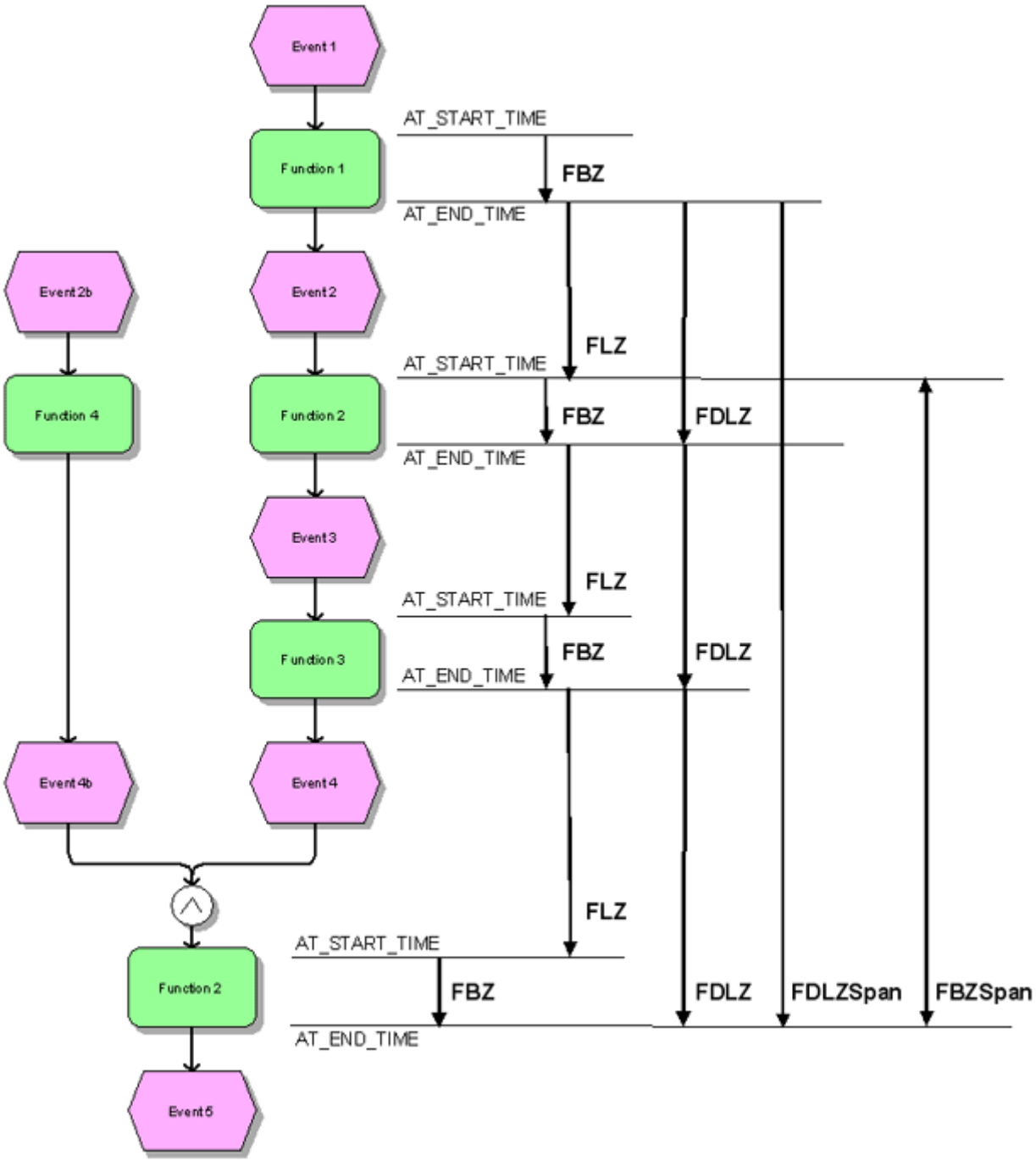
Wenn Sie an der Kommandozeile Argumente zur Steuerung der log-Ausgaben angeben, haben diese Vorrang gegenüber den Angaben in den **Server_settings.properties**. (Kapitel **Gemeinsame Argumente**)

7.1.3.1.2. Zeitenkennzahlen

Zur Berechnung von Zeitenkennzahlen werden die Funktionsattribute **AT_START_TIME** und **AT_END_TIME** verwendet. Diese Attribute werden durch Mapping eines geeigneten Quellsystemattributs während des XML-Imports erzeugt.

7.1.3.1.3. Funktionskennzahlen

Die folgende Abbildung veranschaulicht die Berechnung der Funktionskennzahlen **Bearbeitungszeit** (FBZ), **Bearbeitungsspanne** (FBZSpan), **Durchlaufzeit** (FDLZ), **Durchlaufzeitspanne** (FDLZSpan) und **Liegezeit** (FLZ):



Die Berechnungsvorschriften gelten auch für Beziehungen zwischen verzweigenden Regeln. Für die Betrachtung der Kennzahlen **Durchlaufzeit** und **Liegezeit** von Funktion 2 wird angenommen, dass Start- und Endzeitpunkt von Funktion 3 nach Start- und Endzeitpunkt von Funktion 4 liegen.

Das berechnete Kennzahlattribut wird an die Funktion geschrieben, an der der Pfeil in der Abbildung endet. Negative Zeitdifferenzen werden als Kennzahlwert **0** zurückgeliefert.

Bei der Berechnung der Kennzahlen **Bearbeitungsspanne** und **Durchlaufzeitspanne** werden alle Funktionen mit demselben Namen berücksichtigt (Funktionsattribut **AT_OBJNAME**), auch wenn diese in unabhängigen Prozessketten innerhalb einer Prozessinstanz vorkommen.

Warnung
Können aus dem Quellsystem nur Endzeitpunkte ausgelesen werden, lassen sich nur die Kennzahlen **Durchlaufzeit** und **Durchlaufspanne** bestimmen.

FBZ

Name	Funktionsbearbeitungszeit
Typ	Zeitspanne
Quellattribute	AT_START_TIME
	AT_END_TIME
Ergebnis	Differenz zwischen Endzeitpunkt und Startzeitpunkt einer Funktionsinstanz

FBZWK

Wie **FBZ** unter Berücksichtigung des Werkskalenders

FBZSpan

Name	Funktionsbearbeitungsspanne
Typ	Zeitspanne
Quellattribute	AT_START_TIME
	AT_END_TIME
Ergebnis	Differenz zwischen spätestem Endzeitpunkt und frühestem Startzeitpunkt aller Funktionsinstanzen mit demselben Namen
Hinweis	Das Ergebnis wird an jeder der Funktionsinstanzen mit demselben Namen gespeichert. Kommt eine Funktion nur einmal in einer Prozessinstanz vor, ist **FBZSpan** gleich **FBZ**.

FBZSpanWK

Wie **FBZSpan** unter Berücksichtigung des Werkskalenders

FDLZ

Name	Funktionsdurchlaufzeit
Typ	Zeitspanne
Quellattribute	AT_END_TIME
Ergebnis	Differenz zwischen dem Endzeitpunkt einer Funktionsinstanz und dem spätesten Endzeitpunkt ihrer vorangehenden Funktionsinstanzen
Hinweis	Bei zusammenführenden Regeln werden die Endzeitpunkte aller vorangehenden Funktionsinstanzen berücksichtigt.

FDLZWK

Wie **FDLZ** unter Berücksichtigung des Werkskalenders

FDLZWKByParam

Wie **FDLZWK** unter Verwendung beliebiger, selbstdefinierter Werkskalender. Die entsprechenden Konfigurationseinstellungen werden in Form von Parametern beim Aufruf der Klasse übergeben.

> **Warnung**
> Beachten Sie, dass die Bezeichner (**key**) der jeweiligen Parameter groß geschrieben werden.

```
<calcattr name="AT_KI_FDLZWKByParam" type="OT_FUNC">
<calcclass name="com.idsscheer.ppm.server.keyindicator.
attributecalculator.ZAttributeCalculatorFDLZWKByParam">
<calcparam key="FC_ATTRIBUTENAME" value="AT_FC_NAME"/>
<calcparam key="FC_DIRECTORY" value="calculations\fc"\>
</calcclass>
</calcattr>
```

Mit Hilfe des Parameters **FC_ATTRIBUTENAME** wird der Attributtyp **AT_FC_NAME** angegeben, der den Namen der zu verwendenden Werkskalenderdatei enthält. Der Attributtyp muss für die jeweilige Funktion bzw. Prozessinstanz, für die die Berechnung ausgeführt wird, angegeben sein und in den zu importierenden Datendateien in der Form **<attribute type="AT_FC_NAME">ExampleFactoryCalendar.xml</attribute>** vorkommen.

Das entsprechende Verzeichnis mit der zu verwendenden Werkskalenderdatei geben Sie relativ zum PPM-Datenverzeichnis im Parameter FC_DIRECTORY an. Das PPM-Datenverzeichnis data_ppm liegt unter <PPM-Installationsverzeichnis>\ppm\server\bin\work\.

Beispiel 23. *[en-us]*

PPM-Datenverzeichnis: C:\ARIS\ppm\server\bin\work\data_ppm

- absolute Pfadangabe zum Verzeichnis mit Werkskalenderdatei: C:\ARIS\ppm\server\bin\work\data_ppm\calculations\fc
- relative Pfadangabe zum Verzeichnis mit Werkskalenderdatei: calculations\fc.

Die beiden Parameter FC_ATTRIBUTENAME und FC_DIRECTORY müssen immer zusammen angegeben werden.

Die Berechnung der Funktionsdurchlaufzeit **FDLZWKByParam** erfolgt dabei analog zur standardmäßigen Berechnung der Zeitenkennzahl **FDLZWK** mit dem Unterschied, dass zur Berechnung nicht der standardmäßig in PPM importierte Werkskalender sondern der über die angegebenen Parameter spezifizierte Werkskalender verwendet wird.

Konfigurieren Sie die Berechnungen der Zeitenkennzahlen **FBZWKByParam**, **FLZWKByParam** und **PDLZWKByParam** in analoger Weise, wenn Sie zur Berechnung der Standardkennzahlen **FBZWK**, **FLZWK** bzw. **PDLZWK** selbstdefinierte Werkskalenderdateien verwenden möchten.

FDLZSpan

Name	Funktionsdurchlaufzeitspanne
Typ	Zeitspanne

Name	Funktionsdurchlaufzeitspanne
Quellattribute	AT_END_TIME
Ergebnis	Differenz zwischen dem spätesten Endzeitpunkt aller Funktionsinstanzen mit demselben Namen und dem frühesten Endzeitpunkt aller diesen Funktionsinstanzen vorangehenden Funktionsinstanzen
Hinweis	Das Ergebnis wird an jeder der Funktionsinstanzen mit demselben Namen gespeichert. Kommt eine Funktion nur einmal in einer Prozessinstanz vor, ist **FDLZSpan** gleich **0**.

FDLZSpanWK

Wie **FDLZSpan** unter Berücksichtigung des Werkskalenders

FLZ

Name	Funktionsliegezeit
Typ	Zeitspanne
Quellattribute	AT_START_TIME AT_END_TIME
Ergebnis	Differenz zwischen dem Startzeitpunkt einer Funktionsinstanz und dem spätesten Endzeitpunkt ihrer vorangehenden Funktionsinstanzen
Hinweis	Bei zusammenführenden Regeln werden die Endzeitpunkte aller vorangehenden Funktionsinstanzen berücksichtigt.

FLZWK

Wie **FLZ** unter Berücksichtigung des Werkskalenders

7.1.3.1.4. Prozesskennzahlen

PDLZ

Name	Prozesslaufzeit
Typ	Zeitspanne
Quellattribute	AT_START_TIME AT_END_TIME
Ergebnis	Differenz zwischen dem spätesten und dem frühesten Zeitpunkt der End- und Startzeitpunkte aller Funktionsinstanzen der Prozessinstanz
Hinweis	Durch diese Art der Berechnung können auch Prozesslaufzeiten berechnet werden, wenn an Funktionsinstanzen nur AT_END_TIME gepflegt ist, z.B. beim Auslesen aus SAP-Systemen.

PDLZWK

Wie **PDLZ** unter Berücksichtigung des Werkskalenders

PDLZWKByParam

Wie PDLZWK unter Berücksichtigung selbstdefinierter Werkskalender. Definition analog zu **FDLZWKByParam**.

7.1.3.1.5. Funktionskennzahlen

FEDFREQ

Name	Anzahl Bearbeitungen
Typ	Integer
Quellattribute	AT_COUNT_PROCESSINGS
Ergebnis	Wert des Attributes AT_COUNT_PROCESSINGS. Ist das Attribut an der Funktionsinstanz nicht gepflegt, wird die Summe aller Attributwerte AT_COUNT_PROCESSINGS der Kanten zwischen der Funktionsinstanz und den zugeordneten Organisationseinheiten zurückgegeben.
Hinweis	Quellattributwerte kleiner Null werden als 0 zurückgegeben.

FEDFREQSpan

Name	Anzahl Bearbeitungen (Spanne)
Typ	Integer
Quellattribute	AT_COUNT_PROCESSINGS
Ergebnis	Summe der Werte des Attributs AT_COUNT_PROCESSINGS aller Funktionsinstanzen mit demselben Namen. Ist das Attribut an einer Funktionsinstanz nicht gepflegt, wird anstelle des Attributwertes die Summe aller Attributwerte AT_COUNT_PROCESSINGS der Kanten zwischen der Funktionsinstanz und den zugeordneten Organisationseinheiten verwendet.
Hinweis	Quellattributwerte kleiner Null werden als 0 zurückgegeben.

FFREQ

Name	Funktionshäufigkeit
Typ	Integer
Quellattribute	-
Ergebnis	Wird mit dem Wert 1 belegt.
Hinweis	Bei der Kennzahlermittlung werden die Werte summiert und durch die Anzahl von Tagen dividiert, die sich für die gewählte Skalierung einer Zeitdimension ergibt, z.B. (1+1)/365 Tage bei Verwendung einer Zeitdimension mit der Schrittweite jährlich.

FNUM

Name	Funktionsanzahl
Typ	Integer
Quellattribute	-
Ergebnis	Wird mit dem Wert 1 belegt.
Hinweis	Bei der Kennzahlermittlung werden die Werte summiert und durch die Anzahl von Tagen dividiert, die sich für die gewählte Skalierung einer Zeitdimension ergibt, z.B. (1+1)/365 Tage bei Verwendung einer Zeitdimension mit der Schrittweite jährlich.

FOEFREQ

Name	Anzahl unterschiedlicher Bearbeiter
Typ	Integer
Quellattribute	-
Ergebnis	Anzahl der unterschiedlichen Organisationseinheiten, die eine Funktionsinstanz bearbeiten. Zur Unterscheidung wird das Attribut AT_OBJNAME der Organisationseinheiten verwendet.
Hinweis	Sind an einer Funktionsinstanz keine Organisationseinheiten gepflegt, wird der Wert 1 zurückgegeben.

FOEFREQb

Wie **FOEFREQ** mit dem Unterschied, dass der Wert 0 zurückgegeben wird, wenn an einer Funktionsinstanz keine Organisationseinheiten gepflegt sind.

7.1.3.1.6. Prozesskennzahlen

PEDFREQ

Name	Anzahl Bearbeitungen
Typ	Integer
Quellattribute	AT_COUNT_PROCESSINGS aller Funktionsinstanzen der Prozessinstanz
Ergebnis	Summe der Werte des Attributes **AT_COUNT_PROCESSINGS** aller Funktionsinstanzen der Prozessinstanz. Kann die Summe nicht ermittelt werden, ist das Ergebnis **1**.

PFREQ

Name	Prozesshäufigkeit
Typ	Integer
Quellattribute	-
Ergebnis	Wird mit dem Wert **1** belegt.

Name	Prozesshäufigkeit
Hinweis	Bei der Kennzahlermittlung werden die Werte summiert und durch die Anzahl von Tagen dividiert, die sich für die gewählte Skalierung einer Zeitdimension ergibt, z.B. (1+1)/365 Tage bei Verwendung einer Zeitdimension mit der Schrittweite **jährlich**.

PINT, PINTb, PINTc

Sie können für Funktionsinstanzen, denen kein Bearbeiter zugeordnet ist, festlegen, wie diese bei der Berechnung der Kennzahl **Anzahl Bearbeiter** behandelt werden. Geben Sie zur Berechnung des Attributs **AT_KI_PINT** in der Kennzahldefinition eine der folgenden Berechnungsklassen an:

<Klassenname>	Beschreibung
ZAttributeCalculatorPINT	Jede Funktionsinstanz ohne zugeordneten Bearbeiter wird wie eine Funktionsinstanz mit einem neuen, jeweils verschiedenen Bearbeiter behandelt (Voreinstellung).
ZAttributeCalculatorPINTb	Jede Funktionsinstanz ohne zugeordneten Bearbeiter wird bei der Berechnung der Kennzahl nicht berücksichtigt.
ZAttributeCalculatorPINTc	Alle Funktionsinstanzen ohne zugeordneten Bearbeiter werden für die gesamte Prozessinstanz insgesamt nur einmal berücksichtigt, d. h., es wird derselbe Bearbeiter angenommen.

PNUM

Name	Prozessanzahl
Typ	Integer
Quellattribute	-
Ergebnis	Wird mit dem Wert **1** belegt.
Hinweis	Bei einer verdichteten EPK gibt die Prozessanzahl die Anzahl der verdichteten Prozessinstanzen an. EPKn, deren Kennzahl **Prozessanzahl** größer 1 ist, sind EPKn verdichteter Prozessinstanzen.

7.1.3.1.7. Prozesskostensätze

Die Berechnungsklassen **PKSS** und **PKSR** berechnen Prozesskosten für Funktionsinstanzen. Die Prozesskosten für Prozessinstanzen werden durch eine Berechnungsvorschrift des Attributberechners ermittelt.

PKSS

Name	Prozesskosten auf Basis des Leistungsstandards
Typ	Kosten
Quellattribute	AT_COSTRATE (Organisationseinheiten) AT_LS (Funktion) AT_COUNT_PROCESSINGS (Kanten zwischen Organisationseinheiten und Funktionen)
Ergebnis	Durchschnittliche Prozesskosten für die einmalige Ausführung einer Funktion (siehe Kapitel Definition von Prozesskostenkennzahlen [134])
Hinweis	Der Leistungsstandard (Funktionsattribut **AT_LS**) muss an der Funktionsinstanz gepflegt sein.

PKSR

Name	Prozesskosten auf Basis der Bearbeitungszeit
Typ	Kosten

Name	Prozesskosten auf Basis der Bearbeitungszeit
Quellattribute	AT_COSTRATE (Organisationseinheiten) AT_KI-FBZ (Funktion) AT_COUNT_PROCESSINGS (Kanten zwischen Organisationseinheiten und Funktionen)
Ergebnis	Durchschnittliche Prozesskosten für die einmalige Ausführung einer Funktion (siehe Kapitel Definition von Prozesskostenkennzahlen [134])
Hinweis	Die Bearbeitungszeit (Funktionsattribut **AT_KI_FBZ**) wird automatisch berechnet, Start- und Endzeitpunkte (Funktionsattribute **AT_START_TIME** und **AT_END_TIME**) müssen an der Funktionsinstanz gepflegt sein.

7.1.3.1.8. Weitere Prozesskennzahlen

IsGraphConnected

Name	Verbundene EPK
Typ	Wahrheitswert
Quellattribute	-
Ergebnis	Liefert **TRUE**, wenn alle Objekte der EPK untereinander durch eine Kante verbunden sind.
Hinweis	-

Originator

Name	Organisationseinheit
Typ	Text
Quellattribute	AT_OBJ_NAME der Organisationseinheiten
Ergebnis	Liefert den Namen der an der Funktionsinstanz gepflegten Organisationseinheiten, die denselben Namen haben (Attribut **AT_OBJNAME** einer Organisationseinheit).
Hinweis	Wenn an einer Funktionsinstanz keine Organisationseinheit gepflegt ist oder wenn Organisationseinheiten mit verschiedenen Namen gepflegt sind, ist das Ergebnis eine Zeichenkette der Länge 0.

7.1.3.1.9. Umgebungsrelevante Berechnungen

ZAttributeCalculatorTransformUniversalMappingByParam

Diese Berechnungsklasse ermittelt den Wert eines Attributes an Hand einer Mapping-Datei aus dem Wert eines anderen Attributes. Die Attributwerttransformation kann sowohl auf Prozess- als auch auf Objektattribute angewendet werden.

Sie können das Verhalten der Berechnungsklasse durch die Angabe der folgenden Parameter steuern:

Parameter	Beschreibung	Beispielwert
attrname	Quellsystemattribute, deren Werte umgesetzt werden	AT_PLZ
mappingfile	Datei, die die Mapping-Informationen enthält (Schlüssel-Wert-Paare) Das Verzeichnis mit der Mapping-Datei geben Sie relativ zum Verzeichnis data_ppm\bin an. Das Verzeichnis liegt unter <PPM-Installationsverzeichnis>\ppm\server\bin\work\.	..\custom\<ppmclient>\xml\attrtrans\PLZ_Ort.mappings

Parameter	Beschreibung	Beispielwert
defaultcopy	Bestimmt das Verhalten, wenn kein Mapping zum Attributwert gefunden wird. Zulässige Werte: **TRUE**, **FALSE** **TRUE**: Der Wert des Quellattributs wird unverändert in das Zielattribut geschrieben. **FALSE**: Als Wert wird in das Zielattribut **nicht gepflegt** geschrieben.	FALSE

Beachten Sie, dass ein in der Berechnungsvorschrift eventuell angegebener Standardwert (XML-Element **defaultvalue**) unabhängig vom Parameter **defaultcopy** verwendet wird. Ist ein Standardwert angegeben, hat dieser Vorrang.

Beispiel 24.

Im folgenden Beispiel werden die Werte des Prozessattributes **AT_PLZ** auf das Prozessattribut **AT_ORT** umgesetzt. Die Umsetzung ist in der Datei **PLZ_Ort.mappings** angegeben. Wenn für eine Postleitzahl kein Ort gefunden wird, erhält das Attribut **AT_ORT** den Wert **nicht gepflegt**.

Auszug aus der Kennzahlenkonfiguration:

```
...
<calcattr name="AT_ORT" type="PROCESS">
<defaultvalue>Not specified</defaultvalue>
<calcclass name="com.idsscheer.ppm.server.
keyindicator.attributecalculator
ZAttributeCalculatorTransformUniversalMappingByParam">
<calcparam key="attrname" value="AT_PLZ"/>
<calcparam key="mappingfile" value="PLZ_Ort.mappings"/>
</calcclass>
</calcattr>
...
```

Beispiel 25.

Inhalt der Mapping-Datei **PLZ_Ort.mappings**:

```
66115 = Saarbruecken
10117 = Berlin Center
14612 = Falkensee
```

Achten Sie darauf, dass Ihre verwendeten Mapping-Dateien reguläre Java-Property-Dateien sind. Wenn diese Umlaute oder andere Sonderzeichen enthalten, müssen sie mittels **native2ascii** umgewandelt werden. Näheres zum Konvertieren von Dateien mit nativen Inhalten in ASCII-Dateien entnehmen Sie den Hilfeseiten der Webseite von Sun Microsystems, Inc..

ObjectCounterByEpcEnv

Die Berechnungsklasse **ObjectCounterByEpcEnv** berechnet die Anzahl der der aktuellen Funktion vorangehenden bzw. nachfolgenden Funktionen, wobei Ereignisse und Konnektoren nicht berücksichtigt werden. Die aktuelle Funktion ist die Funktion, für die die Berechnungsvorschrift ausgeführt wird.

Sie können das Verhalten der Berechnungsklasse durch die Angabe der folgenden Parameter steuern:

Parameter	Beschreibung	Beispielwert
DIRECTION (einwertig)	Richtung der Suche nach Vorgänger- bzw. Nachfolgerfunktionen von der aktuellen Funktion aus	**FORWARD** (Nachfolgerfunktionen) oder **BACKWARD** (Vorgängerfunktionen) in Bezug auf die im zugehörigen **calcattr**-Tag referenzierte(n) Funktion(en)
ENVTYPE (einwertig)	Suche nach Funktionen in der direkten Nachbarschaft der Funktion oder in der gesamten Prozessinstanz	**DIRECT** (nur unmittelbar benachbarte Funktionen) **INDIRECT** (alle Funktionen in der angegebenen Suchrichtung)
OBJECTNAMEFILTER (optional, mehrwertig)	Limits the search to particular internal function names, with use of the placeholders ***** and **?**. Mehrere Namensmuster werden mittels **key** wie folgt angegeben: **OBJECTNAMEFILTER.0** **OBJECTNAMEFILTER.1** **OBJECTNAMEFILTER.2**, etc.	***AUFT***,**????AUFT*** **FCT_AUFT1**, **FCT_AUFT_2** ***AUFT??**

Warnung
Beachten Sie, dass die Bezeichner (**key**) der jeweiligen Parameter groß geschrieben werden.

Beispiel 26.

Über die parametrisierte Suche werden alle direkt benachbarten Nachfolgerfunktionen der aktuellen Funktion ermittelt. Die Menge der ermittelten Funktionen wird über die angegebenen Filterausdrücke **FCT_CREATE_*** und **FCT_ORDER_*** weiter eingeschränkt. Für die Menge der so ermittelten Funktionen wird die Objektanzahl berechnet und im berechneten Attributtyp **AT_KI_COUNTFUNC_ENV** gespeichert.

```
...
<calcattr name="AT_KI_COUNTFUNC_ENV" type="OT_FUNC"
objectname="FCT_CREATE_ORD">
<calcclass name="com.idsscheer.ppm.server.
keyindicator.attributecalculator.
ZAttributeCalculatorObjectCounterByEpcEnv">
<calcparam key="DIRECTION" value="FORWARD"/>
<calcparam key="ENVTYPE" value="DIRECT"/>
<calcparam key="OBJECTNAMEFILTER.0"
value="FCT_CREATE_*"/>
<calcparam key="OBJECTNAMEFILTER.1"
value="FCT_ORDER_*">
</calcclass>
</calcattr>
...
```

AttributeCopierByEpcEnv

Verwenden Sie diese Berechnungsklasse, um Abhängigkeiten zwischen mehreren Attributberechnungen (**<depends attrname="..." type="..."/>**) zu definieren. Beispielsweise kann es sinnvoll sein, eine bestimmte Attributberechnung erst dann auszuführen, wenn zuvor bestimmte Attributtypwerte mit der Berechnungsklasse **AttributeCopierByEpcEnv** an ausgewählte Funktionen kopiert worden sind.

In der eigentlichen Berechnung werden über eine parametrisierbare Suche ausgehend von der aktuellen Funktion bestimmte benachbarte Funktionen ermittelt.

Anschließend werden gemäß der angegebenen Parameter bestimmte Attributtypwerte von einer Funktion an eine oder mehrere Funktionen kopiert. Dabei werden zwei Hauptfälle unterschieden:

1. Fall

Die aktuelle Funktion soll das Ziel des Kopiervorgangs sein **<calcparam key="COPYROLE" value="DESTINATION"/>**).

Werden mehrere benachbarte Funktionen ermittelt, muss aus der Menge der ermittelten Funktionen eine Funktion bestimmt werden, die als Quelle des Kopiervorgangs verwendet werden soll. Der Parameter **COPYTYPE** muss in diesem Fall den Wert **1-TO-1** haben. Sie sollten die Menge der Funktionen sortieren. Andernfalls wird der Wert einer zufälligen Funktion kopiert oder es wird kein Wert kopiert, wenn das Attribut für die Funktion nicht gepflegt ist.

2. Fall

Die aktuelle Funktion soll die Quelle des Kopiervorgangs sein (**<calcparam key="COPYROLE" value="SOURCE"/>**).

Die angegebenen Attributtypwerte sollen von der Ausgangsfunktion an eine bestimmte Funktion kopiert werden. Der Parameter **COPYTYPE** muss den Wert **1-TO-1** haben, da über die Suche mehrere benachbarte Funktionen ermittelt werden können. Über die Parameter **SORTATTRIBUTE** und **SORTTYPE** wird aus der ermittelten Menge eine Zielfunktion bestimmt.

Sollen die angegebenen Attributtypwerte von der Ausgangsfunktion an alle benachbarten Funktionen kopiert werden, muss der Parameter **COPYTYPE** den Wert **1-TO-N** haben. In diesem Fall ist keine Angabe der Sortierungsparameter **SORTATTRIBUTE** und **SORTTYPE** notwendig, da aus der Menge der ermittelten Funktionen keine näher bestimmt werden muss.

Ermittlung der Kopierquelle

Als Quelle bzw. Ziel des Kopiervorgangs wird mit Hilfe der Parameter **SORTATTRIBUTE** und **SORTTYPE** die Funktion bestimmt, die gemäß angegebener Sortierrichtung an erster Position steht.

Zu kopierende Attributtypwerte

Die Liste der zu kopierenden Attributtypwerte wird in fortlaufender Nummerierung mit Hilfe der Parameter **SOURCEATTRIBUTE.<x>** bzw. **DESTINATIONATTRIBUTE.<x>** angegeben, wobei **<x>** für eine ganze Zahl steht.Ist kein Zielattributtyp angegeben, wird der Quellattributtyp als Zielattributtyp mit entsprechenden Werten an den angegebenen Funktionen erzeugt.

Sie können das Verhalten der Berechnungsklasse **AttributeCopierByEpcEnv** durch Angabe folgender Parameter steuern:

Parameter	Beschreibung	Beispielwert
DIRECTION (einwertig)	Richtung der Suche nach Vorgänger- bzw. Nachfolgerfunktionen von der aktuellen Funktion aus	**FORWARD** (Nachfolgerfunktionen) oder **BACKWARD** (Vorgängerfunktionen) in Bezug auf die im zugehörigen **calcattr**-Tag referenzierte(n) Funktion(en)

Parameter	Beschreibung	Beispielwert
ENVTYPE (einwertig)	Suche nach Funktionen in der direkten Nachbarschaft der Funktion oder in der gesamten Prozessinstanz in der angegebenen Suchrichtung. Alle ermittelten Funktionen werden einer unsortierten Menge hinzugefügt.	**DIRECT** (nur unmittelbar benachbarte Funktionen) **INDIRECT** (alle Funktionen in der angegebenen Suchrichtung)
OBJECTNAMEFILTER (optional, mehrwertig)	Limits the search to particular internal function names, with use of the placeholders ***** and **?**. Mehrere Namensmuster werden mittels **key** wie folgt angegeben: **OBJECTNAMEFILTER.0** **OBJECTNAMEFILTER.1** **OBJECTNAMEFILTER.2**, etc.	***AUFT*,????AUFT*** **FCT_AUFT1**, **FCT_AUFT_2** ***AUFT??**
COPYROLE (einwertig)	Rolle der aktuellen Funktion beim Kopierprozess. Die über die Suche ermittelten, benachbarten Funktionen nehmen jeweils die gegenteilige Rolle ein.	**SOURCE** (Kopierquelle) **DESTINATION** (Kopierziel)
COPYTYPE (einwertig)	Wenn mehrere Funktionen als Kopierziel ermittelt werden, gibt dieser Parameter an, ob der Wert an alle oder nur an eine Funktion kopiert werden soll.	**1-TO-1** (Wert wird an eine Funktion kopiert) **1-TO-N** (Wert wird an alle ermittelten Funktionen kopiert)
SORTATTRIBUTE (optional, einwertig)	Attribut, das als Sortierkriterium für mehrere ermittelte Funktionen verwendet wird. Wird nur berücksichtigt, wenn COPYTYPE 1-TO-1 angegeben ist.	Existierender Attributtyp PPM
SORTTYPE (optional, einwertig)	Sortierrichtung für das gewählte Sortierkriterium. Wird nur berücksichtigt, wenn COPYTYPE 1-TO-1 angegeben ist.	**ASC**(Vorgabewert: aufsteigend) **DESC** (absteigend)
SORTNULLVALUES (optional, einwertig)	Bestimmt, ob Objekte, deren Sortierattribut nicht gepflegt ist, mit minimalem oder maximalen Wert angeordnet werden oder ob eine Fehlermeldung als Warnung ausgegeben werden soll	**MIN**, **MAX**, **DEFAULT** (Vorgabewert)
SOURCEATTRIBUTE.<x> (mehrwertig)	Der Quellattributtyp, dessen Wert kopiert werden soll. <x> ist eine Ganzzahl und entspricht destinationattribute.<x>.	Ein vorhandener Attributtyp PPM
DESTINATION (optional, mehrwertig)	Der Zielattributtyp, an den der Wert kopiert werden soll. <x> ist eine Ganzzahl und entspricht sourceattribute.<x>.	Existierender Attributtyp PPM

Beispiel 27.

Über die parametrisierte Suche werden alle direkt benachbarten Vorgängerfunktionen der aktuellen Funktion, die als Quelle des Kopiervorgangs dienen soll, ermittelt. Werden mehrere Vorgängerfunktionen ermittelt, wird die Menge der ermittelten Funktionen mit Hilfe des Sortierkriteriums **AT_START_TIME** absteigend sortiert. Der Wert des Attributtyps **AT_START_TIME** wird von der spätesten Funktion an den Attributtyp **AT_END_TIME** an der ermittelten Zielfunktion kopiert.

Auszug aus der Kennzahlenkonfiguration

```
...
<calcattr name="AT_END_TIME" type="OT_FUNC" objectname="...">
<calcclass name="com.idsscheer.ppm.server.
keyindicator.attributecalculator.
ZAttributeCalculatorAttributeCopierByEpcEnv">
```

```
<calcparam key="DIRECTION" value="BACKWARD"/>
<calcparam key="ENVTYPE" value="DIRECT"/>
<calcparam key="COPYROLE" value="SOURCE"/>
<calcparam key="COPYTYPE" value="1-TO-1"/>
<calcparam key="SORTATTRIBUTE" value="AT_START_TIME"/>
<calcparam key="SORTTYPE" value="DESC"/>
<calcparam key="SOURCEATTRIBUTE.0"
value="AT_START_TIME"/>
<calcparam key="DESTINATIONATTRIBUTE.0"
value="AT_END_TIME"/>
</calcclass>
</calcattr>
...
```

AttributeAggregatorByEpcEnv

Die Berechnungsklasse **AttributeAggregatorByEpcEnv** summiert den angegebenen numerischen Attributtyp aller benachbarter Funktionen auf. Das Ergebnis wird im angegebenen Attributtyp gespeichert.

Sie können das Verhalten der Berechnungsklasse **AttributeAggregatorByEpcEnv** durch Angabe folgender Parameter steuern:

Parameter	Beschreibung	Beispielwert
DIRECTION (einwertig)	Richtung der Suche nach Vorgänger- bzw. Nachfolgerfunktionen von der aktuellen Funktion aus	**FORWARD** (Nachfolgerfunktionen) oder **BACKWARD** (Vorgängerfunktionen) in Bezug auf die im zugehörigen **calcattr**-Tag referenzierte(n) Funktion(en)
ENVTYPE (einwertig)	Suche nach Funktionen in der direkten Nachbarschaft der Funktion oder in der gesamten Prozessinstanz in der angegebenen Suchrichtung	**DIRECT** (nur unmittelbar benachbarte Funktionen) **INDIRECT** (alle Funktionen in der angegebenen Suchrichtung)
OBJECTNAMEFILTER (optional, mehrwertig)	Limits the search to particular internal function names, with use of the placeholders ***** and **?**. Mehrere Namensmuster werden mittels **key** wie folgt angegeben: **OBJECTNAMEFILTER.0** **OBJECTNAMEFILTER.1** **OBJECTNAMEFILTER.2**, etc.	***AUFT*,????AUFT*** **FCT_AUFT1**, **FCT_AUFT_2** ***AUFT??**
AGGREGATION_ ATTRIBUTE (einwertig)	Attributtyp, der aufsummiert werden soll	Existierender PPM-Attributtyp

Beispiel 28.

Im folgenden Beispiel werden alle Nachfolgerfunktionen der aktuellen Funktion in der Prozessinstanz ermittelt, deren interner Name mit der Zeichenfolge **FCT_ORDER_** beginnt oder der mit **FCT_INVOICING_** beginnt und von vier beliebigen Zeichen gefolgt wird. Für die Menge der ermittelten Funktionen wird der Attributtyp **AT_ORDER_VOL** aufsummiert und im Attributtyp **AT_KI_ORDER_VOL_AGG** gespeichert.

Auszug aus der Kennzahlenkonfiguration

```
...
<calcattr name="AT_KI_ORDER_VOL_AGG" type="OT_FUNC"
objectname="ORDER*">
```

```
<calcclass name="com.idsscheer.ppm.server.
keyindicator.attributecalculator.
ZAttributeCalculatorAttributeAggregatorByEpcEnv">
<calcparam key="DIRECTION" value="FORWARD"/>
<calcparam key="ENVTYPE" value="INDIRECT"/>
<calcparam key="OBJECTNAMEFILTER.0"
value="FCT_ORDER_*"/>
<calcparam key="OBJECTNAMEFILTER.1"
value="FCT_INVOICING_????"/>
<calcparam key="AGGREGATION_ATTRIBUTE"
value="AT_ORDER_VOL"/>
</calcclass>
</calcattr>
...
```

7.1.3.1.10. Beziehungskennzahlen

OrgCopyAttrFromFunc

Die Berechnungsklasse **OrgCopyAttrFromFunc** kopiert das angegebene Funktionsattribut an die jeweils ausführende Organisationseinheit. Die Berechnungsklasse ist nur im Rahmen der **Interaktionsanalyse** verfügbar. Das Ergebnis wird im angegebenen Attributtyp an der jeweiligen Organisationseinheit gespeichert.

Sie müssen für Berechnungen mit der Klasse **OrgCopyAttrFromFunc** den folgenden Parameter angeben.

Parameter	Beschreibung	Wert bzw. Beispiel
attrname	Bezeichner des zu kopierenden Funktionsattributs	**AT_END_TIME**

Warnung
Es kann immer nur ein Funktionsattribut pro XML-Element **calcattr** kopiert werden. Achten Sie darauf, dass die Datentypen von Quell- und Zielattribut kompatibel sind. Grundsätzlich sind alle (auch benutzerdefinierten) numerischen Datentypen (siehe Kapitel zu Datentypen [16]: LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE) are compatible with one another. Mit dem Operator **convert** können Sie geeignete, vorgeschaltete Datentypkonvertierungen vornehmen (Kapitel Logische Operatoren [98]). Beachten Sie dabei, dass die Umwandlung immer in der Basisskalierung des Zieldatentyps erfolgt.

Beispiel 29.

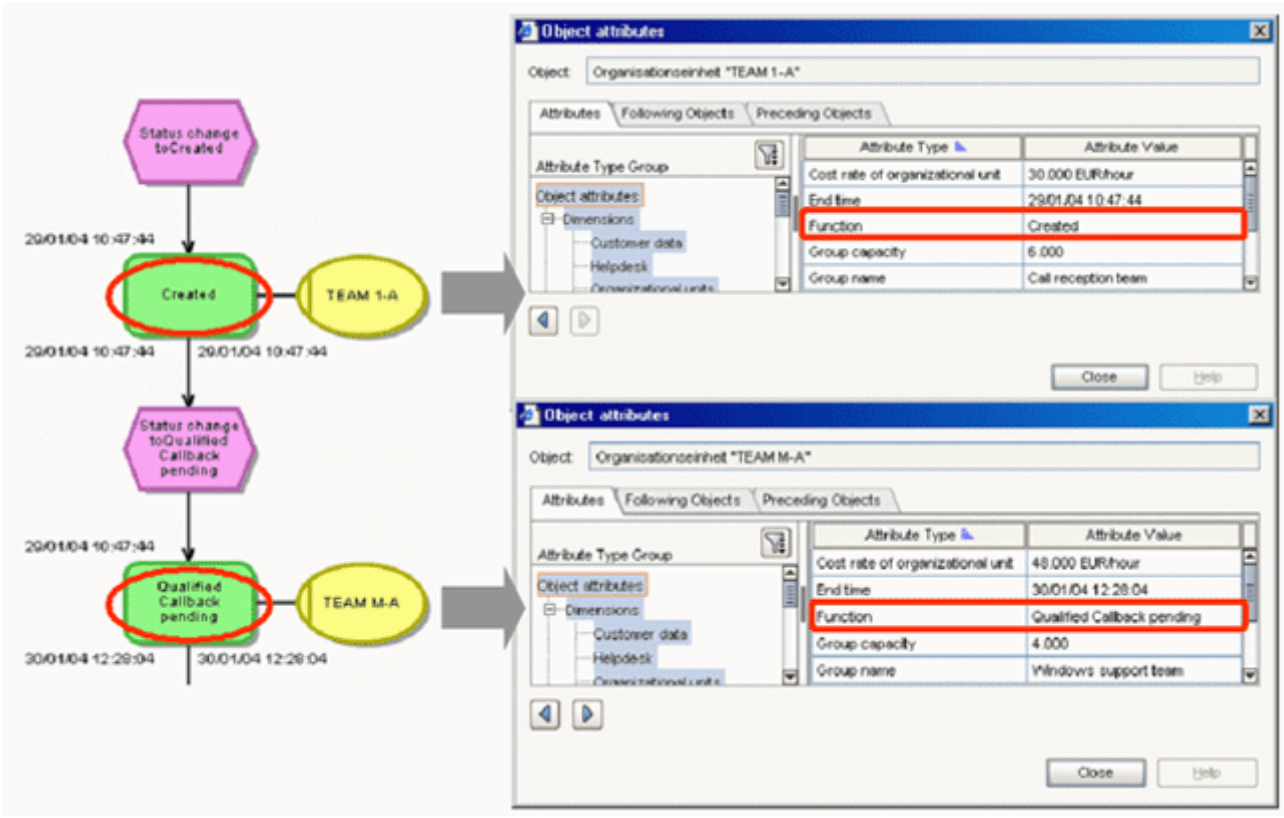
Im folgenden Beispiel aus der Kennzahlenkonfiguration wird das Funktionsattribut **AT_OBJNAME** jeweils als Attribut **AT_FUNC** an die ausführenden Organisationseinheiten (**type="OT_ORG"**) kopiert. Quell- und Zielattribut des Kopiervorganges sind jeweils vom Datentyp **TEXT**.

```
<calcattr name="AT_FUNC" type="OT_ORG">
<calcclass
name="com.idsscheer.ppm.server.keyindicator.attributecalculator.ZAttributeCalcul
atorOrgCopyAttrFromFunc">
<calcparam key="attrname" value="AT_OBJNAME"/>
</calcclass>
</calcattr>
```

Wenn das Attribut **AT_FUNC** bereits an den Organisationseinheiten gepflegt sein sollte, können Sie mittels **delete="yes"** in der **calcattr**-Anweisung bestimmen, dass der Attributwert vor dem Kopiervorgang zuerst gelöscht wird.

Beispiel 30.

In der folgenden Beispielgrafik aus dem Modul **Interaktionsanalyse** ist das Ergebnis des Attributkopiervorgangs in den geöffneten Objektattributdialogen der beiden Organisationseinheiten **TEAM 1-A** und **TEAM M-A** zu sehen. An jede Organisationseinheit wurde der jeweilige Wert des Funktionsattributes **AT_OBJNAME** als Attribut **Funktion** (**AT_FUNC**) der Funktion kopiert, die von der entsprechenden Organisationseinheit ausgeführt wurde.



7.1.3.1.11. Prozesskonformität

Ab Version 10.2 gibt es in PPM eine Prozesskonformitätsprüfung für Prozesse, die in ARIS modelliert wurden und in PPM importiert werden sollen.

In PPM gibt es ein spezielles Konformitätskonfigurationspaket, das alle Customizing-Elemente enthält, die für die Berechnung der Prozesskonformität nötig sind. Unter anderem enthält dieses Paket die Konformitätskennzahl **Konformitätsquote** (KI_CONFORMANCE_RATE [70]), die an den Prozessinstanzen berechnet wird und eine zusätzliche Relation **Konformitätsfehler** (REL_CONFORMANCE_ISSUE [70]), die detaillierte Informationen liefert, warum Prozessinstanzen als konform oder nicht konform eingestuft wurden.

Details zur Prozesskonformitätsprüfung finden Sie im Kapitel **ARIS-Prozesskonformitätsprüfung** der PPM Customizing Toolkit-Dokumentation.

7.1.3.1.12. Kennzahl Konformitätsquote

Das Customizing-Paket zur Konformitätskonfiguration enthält die Kennzahl **KI_CONFORMANCE_RATE** mit dem Quellattribut **AT_KI_CONFORMANCE_RATE**. Das Attribut wird von der Berechnungsklasse **ZAttributeCalculatorConformanceRate** berechnet.

In PPM CTK wird die Berechnungsklasse des Attributs **Konformitätsquote** genannt.

Die XML-Struktur von **AT_KI_CONFORMANCE_RATE** sieht folgendermaßen aus :

```
<calcattr name="AT_KI_CONFORMANCE_RATE" type="PROCESS" delete="no">

  <calcclass
name="com.idsscheer.ppm.server.keyindicator.attributecalculator.ZAttributeCalcul
atorConformanceRate" loglevel="VERBOSE" />

  </calcattr>
```

Das Ergebnis der Berechnung ist ein Wert von 0.0 (nicht konform) oder 1.0 (konform). Die Kennzahl wird nach Mittelwert aggregiert und zeigt das Verhältnis der konformen Prozesse zu allen Prozessen an.

7.1.3.1.13. Relation Konformitätsfehler

Die Relation **Konformitätsfehler** besteht aus einer Quellobjektdimension **Vorgängerfunktion**, einer Zielojektdimension **Nicht konforme Funktion**, eine einstufige Textdimension **Konformitätsfehlerart**, die ein Schlüsselwort für die Art des Fehlers enthält und eine Kennzahl **Anzahl der Konformitätsfehler**, die die Fehler innerhalb einer Prozessinstanz zählt.

Die XML-Struktur der Relation Konformitätsfehler **REL_CONFORMANCE_ISSUE** sieht folgendermaßen aus:

```
<relation name="REL_CONFORMANCE_ISSUE">
<description name="Conformance issue" language="en" />
<sourcedim name="D_PRECEDING_FUNCTION" />
<targetdim name="D_NONCONFORMING_FUNCTION" />
<refki name="RNUM_REL_CONFORMANCE_ISSUE" />
<refdim name="D_CONFORMANCE_ISSUE_TYPE" />
</relation>
```

Aufgrund der internen Abläufe der Berechnung muss die Relation in Abhängigkeit zu AT_KI_CONFORMANCE_RATE [70] stehen. Die Berechnungsregel lautet wie folgt:

```
<calcrel name="REL_CONFORMANCE_ISSUE">
<depends attrname="AT_KI_CONFORMANCE_RATE" type="PROCESS" />
<calcclass name="com.idsscheer.ppm.server.keyindicator.relation.calculator.
ZRelationCalculatorConformanceIssues" loglevel="VERBOSE" />
</calcrel>
```

Die Relation wird von der Berechnungsklasse **ZRelationCalculatorConformanceIssues** berechnet.

Die Fehlerart wird in dem Schlüsselattribut der Relation gespeichert, das für die zugeordnete Dimension konfiguriert wurde. Der Attributname kann frei gewählt werden. Die Dimension muss eine einstufige Textdimension mit dem Schlüsselwort **D_CONFORMANCE_ISSUE_TYPE** sein. Andernfalls wird die Relation nicht berechnet und keine der zugehörige Kennzahlen und Dimensionen besitzt einen Wert.

Wenn Sie ein anderes Passwort verwenden müssen, zum Beispiel weil im Customizing bereits eine Dimension **D_CONFORMANCE_ISSUE_TYPE** existiert, können Sie dieses Passwort der Berechnungsklasse **ZRelationCalculatorConformanceIssues** als Wert des Parameters **issue_type_dimension_keyword** liefern.

7.1.3.1.14. Konvertierung von Zeitspannen in Millisekunden

Die Klasse **ZAttributeCalculatorConvertMillisecondDuration** ist eine parametrisierte Attributberechnerklasse, die verwendet werden kann, um Zeitspannen im internen Format **MillisecondDurationType** in ein PPM-Zeitspannenformat zu konvertieren. Das Format **MillisecondDurationType** besteht aus einem Wert im Float-Format (ohne Einheit), der eine Zeitspanne in Anzahl Millisekunden beinhaltet.

Beispiel 31.

<attribute type="DURATION_IN_MS">12618.0</attribute>

<calcattr name="AT_KI_DURATION" type="OT_FUNC\" >

<calcclass name="com.idsscheer.ppm.server.keyindicator.attributecalculator.

ZAttributeCalculatorConvertMillisecondDuration">

<calcparam key="ATTRIBUTE_MILLISECOND_DURATION" value="AT_DURATION_IN_MS"/>.

</calcclass>

</calcattr>"

In diesem Beispiel ermittelt der Attributberechner an allen Funktionen einer Prozessinstanz den Wert des Attributes **AT_DURATION_IN_MS**, interpretiert diesen als Angabe in Millisekunden und konvertiert ihn in Sekunden. Das Ergebnis wird auf volle Sekunden gerundet und an das Attribut **AT_KI_DURATION** geschrieben.

Voraussetzung für die Verwendung der Berechnerklasse ist, dass das Quellattribut (im Beispiel AT_DURATION_IN_MS) vom Typ **TEXT** oder **DOUBLE** und das Zielattribut (im Beispiel **AT_KI_DURATION**) vom Typ **TIMESPAN** sind.

7.1.3.1.15. Kennzeichnung großer EPK

Der Attributberechner **ZAttributeCalculatorFunctionCount** schreibt die Anzahl der Funktionsknoten in der EPK in ein konfigurierbares Attribut auf Prozessebene.

Beispiel

```
<calcattr name="AT_FCT_COUNT" type="PROCESS" >
<calcclass name="com.idsscheer.ppm.server.keyindicator.
attributecalculator.ZAttributeCalculatorFunctionCount">
</calcclass>
</calcattr>
```

Für komprimierte EPK findet keine Berechnung statt.

Weitere Informationen zum Thema **Behandlung großer EPKs** erhalten Sie in der Dokumentation **PPM Datenimport**.

7.1.3.2. Operanden

Operanden liefern die Eingabewerte (Parameter) für Berechnungsvorschriften. Der Attributberechner unterscheidet drei Typen von Operanden: Wertemenge, Wert und Konstante.

7.1.3.2.1. Wertemengen (XML-Element attribute)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="..." type="...">
<calculation>
...
<attribute name="..." nodetype="..." objectname="..." onerror="..."/>
...
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Das XML-Element **attribute** liefert als Ergebnis eine Menge von Attributwerten. Diese beinhaltet alle in der Prozessinstanz gepflegten Objektattributwerte (**nodetype** ungleich **PROCESS**) des spezifizierten Attributs.

Für Prozessinstanzattribute (**nodetype="PROCESS"**) enthält die Wertemenge nur den Wert des an der Instanz gepflegten Attributes.

Die Wertemenge ist leer, wenn das Attribut innerhalb der Prozessinstanz nicht gepflegt ist.

XML-Tag	Beschreibung
name	Interner Name des Attributs Wird der Attributname mit endständigem Platzhalter ***** angegeben, werden die Werte aller Attribute, deren Name mit der angegebenen Zeichenfolge beginnt, in die Ergebnismenge aufgenommen.
nodetype	Attributtyp: Funktions- (OT_FUNC) oder Prozessinstanzattribut (**PROCESS**)
objectname (optional)	Für Funktionsattribute (**nodetype=**"OT_FUNC") kann die Wertemenge auf Attributwerte des angegebenen Objektnamens eingeschränkt werden. Wird als Objektname **this** angegeben, wird der Attributwert genau der Funktion ermittelt, für die die Berechnung gerade ausgeführt wird. Wird als Objektname **like** angegeben, wird der Attributwert aller gleichnamigen Funktionen ermittelt.
onerror (optional)	Steuert das Verhalten des Kennzahlenberechners, wenn keine Attributwertemenge ermittelt werden kann: **EXIT_WARNING**: Abbruch der aktuellen Attributberechnung mit Ausgabe einer Warnung im Protokoll. **EXIT_NO_WARNING**: Abbruch der aktuellen Attributberechnung ohne Ausgabe einer Warnung im Protokoll. **CONTINUE**: Standardwert. Die aktuelle Attributberechnung wird mit einer Leermenge fortgesetzt. Über weitere Fehlerbehandlungen entscheiden die übergeordneten Operatoren. Eine Ausgabe im Protokoll erfolgt nicht.

Warnung
Die Angabe eines Objektnamens **this** oder **like** im XML-Attribut **objectname** ist nur für die Berechnung von Funktionsattributen (**nodetype="OT_FUNC"**) erlaubt.

Beispiel 32.

Die Wertemenge enthält alle Attributwerte **AT_KI_FDLZ** (Funktionsdurchlaufzeit) von Funktionen, die denselben Namen (Funktionsattribut **AT_OBJNAME_INTERN**) wie die gerade berechnete Funktion haben.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="AT_KI_FDLZSUM" type="OT_FUNC">
<calculation>
<sum>
<attribute name="AT_KI_FDLZ" nodetype="OT_FUNC" objectname="like"/>
</sum>
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Beispiel 33.

Es werden die Werte aller Attribute berücksichtigt, deren Bezeichner mit **AT_SALES_VOLUME_** beginnt.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="AT_KI_XXX" type="PROCESS">
<calculation>
<sum>
<attribute name="AT_SALES_VOLUME_*" nodetype="OT_FUNC"/>
</sum>
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Für den Mustervergleich (Pattern-Matching) der internen Attributnamen gelten folgende Einschränkungen:

- Platzhalter sind bei gefilterten Attributen (XML-Element filteredattribute) nicht zulässig, da dieser Operand nur genau einen Attributwert liefert und sich nur auf genau ein Attribut bezieht.
- Der Platzhalter * wird nur am Ende eines Attributnamens unterstützt.
- Die vom Mustervergleich betroffenen Attribute müssen vom selben Datentyp sein.

7.1.3.2.2. Werte (XML-Element filteredattribute)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="..." type="...">
<calculation>
```

```
  ...
<filteredattribute name="..." nodetype="..." objectname="..." filter="..."
onerror="..."/>
...
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Zur Ermittlung eines konkreten Wertes aus einer Wertemenge können alle in Kapitel Wert erzeugende Operatoren [93] vorgestellten mathematischen Funktionen verwendet werden, die als Ergebnis einen konkreten Wert liefern. Alternativ kann mit dem XML-Element **filteredattribute** ein konkreter Wert aus einer Attributwertemenge bestimmt werden.

Die Angaben **name**, **nodetype**, **objectname** und **onerror** entsprechen denen des XML-Elements **attribute**.

XML-Tag	Beschreibung
name	Interner Name des Attributs Die Verwendung des Mustervergleichs wird beim XML-Element **filteredattribute** nicht unterstützt, da sich die Angabe auf genau ein Attribut bezieht.
nodetype	Attributtyp: Funktionsattribut (**OT_FUNC**), Prozessinstanzattribut (**PROCESS**), oder Beziehungsattribut (**RELATION**)
objectname (optional)	Für Funktionsattribute (**nodetype="OT_FUNC"**) kann die Wertemenge auf Attributwerte des angegebenen Objektnamens eingeschränkt werden. Für Beziehungsattribute (**nodetype="RELATION"**) sind folgende Werte zulässig: **this** Das Attribut wird an der aktuell berechneten Beziehung gesucht (**calcattr name="..." type="RELATION" relname="REL_..."**). **source** Das Attribut wird am Quellobjekt der Beziehung gesucht. **target** Das Attribut wird am Zielobjekt der Beziehung gesucht.
filter (optional)	Filter, der zur Auswahl des Elementes aus der Wertemenge verwendet wird (Nicht bei **objectname="this"**): **EARLY** Es wird der Attributwert desjenigen Objekts übernommen, bei dem eines der Attribute **AT_START_TIME** und **AT_END_TIME** den insgesamt frühesten Zeitpunkt angibt. **LATEST** Es wird der Attributwert desjenigen Objekts übernommen, bei dem eines der Attribute **AT_START_TIME** und **AT_END_TIME** den insgesamt spätesten Zeitpunkt angibt.
onerror (optional)	Steuert das Verhalten des Kennzahlenberechners, wenn kein Attributwert ermittelt werden kann: **EXIT_WARNING**: Abbruch der aktuellen Attributberechnung mit Ausgabe einer Warnung im Protokoll. **EXIT_NO_WARNING**: Abbruch der aktuellen Attributberechnung ohne Ausgabe einer Warnung im Protokoll. **CONTINUE**: Standardwert. Die aktuelle Attributberechnung wird mit NULL fortgesetzt. Über weitere Fehlerbehandlungen entscheiden die übergeordneten Operatoren. Eine Ausgabe im Protokoll erfolgt nicht.

Durch Angabe des Objekttyps (**nodetype**) können gleichnamige Attribute an der Prozessinstanz und an Objekten der Prozessinstanz unterschieden werden.

7.1.3.2.3. Konstanten (XML-Element constant)

Der Wert einer Konstanten wird im Bereich **CDATA** des XML-Elementes **<dataitem>** angegeben. Folgendes Beispiel definiert eine Zeitspannenkonstante von zehn Minuten:

```
<constant>
<dataitem>
10 MINUTE
<datatype name="TIMESPAN"></datatype>
</dataitem>
</constant>
```

Wenn die Angaben zu Datentyp und Wert der Konstanten korrekt sind, werden mögliche Angaben im Attribut **value** des Elements **<dataitem>** ignoriert. Die folgende Definition erzeugt eine Konstante von zwei Stunden:

```
<constant>
<dataitem value="9">
2 HOUR
<datatype name="TIMESPAN"></datatype>
</dataitem>
</constant>
```

Wenn die Angabe im Bereich **CDATA** des Elements **<dataitem>** keinen Wert oder einen Wert mit unzulässigem Datentyp liefert, wird sie ignoriert. Stattdessen werden die Angaben im XML-Attribut **value** verarbeitet. Im folgenden Beispiel ist der Wert der Konstanten mit dem Datentyp **DOUBLE** angegeben, obwohl der Datentyp **LONG** erwartet wird. Die falsche Wertangabe wird ignoriert und stattdessen wird der Wert des Attributs **value** ("**2**") in die Konstante geschrieben:

```
<constant>
<dataitem value="2">
4.0
<datatype name="LONG"></datatype>
</dataitem>
</constant>
```

Wenn auch die Wertangabe im Attribut **value** nicht dem erwarteten Datentyp entspricht, wird die Berechnung abgebrochen:

```
<constant>
<dataitem value="2.4">
4.0
<datatype name="LONG"></datatype>
</dataitem>
</constant>
```

Im folgenden Beispiel wird die Berechnung abgebrochen, da Datentyp und Wertangabe nicht übereinstimmen und kein Eintrag im XML-Attribut **value** vorhanden ist:

```
<constant>
<dataitem>
4.0
<datatype name="LONG"></datatype>
```

```
</dataitem>
</constant>
```

Eine Konstante muss immer mit der Einheit angegeben werden, die für den Datentyp des Attributes zulässig ist. Der Datentyp muss im PPM-System bekannt sein.

Numerische Konstanten bestehen aus der Angabe des Wertes mit einer für den Datentyp zulässigen Einheit und dem Datentyp selbst.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="..." type="...">
<calculation>
...
<constant>
<dataitem>
10 MINUTE
<datatype name="TIMESPAN"></datatype>
</dataitem>
</constant>
...
</calculation>
</calcattr>
```

...

```
</keyindicatorconfig>
```

XML-Element	Beschreibung
dataitem value	Wert der Konstanten mit Einheit
datatype name	Bezeichner des Datentyps. Es können sowohl interne als auch benutzerdefinierte Datentypen verwendet werden.

Alphanumerische Konstanten werden folgendermaßen angegeben:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM´'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="..." type="...">
<calculation>
...
<constant>
<dataitem>
Constant text
<datatype name="TEXT"/>
</dataitem>
</constant>
...
</calculation>
</calcattr>
```

```
...
</keyindicatorconfig>
```

7.1.3.2.4. Attributwertermittlung

Zur Attributberechnung können sowohl Attributwerte mit Objektbezug als auch ohne Objektbezug verwendet werden.

7.1.3.2.5. Attributwerte ohne Objektbezug

Das angegebene Attribut wird an allen Prozessinstanzobjekten des durch **nodetype** spezifizierten Objekttyps verwendet, an denen es gepflegt ist. Hierdurch entsteht eine Wertemenge, die eine dem Auftreten des Attributes entsprechende Anzahl von Elementen enthält.

Beispiel 34.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="..." type="...">
<calculation>

  ...
<attribute name="AT_ABC" nodetype="OT_FUNC"/>
...
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Die Werte des Attributs **AT_ABC** aller Funktionen der gerade berechneten Prozessinstanz werden in die Wertemenge aufgenommen.

7.1.3.2.6. Attributwerte mit Objektbezug

Das angegebene Attribut wird nur an den Funktionen (**nodetype="OT_FUNC"**) mit dem angegebenen Namen (**objectname"FCT_..."**) verwendet. (Der mit **objectname** angegebene Objektname entspricht dem Wert des Funktionsattributs **AT_OBJNAME_INTERN**.) Auch hier kann eine Wertemenge mit mehr als einem Element entstehen, da das angegebene Objekt mehrmals in der Prozessinstanz vorkommen kann.

Beispiel 35.

Die Werte des Attributs **AT_AUFNR** aller Funktionen mit dem Namen **FCT_AUFTRAG_ANLEGEN** der gerade berechneten Prozessinstanz werden in die Wertemenge aufgenommen.

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

```
<keyindicatorconfig>
...
<calcattr name="..." type="...">
<calculation>

  ...
<attribute name="AT_AUFNR" nodetype="OT_FUNC objectname="FCT_AUFTRAG_ANLEGEN"/>
...
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Die Farbe von Funktionen oder Ereignissen in der EPK-Ansicht können Sie über das Standardattribut **AT_BGND_COLOUR** bestimmen. Die folgende Berechnungsvorschrift weist allen Funktionen mit dem internen Namen **SAP.WAUS** eine rote Farbe zu:

```
<calcattr name="AT_BGND_COLOUR" type="OT_FUNC" objectname="SAP.WAUS">
<calculation>
<constant>
<dataitem>
<datatype name="TEXT">255,0,0</datatype>
</dataitem>
</constant>
</calculation>
</calcattr>
```

Sie können diese Berechnungsvorschrift z.B. innerhalb einer bedingten Attributberechnung verwenden. Der jeweilige Farbwert wird als RGB-Wert angegeben. Eine Farbzuweisung für bestimmte Objekte oder Objekttypen kann auch im Attribut-Mapping erfolgen.

Beispiel 36.

Die Berechnungsvorschrift summiert die Durchlaufzeit der Funktion (**AT_KI_FDLZ**) für gleichnamige Funktionen (identischer Wert des Attributs **AT_OBJNAME_INTERN**).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="AT_KI_FDLZSUM" type="OT_FUNC">
<calculation>
<sum>
<attribute name="AT_KI_FDLZ" nodetype="OT_FUNC" objectname="like"/>
</sum>
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Warnung
Die Angabe eines Objektnamens bzw. der Optionen **this** oder **like** im XML-Attribut **objectname** ist nur bei der Berechnung von Funktionsattributen (OT_FUNC) zulässig.

Wenn Sie einen Operator, der einzelne Werte als Operanden erwartet (z.B. **<plus>**), mit einem Operanden verwenden möchten, der Wertemengen liefert (z.B. **<attribute>**), müssen Sie durch geeignete Operatoren aus einer Wertemenge einen einzelnen Wert bestimmen (z.B. **<min>** oder **<max>**). Alternativ können Sie mit dem XML-Element **<filteredattribute>** aus einer Wertemenge einen Wert bestimmen, der für die weitere Attributberechnung verwendet werden soll.

7.1.3.3. Bedingter Attributtypzugriff

Sie können innerhalb einer Berechnungsvorschrift für die Berechnung eines Funktionsattributs die Menge der zu berücksichtigenden Attributtypwerte einschränken, indem Sie eine Bedingung angeben, die sich auf andere Attributtypen an der gleichen Funktion (**objectname="this"**) bezieht. Für die Konfiguration der Bedingung müssen Sie einen booleschen Operator als Wurzeloperator verwenden (siehe Kapitel Logische Operatoren [98]). Die Verschachtelungstiefe der Bedingung ist beliebig. Wenn die Bedingungsprüfung den Wahrheitswert **TRUE** ergibt, fließt der Wert des Attributtyps, für den die Bedingung definiert ist, in die weitere Berechnung mit ein.

Sie können Bedingungen für **<attribute>** und **<filteredattribute>** angeben.

Beispiel 37.

Aus der Wertemenge des Funktionsattributs **AT_HRMODUL** werden nur die Attributtypwerte von Funktionen berücksichtigt, an denen zusätzlich (**objectname="this"**) der Attributtyp **AT_VORG_TYPE** mit Wert **019** gepflegt ist.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="AT_KI_HRMODUL" type="PROCESS">
<calculation>
<max>
<attribute name="AT_HRMODUL" nodetype="OT_FUNC">
<in>
<constant>
<dataitem>
019
<datatype name="TEXT">Text</datatype>
</dataitem>
</constant>
<attribute name="AT_VORG_TYPE" nodetype="OT_FUNC" objectname="this"/>
</in>
</attribute>
</max>
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Definieren Sie bedingte Attributtypzugriffe im PPM Customizing Toolkit um Syntaxfehler zu vermeiden. Die entsprechenden Berechnungsvorschriften erstellen Sie im Modul **Kennzahlen und Dimensionen** im Menü **Berechnete Attributtypen**. Speziell die inkorrekte Verwendung der Operanden **attribute** bzw. **filteredattribute** mit den jeweiligen logischen Operatoren wird darin abgefangen.

7.1.3.4. Operatoren

In Berechnungsvorschriften für Attributtypen bzw. Berechnungsfunktionen werden die einzelnen Operandentypen (Wertemengen, Werte, Konstanten) mittels Operatoren untereinander verknüpft. Wenn beim Verknüpfen von Attributtypen alle Operanden denselben Datentyp haben, werden die Ergebnisse der Attributtypberechnungen in diesem Datentyp zurückgegeben.

Sie können für jeden Operator über das XML-Attribut **mode** das Verhalten der Ausnahmebehandlung angeben (z.B. **<addtimespan mode="PPM4">**). Mögliche Werte sind **PPM3** für das Verhalten bis einschließlich PPM 3.2.1 und **PPM4** für das fehlertolerantere Verhalten ab PPM 4.0.

Aus Gründen der Abwärtskompatibilität ist der Vorgabewert **PPM3**.

In Berechnungsvorschriften, die Sie mit PPM Customizing Toolkit erstellen, wird den verwendeten Operatoren standardmäßig der Wert **PPM4** zugewiesen.

Das Berechnungs- bzw. Fehlerverhalten der beiden verschiedenen Modi ist ab Kapitel Mathematische Operatoren [82] für jeden Operator beschrieben.

Definieren Sie in Ihren Attributtypberechnungen jeweils einen Standardrückgabewert **defaultvalue**, der dem zu berechnenden Attributtyp zugewiesen wird, wenn die Attributtypberechnung fehlschlägt.

> **Warnung**
> Kombinieren Sie in einer Berechnungsvorschrift keine numerischen Werte mit nicht-numerischen Werten (z. B. **TEXT** mit **DOUBLE**), da solche Berechnungsvorschriften zum Abbruch der Berechnung führen.

Numerische Datentypen sind untereinander frei kombinierbar (z. B. mit dem Operator **set**). Die Bildung von Werten erfolgt immer in der Basiseinheit des Attributtyps. Bei Attributtypberechnungen mit gemischten numerischen Datentypen (z.B. **DOUBLE**, **TIMESPAN**, **FACTORYTIMESPAN**) werden alle Werte ohne Einheiten verwendet und das Ergebnis wird im Datentyp **DOUBLE** gespeichert. Diesen Wert können Sie anschließend in einem PPM-Zielattributtyp eines anderen Datentyps in der jeweiligen Basiseinheit speichern.

Die Angabe einer Verknüpfung wird in Form der Umgekehrten Polnischen Notation (UPN) spezifiziert, das heißt, zuerst wird die Verknüpfungsart (Operator) und anschließend werden die Operanden angegeben. In XML-Notation sieht dies folgendermaßen aus:

```
<operator 1>
<operand m>
...
</operand m>
<operand m+1>
...
</operand m+1>
<operator 2>
<operand n>
...
</operand n>
<operand n+1>
...
</operand n+1>
</operator 2>
</operator 1>
```

Das XML-Element **operator** liefert den berechneten Zahlenwert (numerisches Ergebnis der durch den Operator verknüpften Operanden). Die Einheit des Ergebnisses wird durch den Datentyp des Attributtyps bestimmt, dem der Ergebniswert zugewiesen wird. Operatoren können selbst Teil eines übergeordneten Operators sein.

Beispiel 38.

Berechnung des Umfanges eines Kreises mit einem Radius von 6

(Kreisumfang = 2 * p * Radius):

```
<times>
<constant>
<dataitem value="2">
<datatype name="DOUBLE"/>
</dataitem>
</constant>
<constant>
<dataitem value="3.1415">
<datatype name="DOUBLE"/>
</dataitem>
</constant>
<constant>
<dataitem value="6">
<datatype name="DOUBLE"/>
</dataitem>
</constant>
</times>
```

Beispiel 39.

Eine alternative Möglichkeit, die zum selben Ergebnis führt, ist, aus den Operanden eine Menge zu bilden und alle Elemente der Menge miteinander zu multiplizieren:

```
<product>
<set>
<constant>
<dataitem value="2">
<datatype name="DOUBLE"/>
</dataitem>
</constant>
<constant>
<dataitem value="3.1415">
<datatype name="DOUBLE"/>
</dataitem>
</constant>
<constant>
<dataitem value="6">
<datatype name="DOUBLE"/>
</dataitem>
</constant>
</set>
<product>
```

Zur Berechnung von Attributtypwerten stehen die in den folgenden Kapiteln beschriebenen Operatoren zur Verfügung.

7.1.3.4.1. Mathematische Operatoren

Folgende Operatoren stehen zur Verfügung: plus, minus, timespan, times, divide, abs, div, mod, squareroot und round.

Addition

XML-Tag:	**plus**	
Operanden:	mindestens zwei Werte	
Synopsis:		

	`<plus>`
	`<value 1>`
	`<value 2>`
	`<value n>`
	`</plus>`	

Operanden:	Numerische Datentypen (LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE)	
Ergebnis:	Wert	
Ergebnistyp:	Datentyp der Operanden bei einheitlichem Datentyp. DOUBLE bei gemischten, numerischen Datentypen, die in diesem Fall automatisch nach DOUBLE konvertiert werden.	
Description:	Addiert die im XML-Element angegebenen Werte	
Berechnung (PPM3)	Ergebnis	Summe aller Operanden
	Fehler	Wenn mind. ein Operand gleich NULL oder mind. ein Operand mit nicht numerischem Datentyp
Berechnung (PPM$)	Ergebnis	NULL, wenn mind. ein Operand gleich NULL, sonst Summe aller Operanden
	Fehler	Nur bei nicht numerischem Datentyp
Beispiel:	-	

Subtraktion

XML-Tag:	**minus**
Operanden:	genau zwei Werte
Synopsis:	

	`<minus>`
	`<value 1>`
	`<value 2>`
	`</minus>`

Operanden:	numerische Datentypen (LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE)
Ergebnis:	Wert (Differenz)
Ergebnistyp:	Datentyp der Operanden bei einheitlichem Datentyp. DOUBLE bei gemischten, numerischen Datentypen, die in diesem Fall automatisch nach DOUBLE konvertiert werden.

Description:	Subtrahiert Wert 2 von Wert 1	
Berechnung (PPM3)	Ergebnis	Ergebnis der Subtraktion des Operanden 2 von Operand 1
	Fehler	Wenn mind. ein Operand gleich NULL oder mind. ein Operand mit nicht numerischem Datentyp
Berechnung (PPM$)	Ergebnis	NULL, wenn mind. ein Operand gleich NULL, sonst Ergebnis der Subtraktion des Operanden 2 von Operand 1
	Fehler	Nur bei nicht numerischem Datentyp
Beispiel:	-	

Zeitspanne

XML-Tag:	**timespan**	
Operanden:	genau zwei Werte (Zeitpunkte)	
Synopsis:		

```
<timespan>
        <time 1>
        <time 2>
        </timespan>
```

Operanden:	TIME (TIMESTAMP, DATE)	
Ergebnis:	Wert (Zeitspanne)	
Ergebnistyp:	TIMESPAN oder FACTORYTIMESPAN bei Verwendung eines Werkskalenders	
Description:	Berechnet die Zeitdifferenz zwischen Zeitpunkt 1 und Zeitpunkt 2. Entsteht eine negative Differenz, wird der Wert 0 zurückgegeben. Wenn Sie zur Berechnung der Zeitdifferenz den Werkskalender verwenden möchten, geben Sie dem optionalen XML-Attribut **type** den Wert **FACTORYCALENDAR**. Standardwert: **NORMAL**	
Berechnung (PPM3)	Ergebnis	Zeitspanne zwischen Operand 1 und Operand 2 (Operand 1 minus Operand 2)
	Fehler	Wenn mind. ein Operand gleich NULL oder mind. ein Operand mit unzulässigem Datentyp
Berechnung (PPM$)	Ergebnis	NULL, wenn mind. ein Operand gleich NULL, sonst Zeitspanne zwischen Operand 1 und Operand 2 (Operand 1 minus Operand 2)
	Fehler	Nur bei unzulässigem Datentyp

Beispiel:

```
<timespan type="FACTORYCALENDAR"
        directoryname="custom/client/factorycal"
        attributename="AT_FC_XYZ">
        <max>
        <attribute name="AT_GOODS_RECEIPT_DATE"
        nodetype="OT_FUNC"
        objectname="SAP.MM_WE_ANLEG"/>
        </max>
        <min>
        <attribute name="AT_END_TIME" nodetype="OT_FUNC"
        objectname="SAP.MM_BANF_ANLEG"/>
        </min>
        </timespan>
```

Bei Verwendung eines Werkskalenders können durch Angabe des optionalen XML-Attributs **negfactorytimespan="TRUE"** auch nicht positive Zeitspannen berechnet werden.

Standardwert: **FALSE**

Sie können Zeitspannenberechnungen auch auf der Basis externer Werkskalender durchführen, indem Sie eine Werkskalender XML-Datei angeben. Im optionalen XML-Attribut **directoryname** spezifizieren Sie das Verzeichnis mit der zu verwendenden Werkskalenderdatei. Im Attribut **attributename** spezifizieren Sie den Namen des Attributtyps, der den Namen der zu verwendenden Werkskalenderdatei enthält. Der Attributtyp muss am entsprechenden Objekt bzw. der Prozessinstanz gepflegt sein. Die beiden XML-Attribute müssen immer zusammen angegeben werden.

Multiplikation

XML-Tag:	**times**
Operanden:	mindestens zwei Werte
Synopsis:	

```
<times>
        <value 1>
        <value 2>
        <value n>
        </times>
```

Operanden:	numerische Datentypen (LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE)
Ergebnis:	Wert (Produkt)
Ergebnistyp:	Datentyp der Operanden bei einheitlichem Datentyp. DOUBLE bei gemischten, numerischen Datentypen, die in diesem Fall automatisch nach DOUBLE konvertiert werden.

Description:	Multipliziert die im XML-Element angegebenen Werte	
Berechnung (PPM3)	Ergebnis	Ergebnis der Multiplikation der Operanden 1 bis n
	Fehler	Wenn mind. ein Operand gleich NULL oder mind. ein Operand mit nicht numerischem Datentyp
Berechnung (PPM$)	Ergebnis	NULL, wenn mind. ein Operand gleich NULL, sonst Ergebnis der Multiplikation aller Operanden
	Fehler	Nur bei nicht numerischem Datentyp
Beispiel:	-	

Division

XML-Tag:	**divide**	
Operanden:	genau zwei Werte	
Synopsis:		

```
<divide>
        <value 1>
        <value 2>
        </divide>
```

Operanden:	numerische Datentypen (LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE)	
Ergebnis:	Wert (Quotient)	
Ergebnistyp:	immer DOUBLE	
Description:	Dividiert Wert 1 durch Wert 2.	
Berechnung (PPM3)	Ergebnis	Ergebnis der Division von Operand 1 durch Operand 2
	Fehler	Wenn mind. ein Operand gleich NULL oder mind. ein Operand mit nicht numerischem Datentyp oder Operand 2 = 0
Berechnung (PPM$)	Ergebnis	NULL, wenn mind. ein Operand gleich NULL, sonst Ergebnis der Division von Operand 1 durch Operand 2
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp oder Operand 2 = 0
Beispiel:	-	

Betrag

XML-Tag:	**abs**
Operanden:	genau ein Wert
Synopsis:	

```
<abs>
        <value 1>
        </abs>
```

Operanden:	Numerische Datentypen (LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE)
Ergebnis:	Wert (Betrag)
Ergebnistyp:	Datentyp des Operanden
Description:	Bildet den Betrag eines Wertes.

Berechnung (PPM3)	Ergebnis	Absoluter Betrag des Operanden
	Fehler	Wenn Operand gleich NULL oder mit nicht numerischem Datentyp
Berechnung (PPM$)	Ergebnis	NULL, wenn Operand gleich NULL, sonst absoluter Betrag des Operanden
	Fehler	Nur bei nicht numerischem Datentyp
Beispiel:	-	

Ganzzahlige Division

XML-Tag:	**div**	
Operanden:	genau zwei ganzzahlige Werte	
Synopsis:		

```
<div>
        <value 1>
        <value 2>
        </div>
```

Operanden:	LONG	
Ergebnis:	Ganzzahliger Wert der Division	
Ergebnistyp:	LONG	
Description:	Liefert den ganzzahligen Wert zurück, wie oft Wert 2 in Wert 1 enthalten ist. Divisionsreste werden ignoriert. Bei echten Brüchen wird 0 zurückgegeben.	
Berechnung (nur PPM4)	Ergebnis	NULL, wenn mind. ein Operand gleich NULL, sonst ganzzahliges Ergebnis der Division von Operand 1 durch Operand 2
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp (nicht LONG) oder Operand 2 = 0
Beispiel:		

```
<div>
        <max>
        <attribute name="AT_COST" nodetype="FUNCTION"/>
        </max>
        <constant>
        <dataitem>
        5
        <datatype name="LONG">
        Long
        </datatype>
        </dataitem>
        </constant>
        </div>
```

Modulo

XML-Tag:	mod	
Operanden:	genau zwei ganzzahlige Werte	
Synopsis:	<mod> <value 1> <value 2> </mod>	
Operanden:	LONG	
Ergebnis:	Ganzzahliger Divisionsrest	
Ergebnistyp:	LONG	
Description:	Liefert als Ergebnis den Rest der ganzzahligen Division von Wert 1 durch Wert 2 zurück. Bei echten Brüchen wird der Wert des 1. Operanden zurückgegeben. Wenn value 1= value 2, wird 0 zurückgegeben.	
Berechnung (nur PPM4)	Ergebnis	NULL, wenn mind. ein Operand gleich NULL, sonst Rest der ganzzahligen Division von Operand 1 durch Operand 2
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp (nicht LONG) oder Operand 2 = 0
Beispiel:	<mod> <filteredattribute name="AT_COST" nodetype="FUNCTION"/> <constant> <dataitem> 3 <datatype name="LONG">Long</datatype> </dataitem> </constant> </mod>	

Quadratwurzel

XML-Tag:	squareroot
Operanden:	genau ein Wert
Synopsis:	<squareroot> <value 1> </squareroot >
Operanden:	numerische Datentypen (LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE und benutzerdefinierte Typen, z.B. COST)
Ergebnis:	Quadratwurzel

Ergebnistyp:	Datentyp DOUBLE	
Description:	Berechnet die Quadratwurzel aus dem angegebenen Wert	
Berechnung (nur PPM4)	Ergebnis	NULL, wenn der Operand gleich NULL, sonst Quadratwurzel des numerischen Operanden
	Fehler	Wenn Operand mit unzulässigem Datentyp (nicht numerisch) oder Wert des Operanden kleiner 0
Beispiel:		
	–	

Runden

XML-Tag:	**round**	
Operanden:	Genau ein Wert vom Typ TIMESPAN	
Synopsis:	<round> <value> </round>	
Operanden:	TIMESPAN	
Ergebnis:	Gerundeter Zeitspannenwert	
Ergebnistyp:	TIMESPAN	
XML-Attribute	scale (MINUTE|HOUR|DAY|WEEK|MONTH|YEAR) roundingkind (ROUND|FLOOR|CEIL) "ROUND"	
Description:	Liefert den gerundeten Wert von Zeitspannen zurück. Gerundet werden kann nur ein Wert vom Datentyp TIMESPAN. Die Skalierung (scale), in der gerundet werden soll, muss angegeben werden. Es gibt folgende Rundungsarten: ROUND (Nachkommastellen < 5 werden abgerundet bzw. wenn >= 5 aufgerundet) CEIL (unabhängig vom Wert der Nachkommastelle wird zur nächsten Ganzzahl aufgerundet) FLOOR (unabhängig vom Wert der Nachkommastelle wird zur aktuellen Ganzzahl abgerundet) Der Standardwert ist ROUND.	
Berechnung (nur PPM4)	Ergebnis	Der gerundete Wert in der angegebenen Skalierung
	Fehler	Wenn Operand von unzulässigem Datentyp bzw. unzulässige Anzahl von Operanden

Beispiel:

```
<round scale="MINUTE" roundingkind="CEIL">
        <constant>
        <dataitem value="4284.0">
        1,19
        <datatype name="TIMESPAN">
        Time span
        </datatype>
        <scale name="HOUR" factor="3600.0">
        Hours
        </scale>
        </dataitem>
        </constant>
        </round>
```

Der Zeitspannenwert 1,19 Stunden wird gemäß Skalierungsangabe in 71,4 Minuten umgerechnet und gemäß angegebener Rundungsart auf 72 Minuten (Rückgabewert) aufgerundet.

Nullwert

XML-Tag:	**nullvalue**	
Operanden:	genau zwei Werte	
Synopsis:	<nullvalue> <value 1> <value 2> </nullvalue>	
Operanden:	Beliebiger Datentyp, beide Operanden müssen vom gleichen Datentyp sein.	
Ergebnis:	Wert des ersten Operanden, wenn dieser nicht null ist, sonst Wert des zweiten Operanden	
Ergebnistyp:	Datentyp des Operanden	
Description:	Ersetzt den ggf. nicht vorhandenen Wert des ersten Operators (Wert null) durch den Wert des zweiten Operators. Liefert der erste Operand einen Wert, wird dieser zurückgegeben, sonst der Wert des zweiten Operanden. Wenn beide Operatoren keinen Wert liefern, wird null als Wert zurückgegeben. Der zweite Operator sollte also immer einen Wert liefern.	
Berechnung (nur PPM4)	Ergebnis	Wert des ersten Operanden, wenn dieser nicht null ist, sonst Wert des zweiten Operanden.
	Fehler	Wenn Operanden unterschiedlichen Datentyp aufweisen

Beispiel:

```
<nullvalue>
<subtext beginindex="3">
<filteredattribute name="AT_XYZ"

        nodetype="PROCESS"/>
</subtext>
<constant>
<dataitem>
ABC
<datatype name="TEXT">Text</datatype>
</dataitem>
</constant>
</nullvalue>
```

Falls der **subtext**-Operator keinen Wert liefert, wird die Konstante Zeichenkette **ABC** zurückgegeben.

7.1.3.4.2. Wertemenge erzeugende Operatoren

Folgende Operatoren stehen zur Verfügung: set, union, intersect, removeduplicates.

Mengenbildung

XML-Tag:	**Set**	
Operanden:	mindestens ein Wert	
Synopsis:	<set> <value 1> ... <value n> </set>	
Operanden:	Alle Datentypen, jedoch ist bei nicht numerischen Datentypen ein einheitlicher Datentyp innerhalb der Operandenliste notwendig. Unterschiedliche numerische Datentypen werden automatisch in den Datentyp DOUBLE konvertiert.	
Ergebnis:	Wertemenge	
Ergebnistyp:	DOUBLE bei gemischten, numerischen Operanden, Datentyp des ersten Operanden bei nicht numerischen Datentypen	
Description:	Bildet aus den angegebenen Werten eine Wertemenge.	
Berechnung (PPM3/PPM4)	Ergebnis	Leermenge, wenn alle Operanden NULL liefern, d. h. die Ergebnismenge enthält niemals NULL
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp (Wertemenge oder ungleich Datentyp des ersten Operanden)

Beispiel:

```
<set>

        <constant>

        <dataitem value="2">

        <datatype name="DOUBLE"/>

        </dataitem>

        </constant>

        <value 1>

        <value 2>

        <value n>

        </set>
```

Vereinigungsmenge

XML-Tag:	**Union**	
Operanden:	Mindestens zwei Wertemengen (**<attribute ... />** oder **<set>...</set>** oder **<union>...</union>** oder **<intersect>...</intersect>**)	
Synopsis:	<union> <Set of values 1> <Set of values 2> ... <Set of values n> </union>	
Operanden:	alle Datentypen, jedoch keine Mischung von numerischen und nicht numerischen Datentypen	
Ergebnis:	Wertemenge	
Ergebnistyp:	DOUBLE bei gemischten, numerischen Operanden, Datentyp des ersten Operanden bei nicht numerischen Datentypen	
Description:	Bildet die Vereinigungsmenge aus den angegebenen Wertemengen.	
Berechnung (PPM3)	Ergebnis	Leermenge, wenn alle Operanden Leermengen sind
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp oder mind. ein Operand gleich NULL
Berechnung (PPM$)	Ergebnis	Leermenge, wenn alle Operanden Leermengen sind. NULL, wenn mind. ein Operand gleich NULL ist.
	Fehler	Wenn mind. ein Operand von unzulässigem Datentyp

Beispiel:

```
<union>
        <attribute name="AT_START_TIME" nodetype="PROCESS"/>
        <attribute name="AT_END_TIME" nodetype="PROCESS"/>
        <attribute name="AT_START_TIME" nodetype="OT_FUNC"/>
        <attribute name="AT_END_TIME" nodetype="OT_FUNC"/>
        </union>
```

Schnittmenge

XML-Tag:	Intersect	
Operanden:	mindestens zwei Wertemengen	
Synopsis:		

```
<intersect>
        <Set of values 1>
        <Set of values 2>
        ...
        <Set of values n>
        </intersect>
```

Operanden:	alle Datentypen, jedoch keine Mischung von numerischen und nicht numerischen Datentypen	
Ergebnis:	Wertemenge, die alle Elemente enthält, die in allen Ausgangsmengen enthalten sind	
Ergebnistyp:	DOUBLE bei gemischten, numerischen Operanden, Datentyp des ersten Operanden bei nicht numerischen Datentypen	
Description:	Bildet die Schnittmenge der angegebenen Wertemengen.	
Berechnung (PPM3)	Ergebnis	Leermenge, wenn ein Operand Leermenge.
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp oder mind. ein Operand gleich NULL
Berechnung (PPM$)	Ergebnis	Leermenge, wenn ein Operand Leermenge. NULL, wenn mind. ein Operand gleich Null ist.
	Fehler	Wenn mind. ein Operand von unzulässigem Datentyp
Beispiel:	-	

Duplikatentferner

XML-Tag:	Removeduplicates
Operanden:	genau eine Wertemenge
Synopsis:	

```
<removeduplicates>
        <Set of values>
        </removeduplicates>
```

Operanden:	beliebige Datentypen	
Ergebnis:	Wertemenge	
Ergebnistyp:	Datentyp des Operanden	
Description:	Entfernt Elemente mit identischen Werten aus einer Wertemenge.	
Berechnung (nur PPM4)	Ergebnis	Wertemenge, die alle Elemente enthält, die in der Ausgangsmenge enthalten waren, jedes aber nur einmal. Leermenge, wenn Operand gleich Leermenge ist.
	Fehler	Wenn mind. ein Element der Wertemenge einen unzulässigen Datentyp hat
Beispiel:	Zählen der am Prozess beteiligten Werke:	

```
<card>
        <removeduplicates>
        <attribute name="AT_WERK" nodetype"OT_FUNC" />
        </removeduplicates>
        </card>
```

7.1.3.4.3. Wert erzeugende Operatoren

Folgende Operatoren stehen zur Verfügung: sum, product, card, min, max, mean, convert.

Summe

XML-Tag:	**Summe**	
Operanden:	genau eine Wertemenge	
Synopsis:		

```
<sum>
        <Set of values>
        </sum>
```

Operanden:	numerische Datentypen (LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE)	
Ergebnis:	Wert	
Ergebnistyp:	Datentyp der verwendeten Wertemenge, bei gemischten Datentypen innerhalb der Menge immer DOUBLE	
Description:	Bildet die Summe aller Elemente der Wertemenge.	
Berechnung (PPM3/PPM4)	Ergebnis	Summe der Werte, die in der Wertemenge enthalten sind. NULL, wenn die übergebene Menge leer ist.
	Fehler	Wenn mind. ein Element der Wertemenge einen unzulässigen Datentyp hat.
Beispiel:	-	

Produkt

XML-Tag:	**Product**
Operanden:	genau eine Wertemenge

Synopsis:		
	<product> <Set of values> </product>	
Operanden:	numerische Datentypen (LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE)	
Ergebnis:	Wert	
Ergebnistyp:	Datentyp der verwendeten Wertemenge, bei gemischten Datentypen innerhalb der Menge immer DOUBLE	
Description:	Bildet das Produkt aller Elemente der Wertemenge.	
Berechnung (PPM3/PPM4)	Ergebnis	Multiplikation der Werte, die in der Wertemenge enthalten sind. NULL, wenn die übergebene Menge leer ist.
	Fehler	Wenn mind. ein Element der Wertemenge einen unzulässigen Datentyp hat.
Beispiel:	-	

Kardinalität

XML-Tag:	**Card**	
Operanden:	genau eine Wertemenge	
Synopsis:		
	<card> <Set of values> </card>	
Operanden:	Alle Datentypen der spezifizierten Wertemenge	
Ergebnis:	Wert	
Ergebnistyp:	immer LONG	
Berechnung (PPM3/PPM4)	Ergebnis	Ermittelt die Anzahl aller Elemente der Wertemenge. Bei leerer Menge ist der Rückgabewert **0**.
	Fehler	Nichts
Beispiel:	-	

Minimum

XML-Tag:	**Min**
Operanden:	genau eine Wertemenge
Synopsis:	
	<min> <Set of values> </min>
Operanden:	Numerische Datentypen und TIME (TIMESTAMP, DATE), DAY, TIMEOFDAY
Ergebnis:	Wert

Ergebnistyp:	Datentyp der Wertemenge	
Description:	Liefert den kleinsten Wert aus der Wertemenge.	
Berechnung (PPM3/PPM4)	Ergebnis	NULL bei leerer Wertemenge
	Fehler	Nichts
Beispiel:	-	

Maximum

XML-Tag:	**Max**	
Operanden:	genau eine Wertemenge	
Synopsis:		

	`<max>`
	`<Set of values>`
	`</max>`	

Operanden:	Numerische Datentypen und TIME (TIMESTAMP, DATE), DAY, TIMEOFDAY	
Ergebnis:	Wert	
Ergebnistyp:	Datentyp der Wertemenge	
Description:	Liefert den größten Wert aus der Wertemenge.	
Berechnung (PPM3/PPM4)	Ergebnis	NULL bei leerer Wertemenge
	Fehler	Nichts
Beispiel:	-	

Mittel

XML-Tag:	**Mean**	
Operanden:	genau eine Wertemenge	
Synopsis:		

	`<mean>`
	`<Set of values>`
	`</mean>`	

Operanden:	numerische Datentypen (LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, FREQUENCY, PERCENTAGE)	
Ergebnis:	Wert	
Ergebnistyp:	Datentyp des Operanden	
Berechnung (PPM3)	Ergebnis	Durchschnitt der in der Menge enthaltenen numerischen Werte. NULL, wenn Operand gleich leere Menge.
	Fehler	Bei unzulässigem Datentyp oder wenn mind. ein Operand gleich NULL ist.
Berechnung (PPM$)	Ergebnis	Durchschnitt der in der Menge enthaltenen numerischen Werte. NULL, wenn Operand gleich leere Menge oder NULL.
	Fehler	Wenn mind. ein Element der Wertemenge einen unzulässigen Datentyp hat.

Beispiel:

```
<mean>
        <union>
        <attribute name="AT_ANZAHL_POS1" nodetype="OT_FUNC"/>
        <attribute name="AT_ANZAHL_POS2" nodetype="OT_FUNC"/>
        </union>
        </mean>
```

Datentypkonvertierung

XML-Tag:	convert
Operanden:	genau ein Wert
Synopsis:	

```
<convert datatype="...">
        <value>
        </convert>
```

Operanden:	TEXT bzw. numerischer Datentyp, liefert den Eingabewert für die Konvertierung.
Attribut:	Attribut **datatype** legt fest, in welchen Datentyp der Eingabewert konvertiert werden soll.
Ergebnis:	In den angegebenen Datentyp konvertierter Wert
Ergebnistyp:	LONG, DOUBLE, FREQUENCY, BOOLEAN, TEXT, TIME, TIMESPAN, FACTORYTIMESPAN, DAY, PERCENTAGE

Description:

Konvertieren eines numerischen Datentyps (z. B. LONG, DOUBLE, TIMESPAN, FACTORYTIMESPAN, PERCENTAGE) in einen anderen numerischen Datentyp.

Konvertieren des Datentyps TEXT in einen dieser Datentypen: LONG, DOUBLE, BOOLEAN, TIMESPAN, or FREQUENCY.

Nach der Konvertierung wird das Ergebnis in der Basisskalierung an das Ergebnisattribut geschrieben.

Beim Konvertieren wird das interne PPM-Format verwendet. Es kann kein eigenes Format angegeben werden.

Konvertierung des Datentyps LONG nach TEXT, wobei führende Nullen und Trennzeichen entfernt werden. Das Resultat der Konvertierung ist die in einen String konvertierte Zahl ohne Trennzeichen. In folgendem Bsp. wird der LONG-Wert 000300080191 in den TEXT-Wert 300080191 konvertiert:

```
<convert datatype="TEXT">

        <constant>

        <dataitem>

        000300080191

        <datatype name="LONG"/>

        </dataitem>

        </constant>

        </convert>
```

Es ist möglich beliebige Datentypen nach TEXT zu konvertieren, z. B.:

Beispiel 40.

- CONVERT(DOUBLE(-300080191)) -> TEXT(„3.00080191E8")
- CONVERT(TIME(07.01.1971 00:01)) -> TEXT("7.1.1971 0:01")
- CONVERT(TIME(07.01.2000)) -> TEXT(„07.1.1971")
- CONVERT(BOOLEAN(1)) -> TEXT(„FALSE") (Alles was nicht true ist, ist false.)

Die bei der Ausgabe verwendete Skalierung entspricht der am Objekt vorgefundenen, z. B.:

Beispiel 41.

- convert(<dataitem>1 YEAR<datatype name='TIMESPAN'/></dataitem>) -> TEXT("1.0 YEAR")
- convert(<dataitem>1 YEAR<datatype name='TIMESPAN'/><scale name='MONTH'/></dataitem>) -> TEXT("12.166666666666666 MONTH") (== 365 days/30 days)

Berechnung (nur PPM4)		
	Ergebnis	Liefert den konvertierten Wert des Operanden. NULL, wenn Operand NULL liefert
	Fehler	Bei fehlgeschlagener Konvertierung

Beispiel:

```
<convert datatype="LONG">

        <filteredattribute name="AT_ABC"

        nodetype="PROCESS"/>

        </convert>
```

Ein angenommener Wert **456** für das Attribut **AT_ABC** vom Typ **TEXT** wird in den Datentyp **LONG** konvertiert.

7.1.3.4.4. Logische Operatoren

Folgende Operatoren stehen zur Verfügung: eq, eqset, lt, gt, gteq, lteq, ne, exists, filled, in, and, or, xor, not, containstext.

Gleichheit (Wert)

XML-Tag:	**eq**	
Operanden:	mindestens zwei Werte	
Synopsis:		
	<eq> <Value 1> <Value 2> ... <Value n> </eq>	
Operanden:	Alle Datentypen	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Description:	Gleichheit aller angegebenen Werte liefert TRUE.	
Berechnung (PPM3)	Ergebnis	TRUE, wenn die Operanden 1 bis n den gleichen Datentyp und Wert haben, sonst FALSE.
	Fehler	Wenn ein Operand NULL ist.
Berechnung (PPM$)	Ergebnis	TRUE, wenn die Operanden 1 bis n den gleichen Datentyp und Wert haben, sonst FALSE. NULL, wenn ein Operand NULL ist
	Fehler	Nichts

Beispiel:

```
<eq>
        <timespan type="NORMAL">
        <max>
        <attribute name="AT_CUSTDATE_WISH" nodetype="PROCESS"
        onerror="EXIT_NO_WARNING"/>
        </max>
        <min>
        <attribute name="AT_END_TIME" nodetype="OT_FUNC"
        objectname="SAP.WAUS" onerror="EXIT_NO_WARNING"/>
        </min>
        </timespan>
        <constant>
        <dataitem value="0">
        <datatype name="TIMESPAN"/>
        </dataitem>
        </constant>
        </eq>
```

Gleichheit (Wert)

XML-Tag:	**ne**	
Operanden:	mindestens zwei Werte	
Synopsis:	<ne> <Value 1> <Value 2> ... <Value n> </ne>	
Operanden:	Alle Datentypen	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Description:	Ungleichheit aller angegebenen Werte liefert TRUE	
Berechnung (PPM$)	Ergebnis	TRUE, wenn alle Operanden 1 bis n ungleich sind (z.B. Operand 1 != Operand 2), sonst FALSE. NULL, wenn ein Operand NULL ist
	Fehler	Nichts

Beispiel:

```
<ne>

        <timespan type="NORMAL">

        <max>

        <attribute name="AT_CUSTDATE_WISH" nodetype="PROCESS"
onerror="EXIT_NO_WARNING"/>

        </max>

        <min>

        <attribute name="AT_END_TIME"

        nodetype="OT_FUNC"

        objectname="SAP.WAUS"

        onerror="EXIT_NO_WARNING"/>

        </min>

        </timespan>

        <constant>

        <dataitem value="0">

        <datatype name="TIMESPAN"/>

        </dataitem>

        </constant>

        </ne>
```

Gleichheit (Wertemenge)

XML-Tag:	**eqset**
Operanden:	mindestens zwei Wertemengen

Synopsis:	`<eqset>`
	`<Set of values 1>`
	`<Set of values 2>`
	`...`
	`<Set of values n>`
	`</eqset>`

Operanden:	Alle Datentypen
Ergebnis:	Wahrheitswert
Ergebnistyp:	BOOLEAN
Description:	Gleichheit der angegebenen Wertemengen liefert TRUE. Die Operanden 2 bis n werden nacheinander mit Operand 1 verglichen.
Berechnung (PPM3)	Ergebnis TRUE, wenn die zu vergleichenden Mengen gleich groß sind und alle Werte darin gleich sind, sonst FALSE.

	Fehler	Wenn ein Operand NULL ist.
Berechnung (PPM$)	Ergebnis	TRUE, wenn die zu vergleichenden Mengen gleich groß sind und alle Objekte darin gleich sind, sonst FALSE. NULL, wenn ein Operand NULL ist
	Fehler	Nichts
Beispiel:	-	

"Kleiner als"-Vergleich

XML-Tag:	**lt**	
Operanden:	genau zwei Werte	
Synopsis:		
	<lt> <Value 1> <Value 2> </lt>	
Operanden:	Einheitlicher numerischer Datentyp	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Berechnung (PPM3)	Ergebnis	TRUE, wenn Operand 1 und Operand 2 den gleichen Datentyp haben und Operand 1 kleiner als Operand 2 ist, sonst FALSE
	Fehler	Wenn mind. ein Operand gleich NULL oder mind. ein Operand mit unzulässigem Datentyp
Berechnung (PPM$)	Ergebnis	TRUE, wenn Operand 1 und Operand 2 den gleichen Datentyp haben und Operand 1 kleiner als Operand 2 ist, sonst FALSE NULL, wenn mind. ein Operand gleich NULL ist.
	Fehler	Nur bei unzulässigem Datentyp
Beispiel:	-	

"Größer als"-Vergleich

XML-Tag:	**gt**	
Operanden:	genau zwei Werte	
Synopsis:		
	<gt> <Value 1> <Value 2> </gt>	
Operanden:	Einheitlicher numerischer Datentyp	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Berechnung (PPM3)	Ergebnis	Liefert TRUE, wenn Wert 1 größer als Wert 2 ist und die Operanden von einheitlichem Datentyp sind, sonst FALSE
	Fehler	Wenn mind. ein Operand NULL oder mind. ein Operator mit unzulässigem Datentyp

Berechnung (PPM$)	Ergebnis	Liefert TRUE, wenn Wert 1 größer als Wert 2 ist und die Operanden von einheitlichem Datentyp sind, sonst FALSE NULL, wenn mind. ein Operand gleich NULL ist.
	Fehler	Nur bei unzulässigem Datentyp
Beispiel:	-	

"Größer oder gleich"-Vergleich

XML-Tag:	**gteq**	
Operanden:	genau zwei Werte	
Synopsis:		
	<gteq> <Value 1> <Value 2> </gteq>	
Operanden:	Einheitlicher numerischer Datentyp	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Berechnung (PPM$)	Ergebnis	Liefert TRUE, wenn Wert 1 größer oder gleich Wert 2 ist und die Operanden von einheitlichem Datentyp sind, sonst FALSE. NULL, wenn mind. ein Operand gleich NULL ist.
	Fehler	Nur bei unzulässigem Datentyp
Beispiel:	-	

"Kleiner oder gleich"-Vergleich

XML-Tag:	**lteq**	
Operanden:	genau zwei Werte	
Synopsis:		
	<glteq> <Value 1> <Value 2> </lteq>	
Operanden:	Einheitlicher numerischer Datentyp	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Berechnung (PPM$)	Ergebnis	Liefert TRUE, wenn Wert 1 kleiner oder gleich Wert 2 ist und die Operanden von einheitlichem Datentyp sind, sonst FALSE. NULL, wenn mind. ein Operand gleich NULL ist.
	Fehler	Nur bei unzulässigem Datentyp
Beispiel:	-	

Existenzprüfung

XML-Tag:	exists	
Operanden:	mindestens ein Attributbezeichner (attribute, filteredattribute)	
Synopsis:		
	<exists> <Attribute 1> <Attribute 2> ... <Attribute n> </exists>	
Operanden:	Alle Datentypen	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Description:	Liefert TRUE, wenn die angegebenen Attribute vorhanden sind, unabhängig davon, ob den Attributen Werte zugewiesen ist.	
Berechnung (PPM3/ PPM4)	Ergebnis	TRUE, wenn alle angegebenen Attribute existieren, sonst FALSE
	Fehler	Nichts
Beispiel:		
	<exists> <attribute name="AT_ORDER_VOL" nodetype="OT_FUNC"/> </exists>	

Inhaltsprüfung

XML-Tag:	filled
Operanden:	mindestens ein Wert oder eine Wertemenge

Synopsis:	<filled> <Value 1> <Value 2> ... <Value n> </filled> or <filled> <Set of values 1> <Set of values 2> ... <Set of values n> </filled> or <filled> <Value 1> <Value 2> ... <Value n> <Set of values 1> <Set of values 2> ... <Set of values n> </filled>	
Operanden:	Alle Datentypen	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Berechnung (PPM3/PPM4)	Ergebnis	TRUE, wenn alle angegebenen Werte bzw. Wertemengen gepflegt sind, sonst FALSE
	Fehler	Nichts
Beispiel:	-	

Inhaltsprüfung Mengen

XML-Tag:	**in**
Operanden:	1. Operand: Wert oder Wertemenge 2. Operand: Wertemenge

Synopsis:		
	<in> <Value 1> <Set of values 2> </in> or <in> <Set of values 1> <Set of values 2> </in>	
Operanden:	Alle Datentypen	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Berechnung (nur PPM4)	Ergebnis	TRUE, wenn der Wert bzw. die Wertemenge des 1. Operanden in der durch den 2. Operanden angegebenen Wertemenge enthalten ist. NULL, wenn ein Operand NULL liefert.
	Fehler	Nur bei unverträglichen Datentypen
Beispiel:		

```
<calcattr name="AT_KI_ABL" type="PROCESS">
<calculation>
<in>
<constant>
<dataitem>
HR-ABL
<datatype name="TEXT">Text</datatype>
</dataitem>
</constant>
<attribute name="AT_HRMODUL" nodetype="OT_FUNC" />
</in>
</calculation>
</calcattr>
```

Der **in**-Operator liefert TRUE, wenn es an mindestens einer Funktion der EPK ein Attribut **AT_HRMODUL** mit dem Wert **HR-ABL** gibt.

Logisches UND

XML-Tag:	**und**
Operanden:	mindestens zwei Wahrheitswerte

Synopsis:		
	<and> <Logical value 1> <Logical value 2> ... <Logical value n> </and>	
Operanden:	BOOLEAN	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Description:	Liefert TRUE, wenn alle Wahrheitswerte TRUE sind. Beim ersten Operanden, der FALSE liefert, wird die Auswertung der Operandenliste abgebrochen und FALSE zurückgeliefert.	
Berechnung (PPM3)	Ergebnis	TRUE, wenn alle Operanden TRUE liefern, sonst FALSE
	Fehler	Wenn ein Operand NULL liefert oder wenn mind. ein Operand einen unzulässigen Datentyp hat
Berechnung (PPM$)	Ergebnis	TRUE, wenn alle Operanden TRUE liefern. FALSE, wenn ein Operand FALSE liefert und alle davor TRUE. NULL, wenn ein Operand NULL liefert und alle davor TRUE
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp (nicht BOOLEAN)
Beispiel:	-	

Logisches ODER

XML-Tag:	**or**	
Operanden:	mindestens zwei Wahrheitswerte	
Synopsis:		
	<or> <Logical value 1> <Logical value 2> ... <Logical value n> </or>	
Operanden:	BOOLEAN	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Description:	Liefert TRUE, wenn mindestens ein Wahrheitswert TRUE ist. Beim ersten Operanden, der TRUE liefert, wird die Auswertung der Operandenliste abgebrochen und TRUE zurückgeliefert.	
Berechnung (PPM3)	Ergebnis	TRUE, wenn ein Operand TRUE liefert und alle davor nicht NULL liefern, sonst FALSE
	Fehler	Wenn ein Operand NULL liefert und alle anderen Operanden davor FALSE oder wenn mind. ein Operand einen unzulässigen Datentyp hat

Berechnung (PPM$)	Ergebnis	TRUE, wenn ein Operand TRUE liefert und alle davor nicht NULL liefern. FALSE, wenn alle Operanden FALSE liefern. NULL, wenn ein Operand NULL liefert und alle davor FALSE
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp (nicht BOOLEAN)
Beispiel:	-	

Logisches EXKLUSIV ODER

XML-Tag:	**xor**	
Operanden:	mindestens zwei Wahrheitswerte	
Synopsis:		

```
<xor>
        <Logical value 1>
        <Logical value 2>
        ...
        <Logical value n>
        </xor>
```

Operanden:	BOOLEAN	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Description:	Liefert TRUE, wenn genau ein Wahrheitswert TRUE ist	
Berechnung (PPM3)	Ergebnis	TRUE, wenn genau ein Operand TRUE liefert, sonst FALSE
	Fehler	Wenn ein Operand NULL liefert oder wenn mind. ein Operand einen unzulässigen Datentyp hat
Berechnung (PPM$)	Ergebnis	TRUE, wenn genau ein Operand TRUE liefert. FALSE, wenn kein Operand oder mehr als ein Operand TRUE liefern. NULL, wenn ein Operand NULL liefert
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp (nicht BOOLEAN)
Beispiel:	-	

Logisches NICHT

XML-Tag:	**not**	
Operanden:	genau ein Wahrheitswert	
Synopsis:		

```
<not>
        <Logical value>
        </not>
```

Operanden:	BOOLEAN	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Description:	Kehrt den angegebenen Wahrheitswert um.	
Berechnung (PPM3)	Ergebnis	TRUE, wenn Operand FALSE liefert, sonst FALSE

	Fehler	Wenn Operand NULL liefert oder wenn mind. ein Operand einen unzulässigen Datentyp hat
Berechnung (PPM$)	Ergebnis	TRUE, wenn Operand FALSE liefert. FALSE, wenn Operand TRUE liefert. NULL, wenn Operand NULL liefert
	Fehler	Wenn Operand mit unzulässigem Datentyp (nicht BOOLEAN)
Beispiel:		

```
<not>
        <exists>
        <attribute name="AT_ORDER_VOL" nodetype="OT_FUNC"/>
        </exists>
        </not>
```

Prüfung auf Text in Text

XML-Tag:	**containstext**	
Operanden:	genau zwei Werte vom Typ TEXT	
Synopsis:		

```
<containstext>
         <value 1>
         <value 2>
         </containstext>
```

Operanden:	TEXT	
Ergebnis:	Wahrheitswert	
Ergebnistyp:	BOOLEAN	
Berechnung (nur PPM4)	Ergebnis	TRUE, wenn der vom 2. Operanden gelieferte Text eine Teilzeichenkette des vom 1. Operanden gelieferten Wertes ist, sonst FALSE. NULL, wenn ein Operand NULL liefert
	Fehler	Wenn mind. ein Operand mit unzulässigem Datentyp (nicht TEXT)

Beispiel:

```
<containstext>

        <filteredattribute name="AT_ABCDEF"

        nodetype="OT_FUNC" objectname="this"

        onerror="EXIT_NO_WARNING"/>

        <constant>

        <dataitem>

        abc

        <datatype name="TEXT">Text</datatype>

        </dataitem>

        </constant>

        </containstext>
```

Operator liefert **TRUE**, wenn die Zeichenkette **abc** im Wert des Attributs **AT_ABCDEF** enthalten ist.

7.1.3.4.5. Bedingungsoperator

Folgender Operator steht zur Verfügung: if - then - else

Bedingungsprüfung mit optionaler Verzweigung (else)

XML-Tag:	**if - then [- else]**	
Operanden:	genau ein Wahrheitswert	
Synopsis:		

```
<if>

        <Logical value>

        </if>

        <then>

        <Value>

        </then>

        <else>

        <Value>

        </else>
```

Operanden:	BOOLEAN	
Ergebnis:	Wahrheitswert - Wert [- Wert]	
Ergebnistyp:	BOOLEAN - Datentyp des Operanden [- Datentyp des Operanden]	
Berechnung (PPM3/PPM4)	Ergebnis	Wert des 2. Operanden, wenn der 1. Operand TRUE liefert. Wert des 3. Operanden, wenn der 1. Operand FALSE liefert. NULL, wenn 1. Operand FALSE und 3. Operand nicht definiert ist.

	Fehler	Wenn 1. Operand nicht vom Datentyp BOOLEAN
Beispiel:		

```
<if>
        <exists>
        <filteredattribute name="AT_OS" nodetype="OT_FUNC"
        objectname="SAP.WAUS filter="LATEST"/>
        </exists>
        <then>
        <filteredattribute name="AT_CT" nodetype="OT_FUNC"
        filter="EARLY"/>
        </then>
        <else>
        <filteredattribute name="AT_KT" nodetype="OT_FUNC"
        filter="LATEST"/>
        </else>
        </if>
```

Wenn die Bedingung zutrifft (d.h., das Attribut **AT_OS** ist an mindestens einer Funktion mit dem internen Namen **SAP.WAUS** gepflegt), wird der Wert des Attributs **AT_CT** weitergegeben. Wenn die Bedingung nicht zutrifft, wird für die weitere Berechnung der Wert des Attributs **AT_KT** verwendet.

7.1.3.4.6. Kettenoperatoren

Die folgende Operatoren stehen Ihnen zur Verfügung: concat, subtext, indexof.

Zusammenfügen von Ketten

XML-Tag:	**Concat**	
Operanden:	genau eine Wertemenge (Ketten)	
Synopsis:		

```
<concat>
<Value set 1>
</concat>
```

Operanden:	TEXT	
Ergebnis:	Wert (Kette)	
Ergebnistyp:	TEXT	
Berechnung (PPM3/PPM4)	Ergebnis	Ergebnis der Konkatenation aller in der Wertemenge enthaltenen Ketten. NULL, wenn Operand Leermenge
	Fehler	Wenn mind. ein Wert von unzulässigem Datentyp
Beispiel:	-	

Extraktion von Teilzeichenketten

XML-Tag:	Subtext	
Operanden:	Genau ein Wert (Kette)	
Synopsis:		

```
<subtext beginindex="..." [endindex=" "]>
<Value>
</subtext>
Example
<subtext>
<Value>
<beginindex>...</beginindex>
<endindex>...</endindex>
</subtext>
```

Verwenden Sie immer nur eine der beiden gezeigten Varianten (Index-Angabe entweder als XML-Attribute oder als XML-Elemente)

Operanden:	TEXT	
Ergebnis:	Wert (extrahierte Zeichenkette)	
Ergebnistyp:	TEXT	
Description:	Extrahiert aus einer Teilzeichenkette durch Angabe positiver Indizes (vom Stringanfang) bzw. negativer Indizes (vom Stringende).	
Berechnung (PPM3)	Ergebnis	Liefert eine Teilzeichenkette der vom Operanden übergebenen Zeichenkette. NULL, wenn angegebene Indizes ungültig
	Fehler	Wenn Operand NULL oder von unzulässigem Datentyp
Berechnung (PPM$)	Ergebnis	Liefert eine Teilzeichenkette der vom Operanden übergebenen Zeichenkette. NULL, wenn angegebene Indizes ungültig oder Operand NULL
	Fehler	Bei unzulässigem Datentyp (nicht TEXT)
Beispiel:		

```
<subtext beginindex="-3" endindex="-1">

        <filteredattribute name="AT_XYZ" nodetype="OT_FUNC"/>

        </subtext>
Example
<subtext>

        <beginindex>-3</beginindex>

        <endindex>-1</endindex>

        <filteredattribute name="AT_XYZ" nodetype="OT_FUNC"/>

        </subtext>
```

Bei einem angenommenen Wert **ABCDE** des Attributs **AT_XYZ** wird die Teilzeichenkette **CD** extrahiert.

Position eines Teiltextes in einem anderen Text bestimmen

XML-Tag	<indexof>
Beschreibung	Liefert den Index in einem Text (Operand 1) an dem das erste Vorkommen eines Teiltextes (Operand 2) steht, beginnend an einem angegebenen Index (Operand 3).
Operanden	1 / TEXT
(Position/	2 / TEXT
Datentyp)	[3 / LONG], optional, Standardwert ist 0

Berechnung	ab PPM 9.0	
Berechnung (PPM4)	Ergebnis	-1, wenn mind. 1 Operand gleich NULL oder der Teilstring nicht gefunden wird Index des ersten Vorkommen des durch Operand 2 bestimmten Textes im durch Operand 1 bestimmten Text beginnend ab dem in Operand 3 angegebenen Index (äquivalent zu Java String.indexOf(String, int)).
	Datentyp	LONG
	Exception	Wenn mind. 1 Operand mit falschem Datentyp oder Operand 3 < 0
Beispiel		

```
<subtext mode="PPM4">
<filteredattribute name="AT_TEXT" … />
<beginindex>
<indexof>
<filteredattribute name="AT_TEXT" … />
<constant>
<dataitem>
XYZ
<datatypename="TEXT">Text</datatype>
</dataitem>
</constant>
</beginindex>
</subtext>
```

In dem Beispiel werden die Operatoren **indexof** und **subtext** zusammen verwendet. Hier ermittelt indexof die Position des Textes **XYZ** im Attributwert des Attributs **AT_TEXT**. Anschließend erkennt der Operator **subtext** den Teilstring ab dieser Position.

Wenn das Attribut **AT_TEXT** den Wert **ABCDEXYZAC** hat würde obige Berechnungsvorschrift den Wert **XYZAC** liefern.

7.1.3.4.7. Zeitoperatoren

Folgende Operatoren stehen zur Verfügung: createday, createtimeofday, createtimestamp, addtimespan, addfactorytimespan und weekday.

Format-Umwandlung (Datum)

XML-Tag:	**createday**	
Operanden:	genau ein Wert	
Synopsis:		

```
<createday>
        <Value>
        </createday>
```

Operanden:	TIME (TIMESTAMP, DATE)	
Ergebnis:	Wert (Datum im Format **dd.MM.yyyy**)	
Ergebnistyp:	DAY	
Description:	Extrahiert ein Datum aus einem PPM-Zeitstempel	
Berechnung (PPM3)	Ergebnis	Datum, das vom Operanden geliefert wird
	Fehler	Wenn Operand von unzulässigem Datentyp oder gleich NULL

Berechnung (PPM$)	Ergebnis	Datum, das vom Operand NULL geliefert wird, wenn der Operand gleich NULL ist
	Fehler	Wenn Operand von unzulässigem Datentyp

Beispiel:

```
<calcattr name="AT_DAY" type="PROCESS">
        <calculation>
        <createday>
        <filteredattribute name="AT_TIME"
        nodetype="OT_FUNC"
        objectname="this" filter="EARLY"/>
        </createday>
        </calculation>
        </calcattr>
```

Format-Umwandlung (Uhrzeit)

XML-Tag:	**createtimeofday**	
Operanden:	genau ein Wert	
Synopsis:		

```
<createtimeofday>
        <Value>
        </createtimeofday>
```

Operanden:	TIME (TIMESTAMP, DATE)	
Ergebnis:	Wert (Uhrzeit im Format **hh:mm:ss**)	
Ergebnistyp:	TIMEOFDAY	
Description:	Extrahiert die Uhrzeit aus einem PPM-Zeitstempel	
Berechnung (PPM3)	Ergebnis	Uhrzeit, die vom Operanden definiert wird
	Fehler	Wenn Operand von unzulässigem Datentyp oder gleich NULL
Berechnung (PPM$)	Ergebnis	Uhrzeit, die vom Operanden NULL definiert wird, wenn der Operand NULL liefert
	Fehler	Wenn Operand von unzulässigem Datentyp

Beispiel:

```
<calcattr name="AT_DAY" type="PROCESS">
        <calculation>
        <createtimeofday>
        <filteredattribute name="AT_TIME" nodetype="OT_FUNC"
        objectname="this" filter="EARLY"/>
        </createtimeofday>
        </calculation>
        </calcattr>
```

Format-Umwandlung (Zeitstempel)

XML-Tag:	**createtimestamp**	
Operanden:	ein oder zwei Werte (**Datum** oder **Datum und Uhrzeit**)	
Synopsis:		

```
<createtimestamp>
        <Date>
        <Time> [optional]
        </createtimestamp>
```

Operanden:	DAY, TIMEOFDAY	
Ergebnis:	Wert (Zeitstempel im Format **dd.MM.yyyy hh:mm:ss**)	
Ergebnistyp:	TIME (TIMESTAMP, DATE)	
Description:	Erzeugt einen PPM-Zeitstempel aus einem Datum bzw. aus einem Datum und einer Uhrzeit	
Berechnung (PPM3)	Ergebnis	Zeitstempel, der durch die Operanden definiert wird
	Fehler	Wenn mind. ein Operand von unzulässigem Datentyp oder gleich NULL
Berechnung (PPM$)	Ergebnis	Zeitstempel, der durch die Operanden definiert wird. NULL, wenn Operand vom Typ DAY NULL liefert oder wenn erster Operand vom Datentyp TIMEOFDAY und zweiter Operand NULL liefern
	Fehler	Wenn mind. ein Operand von unzulässigem Datentyp

Beispiel:

```
<createtimestamp>
        <constant>
        <dataitem>
        <datatype name="DAY">
        25.01.2004
        </datatype>
        </dataitem>
        </constant>
        </createtimestamp>
```

Erzeugt den Zeitstempel **25.01.2004 00:00:00**.

Addition einer Zeitspanne

XML-Tag:	**addtimespan**	
Operanden:	genau zwei Werte (Zeitstempel und Zeitspanne, Datum und Zeitspanne oder Uhrzeit und Zeitspanne)	
Synopsis:		

```
<addtimespan>
        <Time stamp or date or time>
        <Time span>
        <addtimespan>
```

Operanden:	Operand 1: TIME (TIMESTAMP, DATE) oder DAY oder TIMEOFDAY Operand 2: TIMESPAN	
Ergebnis:	Wert (Zeitstempel im Format **dd.MM.yyyy hh:mm:ss**)	
Ergebnistyp:	Zeitpunkt: TIME (TIMESTAMP, DATE) oder DAY oder TIMEOFDAY	
Description:	Addiert zu einem PPM-Zeitstempel eine Zeitspanne in der Basisskalierung (SECOND). Das Ergebnis ist ein Zeitstempel.	
Berechnung (PPM3)	Ergebnis	Zeitpunkt, der sich aus der Addition der angegebenen Zeitspanne (Operand 2) zum angegebenen Zeitpunkt (Operand 1) ergibt
	Fehler	Wenn mind. ein Operand gleich NULL oder mind. ein Operand von unzulässigem Datentyp
Berechnung (PPM$)	Ergebnis	NULL, wenn mind. ein Operand NULL; Zeitpunkt, der sich aus der Addition der angegebenen Zeitspanne (Operand 2) zum angegebenen Zeitpunkt (Operand 1) ergibt
	Fehler	Bei unzulässigem Datentyp

Beispiel:

```
<calcattr name="AT_NTOFD" type="PROCESS">
<calculation>
<addtimespan>
<constant>
<dataitem>
<datatype name="TIMEOFDAY">

        08:35:41

        </datatype>

        </dataitem>

        </constant>

        <constant>

        <dataitem>

        <datatype name="TIMESPAN">

        -30 MINUTE

        </datatype>
</dataitem>
</constant>
</addtimespan>
</calculation>
</calcattr>
```

Zur angegebenen Uhrzeit wird eine negative Zeitspanne von dreißig Minuten addiert. Der Ergebniswert **08:05:41** wird im Zielattribut **AT_NTOFD** gespeichert.

Addition einer Zeitspanne mit Werkskalender

Addiert zu einem PPM-Zeitstempel eine Werkskalenderzeitspanne. Das Ergebnis ist ein Zeitstempel. Konfiguration und Verwendung des Operators **addfactorytimespan** sind ähnlich zu **addtimespan**. Bei der Berechnung wird von einem Startzeitpunkt ausgehend die angegebene Werkskalenderzeitspanne addiert. Standardmäßig unterstützt der Operator nur die Addition positiver Werkskalenderzeitspannen. Wenn Sie auch Zeitpunkte in der Vergangenheit berechnen möchten, können Sie durch Angabe des optionalen XML-Attributs **negfactorytimespan="TRUE"** auch nicht positive Zeitspannen addieren (Standardwert: **FALSE**). Liegt der berechnete Zeitpunkt genau auf einer Arbeitszeitgrenze, liefert der Operator den frühest möglichen Zeitpunkt.

Beispiel 42.

Unter der vereinfachten Voraussetzung einer täglichen Arbeitszeit von 9:00-17:00 Uhr:

- addFactoryTimeSpan("01.12.2011 12:00:00", "8 FACTORY_HOUR") = "02.12.2011 12:00:00"
- addFactoryTimeSpan("01.12.2011 12:00:00", "5 FACTORY_HOUR") = "01.12.2011 17:00:00"
- addFactoryTimeSpan{negfactorytimespan="TRUE"}("02.12.2011 12:00:00", "-8 FACTORY_HOUR") = "01.12.2011 12:00:00"
- addFactoryTimeSpan{negfactorytimespan="TRUE"}("02.12.2011 12:00:00", "-3 FACTORY_HOUR") = "01.12.2011 05:00:00 PM"

Wenn Sie einen anderen als den Standardwerkskalender (**factorycalendar.xml**) verwenden möchten, können Sie optional eine XML-Datei angeben, die den zu verwendenden Werkskalender enthält. Im XML-Attribut **attributename** geben Sie das Funktions- bzw. Prozessinstanzattribut an, das den Namen der zu verwendenden XML-Werkskalenderdatei bestimmt. Das Attribut muss an der jeweiligen Funktions- bzw. Prozessinstanz gepflegt sein, für die die Berechnung durchgeführt wird. Im XML-Attribut **directory** geben Sie das Verzeichnis an, in dem die angegebene Werkskalenderdatei gesucht wird. Die beiden XML-Attribute **attributename** und **directory** müssen immer gemeinsam angegeben werden.

Das entsprechende Verzeichnis mit der zu verwendenden Werkskalenderdatei geben Sie relativ zum PPM-Datenverzeichnis an. Das PPM-Datenverzeichnis **data_ppm** liegt unter <PPM-Installationsverzeichnis>\ppm\server\bin\work\.

```
...
<addfactorytimespan directory="calc\fc" attributename="AT_FC_NAME">
...
```

Wenn das Attribut **AT_FC_NAME** den Wert **myFactoryCalendar.xml** enthält, wird zur Berechnung der in der Datei **myFactoryCalendar.xml** definierte Werkskalender verwendet. Die Datei liegt im Verzeichnis **<PPM-Installationsverzeichnis>\ppm\server\bin\work\data_ppm\calc\fc**.

Die Addition der Werkskalenderzeitspannen erfolgt immer in der Basiseinheit **Personensekunde**. Die hierfür verwendeten Umrechnungsfaktoren sind unabhängig vom Werkskalender und werden in der mandantenspezifischen Konfigurationsdatei **transformationfactors.xml** definiert. Wenn Sie diese nicht verwenden möchten, dürfen Sie zur Addition einer Zeitspanne unter Berücksichtigung eines Werkskalenders nur Werkskalenderzeitspannen in der Einheit Personensekunde, -minute oder -stunde verwenden.

Wochentagermittlung (aus Datum)

XML-Tag:	**weekday**	
Operanden:	genau ein Wert	
Synopsis:		

	`<weekday>`
	`<Value>`
	`</weekday>`	

Operanden:	Genau ein Operand: TIME oder DAY	
Ergebnis:	Zeichenkette im Format **MO**, **TU**, **WE**, **TH**, **FR**, **SA** bzw. **SU**	
Ergebnistyp:	TEXT	
Description:	Ermittelt den Wochentag aus einem PPM-Datumstyp und liefert ihn als Zeichenkette zurück.	
Berechnung (PPM$)	Ergebnis	Eine der Konstanten MO, TU, WE, TH, FR, SA oder SU, entsprechend dem Wochentag des übergebenen Datums
	Fehler	Wenn Operand von unzulässigem Datentyp bzw. unzulässige Anzahl von Operanden

Beispiel:

```
<calcattr name="AT_WEEKDAY" type="PROCESS">
        <calculation>
        <weekday>
        <constant>
        <dataitem>
        <datatype name="DAY">
        25.08.2007
        </datatype>
        </dataitem>
        </constant>
        </weekday>
        </calculation>
        </calcattr>
```

Ermittelt den Wochentag zum angegebenen Datum ('saturday') und liefert ihn als TEXT-Zeichenkette **SA** zurück.

Werte von Textdimensionen, die Ergebnisse des Operators **weekday** verwenden, können nicht sortiert werden.

7.1.3.4.8. Bedingte Attributtypberechnung

Die bedingte Berechnung von Attributtypen ermöglicht die Steuerung der Attributtypberechnung. Die Steuerung basiert dabei auf der Existenzprüfung von Attributtypen oder dem Ergebnis von Vergleichen. Die Existenzprüfung unterscheidet die beiden Fälle **Attributtyp vorhanden** (XML-Element **exists**) und **Attributtyp gepflegt** (XML-Element **filled**).

Im folgenden Beispiel liefert das Berechnungselement **<if>** den Wert **Null**, wenn es in der Prozessinstanz kein Funktionsattribut **AT_B** gibt. Die vom XML-Element **attribute** erzeugte Ergebnismenge ist in diesem Fall leer.

Existiert mindestens ein Attribut **AT_B** an einer beliebigen Funktion der Prozessinstanz, wird der Wert des XML-Elements **filteredattribute** weitergegeben.

```
<if>
<exists>
<attribute name="AT_B" nodetype="OT_FUNC"/>
</exists>
<then>
<filteredattribute name="AT_C" nodetype="OT_FUNC" filter="EARLY"/>
</then>
</if>
```

Durch Verknüpfen von Bedingungen mittels logischer Operatoren lassen sich auch komplexere Bedingungen formulieren. Das gezeigte Beispiel soll so erweitert werden, dass auch auf einen vorhandenen Attributwert getestet wird.

```
<if>
<and>
<exists>
```

```
<attribute name="AT_B" nodetype="OT_FUNC"/>
</exists>
<filled>
<attribute name="AT_B" nodetype="OT_FUNC"/>
</filled>
</and>
<then>
<filteredattribute name="AT_C" nodetype="OT_FUNC" filter="EARLY"/>
</then>
</if>
```

Da ein existierender Attributwert die Existenz des entsprechenden Attributtyps voraussetzt, kann die Prüfung auf Existenz entfallen und die Bedingung optimiert werden.

Im folgenden Beispiel liefert das Berechnungselement **<if>** den Wert **NULL**, wenn an keiner der vorkommenden Funktionen ein Attributtyp mit dem Bezeichner **AT_G** gepflegt ist.

```
<if>
<exists>
<filteredattribute name="AT_G" nodetype="OT_FUNC" filter="LATEST"/>
</exists>
</if>
```

7.1.3.5. Verschachtelung von Operatoren

Operatoren können in beliebiger Tiefe verschachtelt werden. Wenn Sie Operatoren kombinieren, müssen Sie die in der DTD vorgegebenen Regeln einhalten.

> **Warnung**
> Berechnungsvorschriften, die auf unzulässigen Verschachtelungen von Operatoren beruhen, führen beim Einlesen der Kennzahlenkonfiguration zum Abbruch des Imports. Aufgrund der komplexen Abhängigkeiten können fehlerhafte Berechnungsvorschriften unter Umständen zur völligen Unbrauchbarkeit des Datenbankinhalts führen.

In der Datei **KeyindicatorConfiguration.dtd** im Verzeichnis **dtd** Ihrer PPM-Installation können Sie überprüfen, welche Verschachtelungen der Operatoren untereinander erlaubt sind.

Beispiel (Auszug aus der DTD):

```
<!ELEMENT abs (%numericoperator; | %setoperator; | %caseoperator; |
filteredattribute | constant)>
```

Der Operator **<abs>** kann mit einem der XML-Elemente **<filteredattribute>** bzw. **<constant>** oder mit einem Operator der angegebenen Entitäten (deklarierte Einheiten in XML-Notation, denen bestimmte XML-Elemente zugeordnet sind) verschachtelt werden:

- % numericoperator (Einheit aller mathematischen Operatoren)
- % setoperator (Einheit aller Wert erzeugenden Operatoren)
- % caseoperator (Einheit aller Bedingungsoperatoren)

Welche Operatoren welcher Entität zugeordnet sind, können Sie der Deklaration der Entität entnehmen.

Beispiel (Auszug aus der DTD):

```
<!ENTITY % setoperator "sum|product|card|min|max|mean">
```

Die Entität **% setoperator** steht für einen der Operatoren **<sum>**, **<product>**, **<card>**, **<min>**, **<max>** oder **<mean>**.

Das folgende Beispiel zeigt eine DTD-konforme Berechnungsvorschrift:

```
<calcattr name="..." type="...">
<calculation>
<abs>
<minus>
<filteredattribute name="AT_KI_BSP1" nodetype="OT_FUNC" objectname="this"
filter="LATEST" onerror="EXIT_NO_WARNING"/>
<filteredattribute name="AT_KI_BSP2" nodetype="OT_FUNC" objectname="this"
filter="EARLY" onerror="EXIT_NO_WARNING"/>
</minus>
</abs>
</calculation>
</calcattr>
```

7.1.3.6. Berechnungsfunktionen

Definieren Sie komplexe Teilberechnungen von Berechnungsvorschriften, die in mehreren Attributberechnungen verwendet werden, als Berechnungsfunktionen. Eine Berechnungsfunktion wird über den Aufruf **usefunction** in der Berechnungsvorschrift einer Attributberechnung oder Berechnungsfunktion verwendet.

> **Warnung**
> Vermeiden Sie beim Aufruf von Berechnungsfunktionen aus anderen Berechnungsfunktionen zyklische Abhängigkeiten. Der Import einer solchen Kennzahlenkonfiguration wird mit Ausgabe einer Fehlermeldung abgebrochen.

XML-Tag	Beschreibung
function name	Interner Name der Berechnungsfunktion. Wird im Funktionsaufruf referenziert.
Resulttype	Ergebnistyp (für die Verwendung mit anderen Operatoren). Zulässige Werte: Wert (**VALUE**) Wertemenge (**VALUELIST**) Wahrheitswert (**BOOLEAN**)
Datatype	Datentyp des Berechnungsergebnisses
Usefunction	Funktionsaufruf

Sowohl bei der Definition als auch beim Aufruf einer Berechnungsfunktion müssen Ergebnistyp (**resulttype**) und Datentyp (**datatype**) mit angegeben werden.

Beispiel 43.

Das folgende Beispiel zeigt die Definition der Berechnungsfunktion getPrincipal, die als Ergebnis einen Wert vom Datentyp **TEXT** zurückliefert.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM "KeyindicatorConfiguration.dtd">
<keyindicatorconfig>
...
<function name="getPrincipal" resulttype="VALUE" datatype="TEXT">
<if>
<exists>
<attribute name="AT_PRINCIPAL_NAME" nodetype="PROCESS"/>
</exists>
<then>
<max>
<attribute name="AT_PRINCIPAL_NAME" nodetype="PROCESS"/>
</max>
</then>
<else>
<max>
<attribute name="AT_PRINCIPAL_ID" nodetype="PROCESS"/>
</max>
</else>
</if>
</function>
...
</keyindicatorconfig>
```

Beispiel 44.

Aufruf der Berechnungsfunktion

Die zuvor definierte Berechnungsfunktion getPrincipal wird in der Berechnungsvorschrift des Attributs **AT_EXP** mit **usefunction** aufgerufen. Zu beachten ist, dass der Ergebnistyp der Berechnungsfunktion zum verarbeitenden Operator passt. Im Beispiel wird syntaktisch korrekt der Ergebnistyp **VALUE** (Wert) mit dem Werte verarbeitenden Operator **eq** kombiniert.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM "KeyindicatorConfiguration.dtd">
<keyindicatorconfig>
...
<calcattr name="AT_EXP" type="PROCESS">
<calculation>
<if>
<eq>
<usefunction name="getPrincipal" resulttype="VALUE" datatype="TEXT"/>
<constant>
<dataitem>
KTD
<datatype name="TEXT">Text</datatype>
</dataitem>
</constant>
</eq>
```

```
<then>
...
</then>
...
</if>
</calculation>
</calcattr>
...
</keyindicatorconfig>
```

Erstellen Sie Berechnungsfunktionen mit Hilfe des PPM Customizing Toolkit. Im Menü **Berechnete Attribute** des Moduls **Kennzahlen und Dimensionen** rufen Sie den Dialog zum Anlegen, Bearbeiten oder Löschen von Berechnungsfunktionen über die Schaltfläche **Berechnungsfunktionen konfigurieren** auf. Wenn Berechnungsfunktionen im System gepflegt sind, sind diese im Dialog **Berechnungsvorschrift definieren** sowohl für die Definition weiterer Berechnungsfunktionen als auch für die Definition von Attributberechnungen verfügbar.

7.1.3.7. Wandlung des Attributtyps

Mathematische Berechnungen werden intern mit dem Datentyp **DOUBLE** ausgeführt. Die arithmetische Verknüpfung beliebiger numerischer Datentypen wird korrekt berechnet und anschließend in den Datentyp des Ergebnisattributtyps umgewandelt.

Auch die Verknüpfung eines Zeitspannenattributtyps mit einem Kostenattributtyp wird numerisch korrekt durchgeführt. Die Einheit des Ergebnisses ist hierbei durch die Basiseinheit des Ergebnisattributtyps vorgegeben.

7.1.3.8. Zusammenfassung

Ein durch **calcattr** neu berechneter Attributtyp enthält den Ergebniswert in der Basiseinheit.

Eine angegebene Berechnungsvorschrift wird nur dann ausgeführt, wenn der spezifizierte Attributtyp in der Definition einer Kennzahl oder Dimension (XML-Attribut **attrname**) als zu berechnendes Attribut angegeben ist (**calculated=TRUE**).

Wird als Knotentyp (**nodetype**) **PROCESS** angegeben, wird der spezifizierte Attributtyp genau einmal berechnet und an die Prozessinstanz kopiert.

Wird als Knotentyp einer Berechnungsvorschrift **OT_FUNC** angegeben, wird der spezifizierte Attributtyp für jede Funktion der Prozessinstanz berechnet. Dabei wird er an jede Funktion kopiert.

Innerhalb einer Berechnungsvorschrift (**calculation**) kann auf beliebige existierende Attributtypen Bezug genommen werden. Soll bei dieser Berechnung auf einen Attributtyp derjenigen Funktion zugegriffen werden, für die die Berechnung gerade ausgeführt wird, wird als Objektname **this** verwendet.

7.1.3.9. Beispiele Attributberechnungen

Beispiel 45.

Die Kennzahl Wunschliefertermintreue vergleicht den tatsächlichen Liefertermin (Endzeitpunkt der Funktion **SAP.WAUS** einer Prozessinstanz) mit einer aus dem Quellsystem eingelesenen Vorgabe. Liegt der

tatsächliche Liefertermin vor dem Vorgabewert, ist der Kennzahlenwert **0**. Der Wert **0** wird als termingetreue Lieferung interpretiert. Anderenfalls gibt der Kennzahlenwert die zeitliche Abweichung vom Vorgabewert an. Der Vorgabewert ist im Prozessinstanzattribut **AT_CUSTDATE_WISH** gespeichert. Für den Fall, dass die Funktion **SAP.WAUS** mehrfach in der Prozessinstanz vorkommt, wird der früheste Wert ermittelt.

```
...
<!-Delivery performance -->
<calcattr name="AT_KI_WLFTREU" type="PROCESS">
<calculation>
<max>
<set>
<constant>
<dataitem value="0 SECOND">
<datatype name="TIMESPAN"/>
</dataitem>
</constant>
<timespan>
<max>
<attribute name="AT_CUSTDATE_WISH" nodetype="PROCESS"/>
</max>
<min>
<attribute name="AT_END_TIME" nodetype="OT_FUNC" objectname="SAP.WAUS"/>
</min>
</timespan>
</set>
</max>
</calculation>
</calcattr>
...
```

Beispiel 46.

Die Kennzahl erhält als Wert das Maximum (**max**) einer Wertemenge (**set**). Die Wertemenge enthält die Elemente **0** (**constant**) und die zeitliche Differenz zwischen tatsächlichem Liefertermin und Wunschliefertermin (**timespan**). Da das XML-Element **attribute** eine Wertemenge bildet, muss mit geeigneten Operatoren zunächst ein Attributwert zur weiteren Berechnung bestimmt werden. Bei Ermittlung des Attributwertes der Funktion **SAP.WAUS** wird durch Verwendung des Operators **min** zusätzlich der früheste tatsächliche Liefertermin bestimmt. Die mit **set** erzeugte Wertemenge erhält 2 Elemente: {**0**, (**Wunschtermin - Liefertermin**)}. Bei Ermittlung des Maximums der Wertemenge wird bei negativer Zeitspanne **0** zurückgegeben und bei positiver Zeitspanne die Abweichung des Endzeitpunkts der Funktion **SAP.AUS** vom Prozessattribut **AT_CUSTDATE_WISH** in Sekunden (Basiseinheit des Datentyps **timespan**).

Da dieses neue Attribut ein Prozessinstanzattribut ist, wird es für jede Prozessinstanz genau einmal berechnet. Folgende Ergebnisse der Berechnung können auftreten:

Das neue Prozessinstanzattribut erhält die berechnete, positive Zeitspanne in der Einheit **Sekunden**. Ist die berechnete Zeitspanne negativ, erhält das neue Prozessinstanzattribut die Zeitspanne **0 Sekunden**.

Das neue Prozessinstanzattribut wird nicht an die Prozessinstanz geschrieben, wenn die Berechnung aus einem oder mehreren der folgenden Gründe fehlschlägt und kein Standardwert angegeben ist:

- Das Attribut **AT_CUSTDATE_WISH** existiert nicht an der Prozessinstanz.
- Es gibt keine Funktion **SAP.WAUS** an der Prozessinstanz.
- Das Attribut **AT_END_TIME** existiert nicht an der Funktion **SAP.WAUS**.

Beispiel 47.

An jeder Funktion einer Prozessinstanz soll das Attribut **AT_KI_COMPETENCE** angeben, ob die Werte der Attribute **AT_COMPETENCE** und **AT_CREDIT_AMOUNT** einer Funktion übereinstimmen. Bei Übereinstimmung soll das Attribut den Wert **1** haben, anderenfalls den Wert **0**.

```
<calcattr name="AT_KI_COMPETENCE" type="OT_FUNC">
<calculation>
<if>
<eq>
<min>
<attribute name="AT_COMPETENCE" nodetype="OT_FUNC" objectname="this"/>
</min>
<max>
<attribute name="AT_CREDIT_AMOUNT" nodetype="OT_FUNC" objectname="this"/>
</max>
</eq>
<then>
<constant>
<dataitem value="1">
<datatype name="DOUBLE"/>
</dataitem>
</constant>
</then>
<else>
<constant>
<dataitem value="0">
<datatype name="DOUBLE"/>
</dataitem>
</constant>
</else>
</if>
</calculation>
</calcattr>
```

Beispiel 48.

Die Angabe des Knotentyps **OT_FUNC** und das Fehlen eines Objektnamens führen dazu, dass das berechnete Attribut **AT_KI_COMPETENCE** an alle Funktionen der Prozessinstanz geschrieben wird. Bei der Berechnung des Attributes bewirkt die Angabe des Objektnamens **this**, dass jede Funktion auf ihre eigenen Attribute zugreift. Die umklammernden Operatoren **min** und **max** liefern in diesem Fall den Wert des referenzierten Attributes, da durch den Objektnamen **this** eine Attributmenge mit nur einem Element entsteht.

Standardmäßig wird als Startzeit einer Prozessinstanz der früheste Startzeitpunkt einer Funktion der Prozessinstanz verwendet:

```
<calcattr name="AT_START_TIME" type="PROCESS">
<calculation>
<min>
<attribute name="AT_START_TIME" nodetype="OT_FUNC"/>
</min>
</calculation>
```

```
</calcattr>
```

Beispiel 49.

Falls nur die Startzeitpunkte bestimmter Funktionen verwendet werden sollen, müssen diese Funktionen auf ein bestimmtes Kriterium überprüft werden. Im folgenden Beispiel wird das Hilfsattribut **AT_TEMP_TIME** dazu genutzt, das Kriterium **Auftragsart** "**Sofortauftrag**" (Funktionsattributwert **AT_AUFTRAGSART**) zu filtern. Der eigentliche Startzeitpunkt der Prozessinstanz wird anschließend aus den gefilterten Startzeitpunkten der Funktionen ermittelt.

```
<calcattr name="AT_TEMP_TIME" type="OT_FUNC">
<calculation>
<if>
<eq>
<filteredattribute name="AT_AUFTRAGSART" nodetype="OT_FUNC" objectname="this"
filter="EARLY"/>
<constant>
<dataitem>
Rush order
<datatype name="TEXT"/>
</dataitem>
</constant>
</eq>
<then>
<filteredattribute name="AT_START_TIME" nodetype="OT_FUNC" objectname="this"
filter="EARLY"/>
</then>
</if>
</calculation>
</calcattr>
<calcattr name="AT_START_TIME" type="PROCESS">
<depends attrname="AT_TEMP_TIME" nodetype="OT_FUNC">
<calculation>
<min>
<attribute name="AT_TEMP_TIME" nodetype="OT_FUNC"/>
</min>
</calculation>
</calcattr>
```

Beispiel 50.

Im Attribut **AT_KI_AUFTR_GRUPPE** soll die Auftragsgruppe als Funktionsattribut gespeichert werden. Die Auftragsgruppe ergibt sich aus den beiden ersten Zeichen der Auftragsnummer (**AT_AUFTNR**). Zur Auftragsnummer **40268755** gehört z. B. die Auftragsgruppe **40**.

Der Operator **subtext** extrahiert aus der Zeichenfolge **40268755** des Funktionsattributs **AT_AUFTNR** die Zeichenfolge **40**:

```
<calcattr name="AT_KI_AUFTR_GRUPPE" type="OT_FUNC">
<calculation>
<subtext beginindex="0" endindex="2">
<filteredattribute name="AT_AUFTNR" nodetype="OT_FUNC" objectname="this"
```

```
filter="EARLY"/>
</subtext>
</calculation>
</calcattr>
```

XML-Attribut	Beschreibung
beginindex	Startindex (inklusiv, beginnend mit **0**)
endindex (optional)	Endindex (exklusiv)

Wird kein Endindex angegeben, beginnt die Ergebniszeichenfolge am angegebenen Startindex und endet am Ende der Quellzeichenfolge.

Warnung
Der Operator **subtext** kann nur auf Attribute und Konstanten vom Datentyp **TEXT** angewendet werden. Wenn Sie ihn auf eine Zeichenfolge anwenden, die weniger als die in **beginindex** bzw. **endindex** angegebene Zeichenanzahl enthält, liefert der Operator den Wert **NULL** zurück.

Beispiel 51.

Aus dem Zeitstempel **07.04.2003 17:30:58** wird das Datum **07.04.2003** extrahiert und als Wert des Attributs **AT_CALEN_DAY** an alle Funktionen der Prozessinstanz geschrieben.

```
<calcattr name="AT_CALEN_DAY" type="OT_FUNC">
<calculation>
<createday>

  <constant>
<dataitem value="07.04.2003 17:30:58">
<datatype name="TIME"/>
</dataitem>
</constant>
</createday>
</calculation>
</calcattr>
```

Beispiel 52.

Zu dem Zeitstempel **22.01.2002 14:55:21** wird eine Zeitspanne von einer Stunde (in der Basiseinheit **3600** Sekunden) hinzu addiert und als Zeitstempelwert **22.01.2002 15:55:21** des Attributs **AT_ADD_TSP** an alle Funktionen der Prozessinstanz kopiert.

```
<calcattr name="AT_ADD_TSP" type="OT_FUNC">
<calculation>
<addtimespan>
<!-- Time stamp -->
<constant>
<dataitem value="22.01.2002 14:55:21">
```

```
<datatype name="TIME"/>
</dataitem>
</constant>
<!— Time span 3600 seconds -->
<constant>
<dataitem value="3600">
<datatype name="TIMESPAN"/>
</dataitem>
</constant>
</addtimespan>
</calculation>
</calcattr>
```

7.1.3.10. Besonderheiten der Attributberechnung

7.1.3.10.1. Funktionsattribut AT_INTERNAL_NO_CUBE_ENTRY

Für bestimmte Funktionen können Sie angeben, dass diese nicht im Funktions-Cube gespeichert werden sollen. Wenn an einer Funktion das Attribut **AT_INTERNAL_NO_CUBE_ENTRY** existiert und den Wert **true** hat, wird diese Funktionsinstanz nicht in die Cube-Tabelle geschrieben. Auf die Kennzahlenberechnung dieser Funktion hat die Existenz des Attributes keinen Einfluss, das heißt Sie können das Attribut auch durch eine Berechnungsvorschrift an der Funktionsinstanz erzeugen.

Die Auswertung des Attributes erfolgt instanzweise, das heißt fehlt das Attribut an einzelnen Funktionsinstanzen (oder Atrributwert ungleich **true**), werden diese im Funktions-Cube gespeichert. Wenn Sie diese Funktionsinstanzen jedoch mit existierendem Attribut **AT_INTERNAL_NO_CUBE_ENTRY** und Wert **true** beim Reimport überschreiben, werden auch der Einträge im Funktions-Cube gelöscht.

Attributdefinition

Das Attribut **AT_INTERNAL_NO_CUBE_ENTRY** ist in der Standardkonfiguration der PPM-Attribute nicht enthalten. Wenn Sie dieses Feature verwenden möchten, müssen Sie zunächst das Attribut **AT_INTERNAL_NO_CUBE_ENTRY** mit Datentyp **boolean** definieren.

Die im Prozessbaum der Analyse angezeigten Funktionen basieren auf den Einträgen im Funktions-Cube. Funktionen, deren Instanzen mittels Attributwert **true** des Attribut **AT_INTERNAL_NO_CUBE_ENTRY** nicht in den Funktions-Cube geschrieben wurden, werden im Prozessbaum nicht angezeigt.

Auf die Verwendung von Prozesshierarchieen hat das beschriebene Feature keine Auswirkungen, da auch nicht berechnete Funktionen mit Prozessinstanzen hinterlegt werden können. Wegen dem zuvor beschriebenen Sachverhalt werden die nicht im Funktions-Cube gespeicherten Funktionen im Prozessbaum des hinterlegten Prozesstyps ebenfalls nicht angezeigt.

7.1.4. Typisierungsregeln im CTK

Typisierungsregeln können Sie im PPM CTK-Modul **Prozesse** definieren. Um eine Regel für einen bestimmten Prozesstyp zu erstellen, zu bearbeiten bzw. zu löschen, wählen Sie einfach den entsprechenden Prozesstyp aus dem Prozessbaum aus und wählen den gewünschten Kontextmenüeintrag. Es ist auch möglich, eine Typisierungsregel auf Basis einer Vorlage zu erstellen. Als Vorlagen können

alle bereits definierten Regeln verwendet werden. Die Definition der Berechnungsvorschrift einer Typisierungsregel erfolgt über die bekannten Operanden und Operatoren aus der Attributberechnung (Kapitel Definition von Attributberechnungen [50]).

Warnung
Sie müssen in der Definition der entsprechenden Berechnungsvorschrift einer Typisierungsregel sicherstellen, dass diese einen Rückgabewert vom Typ **BOOLEAN** liefert.

Im Dialog **Typisierungsregel "typifierrule_<processtypegroup>_<processtype>" konfigurieren** wird jede Berechnungsvorschrift automatisch auf korrekte Syntax überprüft.

Sobald Sie Ihre Änderungen speichern, werden diese persistent in der Prozessbaum- und Kennzahlenkonfiguration hinterlegt. Wenn Sie die geänderte Konfiguration aktivieren, wird diese ins PPM-System übernommen.

7.2. Typisierung durch Attributberechnung

Die Typisierung kann durch Verwendung von Typisierungsregeln oder alternativ durch den Import von Werten in spezifische Attribute erfolgen, die sogenannte „Vortypisierung“.

Die Attribute können immer getrennt von Typisierung und Prozesszuordnung berechnet werden. Dies ist durch Zuordnung der Attribute zur Prozessbaumwurzel möglich. Diese über **runppmimport** durchgeführte Attributberechnung wird zwischen Merge und Typisierung/Kennzahlberechnung verarbeitet.

Auf diese Weise kann eine EPK-Typisierung angewandt werden, indem die Attribute **AT_PROCTYPEGROUP** und **AT_PROCTYPE** importiert oder berechnet werden. Wenn diese eingestellt sind, verwendet die Typisierung ihre Werte anstelle der oben beschriebenen Typisierungsregeln.

Attribute **AT_PROCTYPE** und **AT_PROCTYPEGROUP**:

Attribute	Beschreibung	Verwendung
AT_PROCTYPE	Prozesstypen	Importierte oder eingestellte Typisierungsregeln
AT_PROCTYPEGROUP	Prozesstypgruppe	Importierte oder eingestellte Typisierungsregeln

8. Definition von Kennzahlen, Dimensionen, Attributberechnungen und Beziehungen

Kennzahlen des PPM-Systems liefern messbare Werte berechenbarer Eigenschaften von Prozess- bzw.Funktionsinstanzen, wie z.B.Prozess- bzw. Funktionsdurchlaufzeiten in Stunden oder Auftragsvolumina in Euro.

Dimensionen konkretisieren die berechneten Kennzahlenwerte von Prozess- und Funktionsinstanzen anhand bestimmter Kriterien, wie bspw.Auftragsnummer, Auftraggeber und so weiter.

Wie Sie Kennzahlen und Dimensionen oder Attributberechnungen und Beziehungen definieren und mittels spezieller Konfigurationen des Prozessbaums (siehe Kapitel Anmeldung von Kennzahlen und Dimensionen am _ppm-System [181]) für das PPM-System verfügbar machen, erfahren Sie in den folgenden Kapiteln.

8.1. Begriffserläuterungen

Nachfolgend werden zentrale Begriffe des Kapitels **Definition von Kennzahlen, Dimensionen, Attributberechnungen und Beziehungen** näher erläutert.

8.1.1. Kennzahlen

Im PPM-System werden verschiedene Kategorien von Kennzahlen unterschieden:

Unterscheidung nach Kennzahltyp

Nach dem Objekttyp, auf den sich die jeweilige Kennzahl bezieht, werden die folgenden Kennzahltypen unterschieden:

- **Prozesskennzahlen** sind Kennzahlen, deren Werte an der gesamten Prozessinstanz für die Analyse zur Verfügung stehen.
- **Funktionskennzahlen** werden dagegen auf der Basis von Funktionsinstanzen ausgewertet.
- **Relationskennzahlen** sind Kennzahlen, die für die Auswertung von Beziehungen zur Verfügung stehen.
- **Kardinalitätskennzahlen** stehen für spezifische Auswertungen von Textdimensionen zur Verfügung.

Unterscheidung nach Prozessbezug

- **Prozessinstanzabhängige Kennzahlen** sind Kennzahlen, deren Werte mit einem Bezug zu Prozessinstanzen berechnet werden.
- **Prozessinstanzunabhängige Kennzahlen** sind Kennzahlen, deren Werte ohne Bezug zu Prozessinstanzen berechnet werden.

Unterscheidung nach Definitionsweise

- **Standardkennzahlen** werden in der mandantenspezifischen Kennzahlenkonfigurationsdatei definiert. Ein Großteil dieser Kennzahlen ist PPM-seitig vorkonfiguriert.

- **Benutzerdefinierte Kennzahlen** werden von Anwendern in einem speziellen Modul der PPM-Benutzeroberfläche auf der Basis von Standardkennzahlen definiert und in einer speziellen XML-Konfigurationsdatei gespeichert. Vorkonfigurierte, benutzerdefinierte Kennzahlen gehören ebenso zum Auslieferungsstandard von ARIS Process Performance Manager.

Die aufgelisteten Kategorien von Kennzahlen sind untereinander kombinierbar, so können beispielsweise prozessinstanzunabhängige Prozesskennzahlen definiert werden.

Allen Kennzahlkategorien ist gemein, dass der jeweils konkrete Wert einer Kennzahl eine bestimmte, messbare Eigenschaft z.B.einer Prozessinstanz beschreibt, bspw.Ausführungsdauer oder Anzahl Bearbeiter.

Des Weiteren lassen sich Kennzahlen logisch gruppieren. Die Zuordnung einer Kennzahl zu einer Gruppe muss eindeutig sein. D. h. jede Kennzahl kann nur einer Gruppe zugeordnet werden. Die Gruppenstruktur ist hierarchisch und beliebig tief.

8.1.1.1. Prozessinstanzabhängige Kennzahlen

Standardkennzahlen

ie Werte von Standardkennzahlen werden auf der Basis von Attributen auf Prozess-, Funktions- bzw. Relationsinstanzebene mittels der Attributberechnerkomponente des Kennzahlenberechners berechnet.

Zur Berechnung wird entweder ein fest in der PPM-Software programmierter Algorithmus oder ein vom Anwender in der XML-Kennzahlenkonfigurationsdatei angegebener Algorithmus verwendet.

Benutzerdefinierte Kennzahlen

Der Algorithmus zur Berechnung von benutzerdefinierten Kennzahlen wird komfortabel mit Hilfe des PPM-Frontend erstellt.

Der wesentliche Unterschied zu den Standardkennzahlen ist, dass die Berechnung nicht instanzbezogen auf Attributen, sondern auf Wertemengen bereits berechneter Kennzahlen basiert. Die Ergebnisse sind nicht in der PPM-Datenbank gespeichert und werden bei jedem Aufruf der Kennzahl neu berechnet. Änderungen des Algorithmus werden zeitnah durch erneutes Aufrufen der Kennzahl in der PPM-Benutzeroberfläche angezeigt.

8.1.1.2. Prozessinstanzunabhängige Kennzahlen (PIKIs)

Die Werte prozessinstanzunabhängiger Kennzahlen werden auf der Basis von Daten berechnet, die nicht prozessorientiert sind. Prozessinstanzunabhängige Kennzahlen können im PPM-System wie prozessinstanzabhängige Prozesskennzahlen analysiert und bspw.in Berechnungsvorschriften von benutzerdefinierten Kennzahlen verwendet werden.

Prozessinstanzunabhängige Kennzahlen werden nicht aus Prozessinstanzdaten berechnet. Der konkrete Kennzahlwert hat keinen Prozessinstanzbezug.

Wie prozessinstanzunabhängige Kennzahlen definiert werden, lesen Sie im Kapitel **Definition prozessinstanzunabhängiger Kennzahlen [137]**.

Die Konfiguration der Datenimportformate (XML, CSV, XLS) prozessinstanzunabhängiger Kennzahlen ist in der Technischen Referenz **PPM Datenimport** beschrieben.

8.1.2. Dimensionen

Dimensionen sind Kriterien zur Unterscheidung von Prozess- und Funktionsinstanzen. Dimensionswerte basieren auf Attributwerten, die entweder direkt aus dem Quellsystem übernommen (z.B. Standort, Produktsparte) oder berechnet werden (z.B. Prozesstyp).

Es werden die folgenden Dimensionstypen unterschieden:

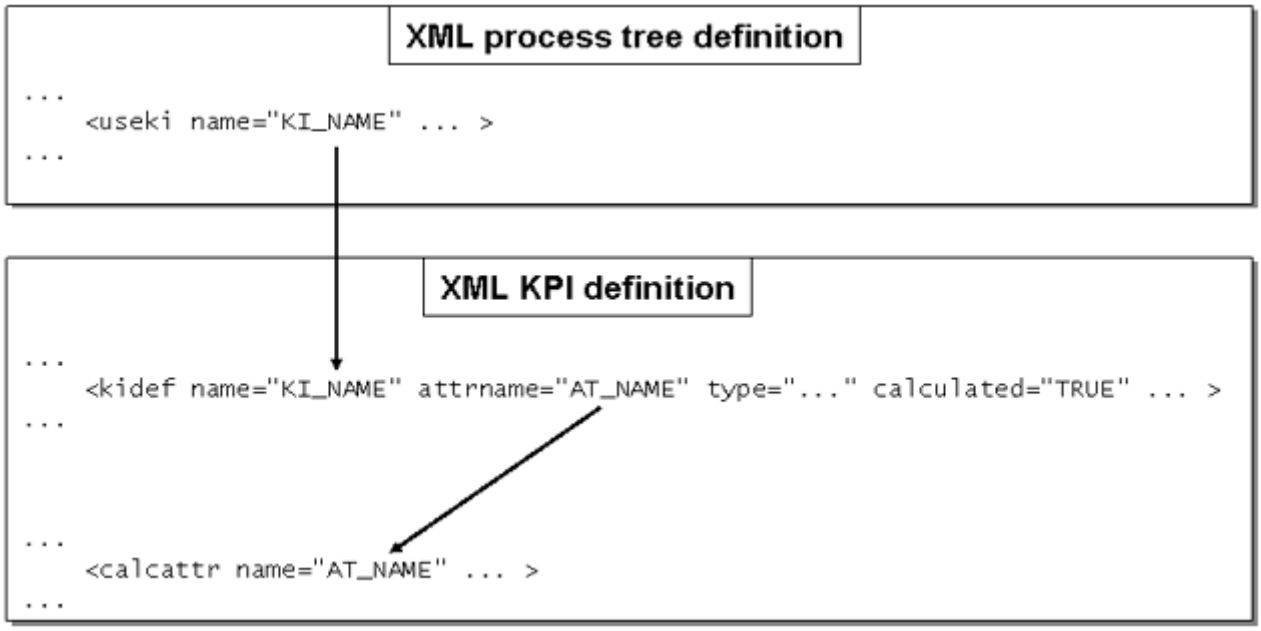
- Textdimensionen [154] (einstufig, zweistufig, n-stufig)
- Fließkommadimensionen [163] (Fließkommazahlenformat)
- Dimension Zeit [164]
- Tageszeitdimension [170]
- Zeitraumdimension [168]
- Suchdimensionen [171]
- Shared Function Dimension [174]

Die Standardschrittweite zur Anzeige einer Dimension wird entweder in der Konfigurationsdatei explizit angegeben oder vom System automatisch für eine optimale Darstellung berechnet.

8.2. Definition von Kennzahlen

Ausgangspunkt für die Berechnung von Kennzahlen und Dimensionen ist der Prozessbaum. Beim Berechnen wird für alle in der Konfigurationsdatei des Prozessbaumes angegebenen Kennzahlen (XML-Element **useki**) die Definition in der Kennzahlenkonfiguration (XML-Element **kidef**) ermittelt und die zugehörige Berechnungsvorschrift (XML-Element **calcattr**) ausgeführt. Durch diese Vorgehensweise werden nur Kennzahlen berechnet, die im Prozessbaum verwendet werden, und die Performance des Kennzahlenberechners unabhängig von der Anzahl definierter Kennzahlen optimiert. Die Berechnung von Dimensionen erfolgt analog (XML-Tag **usedim**).

Die folgende Grafik veranschaulicht die Zusammenhänge der Kennzahlendefinition und Prozessbaumdefinition:



Die mit dem PPM-System ausgelieferten Kennzahlenkonfigurationen (Dateien ***_keyindicator.xml** in den Verzeichnissen **<PPM-Installationsverzeichnis>\ppm\server\bin\agentLocalRepo\.unpacked\<Installationszeit>_ppm-client-run-prod-<Version>-runnable.zip\ppm\ctk\ctk\examples\custom\<Mandantenvorlage>\xml**) enthalten Definitionen der gebräuchlichsten, allgemein gültigen Kennzahlen und Dimensionen. Diese Standardkonfigurationen lassen sich einfach im PPM Customizing Toolkit-Modul **Kennzahlen und Dimensionen** um projektspezifische Kennzahlen erweitern.

Warnung
Vermeiden Sie bei der Vergabe von internen Kennzahlnamen die Suffixe **_NUM** und **_SUM**. Diese Suffixe werden intern vom Kennzahlenberechner verwendet.

8.2.1. Definition von Standardkennzahlen

Eine Kennzahl wird in der mandantenspezifischen XML-Konfigurationsdatei mit der Dokumenttypdefinition **KeyindicatorConfiguration.dtd** durch folgendes Element definiert:

```
...
<kidef name="..." attrname="..." type="..."
calculated="..." distribution="..."
standarddeviation="..." retrievertype="..."
kigroup="..." sharedfunctionki="..."
functionspanki="..." colname="..."
importmode="OPTIONAL">
<description language="..." name="...">
Description text...
</description>
</kidef>
...
```

XML-Attribut	Beschreibung
name	Interner Name der Kennzahl. Wird im XML-Tag **useki** der Prozessbaumdefinition referenziert.
type	Kennzahlart **PROCESS**: Prozesskennzahl **FUNCTION** (veraltet): Funktionskennzahl **OT_FUNC**: Funktionskennzahl **OT_ORG**: Organisationskennzahl **RELATION**: Beziehungskennzahl
location (optional)	Nur bei **type="RELATION"** Gültige Werte: **SOURCE** (Attributplatzierung am Quellbezugsobjekt der Beziehung) **TARGET** (Attributplatzierung am Zielbezugsobjekt der Beziehung) **THIS** (Standardwert: Attribut wird an der Beziehung selbst platziert)
description	Sprachabhängige Beschreibung einer Gruppe, optional mit Tooltip (Bereich **#PCDATA** des Elements **description**). Mindestens die Beschreibung in der Standardsprache muss gepflegt sein.
attrname	Bezeichner des Attributes, auf dem die Kennzahl basiert. Dies kann ein bereits existierender Attributwert (**calculated=FALSE**) oder ein zu berechnender Attributwert (**calculated=TRUE**) sein.

XML-Attribut	Beschreibung
calculated	**TRUE**: Der Wert des referenzierten Attributes wird durch die mit **calcattr** angegebene Berechnungsvorschrift berechnet. **FALSE**: Der Wert des referenzierten Attributes wird nicht berechnet.
distribution	**TRUE**: Die Kennzahl kann als Dimension verwendet werden. **FALSE**: Die Kennzahl kann nicht als Dimension verwendet werden.
standarddeviation (optional)	**TRUE**: Für die Kennzahl kann die Standardabweichung berechnet werden. Die Standardabweichung kann für alle Kennzahlen außer **Prozessanzahl** und **Funktionsanzahl** berechnet werden. Vorgabewert ist **TRUE**.
sharedfunctionki (optional)	**TRUE**: Die Kennzahl wird bei der Kennzahlenermittlung als Shared-Function-Kennzahl behandelt. Die Kennzahl einer Shared Function wird nur einmal berechnet und gilt für alle Instanzen der Shared Function. Vorgabewert ist **FALSE**.
functionspanki (optional)	**TRUE**: Die Kennzahl ist eine Funktionsspannenkennzahl (z.B. Durchlaufspanne). Bei mehrfachem Auftreten der Funktion innerhalb einer Prozessinstanz wird der ermittelte Kennzahlwert nur einmal pro Instanz gewertet. Vorgabewert ist **FALSE**.
retrievertype (optional)	Typ des verwendeten Kennzahlenermittlers. Legt fest, wie die Kennzahlwertemenge der an einer bestimmten Analyse beteiligten Prozessinstanzen aggregiert wird. Standardwert: **KEYINDICATOR**. **KEYINDICATOR**: Berechnet den Mittelwert (z.B. Durchlaufzeit). Als Datentyp sind alle numerischen Typen außer **LONG** zulässig. **NUM_KEYINDICATOR**: Aggregiert numerische Kennzahlen (z.B. Prozessanzahl, Funktionsanzahl) durch Addition der Werte. Als Datentyp sind alle numerischen Datentypen zulässig. **FREQ_KEYINDICATOR**: Aggregiert Häufigkeiten (z.B. Prozesshäufigkeit, Funktionshäufigkeit). Die Werte werden addiert und anschließend durch die Zeitspanne dividiert, die sich durch die gewählte Schrittweite der Dimension und den eingestellten Zeitfilter ergibt. **FACTORY_KEYINDICATOR**: Aggregiert Kennzahlen durch Bildung des Mittelwertes unter Verwendung des Werkskalenders. Als Datentyp ist ausschließlich **FACTORY_TIMESPAN** zulässig.
dimreferring	Art des Dimensionsbezugs **LOOSE**: lose **STRICT**: strikt Standardwert: **LOOSE**
kigroup (optional)	Kennzahlengruppe
importmode (optional)	Ausgabe von Fehlermeldungen beim Berechnen von Kennzahlwerten. **OPTIONAL**: Berechnungsfehler werden nicht ausgegeben. **MANDATORY**: Berechnungsfehler werden ausgegeben Standardwert: OPTIONAL

Von den Attributen **sharedfunctionki** und **functionspanki** darf nur eines den Wert **TRUE** haben. Wenn eines der beiden Attribute den Wert **TRUE** hat, muss der Kennzahlentyp **type** den Wert **FUNCTION** haben (Funktionskennzahl).

Der Kennzahlenermittlertyp **FACTORY_KEYINDICATOR** wird ab PPM 3.x nicht mehr verwendet, aber aus Kompatibilitätsgründen weiterhin unterstützt. Beim Einlesen der Konfiguation wird er durch den Ermittlertyp **KEYINDICATOR** ersetzt.

8.2.1.1. Formatierung von Kennzahlwerten

Standardmäßig werden Kennzahlwerte auf drei Nachkommastellen gerundet und mit Tausendertrenner bzw. gemäß den Angaben zu den Schlüsseln **MINIMUM_FRACTION_DIGITS** bzw. **MAXIMUM_FRACTION_DIGITS** in der Datei **Keyindicator_settings.properties** angezeigt. Sie können davon abweichend mittels XML-Element **format** für jede Kennzahl gesondert eine andere Formatierung angeben, wenn es sich nicht um eine Zielerreichungskennzahl handelt.

Die Werte von Zielerreichungskennzahlen werden immer auf eine Nachkommastelle und eine signifikate Stelle gerundet dargestellt.

```
...
<kidef name="PDLZ" attrname="..." type="..."
calculated="..." distribution="..."
standarddeviation="..." retrievertype="...">
<description name="Process cycle time" language="de"/>
<format fractiondigits="1" significantdigits="1" />
</kidef>
...
```

In der Analyse in PPM werden die Werte der Kennzahl **Prozesslaufzeit** auf eine Nachkommastelle gerundet (**fractiondigits="1"**) angezeigt. Für den jeweiligen Kennzahlwert in Tooltips und Modellattributen der EPK-Ansicht soll dabei genau eine signifikante Stelle (**significantdigits="1"**) angezeigt werden.

Die Definition der Formatangaben ist in der Datei **_formatinfo.dtd** vorgegeben:

XML-Tag	Beschreibung
format	Formatangabe für Kennzahlwerte
fractiondigits (optional)	Anzahl der anzuzeigenden Nachkommastellen für Kennzahlwerte in Tabellen, an EPK-Objektkanten und in Filterdialogen. Standardwert: **3**
significantdigits (optional)	Nur für Kennzahlwerte in Tooltips und Modellattributen der EPK-Ansicht: Anzahl der anzuzeigenden, signifikanten Stellen (Vor- und Nachkommastellen ungleich 0) bis zu maximal zehn Nachkommastellen insgesamt. Bspw. wird der Wert **1453,03500125** bei Angabe von **significantdigits="6"** und unabhängig von der Angabe zu **fractiondigits** als **1453,035** angezeigt.
usegrouping (optional)	**TRUE**: Tausendertrenner werden angezeigt. **FALSE**: Tausendertrenner werden nicht angezeigt. Standardwert: **TRUE**

8.2.1.2. Definition von Prozesskostenkennzahlen

Die Prozesskosten einer Prozessinstanz ergeben sich aus der Summe der Prozesskosten aller Funktionsinstanzen der Prozessinstanz. Die Prozesskosten von Funktionsinstanzen werden mit Hilfe der Kostensätze der den Funktionen zugeordneten Organisationseinheiten (Kapitel Anonymisieren [45]) und der Ausführungszeiten der Funktionen berechnet. Die Anzahl der Bearbeitungen einer Funktion durch

eine Organisationseinheit ergibt sich aus dem Attribut **AT_COUNT_PROCESSINGS** der Kante zwischen Organisationseinheit und Funktion. Sind einer Funktion mehrere Organisationseinheiten zugeordnet, wird dies als mehrfaches Ausführen der Funktion gewertet.

Um Kostenkennzahlen zu berechnen, müssen die Datentypen **Kosten** und **Kostensatz** bekannt sein. Die Definition dieser Datentypen ist in der XML-Konfigurationsdatei ***_datatypes.xml** der jeweiligen PPM CTK-Mandantenvorlage (unter <PPM-Installationsverzeichnis>\ppm\server\bin\agentLocalRepo\.unpacked\<Installationszeit>_ppm-client-run-prod-<Version>-runnable.zip\ppm\ctk\ctk\examples\custom\) enthalten, die Sie projektspezifisch anpassen können.

Die zur Berechnung von Kostenkennzahlen benötigten Ausführungszeiten von Funktionen können auf zwei unterschiedliche Arten berechnet werden. Je nach gewählter Berechnungsart werden die berechneten Kosten in unterschiedlichen Kennzahlen gespeichert. Welche Berechnungsart für die Ausführungszeiten verwendet wird, hängt davon ab, welche Informationen aus dem Quellsystem ausgelesen werden.

Kennzahlen FPKS_R und PK_R

Zur Berechnung des Kostensatzes auf Basis der Bearbeitungszeit wird die aus den Attributen **AT_START_TIME** und **AT_END_TIME** einer Funktion berechnete Bearbeitungszeit (Funktionsattribut **AT_KI_FBZ**) verwendet. Der berechnete Kostensatz wird im Funktionsattribut **AT_PKS_R** gespeichert.

Kennzahlen FPKS_S und PK_S

Zur Berechnung des Kostensatzes auf Basis des Leistungsstandards wird aus dem Quellsystem eine geschätzte Standardbearbeitungszeit ausgelesen und als Attribut **AT_LS** an die Funktionen geschrieben. Der berechnete Kostensatz wird im Funktionsattribut **AT_PKS_S** gespeichert.

Der Prozesskostensatz einer Funktion gibt die durchschnittlichen Kosten für die einmalige Bearbeitung der Funktion an und wird für die beiden beschriebenen Berechnungsarten mit folgender Berechnungsvorschrift berechnet: Das Produkt der Ausführungszeit einer Funktion und der Summe der gewichteten Kostensätze aller der Funktion zugeordneten Organisationseinheiten wird durch die Gesamtanzahl der Bearbeitungen dividiert.

Folgende Formel veranschaulicht die Berechnungsvorschrift:

FPKS Prozesskostensatz (Funktion)

FT Funktionsausführungszeit

KS Prozesskostensatz

FREQ Bearbeitungshäufigkeit

$$\mathbf{FPKS} = \frac{\mathbf{FT} * \mathrm{sum}(\mathbf{KS} * \mathbf{FREQ})}{\mathrm{sum}(\mathbf{FREQ})}$$

Die Art der Berechnung von Kostenkennzahlen wählen Sie durch Anmeldung der entsprechenden Kennzahlen am Prozessbaum. Die Standardkonfiguration von PPM berechnet Prozesskostensätze auf Basis des Leistungsstandards.

Auszug aus der Datei ***_processtree.xml**:

```
...
<useki name="FPKS_S" scale="EUR" assessment="NEG"/>
<useki name="PK_S" scale="EUR" assessment="NEG"/>
...
```

8.2.2. Kennzahldefinition in Multibyte-Zeichensätzen

Der folgende Auszug aus der Kennzahlenkonfigurationsdatei zeigt beispielhaft die Definitionsmöglichkeiten für benutzerdefinierte Kennzahlen unter Verwendung eines Multibyte-Zeichensatzes:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE keyindicatorconfig SYSTEM "KeyindicatorConfiguration.dtd">
<keyindicatorconfig>
...
<calcattr name="ΙΔ_ΟΓΚ_ENT" type="PROCESS">
<calculation>
...
</calculation>
</calcattr>
...
<!-- Ορισμός του δείκτη όγκου των εντολών -->
<kidef name="ORDERVOL" attrname="ΙΔ_ΟΓΚ_ENT"
type="PROCESS" calculated="FALSE"
distribution="FALSE" standarddeviation="FALSE"
retrievertype="NUM_KEYINDICATOR"
kigroup="KI_GROUP_COST" dimreferring="LOOSE"
importmode="OPTIONAL" sharedfunctionki="FALSE"
functionspanki="FALSE">
<description name="Auftragsvolumen" language="de"/>
Order volume
<description name="Order volume" language="en">
Order volume
<description name="Όγκος εντολών" language="el">
Όγκος εντολών κατα αύξοντα αριθμό
</description>
</kidef>
...
</keyindicatorconfig>
```

8.2.3. Definition von Kardinalitätskennzahlen

Der Wert einer Kardinalitätskennzahl ergibt sich aus der Anzahl der unterschiedlichen Werte (= max. mögliche Schritte) der referenzierten Textdimension für die angegebene Schrittweite (Stufe). Eine Kardinalitätskennzahl kann für ein-, zwei- und n-stufige Dimensionen definiert werden und wird in der Kennzahlenkonfigurationsdatei durch folgendes XML-Element definiert:

```
...
<crdkidef name="..." dimreferring="...">
<description language="de" name="..."/>
<description language="en" name="..."/>
<refdim name="..." refinement="..."/>
</crdkidef>
...
```

XML-Tag	Beschreibung
name	Interner Name der Kennzahl. Wird im XML-Tag **useki** der Prozessbaumdefinition referenziert.
dimreferring	Art des Dimensionsbezugs. **LOOSE**: lose **STRICT**: strikt Standardwert: **LOOSE**
refdim	Im XML-Attribut **name** wird der Name der Dimension angegeben, auf die sich die berechnete Kardinalität bezieht. Optional wird im XML-Attribut **refinement** die Schrittweite der Dimension angegeben, für die die Kardinalität berechnet wird. Fehlt die Angabe, wird die Standardschrittweite der Dimension verwendet. Für n-stufige Dimensionen ist die Angabe der Schrittweite zwingend. Zulässige Werte: Einstufige Dimension: **BY_LEVEL_1** Zweistufige Dimension: **BY_LEVEL_1** (grob) oder **BY_LEVEL_2** (detailliert) N-stufige Dimension: Nur **BY_LEVEL<1_N>** (gröbste Stufe, z.B. **BY_LEVEL1_12**) oder **BY_LEVEL<N_N>** (detaillierteste Stufe, z. B., **BY_LEVEL12_12**) Vorgabewert ist die Standardschrittweite der referenzierten Dimension.
kigroup (optional)	Kennzahlengruppe Standardwert: Gruppe **Alle Kennzahlen**

Für eine Kardinalitätskennzahl können neben der Kennzahl selbst nur Ranking, Vorperioden und Planwerte ermittelt werden. Statistische Bewertungen (Minimum, Maximum, Summe und Standardabweichung) können nicht angezeigt werden. Kardinalitätskennzahlen können nicht als Dimension verwendet werden. Für Kardinalitätskennzahlen können keine Filter angegeben werden.

Bei der Berechnung von Kardinalitätskennzahlen werden optional durch prozessinstanzunabhängigen Kennzahlen hinzukommende Dimensionswerte nicht berücksichtigt. Die Kardinalität von Dimensionen, die ausschließlich von prozessinstanzunabhängigen Kennzahlen verwendet werden, liefern immer den Wert **0**.

8.2.4. Definition prozessinstanzunabhängiger Kennzahlen

Prozessinstanzunabhängige Kennzahlen werden in der mandantenspezifischen Kennzahlenkonfiguration (XML-Datei mit der Dokumenttypdefinition **keyindicatorconfiguration.dtd**) im Rahmen von Datenreihen definiert.

Definition von Datenreihen

Eine Datenreihe (XML-Element **pikicube**) besteht aus prozessinstanzunabhängigen Kennzahlen und referenzierten Dimensionen. Sie muss mindestens eine prozessinstanzunabhängige Kennzahl (**pikidef**) und mindestens eine referenzierte Dimension (**refdim**) enthalten. Referenzierte Dimensionen müssen im PPM-System konfigurierte Dimensionen sein.

Prozessinstanzunabhängige Kennzahlen in Datenreihen sind immer vom Typ **Prozess**, um eine maximale Verwendbarkeit im PPM-System sicherzustellen. Dabei spielt es keine Rolle, von welchem Typ die Datenreihe selber ist, siehe Kapitel Verwendung (Typ) einer Datenreihe [141].

Pro Datenreihe muss mindestens eine referenzierte Dimension als Schlüsseldimension (**refdim ... iskeydimension="TRUE"**) gekennzeichnet sein. Standardmäßig sind alle referenzierten Dimensionen Schlüsseldimensionen. Eine bestimmte Wertkombination der angegebenen Schlüsseldimension(en) liefert dabei eine eindeutige Datenzeile innerhalb einer Datenreihe, d. h., eine bestimmte Wertkombination kommt innerhalb einer Datenreihe nur einmal vor.

Beispiel 53.

Die folgende Datenreihe enthält drei Datenzeilen, die anhand der Wertkombinationen der angegebenen Schlüsseldimensionen (*****) unterschieden werden:

Beispiel 54.	Beispiel 55.	Beispiel 56.	Beispiel 57.	Beispiel 58.	Beispiel 59.
D_COUNTRY*	D_PLANT*	D_DEPARTMENT*	D_RECORDED BY	SALES	COSTS
Beispiel 60.	**Beispiel 61.**	**Beispiel 62.**	**Beispiel 63.**	**Beispiel 64.**	
Deutschland	Hamburg	42	Schmidt	400000	
Beispiel 65.	**Beispiel 66.**	**Beispiel 67.**	**Beispiel 68.**	**Beispiel 69.**	**Beispiel 70.**
Deutschland	Frankfurt	17	Hartmann	510000	360000
Beispiel 71.	**Beispiel 72.**	**Beispiel 73.**	**Beispiel 74.**		**Beispiel 75.**
USA	Pittsburgh	53	Fox		410000

Beispiel 76.

Jede der drei Datenzeilen kann innerhalb der Datenreihe nur einmal vorkommen. Die spezifische Wertkombination der Schlüsseldimensionen **D_LAND**, **D_WERK** und **D_ABTEILUNG** (z.B. **Deutschland; Hamburg; 42**) bildet dabei den Identifizierer einer Datenzeile.

Konfiguration

Die Konfiguration einer Datenreihe (XML-Element **pikicube**) veranschaulicht die nachstehende, allgemeine Dateistruktur:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM "KeyindicatorConfiguration.dtd">
<keyindicatorconfig>
...
<pikicube name="...">
<description language="de" name="..."/>
...
<pikidef name="..." retrievertype="..." dimreferring="...">
<description language="de" name="...">
Descriptive text (tooltip)
</description>
...
<datatype name="..."/>
</pikidef>
<refdim name="..." refinement="..." iskeydimension="TRUE"/>
...
</pikicube>
...
</keyindicatorconfig>
```

Die folgenden Tabellen erläutern die Konfiguration einer prozessinstanzunabhängigen Datenreihe:

ELEMENT und ATTLIST pikicube	Beschreibung
pikicube	Prozessinstanzunabhängige Datenreihe
name	Systemweit eindeutiger Name der Datenreihe. Dieser wird auch für den Namen des Cubes in der Datenbank verwendet.
comment (optional)	Kommentar zur Datenreihe, wird in PPM Customizing Toolkit verwendet.
editable	Mit **editable="TRUE"** (Vorgabewert) wird die Dateneingabe für die Datenreihe im Modul **Konfiguration/ Dateneingabe** der PPM-Oberfläche ermöglicht.
type	Verwendung (Typ der Datenreihe), die festlegt, welche Dimensionen in der Datenreihe als referenzierte Dimensionen verwendet werden dürfen. Vorgabewert ist **PROCESS**, das heißt, es dürfen nur Prozessdimensionen (**dimtype="PROCESS"** in der Dimensionsdefinition) in der Datenreihe angegeben werden. Weitere, mögliche Werte: **OT_FUNC** (Nur Funktionsdimensionen in der Datenreihe erlaubt) **RELATION** (Nur Dimensionen vom Typ **RELATION** in der Datenreihe erlaubt) Nähere Angaben siehe Kapitel **Verwendung (Typ) einer Datenreihe [141]**.
relname	Nur bei **type="RELATION"**. Es ist genau eine im PPM-System bestehende Relation mit ihrem Namen anzugeben, z. B. **relname="REL_WORKS_TOGETHER"**. Der angegebenen Relation wird der Datenreihe fest zugeordnet.
deletedata onredefinition	Veraltet, wird nicht mehr verwendet.
description	Sprachabhängige Beschreibung der Datenreihe. Mindestens die Beschreibung in der Standardsprache muss gepflegt sein.
pikidef	Definition einer prozessinstanzunabhängigen Kennzahl, mindestens eine pro Datenreihe, s. u.
refki	Veraltet, wird nicht mehr verwendet.
refdim	Referenzierte Dimension, s. u.

ELEMENT und ATTLIST refdim	Beschreibung
refdim	Im PPM-System existierende Dimension, auf welche sich die prozessinstanzunabhängigen Kennzahlen der Datenreihe beziehen. Es ist mindestens eine referenzierte Dimension pro Datenreihe anzugeben. Bei prozessinstanzunabhängigen Kennzahlen werden interne Dimensionen [159] nicht als referenzierte Dimensionen (refdim) unterstützt.
name	Interner Name der im PPM-System existierenden Dimension.
refinement	Dimensionsschrittweite, mit der der Datenimport erfolgen soll. Die zu importierenden Dimensionswerte müssen genau in dieser Schrittweite angegeben werden.
iskeydimension	Mit **iskeydimension="TRUE"** (Vorgabewert) wird festgelegt, dass die referenzierte Dimension eine Schlüsseldimension der Datenreihe ist. Die Wertkombinationen aller angegebenen Schlüsseldimensionen machen jede Datenzeile einer Datenreihe eindeutig.

ELEMENT und ATTLIST pikidef	Beschreibung
pikidef	Definition einer prozessinstanzunabhängigen Kennzahl. Pro Datenreihe ist mindestens eine Definition anzugeben. Eine prozessinstanzunabhängige Kennzahl kann nur in genau einer Datenreihe verwendet werden.
name	Systemweit eindeutiger Name der prozessinstanzunabhängigen Kennzahl.
type	Veraltet, wird nicht mehr verwendet.

ELEMENT und ATTLIST pikidef	Beschreibung
retrievertype	Kennzahlwert-Ermittlertyp. Standardwert: **KEYINDICATOR** (Mittelwertbildung) Weitere Werte: **NUM_KEYINDICATOR** (Summenbildung) **FREQ_KEYINDICATOR** (veraltet, wird nicht mehr verwendet) **FACTORY_KEYINDICATOR** (wird nicht mehr verwendet)
dimreferring	Art des Dimensionsbezugs **LOOSE**: lose **STRICT**: strikt Standardwert: **LOOSE**
kigroup (optional)	Zuordnung der prozessinstanzunabhängigen Kennzahl zu einer bestehenden Kennzahlengruppe
description	Sprachabhängige Beschreibung einer prozessinstanzunabhängigen Kennzahl. Mindestens die Beschreibung in der Standardsprache muss gepflegt sein.
datatype	Datentyp der Werte einer prozessinstanzunabhängigen Kennzahl

Sie können beliebig viele Datenreihen (XML-Elemente **pikicube**) innerhalb einer Kennzahlenkonfiguration definieren. In einer Datenreihe können beliebig viele prozessinstanzunabhängige Kennzahldefinitionen (XML-Elemente **pikidef**) angegeben werden. Für alle prozessinstanzunabhängige Kennzahldefinitionen einer Datenreihe gilt derselbe Dimensionsbezug (XML-Elemente **refdim**).

Prozessinstanzunabhängige Datenreihen konfigurieren Sie bequem im PPM CTK-Modul **Kennzahlen und Dimensionen** mittels Untermodul **Prozessinstanzunabhängige Kennzahlen**.

Die Gesamtheit der in einem PPM-System konfigurierten, prozessinstanzunabhängigen Datenreihen können Sie über die XML-Schnittstelle mittels Kommandozeilenprogramm **runppmconfig** mit dem Parameter **-keyindicator** in eine XML-Datei exportieren.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM "KeyindicatorConfiguration.dtd">
<keyindicatorconfig>
...
<pikicube name="PIKICUBE_TURNOVER_PROD_GROUP">
<pikidef name="TURNOVER_PROD_GROUP" retrievertype="KEYINDICATOR"
dimreferring="STRICT" kigroup="KI_GROUP_COST">
<description language="de" name="Umsatz pro Produktgruppe"/>
<description language="en" name="Turnover by product group"/>
<datatype name="DOUBLE"/>
</pikidef>
<refdim name="TIME" refinement="BY_MONTH"/>
<refdim name="D_PRODUCT_GROUP"/>
<refdim name="PROCESSTYPE" refinement="BY_LEVEL2"/>
</pikicube>
...
</keyindicatorconfig>
```

Beispiel 77.

Es wird eine Datenreihe mit dem internen Namen **PIKICUBE_TURNOVER_PROD_GROUP** angelegt.

In der Definition (**pikidef**) der prozessinstanzunabhängigen Kennzahl **TURNOVER_PROD_GROUP** werden ein strikter Dimensionsbezug (**dimreferring="STRICT"**) und eine Zuordnung der prozessinstanzunabhängigen Kennzahl zur Kennzahlengruppe **KI_GROUP_COST** angegeben.

Die prozessinstanzunabhängigen Kennzahl bezieht sich (**refdim="..."**) strikt auf die Dimensionen **TIME**, **D_PRODUCT_GROUP** und **PROCESSTYPE**. Der Bezug auf die Dimension **PROCESSTYPE** wird mit der Schrittweite **fein** (**refinement="BY_LEVEL2"**) festgelegt.

Da in den **refdim**-Elementen keine Angaben zum Attribut **iskeydimension** gemacht werden, wird der Standardwert **TRUE** des Attributs verwendet, d. h., alle referenzierten Dimensionen werden als Schlüsseldimensionen der Datenreihe verwendet.

```
...
<pikicube name="PIKICUBE_COSTS">
<description language="en" name="Costs"/>
<pikidef name="OVERHEAD_COSTS" retrievertype="KEYINDICATOR"
dimreferring="LOOSE">
<description language="en" name="Overhead costs"/>
<datatype name="COST"/>
</pikidef>
<refdim name="PROCESSTYPE" refinement="BY_LEVEL2" iskeydimension="FALSE"/>
<refdim name="TIME" refinement="BY_MONTH" iskeydimension="TRUE"/>
<refdim name="MATERIAL" refinement="BY_LEVEL2" iskeydimension="TRUE"/>
</pikicube>
...
```

Beispiel 78.

Der gezeigte Dateiauszug definiert die Datenreihe **Kosten** mit der prozessinstanzunabhängigen Kennzahl **Gemeinkosten** des Datentyps **COST** mit dem PPM-systemweit eindeutigen, internen Bezeichner **OVERHEAD_COSTS**.

Der Kennzahlwert-Ermittlertyp ist Mittelwertbildung (**KEYINDICATOR**), der Dimensionsbezug ist lose (**LOOSE**).

In den XML-Elementen **refdim** sind die PPM-Dimensionen **TIME** und **MATERIAL** als Schlüsseldimensionen (**iskeydimension="TRUE"**) des Dimensionsbezugs der prozessinstanzunabhängigen Datenreihe angegeben.

Zusätzlich sind für die zu importierenden Dimensionswerte von den Standardschrittweiten abweichende Schrittweiten festgelegt.

Anmeldung von prozessinstanzunabhängigen Kennzahlen am PPM-System

Prozessinstanzunabhängige Kennzahlen werden im Prozessbaum (Element **useki** in der XML-Datei mit der Dokumenttypdefinition **keyindicatorprocesstree.dtd**) an Prozesstypgruppen und Prozesstypen angemeldet.

Nähere Informationen zur Anmeldung prozessinstanzunabhängiger Kennzahlen am Prozessbaum erhalten Sie im Kapitel Kennzahlen und Dimensionen prozessinstanzunabhängige Datenreihen anmelden [182].

8.2.4.1. Verwendung (Typ) einer Datenreihe

Für eine Datenreihe ist eine der folgenden Verwendungen (**pikicube type="..."**) zu wählen, die festlegt, welche Dimensionen im PIKI-Cube verwendet werden dürfen:

- Prozess (**PROCESS** - Standardwert)
- Funktion (**OT_FUNC**)
- Relation (**RELATION**)

Prozessinstanzunabhängige Kennzahlen sind unabhängig vom gewählten Typ der Datenreihe immer vom Typ **PROCESS**, d.h., sie werden generell wie Prozesskennzahlen behandelt.

Die Auswirkungen der jeweiligen PIKI-Cube-Typen sind wie folgt.

PROCESS

Es sind nur Prozessdimensionen (**dimtype="PROCESS"** in der Definition der jeweiligen Dimension) als referenzierte Dimensionen (**refdim="..."**) in der Datenreihe erlaubt.

OT_FUNC

Es sind nur Prozess- sowie Funktionsdimensionen (**dimtype="PROCESS"** bzw. **"OT_FUNC"** oder **"FUNCTION"**) als referenzierte Dimensionen in der Datenreihe erlaubt.

RELATION (mit Relationsname <x>)

Als referenzierte Dimensionen der Datenreihe sind nur Prozessdimensionen, Beziehungsdimensionen der Relation <x> sowie Quell- und Zielbezugsdimension der Relation <x> (also **FPROCESSTYPE**, **FROMORG**, **TOORG**, **FUNCTION**, **ORGUNIT**) erlaubt.

8.2.4.2. Dimensionsbezug

Prozessinstanzunabhängige Kennzahlen können einen losen oder strikten Dimensionsbezug haben (XML-Attribut **dimreferring**). Der Vorgabewert ist loser Dimensionsbezug (**dimreferring="LOOSE"**).

Loser Dimensionsbezug

Eine prozessinstanzunabhängige Kennzahl mit einem losen Dimensionsbezug kann für alle vorhandenen Dimensionen analysiert werden. Die prozessinstanzunabhängige Kennzahl liefert auch Werte für Anfragen mit anderen als den für die prozessinstanzunabhängige Kennzahl angegebenen Schrittweiten (XML-Attribut **refinement**) sowie für Dimensionen, für die kein Bezug definiert ist.

Wenn Sie eine prozessinstanzunabhängige Kennzahl mit einer Dimension analysieren, für die kein Dimensionsbezug (XML-Element **refdim**) definiert wurde, wird diese Dimension bei der Wertberechnung der prozessinstanzunabhängigen Kennzahl ignoriert. Die gezeigten Werte der prozessinstanzunabhängige Kennzahl gelten nur für die Dimensionen, auf die in der Definition der prozessinstanzunabhängige Kennzahldatenreihe Bezug genommen wird.

Ebenso werden bei Anfragen mit feineren Schrittweiten die Werte der prozessinstanzunabhängigen Kennzahl zurückgegeben, die auf die definierte Schrittweiten Bezug nehmen. Das heißt, andere Schrittweiten werden in der Analyse ignoriert.

Beispiel

Gemeinkosten	Gesamtkosten	Auftraggeber (grob, detailliert)	Zeit (monatsweise)
1000 €	25000 €	Deutschland, Becker	Jan 2001
3000 €	68000 €	Deutschland, Schmidt	Jan 2001
1500 €	13000 €	Frankreich, Leclerc	Jan 2001

Gemeinkosten	Gesamtkosten	Auftraggeber (grob, detailliert)	Zeit (monatsweise)
1200 €	12000 €	Deutschland, Becker	Feb 2001
3400 €	78000 €	Deutschland, Schmidt	Feb 2001
...	...	...	...

Beispiel 79.

In der Tabelle sind die prozessinstanzunabhängigen Kennzahlen **Gemeinkosten** und **Gesamtkosten** in Bezug auf die Dimensionen **Auftraggeber** und **Zeit** aufgelistet. Wenn Sie in der Analyse die Gemeinkosten für den Auftraggeber **Deutschland**, **Becker** für den **15. Jan 2001** anfragen, erhalten Sie für die prozessinstanzunabhängige Kennzahl den Rückgabewert **1000 €**. Dieser Wert bezieht sich in Wirklichkeit jedoch auf den gesamten Monat Januar 2001 (**refinement="BY_MONTH"**).

Beachten Sie die definierten Dimensionsbezüge einer prozessinstanzunabhängigen Kennzahl sowie die angegebenen Schrittweiten der referenzierten Dimensionen, um aussagekräftige Analyseergebnisse zu gewährleisten.

Strikter Dimensionsbezug

Eine prozessinstanzunabhängige Kennzahl mit striktem Dimensionsbezug kann nur mit den Dimensionen ausgewertet werden, auf welche sie sich in der Definition der Datenreihe bezieht (XML-Elemente **refdim**). Anfragen für Dimensionen, auf die sich die prozessinstanzunabhängige Kennzahl nicht bezieht, sind nicht möglich. Auch Anfragen mit einer anderen als der definierten Schrittweite sind nicht möglich.

Wird in der Analyse eine andere Schrittweite oder Dimension gewählt, wird ein entsprechender Fehlerdialog angezeigt.

Dimensionsbezug "Prozesstyp" als Sonderfall

Wenn Sie die Dimension **Prozesstyp** (**PROCESSTYPE**) als Dimensionsbezug in einer prozessinstanzunabhängigen Datenreihe angeben, ist beim Datenimport nur die Verwendung bereits im PPM-System existierender Prozesstypen erlaubt. Versuchen Sie dagegen, prozessinstanzunabhängige Daten in einen noch nicht vorhandenen Prozesstyp zu importieren, gibt der Import eine Fehlermeldung inklusive der betroffenen Datenzeilen aus. Eine automatische Prozessbaumerweiterung findet nicht statt. Der Datenimport bricht nicht ab, aber die Datenzeilen mit dem nicht vorhandenen Prozesstyp werden nicht importiert.

8.2.4.3. Definition prozessinstanzunabhängiger Kennzahlen in Multibyte-Zeichensätzen

Der folgende Dateiauszug aus der Kennzahlenkonfiguration zeigt beispielhaft die Definitionsmöglichkeiten prozessinstanzunabhängiger Datenreihen unter Verwendung eines Multibyte-Zeichensatzes:

```
...
<!-- Ορισμος σειράς δεικτών -->
<pikicube name="PIKICUBE_1">
<description name="Umsatz" language="de"/>
<description name="Turnover" language="en"/>
<description name="Τζίρος" language="el"/>
<pikidef name="PIKI_1" retrievertype="NUM_KEYINDICATOR" dimreferring="LOOSE"
kigroup="KI_GROUP_COST">
<description name="Umsatz" language="de"/>
<description name="Turnover" language="en"/>
<description name="Τζίρος" language="el"/>
```

```
<datatype name="COST"/>
</pikidef>
<pikidef name="PIKI_2" retrievertype="KEYINDICATOR" dimreferring="LOOSE">
<description name="Kundenzufriedenheit" language="de"/>
<description name="Customer satisfaction" language="en"/>
<description name="Ευχαρίστηση των πελατών" language="el"/>
<datatype name="DOUBLE"/>
</pikidef>
<refdim name="MATERIAL"/>
</pikicube>
...
```

8.2.4.4. Konfigurationsimport

Der Import prozessinstanzunabhängiger Datenreihen erfolgt zusammen mit dem der übrigen Kennzahlenkonfiguration mit Hilfe des Kommandozeilenprogramms **runppmconfig** (siehe Benutzerhandbuch **PPM Operation Guide**):

```
runppmconfig –user <user name> –password <password>
[–client <client name>]
–mode import
[–overwrite]
–keyindicator <XML measure configuration>
```

Voraussetzung ist, dass der ausführende Benutzer das Funktionsrecht **Konfigurationsimport** besitzt.

Additiver Konfigurationsimport

Standardmäßig, d.h. ohne Angabe der Option **-overwrite**, arbeitet der Import der Kennzahlenkonfiguration additiv, d.h., dass bereits im PPM-System exisitierende Datenreihen erhalten bleiben und nicht verändert werden.

Für jede neu importierte Datenreihe wird eine Datenbanktabelle mit dem internen Namen der Datenreihe (**pikicube name="..."**) angelegt sowie eine entsprechende Datenstruktur auf dem Analyseserver geschaffen.

Überschreibender Konfigurationsimport

Beim nachträglichen Import einer geänderten Konfiguration einer prozessinstanzunabhängigen Datenreihe mit der Kommandozeilenoption **runppmconfig -mode import -overwrite** müssen Sie beachten, ob Ihre Änderungen sich auf die Datenstruktur der bestehenden Datenreihe auswirken oder nicht (siehe unten).

Im ersten Fall müssen bereits in die Datenreihe importierte Daten vor einem Import der geänderten Konfiguration zuerst gelöscht werden, im letzten Fall ist dies nicht notwendig:

Nicht datenstrukturrelevante Änderungen

Durch Angabe der Option **-overwrite** können Sie folgende Änderungen an der Konfiguration bereits im PPM-System bestehender Datenreihen vornehmen, ohne dass Sie bereits importierte Daten vorher löschen müssen:

- Hinzufügen weiterer Schlüsseldimensionen bzw. Nichtschlüsseldimensionen sowie weiterer prozessinstanzunabhängiger Kennzahlen
- Ändern einer referenzierten Dimension in eine Schlüsseldimension (**iskeydimension="TRUE"**)

- Ändern der Beschreibung einer Datenreihe (PIKI-Cube)
- Ändern der Verwendung einer Datenreihe (zum Beispiel **type="PROCESS"** zu **type="FUNCTION"**)
- Zuordnung einer Datenreihe zu einer anderen Relation (**relname="..."**)
- Ändern des Dimensionsbezugs (lose/strikt) bei Nichtschlüsseldimensionen
- Ändern des Kennzahlwert-Ermittlertyps
- Ändern der Option **editable**

Datenstrukturrelevante Änderungen

Möchten Sie dagegen Konfigurationsänderungen importieren, die Auswirkungen auf die Struktur einer Datenreihe haben, müssen zunächst gegebenenfalls zuvor importierte Daten der Datenreihe gelöscht werden (über die PPM-Benutzeroberfläche bzw. mit Hilfe des Kommandozeilenprogramms **runpikidata** mit der Option **-mode delete**). Erst danach kann die geänderte Konfiguration mit dem Importparameter **-overwrite** importiert werden.

Sind beim Import datenstrukturrelevanter Konfigurationsänderungen noch Daten in der betreffenden Datenreihe enthalten, wird eine Fehlermeldung ausgegeben und die neue Definition der Datenreihe wird nicht übernommen. Der weitere Import zulässiger Konfigurationsänderungen wird allerdings nicht abgebrochen.

Datenstrukturrelevante Konfigurationsänderungen können sein:

- Löschen einer referenzierten Dimension (Schlüsseldimension, Nichtschlüsseldimension)
- Löschen einer prozessinstanzunabhängigen Kennzahl
- Ändern einer Schlüsseldimension in eine Nichtschlüsseldimension (**iskeydimension="FALSE"**)
- Ändern der Schrittweite einer referenzierten Dimension bzw. des Datentyps einer prozessinstanzunabhängigen Kennzahl

8.2.4.5. Migration von Datenreihen

Bevor Sie bestehende, prozessinstanzunabhängige Datenreihen aus einem PPM-System der Version 4 in ein PPM-System der Version 9 mittels des Kommandozeilenprogramms **runppmconverter.bat** in <Installationsverzeichnis>\ppm\server\bin\agentLocalRepo\.unpacked\<Installationszeit>_ppm-client-<Version>-runnable.zip\ppm\bin übernehmen, müssen Sie gegebenenfalls folgende Punkte beachten:

- Importdaten prozessinstanzunabhängiger Datenreihen aus einer PPM-Datenbank der Version 4 müssen Sie vor einer Konvertierung mittels Export in eine XML-Datei sichern (vgl. **PPM-Migrationshandbuch**).

  Sind vor der Konvertierung bestehender Datenreihen Konfigurationsänderungen notwendig, die sich auf die Struktur von Datenreihen auswirken, müssen gegebenenfalls noch vorhandene Importdaten vor dem Konvertieren gelöscht werden (vgl. Kap. Konfigurationsimport [144]), andernfalls bricht das Konvertierungsprogramm mit einer Fehlermeldung ab.

Verwenden Sie den Kommandozeilenkonvertierer mit der Option **-ignorepikidata**, werden vor dem Konvertieren generell alle vorhandenen Importdaten bestehender Datenreihen vollständig gelöscht.

Nach dem Konvertieren der Konfiguration für PPM der Version 9 können die exportierten Daten wieder importiert werden, siehe Technische Referenz **PPM Datenimport**.

- Datenreihen, die nicht mehr unterstützte, referenzierte Kennzahlen (XML-Elemente **refki**) enthalten, werden nicht automatisch konvertiert. Es wird ein Hinweis vom Konvertierungsprogramm ausgegeben, bestehende Daten der entsprechenden Datenreihe - wenn gewünscht - zu exportieren, die Konfiguration entsprechend anzupassen sowie einen Datenneuimport mit der angepassten Konfiguration durchzuführen.

- Bei Verwendung des nicht mehr unterstützten XML-Attributs **deletedataonredefiniton** in bestehenden, prozessinstanzunabhängigen Datenreihen, muss dieses vor der Konvertierung manuell aus der Konfiguration entfernt werden.
- Datenreihen im PPM-System der Version **9** sind als Typ **PROCESS** vorgegeben. Sind in den zu konvertierenden Datenreihen andere Kennzahltypen (**RELATION**, **OT_FUNC**, **OT_ORG**) vorhanden, können diese nicht automatisch konvertiert werden. Die Konfiguration muss auch in diesem Fall vor der Konvertierung manuell angepasst werden.

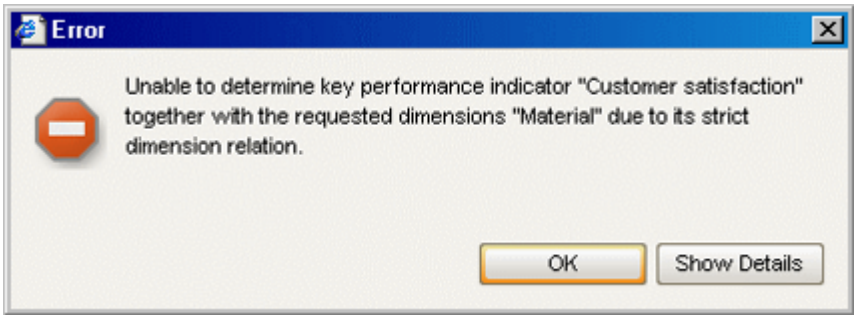
8.2.4.6. Zusätzliche Information: Benutzerdefinierte Kennzahlen auf Basis von prozessinstanzabhängigen Kennzahlen

Wird eine benutzerdefinierte Kennzahl, die auf Basis von prozessinstanzabhängigen Kennzahlen erstellt wurde, in der Analyse mit Dimensionen verwendet, die für alle beteiligten Kennzahlen definiert sind, liefert die benutzerdefinierte Kennzahl nur Werte, wenn für den jeweiligen Dimensionsschritt für alle Kennzahlen Werte ermittelt werden können (Schnittmenge der beteiligten Dimensionswerte).

Beachten Sie bei der Verwendung von prozessinstanzunabhängigen Kennzahlen mit striktem Dimensionsbezug in der Berechnung benutzerdefinierter Kennzahlen die folgenden zusätzlichen Hinweise:

- Werden in einer benutzerdefinierten Kennzahl zwei oder mehr prozessinstanzunabhängige Kennzahlen verwendet, sollte der Wertebereich der einzelnen Dimensionen, auf die sich die prozessinstanzunabhängige Kennzahlen beziehen, identisch sein.
- Um eine prozessinstanzunabhängige Kennzahl in die Berechnungsvorschrift einer benutzerdefinierten Kennzahl einfließen zu lassen, muss mindestens eine Dimension, auf die die prozessinstanzunabhängige Kennzahl Bezug nimmt, im Prozessbaum an derselben Stelle wie die benutzerdefinierte Kennzahl angemeldet sein.

Sind diese beiden Punkte nicht erfüllt, wird beim Aufruf der benutzerdefinierten Kennzahl ein Hinweisdialog ähnlich wie folgendes Beispiel angezeigt:



Beispiel 80.

Die benutzerdefinierte Kennzahl **Gesamtkosten** setzt sich aus Prozesskosten und Gemeinkosten zusammen.

Die Prozesskennzahl **Prozesskosten** liefert Werte auf der Basis von Prozessinstanzen für die Monate Januar bis März 2001 und Juni bis Dezember 2001.

Die prozessinstanzunabhängige Kennzahl **Gemeinkosten** ist mit Werten für Jan bis Jun 2001 gepflegt.

Bei monatsweiser Analyse im Jahr 2001 liefert die benutzerdefinierte Kennzahl **Gesamtkosten** nur Werte für die Monate Januar, Februar, März und Juni 2001.

8.2.5. Definition von Kennzahlengruppen

Kennzahlengruppen werden in der Konfigurationsdatei **KiGroup.xml** definiert. Man unterscheidet zwischen den sichtbaren Kennzahlengruppen (**kigrouproot** bzw. **kigroup**) und genau einer unsichtbaren Kennzahlengruppe (**kigroupinvisible**). Eine Kennzahl kann nur genau einer Gruppe zugeordnet werden.

Das Gruppieren von Kennzahlen stellt keine Hierarchisierung oder Verfeinerung von Kennzahlen dar, sondern verbessert lediglich die Übersichtlichkeit.

Die **kigroup** Elemente können in beliebiger Verschachtelungstiefe angeordnet werden. Dadurch können Sie eine individuelle Ordnerstruktur für Ihre Kennzahlengruppen aufbauen.

Verwenden Sie zur Erstellung von Kennzahlengruppen PPM Customizing Toolkit. Im Modul **Kennzahlen und Dimensionen** können Sie auf einfache Weise die gewünschte Gruppenstruktur definieren.

```
<kigrouproot>
<description language="de" name="Alle Kennzahlen">

  Diese Gruppe umfasst alle angezeigten Kennzahlen.

  </description>

  <description language="en" name="All measures">

  This group includes all displayed measures.

  </description>
<kigroupinvisible>
<description language="de" name="Unsichtbare Kennzahlen">

  Diese Gruppe umfasst alle Kennzahlen, die

  nur zur Berechnung weiterer Kennzahlen verwendet

  werden. Diese Kennzahlen werden nicht in der

  Kennzahlenliste angezeigt.
</description>
<description language="en" name="Invisible measures">

  This group includes all measures

  that are merely used for calculation

  of additional measures. These

  measures are not displayed in

  the measure list.
</description>
</kigroupinvisible>
<kigroup name="KI_GROUP_COST">
<description language="de" name="Kostenkennzahlen"/>
<description language="en" name="Cost KPIs"/>
<kigroup name="KI_GROUP_COST">
```

```
<description language="de" name="Kostenkennzahlen"/>
<description language="en" name="Cost KPIs"/>
<kigroup name="KI_GROUP_COST_ALL">
<description language="de" name="Gesamtkostenkennzahlen"/>
<description language="en" name="Total cost KPIs"/>
</kigroup>
<kigroup name="KI_GROUP_COST_AVERAGE">
<description language="de name="Durchschnittskostenkennzahlen"/>
<description language="en" name="Average cost KPIs"/>
</kigroup>
</kigroup>
</kigroup>
<kigroup name="KI_GROUP_TIME">
<description language="de" name="Zeitenkennzahlen"/>
<description language="en" name="Time KPIs"/>
</kigroup>
<kigroup name="KI_GROUP_QUALITY">
<description language="de" name="Qualitätskennzahlen"/>
<description language="en" name="Quality KPIs"/>
</kigroup>
</kigrouproot>
```

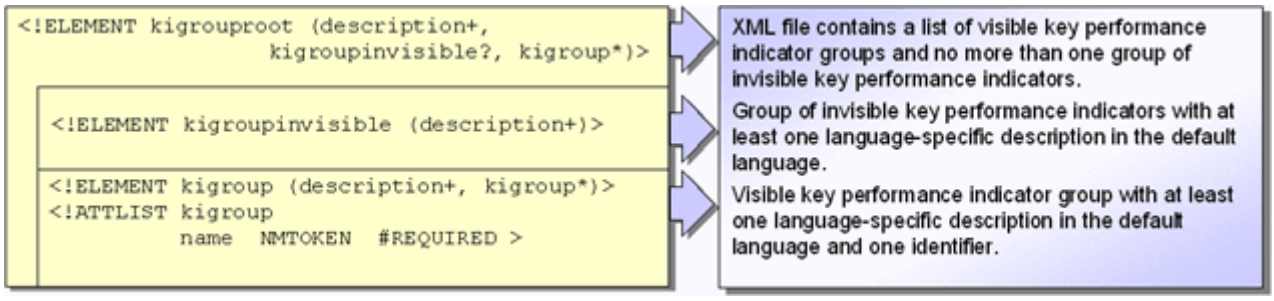
Die Gruppe **Alle Kennzahlen** umfasst die Kennzahlen aller Gruppen und Untergruppen außer den Kennzahlen der Gruppe **Unsichtbare Kennzahlen**. Kennzahlen, die Sie nicht ausdrücklich einer Gruppe zugeordnet haben, werden automatisch der Gruppe **Alle Kennzahlen** zugeordnet. Dieser Gruppe werden auch die Kennzahlen zugeordnet, die Sie einer nicht existierenden Gruppe zugeordnet haben.

Im PPM-Frontend werden auf der Registerkarte **Kennzahlen** alle Kennzahlen der gewählten Kennzahlengruppe und aller Untergruppen angezeigt mit Ausnahme der Kennzahlen der Gruppe **Unsichtbare Kennzahlen**, die nur dann angezeigt werden, wenn diese Gruppe selektiert ist.

Die Gruppe der unsichtbaren Kennzahlen **KI_GROUP_INVISIBLE** befindet sich direkt unter der Wurzel und kann nicht weiter strukturiert werden.

Warnung
Die Gruppenidentifizierer **KI_GROUP_ROOT** und **KI_GROUP_INVISIBLE** sind fest vom PPM-System vergeben und dürfen weder in einem anderen Zusammenhang verwendet noch verändert werden.

Der Aufbau der Konfigurationsdatei **KIGroup.xml** ist durch die DTD **KIGroup.dtd** vorgegeben:



Allgemeine Struktur

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE kigrouproot SYSTEM 'KIGroup.dtd'>
<kigrouproot>
<description language="de" name="...">...</description>
<description language="en" name="...">...</description>
<kigroupinvisible>
<description language="de" name="...">...</description>
<description language="en" name="...">...</description>
</kigroupinvisible>
<kigroup name="...">
<description language="de" name="...">...</description>
<description language="en" name="...">...</description>
<kigroup name="...">
<description language="de" name="..."/>
<description language="en" name="..."/>
<kigroup name="...">
<description language="de" name="...">...</description>
<description language="en" name="...">...</description>
...
</kigroup>
...
</kigroup>
...
</kigroup>
<kigroup name="...">
<description language="de" name="..."/>
<description language="en" name="..."/>
...
</kigroup>
<kigroup name="...">
<description language="de" name="..."/>
<description language="en" name="..."/>
...
</kigroup>
...
</kigrouproot>
```

Beispiel 81.

Das folgende Beispiel veranschaulicht die Anzeige der XML-Datei an der Oberfläche des PPM-Frontend:

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

<kigrouproot>
<description language="de" name="Alle Kennzahlen">

  Diese Gruppe umfasst alle angezeigten Kennzahlen mit

  Ausnahme der unsichtbaren Kennzahlen.

  </description>
<description language="en" name="All measures">
```

```
  This group includes all displayed measures

  except for the invisible ones.

  </description>
<kigroupinvisible>
<description language="de" name="Unsichtbare Kennzahlen">
Diese Gruppe umfasst alle Kennzahlen, die nur

  zur Berechnung weiterer Kennzahlen verwendet werden.

  Diese Kennzahlen werden nicht in der

  Kennzahlenliste angezeigt.

  </description>
<description language="en" name="Invisible KPIs">
This group includes all measures

  that are merely used for calculation of additional

  measures. These measures

  are not displayed in the measure

  list, unless the group of

  invisible measures is selected.

  </description>
</kigroupinvisible>
<kigroup name="KI_GROUP_COST">
<description language="de" name="Kostenkennzahlen"/>
<description language="en" name="Cost KPIs"/>
</kigroup>
<kigroup name="KI_GROUP_TIME">
<description language="de" name="Zeitenkennzahlen"/>
<description language="en" name="Time KPIs"/>
</kigroup>
<kigroup name="KI_GROUP_QUALITY">
<description language="de" name="Qualitätskennzahlen"/>
<description language="en" name="Quality KPIs"/>
</kigroup>
</kigrouproot>
```

8.2.5.1. Sichtbare Kennzahlengruppen

Eine Kennzahlengruppe wird in der Konfigurationsdatei durch folgendes XML-Element definiert:

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

```
<kigrouproot>
<description language="de" name="...">

  Description text...

  </description>
...

  <kigroup name="...">
<description language="de" name="..."/>
<description language="en" name="..."/>
</kigroup>
...
</kigrouproot>
```

XML-Tag	Beschreibung
kigrouproot	Wurzel der Kennzahlengruppen. Wird im PPM-Frontend als oberster Gruppenordner angezeigt.
description	Sprachabhängige Beschreibung der Kennzahlengruppenwurzel. Muss mindestens in der Standardsprache angegeben werden.
kigroup	Zu definierende Kennzahlengruppe. Jede Gruppe kann Untergruppen enthalten. Es können beliebig viele Gruppen und Untergruppen angelegt werden. Die Gruppenstruktur entspricht einer Baumstruktur mit beliebiger Verzweigungstiefe.
name	Interner Name der Gruppe. Wird vom XML-Attribut **kigroup** der Kennzahldefinition (XML-Element **kidef**) in der Kennzahlenkonfiguration referenziert.
description	Sprachabhängige Beschreibung einer Gruppe, optional mit Tooltip (Bereich **#PCDATA** des Elements **description**). Mindestens die Beschreibung in der Standardsprache muss gepflegt sein.

8.2.5.2. Gruppe der unsichtbaren Kennzahlen

Die Gruppe **Unsichtbare Kennzahlen** beinhaltet alle Kennzahlen, die nur dann in der Kennzahlenliste angezeigt werden, wenn die Gruppe selektiert ist. Nur dann stehen sie in der Analyse zur Verfügung. In den Kennzahlenlisten aller anderen Kennzahlengruppen werden die unsichtbaren Kennzahlen nicht angezeigt. Die Gruppe der unsichtbaren Kennzahlen ist eindeutig und nicht strukturierbar.

Ordnen Sie Kennzahlen, die ausschließlich als Zwischenergebnis zur Berechnung weiterer Kennzahlen verwendet werden, der Gruppe der unsichtbaren Kennzahlen zu (**<kidef name="..." kigroup="KI_GROUP_INVISIBLE" ... />**).

Die Gruppe wird in der XML-Konfigurationsdatei ***_kigroup.xml** der jeweiligen PPM CTK-Mandantenvorlage (unter <PPM-Installationsverzeichnis>\ppm\server\bin\agentLocalRepo\.unpacked\<Installationszeit>_ppm-client-run-prod-<Version>-runnable.zip\ppm\ctk\ctk\examples\custom\) durch folgendes Element konfiguriert:

```
...
<kigroupinvisible name="...">
<description language="de" name="..."/>
<description language="en" name=".."/>
</kigroupinvisible>
...
```

8.3. Definition von Dimensionen

Dimensionen werden zusammen mit Kennzahlen in der mandantenspezifischen Kennzahlenkonfiguration definiert. Das PPM-System unterscheidet folgende Dimensionsarten:

Dimension	XML-Element	Beschreibung
Einstufig	**oneleveldim**	Einstufige Dimensionen werden verwendet, wenn die Anzahl der Dimensionswerte klein und keine sinnvolle Gruppierung der Werte möglich ist. Beispiel: Name des Quellsystems, Eingangskanal eines Callcenters (z.B. Anruf, Fax, E-Mail) Attributdatentyp: Alle. Numerische Attributwerte werden in Text umgewandelt.
Zweistufig	**twoleveldim**	Zweistufige Dimensionen werden verwendet, wenn eine sinnvolle Gruppierung der Dimensionswerte möglich ist. Beispiel: Prozesstypgruppe/Prozesstyp, Materialart/Material Attributdatentyp: Alle. Numerische Attributwerte werden in Text umgewandelt. Dimensionswerte werden als Datentyp **TEXTPAIR** gespeichert.
N-stufig	**nleveldim**	Für Textdimensionen mit mehr als zwei Stufen.
Fließ-kommazahl	**floatingdim**	Dimension basiert auf Fließkommawerten. Die Dimensionswerte stellen bestimmte Intervalle dar. Beispiel: Auftragsvolumen Attributdatentyp: **DOUBLE**, **TIMESPAN**, **PERCENTAGE** und alle davon abgeleiteten benutzerdefinierten Typen
Zeit	**timedim**	Zeigt die Veränderung einer Kennzahl über die Zeit an. Attributdatentypen: **DAY**, **TIME**
Tageszeit	**hourdim**	Zeigt die Veränderung einer Kennzahl über die Zeit an. Attributdatentyp: **TIMEOFDAY**
Zeitraum	**timerange**	Zeigt die Veränderung einer Kennzahl innerhalb eines Zeitraumes an.
Suchdimension	**searchdim**	Suche nach Prozessinstanzen anhand von Attributwerten Attributdatentyp: **TEXT** Suchdimensionen können in der Analyse nicht als Dimension angezeigt werden.

Die für die Bildung von Text- und Fließkommazahldimensionen benötigten Attribute werden jeweils im XML-Element **dimitem** angegeben. Handelt es sich bei dem referenzierten Attribut um ein zu berechnendes Attribut, muss dies angegeben werden, damit das Attribut vor dem Bilden der Dimension berechnet werden kann (**calculated="TRUE"**). Zusätzlich muss angegeben werden, ob es sich um eine Prozess- oder Funktionsdimension handelt.

Achten Sie beim Definieren von Dimensionen darauf, dass der Datentyp des referenzierten Attributs zum gewählten Dimensionstyp passt.

8.3.1. Definition von Dimensionengruppen

Dimensionengruppen werden in der Konfigurationsdatei **DimGroup.xml** definiert. Man unterscheidet zwischen sichtbaren Dimensionengruppen (**dimgroup** bzw. **dimgrouproot**) und genau einer unsichtbaren Dimensionengruppe (**dimgroupinvisible**). Eine Dimension kann nur genau einer Gruppe zugeordnet werden.

Das Gruppieren von Dimensionen stellt keine Hierarchisierung oder Verfeinerung von Dimensionen dar, sondern verbessert lediglich die Übersichtlichkeit.

Die **dimgroup** Elemente können in beliebiger Verschachtelungstiefe angeordnet werden. Dadurch können Sie eine individuelle Struktur für Ihre Dimensionengruppen aufbauen.

Verwenden Sie zur Erstellung von Dimensionengruppen PPM Customizing Toolkit. Im Modul **Kennzahlen und Dimensionen** können Sie auf einfache Weise die gewünschte Gruppenstruktur definieren.

Im PPM-Frontend werden auf der Registerkarte **Dimensionen** alle Dimensionen der gewählten Dimensionengruppe und aller Untergruppen angezeigt mit Ausnahme der Dimensionen der Gruppe **Nicht sichtbare Dimensionen**.

Die Gruppe der unsichtbaren Dimensionen **DIM_GROUP_INVISIBLE** befindet sich direkt unter der Wurzel und kann nicht weiter strukturiert werden. Sie enthält alle internen Dimensionen und wird an der Oberfläche nicht angezeigt.

Dimensionen, die Sie keiner Dimensionengruppe zuordnen, werden automatisch der Wurzel **DIM_GROUP_ROOT** (Gruppe **Alle Dimensionen**) zugeordnet.

> **Warnung**
> Die Gruppenidentifizierer **DIM _GROUP_ROOT** und **DIM _GROUP_INVISIBLE** sind fest vom PPM-System vergeben und dürfen weder in einem anderen Zusammenhang verwendet noch verändert werden.

```
<dimgrouproot>
<description name="Alle Dimensionen" language="de">

  Diese Gruppe beinhaltet alle angezeigten Dimensionen.

  </description>
<description name="All dimensions" language="en">
This group includes all displayed dimensions.
</description>
<dimgroupinvisible>
<description name="Nicht sichtbare Dimensionen" language="de">
Diese Gruppe beinhaltet alle internen Dimensionen.
</description>
<description name="Invisible dimensions" language="en">
This group includes all internal dimensions.
</description>
</dimgroupinvisible>
<dimgroup name="DIM_GROUP_CRITERIA">
<description name="Criteria" language="de">
Diese Gruppe beinhaltet Dimensionen, die
als Unterscheidungskriterien dienen.
</description>
<description name="" language="en" />
<dimgroup name="DIM_GROUP_CUST">
<description name="Kundendimensionen" language="de">
Diese Gruppe beinhaltet alle kundenrelevanten
Dimensionen.
</description>
<description name="Customer dimensions" language="en"/>
</dimgroup>
<dimgroup name="DIM_GROUP_PRINC">
<description name="Auftraggeberdimensionen" language="de">
```

```
Diese Gruppe beinhaltet alle
auftraggeberrelevanten Dimensionen.
</description>
<description name="Principal dimensions" language="en"/>
<dimgroup name="DIM_GROUP_USA">
<description name="Auftraggeber in USA" language="de">
Diese Gruppe beinhaltet alle Dimensionen
für Auftraggeber in den USA.
</description>
<description name="Customers USA" language="en" />
</dimgroup>
<dimgroup name="DIM_GROUP_EUROPE">
<description name="Auftraggeber in Europa" language="de">
Diese Gruppe beinhaltet alle Dimensionen für
Auftraggeber in Europa.
</description>
<description name="Principals Europe" language="en"/>
</dimgroup>
</dimgroup>
</dimgroup>
<dimgroup name="DIM_GROUP_TIME">
<description name="Zeitdimensionen" language="de">
Diese Gruppe beinhaltet alle Zeitdimensionen.
</description>
<description name="Time dimensions" language="en" />
</dimgroup>
</dimgrouproot>
```

8.3.2. Textdimensionen

Dieser Dimensionstyp umfasst drei Arten, die für die Definition der jeweiligen Dimensionsstufen gleichartige Konfigurationen verwenden. Es gibt einstufige, zweistufige und n-stufige Textdimensionen. N-stufige Dimensionen können beliebig viele Hierarchiestufen haben.

Jede Definition einer Dimensionsstufe (XML-Element **leveldesc**) setzt sich aus der Angabe eines obligatorischen Schlüssels (erstes XML-Element **dimitem**) und einer optionalen Beschreibung (zweites XML-Element **dimitem**) zusammen. Die sprachabhängigen Oberflächennamen der Dimensionen, Schlüssel und Beschreibungen der einzelnen Stufen (XML-Elemente **description**) müssen in der Standardsprache gepflegt sein. Die einzelnen Schlüssel und Beschreibungen referenzieren Attribute vom Typ **TEXT**, deren Werte von PPM in der Analyse und den Filterdialogen angezeigt werden.

8.3.2.1. Allgemeine XML-Struktur

8.3.2.1.1. Einstufige Dimension

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

```
...
<oneleveldim name="..." dimtype="..." internal="..." importmode="..."
dimgroup="...">
<description language="..." name="..."/>
<leveldesc>
<dimitem attrname="..." colname="..." calculated="..." location="..."
substvalue="...">
<description language="..." name="..."/>
<defaultvalue="..."/>
</dimitem>
<dimitem attrname="..." colname="..." calculated="...">
<description language="..." name="..."/>
</dimitem>
</leveldesc>
</oneleveldim>
...
</keyindicatorconfig>
```

8.3.2.1.2. Zweistufige Dimension

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<twoleveldim name="..." dimtype="..." internal="..." importmode="..."
dimgroup="...">
<description language="..." name="..."/>
<leveldesc>
<dimitem attrname="..." colname="..." calculated="..." location="..."
substvalue="...">
<description language="..." name="..."/>
<defaultvalue="..."/>
</dimitem>
<dimitem attrname="..." colname="..." calculated="...">
<description language="..." name="..."/>
<defaultvalue="..."/>
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="..." colname="..." calculated="...">
<description language="..." name="..."/>
<defaultvalue="..."/>
</dimitem>
<dimitem attrname="..." colname="..." calculated="...">
<description language="..." name="..."/>
<defaultvalue="..."/>
</dimitem>
</leveldesc>
</twoleveldim>
...
</keyindicatorconfig>
```

8.3.2.1.3. N-stufige Dimension

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
<nleveldim name="..." dimtype="..." internal="..." importmode="..."
dimgroup="...">
<description name="..." language="..."/>
<leveldesc>
<dimitem attrname="..." colname="..." caculated="..." location="..."
substvalue="...">
<description language="..." name="..." />
<defaultvalue="..."/>
</dimitem>
<dimitem attrname="..." colname="..." calculated="...">
<description language="..." name="..."/>
<defaultvalue="..."/>
</dimitem>
</leveldesc>
<leveldesc>
...
</leveldesc>
<leveldesc>
...
</leveldesc>
...
</nleveldim>
...
</keyindicatorconfig>
```

Standardmäßig werden in PPM die Werte der einzelnen Dimensionsstufen in der Form **<Beschreibung (Schlüssel)>** angezeigt, wenn Beschreibungen definiert sind. Ansonsten wird nur der Schlüssel als Wert angezeigt.

Beispiel 82.

Beinhaltet das vom ersten **dimitem** referenzierte Attribut die Definition eines Schlüssels **ID** und das vom zweiten **dimitem** referenzierte Attribut die zugehörige Beschreibung **Text**, werden die Dimensionswerte dieser Stufe an der Oberfläche standardmäßig in der Form <**Text**> (**ID**>) angezeigt.

Der Auszug aus der Konfigurationsdatei für ein ähnliches Beispiel sieht folgendermaßen aus.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<oneleveldim name="VTWEG" dimtype="PROCESS" internal="no" importmode="OPTIONAL">
<description language="de" name="Vertriebsweg"/>
<description language="en" name="Distribution channel"/>
<leveldesc>
<dimitem attrname="AT_VTWEG" colname="FIRST_ID" calculated="FALSE">
<description language="de" name="ID des Vertriebsweg"/>
<description language="en" name="ID of distribution channel"/>
```

```
</dimitem>
<dimitem attrname="AT_VTWEG_NAME" colname="FIRST_DESC" calculated="FALSE">
<description language="de" name="Vertriebsweg"/>
<description language="en" name="Distribution channel"/>
</dimitem>
</leveldesc>
</oneleveldim>
...
</keyindicatorconfig>
```

Beispiel 83.

Das XML-Element **dimitem** konfiguriert folgende Einstellungen für den Schlüssel bzw. die Beschreibung einer Dimensionsstufe:

XML-Tag	Beschreibung
attrname	Name des referenzierten Attributes. Es ist nur der Datentyp **TEXT** zulässig.
calculated	**TRUE**: Attributwert wird berechnet. Vorgabewert ist **FALSE**.
location	Nur bei dimtype="RELATION" Gültige Werte: **SOURCE** (Attributplatzierung am Quellbezugsobjekt der Beziehung) **TARGET** (Attributplatzierung am Zielbezugsobjekt der Beziehung) **THIS** (Standardwert: Attribut wird an der Beziehung selbst platziert)
defaultvalue (optional)	Angabe eines Standardwertes, der angezeigt wird, wenn kein Attributwert ermittelt werden kann und auch mittels **substvalue** kein Wert ermittelt wird bzw. werden kann. Gibt es weder Angaben zu **defaultvalue** noch zu **substvalue**, wird bei nicht ermittelbarem Attributwert der Wert des Schlüssels **PPM_NULL** aus der Datei **Database_settings.properties** angezeigt.
substvalue (optional)	Angabe eines Ersatzwertes, der angezeigt wird, wenn kein Attributwert ermittelt werden kann. Als Ersatzwert kann der Attributwert der direkt vorangehenden, gröberen Stufe (**PRED**) oder der direkt nachfolgenden, feineren Stufe (**SUCC**) verwendet werden. Ersetzungen können sich über mehrere, aufeinanderfolgende Stufen erstrecken. Kann mit den Angaben zu **substvalue** kein Wert ermittelt werden (z.B. **substvalue="SUCC"** bei einer einstufigen Dimension), wird kein Ersatzwert angezeigt. Standardwert: **NONE** (kein Ersatzwert)

XML-Element	Beschreibung
compression **wert** (optionales Unterelement zu **dimitem**)	Das interne Verdichtungsattribut **AT_INTERNAL_COMPRESSCRITERION** muss gepflegt sein (Konfiguration des internen Verdichtungsattributes [198]). Nur bei **dimtype="PROCESS"**. Gleiche und unterschiedliche Dimensionswerte werden bei der dauerhaften Verdichtung über Kommandozeile (**runppmcompress**, siehe Benutzerhandbuch **PPM Operation Guide**) gelöscht und durch den angegebenen Verdichtungswert ersetzt (Verdichtungsverhalten ändern [198]).

Ersatz- und Standardwerte bei ein-, zwei- und n-stufigen Dimensionen

Für Schlüssel bzw. Beschreibung jeder einzelnen Dimensionsstufe können Sie Standard- und Ersatzwerte angeben. Diese Werte werden wahlweise angezeigt, wenn kein Attributwert ermittelt werden kann. Die Reihenfolge bei der Wahl des anzuzeigenden Dimensionswertes ist dabei wie folgt:

1. Attributwert
2. Ersatzwert (**substvalue**)
3. Standardwert (**defaultvalue**)

4. DB-Standardwert (Wert des Schlüssels **PPM_NULL** in der Datei **Database_settings.properties**)

Warnung
Die zweistufige Dimension **Prozesstyp** unterstützt weder Standard- noch Ersatzwerte. Wenn Sie in der Konfiguration dazu Angaben machen, werden diese beim Import gelöscht.

```
...
<nleveldim name="SALE" dimtype="PROCESS" dimgroup="DIM_GROUP_CRITERIA">
<description name="Sales" language="en"/>
<leveldesc>
<dimitem attrname="AT_VKORG" colname="NAME_1" calculated="FALSE">
<description language="en" name="Sales organization"/>
</dimitem>
<dimitem attrname="AT_VKORG_NAME" colname="DESC_NAME_1" calculated="FALSE">
<description language="en" name="Name of sales organization"/>
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="AT_DIVISION" colname="NAME_2" calculated="FALSE"
substvalue="SUCC">
<description language="en" name="Division"/>
<defaultvalue>defaultvalue 2nd level ID
</defaultvalue>
</dimitem>
<dimitem attrname="AT_DIVISION_NAME" colname="DESC_NAME_2" calculated="FALSE"
substvalue="SUCC">
<description language="en" name="Division name"/>
<defaultvalue>defaultvalue 2nd level description
</defaultvalue>
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="AT_VTWEG" colname="NAME_3" calculated="FALSE">
<description language="en" name="Distribution channel"/>
</dimitem>
<dimitem attrname="AT_VTWEG_NAME" colname="DESC_NAME_3" calculated="FALSE">
<description language="en" name="Name of distribution channel"/>
</dimitem>
</leveldesc>
</nleveldim>
...
```

Beispiel 84.

Für Schlüssel und Beschreibung der zweiten Stufe der n-stufigen Dimension **SALE** sind Ersatzwerte (**substvalue**) definiert. Diese Ersatzwerte werden von der nachfolgenden dritten Stufe übernommen werden. Kann kein Ersatzwert ermittelt werden, wird der angegebene Standardwert (**defaultvalue**) angezeigt.

Textdimensionen basieren in der Regel auf alphanumerischen Attributtypen. Die Dimensionswerte werden an der Oberfläche in alphanumerischer Sortierung dargestellt.

Bei Verwendung von Attributen, die auf numerischen Datentypen basieren, werden die Dimensionswerte als Zeichenketten in die Datenbank geschrieben und an der Oberfläche numerisch sortiert dargestellt.

8.3.2.2. Konfiguration

8.3.2.2.1. Einstufige Dimensionen

Eine einstufige Dimension (XML-Element **oneleveldim**) wird durch genau eine Stufendefinition (XML-Element **leveldesc**) vollständig beschrieben. Die sprachabhängigen Namen von Dimension, Schlüssel und Beschreibung werden in den XML-Elementen **description** angegeben.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<oneleveldim name="VTWEG" dimtype="PROCESS" internal="no" importmode="OPTIONAL">
<description language="de" name="Vertriebsweg"/>
<description language="en" name="Distribution channel"/>
<leveldesc>
<dimitem attrname="AT_VTWEG" colname="FIRST_ID" calculated="FALSE">
<description language="de" name="ID des Vertriebsweg"/>
<description language="en" name="ID of distribution channel"/>
</dimitem>
<dimitem attrname="AT_VTWEG_NAME" colname="FIRST_DESC" calculated="FALSE">
<description language="de" name="Vertriebsweg"/>
<description language="en" name="Distribution channel"/>
</dimitem>
</leveldesc>
</oneleveldim>
...
</keyindicatorconfig>
```

XML-Attribut	Beschreibung
name	Name der Dimension. Unter diesem Namen wird in der Datenbank eine Tabelle angelegt. Für den angegebenen Namen gelten die in Kapitel **Tabellenname** beschriebenen Richtlinien.
dimtype	Dimensionstyp: Gültige Werte: **PROCESS** (Prozessdimension) **FUNCTION** (Funktionsdimension, veraltet, nur noch aus Kompatibilitätsgründen zu verwenden) **OT_FUNC** (Funktionsdimension) **RELATION** (Beziehungsdimension) **OT_ORG** (Organisationsdimension)
internal	Interne Verwendung der Dimension **yes**: Die Dimension wird intern verwendet und nicht an der Oberfläche angezeigt. Vorgabewert ist **no**.
importmode	Ausgabe von Fehlermeldungen beim Berechnen von Dimensionswerten **OPTIONAL**: Berechnungsfehler werden nicht ausgegeben. **MANDATORY**: Berechnungsfehler werden ausgegeben Standardwert: **OPTIONAL**

XML-Attribut	Beschreibung
dimgroup (optional)	Dimensionengruppe, der die Dimension zugeordnet wird

XML-Element	Beschreibung
compression **wert** (optionales Unterelement zu **dimitem**)	Das interne Verdichtungsattribut **AT_INTERNAL_COMPRESSCRITERION** muss gepflegt sein (Konfiguration des internen Verdichtungsattributes [198]). Nur bei **dimtype="PROCESS"**. Gleiche und unterschiedliche Dimensionswerte werden bei der dauerhaften Verdichtung über Kommandozeile (**runppmcompress**, siehe Benutzerhandbuch **PPM Operation Guide**) gelöscht und durch den angegebenen Verdichtungswert ersetzt (Verdichtungsverhalten ändern [198]).

Wenn Sie den Dimensionsschritt **nicht gepflegt** einer Dimension vermeiden möchten, erreichen Sie durch Angabe von **importmode="MANDATORY"**, dass Prozessinstanzen, die keinem Dimensionsschritt zugeordnet werden können, durch Ausgabe eines entsprechenden Hinweises bereits beim Datenimport erkannt werden.

Interne Dimensionen

Interne Dimensionen (**internal="yes"**) werden nicht an der Oberfläche des PPM-Frontend angezeigt. Sie werden für die Unterscheidung administrativer Prozessinstanzmerkmale verwendet.

Verwendung von Multibyte-Zeichensätzen

Der folgende Dateiauszug aus der Kennzahlenkonfiguration zeigt beispielhaft die Definitionsmöglichkeiten einstufiger Dimensionen unter Verwendung eines Multibyte-Zeichensatzes:

```
...
<oneleveldim name="D_PRODUCT_GR" dimtype="FUNCTION" internal="no"
importmode="OPTIONAL">
<description name="Produktgruppe" language="de"/>
<description name="Product group" language="en"/>
<description name="Ομάδα προϊόντων" language="el"/>
<leveldesc>
<dimitem attrname="ΙΔ_ΣΥΝ_ΠΡΟΪΟΝΤ_ΤΑΥΤ" colname="Column name_3"
calculated="FALSE">
<description language="de" name="ID Produktgruppe"/>
<description language="en" name="Product group ID"/>
<description language="el" name="Ταυτότητα ομάδας προϊόντων"/>
</dimitem>
<dimitem attrname="ΙΔ_ΣΥΝ_ΠΡΟΪΟΝΤ_ΠΕΡΙΓΡ" colname="Column name_4"
calculated="FALSE">
<description language="de" name="Beschreibung Produktgruppe"/>
<description language="en" name="Product group description"/>
<description language="el" name="Περιγραφή της ομάδας προϊόντων"/>
</dimitem>
</leveldesc>
</oneleveldim>
...
```

8.3.2.2.2. Zweistufige Dimensionen

Zweistufige Dimensionen (XML-Element **twoleveldim**) werden analog den einstufigen Dimensionen konfiguriert, mit dem Unterschied, dass sie aus zwei Stufendefinitionen (XML-Elemente **leveldesc**)

bestehen. Die Verwendung von Multibyte-Zeichensätzen erfolgt analog zu einstufigen Dimensionen (Kapitel Einstufige Dimension [154]).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<twoleveldim name="PROCESSTYPE" dimtype="PROCESS" importmode="OPTIONAL">
<description language="de" name="Prozesstyp"/>
<description language="en" name="Process type"/>
<leveldesc>
<dimitem attrname="AT_PROCTYPEGROUP" colname="PROCTYPEGROUP" calculated="FALSE">
<description language="de" name="Prozesstypgruppe"/>
<description language="en" name="Process type group"/>
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="AT_PROCTYPE" colname="PROCTYPE" calculated="FALSE">
<description language="de" name="Prozesstyp"/>
<description language="en" name="Process type"/>
</dimitem>
</leveldesc>
</twoleveldim>
...
</keyindicatorconfig>
```

XML-Attribut	Beschreibung
name	Interner Name der Dimension. Unter diesem Namen wird in der Datenbank eine Tabelle angelegt. Für den angegebenen Namen gelten die in Kapitel **Tabellenname** beschriebenen Richtlinien.
dimtype	Dimensionstyp: Gültige Werte: **PROCESS** (Prozessdimension) **FUNCTION** (Funktionsdimension, veraltet, nur noch aus Kompatibilitätsgründen zu verwenden) **OT_FUNC** (Funktionsdimension) **RELATION** (Beziehungsdimension) **OT_ORG** (Organisationsdimension)
internal	Interne Verwendung der Dimension **yes**: Die Dimension wird intern verwendet und nicht an der Oberfläche angezeigt. Vorgabewert ist **no**.
importmode	Ausgabe von Fehlermeldungen beim Berechnen von Dimensionswerten **OPTIONAL**: Berechnungsfehler werden nicht ausgegeben. **MANDATORY**: Berechnungsfehler werden ausgegeben Standardwert: **OPTIONAL**
dimgroup (optional)	Dimensionengruppe, der die Dimension zugeordnet wird

XML-Element	Beschreibung
compression **wert** (optionales Unterelement zu **dimitem**)	Das interne Verdichtungsattribut **AT_INTERNAL_COMPRESSCRITERION** muss gepflegt sein (Konfiguration des internen Verdichtungsattributes [198]). Nur bei **dimtype="PROCESS"** Gleiche und unterschiedliche Dimensionswerte werden bei der dauerhaften Verdichtung über Kommandozeile (**runppmcompress**, siehe Benutzerhandbuch **PPM Operation Guide**) gelöscht und durch den angegebenen Verdichtungswert ersetzt (Verdichtungsverhalten ändern [198]).

8.3.2.2.3. N-stufige Dimensionen

N-stufige Dimensionen werden analog zu ein- bzw. zweistufigen Dimensionen konfiguriert. Eine n-stufige Dimension besteht mindestens aus einer und maximal aus **<n>** Stufen (XML-Elemente **leveldesc**). Jede Stufe enthält einen Schlüssel (erstes XML-Element **dimitem**) und optional eine Beschreibung (zweites XML-Element **dimitem**). Alle sprachabhängigen Bezeichnungen (Name von Dimension, Schlüssel bzw. Beschreibung) werden mit dem XML-Element **description** angegeben. Die XML-Attribute stimmen mit denen der ein- bzw. zweistufigen Dimensionen überein.

Beispiel 85.

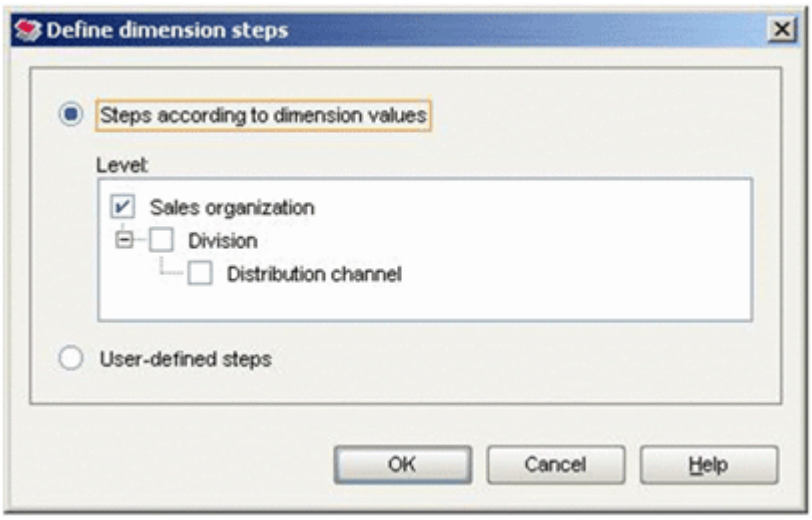
(n-stufige Dimension mit drei Stufen, für die jeweils ein Schlüssel und eine Beschreibung definiert sind)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<nleveldim name="SALE" dimtype="PROCESS" dimgroup="DIM_GROUP_CRITERIA">
<description name="Sales" language="de"/>
<leveldesc>
<dimitem attrname="AT_VKORG" colname="NAME_1"calculated="FALSE">
<description language="de" name="Verkaufsorganisation"/>
</dimitem>
<dimitem attrname="AT_VKORG_NAME" colname="DESC_NAME_1" calculated="FALSE">
<description language="de" name="Name der sales organization"/>
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="AT_DIVISION" colname="NAME_2" calculated="FALSE">
<description language="de" name="Division"/>
</dimitem>
<dimitem attrname="AT_DIVISION_NAME" colname="DESC_NAME_2" calculated="FALSE">
<description language="de" name="Division name"/>
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="AT_VTWEG" colname="NAME_3" calculated="FALSE">
<description language="de" name="Vertriebsweg"/>
</dimitem>
<dimitem attrname="AT_VTWEG_NAME" colname="DESC_NAME_3" calculated="FALSE">
<description language="de" name="Name des Vertriebswegs"/>
</dimitem>
</leveldesc>
</nleveldim>
...
```

```
</keyindicatorconfig>
```

Beispiel 86.

Die Dimension **Vertrieb** wird an der PPM Oberfläche wie folgt angezeigt:



Die Verwendung von Multibyte-Zeichensätzen erfolgt analog zu einstufigen Dimensionen (Kapitel Einstufige Dimension [159]).

8.3.2.3. Import von Dimensionswerten

Für ein-, zwei- bzw. n-stufige Dimensionen können vor dem eigentlichen PPM-Import mit dem Kommandozeilenprogramm **rundimdata** Daten importiert werden. Die Werte werden dabei pro Dimensionsstufe als Paare in der Form **<Schlüssel-Beschreibung>** importiert, wobei die Angabe des Schlüssels obligatorisch, die der Beschreibung optional ist.

Der Import von Dimensionswerten für Textdimensionen vor dem eigentlichen PPM-Import ist im Handbuch PPM Datenimport ausführlich beschrieben.

8.3.3. Fließkommadimensionen

Fließkommadimensionen (XML-Element **floatingdim**) werden analog zu einstufigen Dimensionen konfiguriert, mit dem Unterschied, dass das mittels XML-Element **dimitem** referenzierte Attribut ein numerischer Datentyp sein muss (z.B. **DOUBLE**, **LONG, TIMESPAN**, **FACTORYTIMESPAN, FREQUENCY, PERCENTAGE,** sowie alle benutzerdefinierten Datentypen, wie z.B. **COST**).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<floatingdim name="ORDERVOL" dimtype="PROCESS" importmode="OPTIONAL">
```

```
<description language="de" name="Auftragsvolumen"/>
<description language="en" name="Order size"/>
<dimitem attrname="AT_ORDERVOL" colname="ORDERVOL" calculated="FALSE">
<description language="de" name="Umsatz"/>
<description language="en" name="Sales revenues"/>
</dimitem>
</floatingdim>
...
</keyindicatorconfig>
```

XML-Tag	Beschreibung
name	Interner Name der Dimension. Unter diesem Namen wird in der Datenbank eine Tabelle angelegt. Für den angegebenen Namen gelten die in Kapitel **Tabellenname** beschriebenen Richtlinien.
dimtype	Dimensionstyp: Gültige Werte: **PROCESS** (Prozessdimension) **FUNCTION** (Funktionsdimension, veraltet, nur noch aus Kompatibilitätsgründen zu verwenden) **OT_FUNC** (Funktionsdimension) **RELATION** (Beziehungsdimension) **OT_ORG** (Organisationsdimension)
dimitem location (optional)	Nur bei **dimtype="RELATION"** Gültige Werte: **SOURCE** (Attributplatzierung am Quellbezugsobjekt der Beziehung) **TARGET** (Attributplatzierung am Zielbezugsobjekt der Beziehung) **THIS** (Standardwert: Attribut wird an der Beziehung selbst platziert)
importmode (optional)	Ausgabe von Fehlermeldungen beim Berechnen von Dimensionswerten **OPTIONAL**: Berechnungsfehler werden nicht ausgegeben. **MANDATORY**: Berechnungsfehler werden ausgegeben Standardwert: **OPTIONAL**

8.3.4. Dimensionen Zeit

Die Konfigurationsmerkmale einer Zeitdimension sind im XML-Element **timedim** zusammengefasst:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<timedim name="DAY" dimtype="PROCESS" precision="DAY" attrname="AT_DAY"
calculated="TRUE" importmode="OPTIONAL">
<description language="de" name="Tag"/>
<description language="en" name="Day"/>
</timedim>
...
</keyindicatorconfig>
```

XML-Tag	Beschreibung
name	Interner Name der Dimension, wird im Paramset angezeigt. Unter diesem Namen wird in der Datenbank eine Tabelle angelegt. Für den angegebenen Namen gelten die in Kapitel **Tabellenname** beschriebenen Richtlinien.
dimtype	Dimensionstyp: Gültige Werte: **PROCESS** (Prozessdimension) **FUNCTION** (Funktionsdimension, veraltet, nur noch aus Kompatibilitätsgründen zu verwenden) **OT_FUNC** (Funktionsdimension) **RELATION** (Beziehungsdimension) **OT_ORG** (Organisationsdimension)
location (optional)	Nur bei **dimtype="RELATION"** Gültige Werte: **SOURCE** (Attributplatzierung am Quellbezugsobjekt der Beziehung) **TARGET** (Attributplatzierung am Zielbezugsobjekt der Beziehung) **THIS** (Standardwert: Attribut wird an der Beziehung selbst platziert)
attrname	Interner Name des referenzierten Attributs
precision (optional)	Feinste Schrittweite der Dimension (**DAY**, **HOUR**, **MINUTE**)
calculated	**TRUE**: Attributwert muss berechnet werden. Standardwert: **FALSE**.
internal	Kennzeichnung als interne Dimension mit **yes**. Standardwert:**nein**
earlyalert	Kennzeichnung als kritische Dimension des Frühwarnsystems mit **yes**. Standardwert:**nein**
importmode (optional)	Ausgabe von Fehlermeldungen beim Berechnen von Dimensionswerten **OPTIONAL**: Berechnungsfehler werden nicht ausgegeben. **MANDATORY**: Berechnungsfehler werden ausgegeben Standardwert: **OPTIONAL**
dimgroup (optional)	Name der Dimensionengruppe, der die Zeitdimension zugeordnet werden soll.
deleteon compression	Das interne Verdichtungsattribut **AT_INTERNAL_COMPRESSCRITERION** muss gepflegt sein (Konfiguration des internen Verdichtungsattributes [198]). Nur bei **dimtype="PROCESS"** **TRUE**: Gleiche und unterschiedliche Dimensionswerte werden bei der dauerhaften Verdichtung über Kommandozeile (**runppmcompress**) - (siehe Benutzerhandbuch **PPM Operation Guide**) gelöscht (Verdichtungsverhalten ändern [198]). **FALSE**: Bei der dauerhaften Verdichtung über Kommandozeile werden gleiche Dimensionswerte in die verdichtete EPK übernommen, unterschiedliche Werte dagegen gelöscht. Standardeinstellung: **FALSE**

8.3.4.1. Zeitdimensionen für das Frühwarnsystem

Für die Überprüfung kritischer Zeitpunkte einzelner Prozessinstanzen im Rahmen des Frühwarnsystems definieren Sie in der Kennzahlenkonfiguration spezielle Zeitdimensionen. Diese Zeitdimensionen sind durch das Attribut **earlyalert** mit dem Wert **yes** gekennzeichnet (PPM CTK: Kontrollkästchen **Frühwarnung** wählen). Für alle anderen Zeitdimensionen hat dieses Attribut standardmäßig den Wert **no**.

- Zur besseren Unterscheidung sollte der Bezeichner der Dimension (**name**) mit dem Präfix **CRIT_** beginnen.

- Da kritische Zeitdimensionen nur intern auf Prozessinstanzebene berechnet werden, muss das Attribut **internal** den Wert **yes** (PPM CTK: entsprechendes Kontrollkästchen in Spalte **Intern** wählen) und das Attribut **dimtype** den Wert **PROCESS** haben (PPM CTK: Wert **PROCESS** in Spalte **Verwendung** wählen). Anderenfalls wird beim Konfigurationsimport eine Fehlermeldung ausgegeben.

Zeitdimensionen des Frühwarnsystems werden als interne Dimensionen nicht in der Dimensionenliste der Navigation angezeigt. Aus diesem Grund ist die Zuordnung zu einer Dimensionengruppe nicht sinnvoll und wird ignoriert.

8.3.4.2. Besonderheit der Berechnung kritischer Zeitattribute

Sie müssen in der Berechnungsvorschrift (**calcattr**) für das kritische Zeitattribut dem XML-Attribut **delete** den Wert **yes** zuweisen (PPM CTK: Kontrollkästchen **Attributwert löschen** wählen).

Beispiel 87.

Auszug der Berechnungsvorschrift für das Attribut **Kritischer Warenausgangstermin**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM "KeyindicatorConfiguration.dtd">
<keyindicatorconfig>
...
<calcattr name="AT_CRITICAL_WAUS_DATE" type="PROCESS" delete="yes">
...
</keyindicatorconfig>
```

Verwenden Sie PPM Customizing Toolkit zur Definition, Berechnung und Anmeldung kritischer Zeitdimensionen.

Im Modul **Kennzahlen und Dimensionen** können Sie in der Komponente **Dimensionen** auf einfache Weise die erforderlichen Definitionsangaben machen. Die entsprechenden Definitionen der Berechnungsvorschriften für die kritischen Zeitpunkte können Sie in der Komponente **Berechnete Attribute** anlegen.

Melden Sie die kritischen Zeitdimensionen im Modul **Prozesse** auf der Registerkarte **Prozessanalyse-Dimensionen** der Komponente **Prozessbaum** an den gewünschten Prozesstypgruppen bzw. Prozesstypen an.

Beispiel 88.

Im folgenden Beispiel wird der **kritische Warenausgangstermin** berechnet. Als kritisch sollen dabei Instanzen gelten, in denen spätestens vier Tage nach Ausführen der Funktion **Lieferung anlegen** (**SAP.LIEF**) kein Warenausgang verbucht wurde, das heißt, die Funktion **Warenausgang buchen** (**SAP.WAUS**) kommt in der Prozessinstanz nicht vor. Mit der Berechnungsvorschrift für das Attribut **AT_CRITICAL_WAUS_DATE** wird der kritische Zeitpunkt berechnet als Addition einer Zeitspanne von vier Tagen (**354600 Sekunden**) zum Zeitpunkt des insgesamt frühesten Auftretens der Funktion **Lieferung anlegen**.

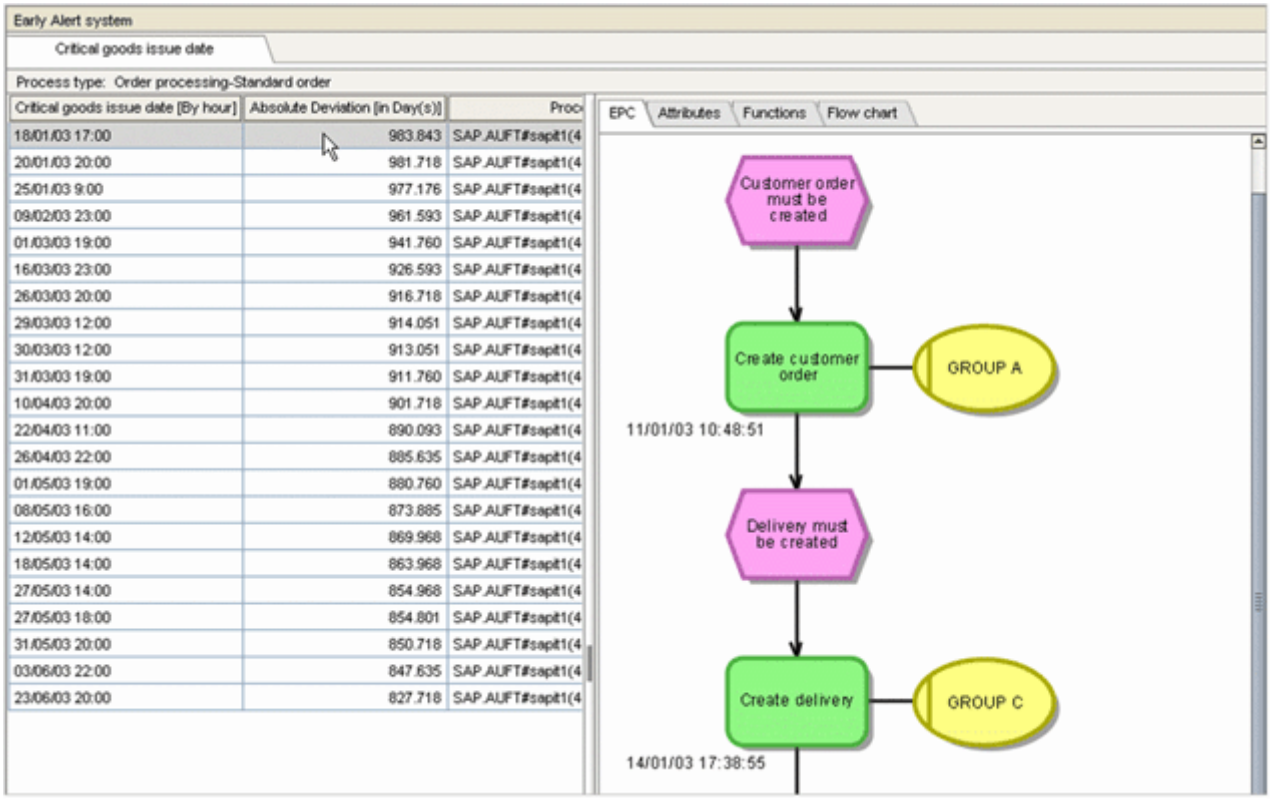
Die Berechnung erfolgt stundengenau (XML-Attribut **precision="hour"** der Dimension **CRT_TIME_WAUS**). Ist der früheste Bezugszeitpunkt (**AT_TIME**) der Funktion **Lieferung anlegen** in einer Prozessinstanz bspw. der **13.07.03 20:26:55**, ergibt sich gemäß obiger Berechnungsvorschrift als kritischer Warenausgangstermin in stundengenauer Angabe der **17.07.03 20:00**.

Das Ergebnis eines berechneten, kritischen Zeitattributs muss immer in einem Zeitstempelformat vorliegen (Datentyp **TIME**).

Folgender Auszug aus der Kennzahlenkonfiguration zeigt die Definition der kritischen Zeitdimension **CRIT_TIME_WAUS**. Durch die Attribut-Wert-Kombination **earlyalert="yes"** ist sie als Frühwarnsystem-Zeitdimension gekennzeichnet.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM "KeyindicatorConfiguration.dtd">
<keyindicatorconfig>
...
<calcattr name="AT_CRITICAL_WAUS_DATE" type="PROCESS" delete="yes">
<calculation>
<if>
<not>
<exists>
<attribute name="AT_OBJNAME_INTERN" nodetype= "OT_FUNC" objectname="SAP.WAUS"
onerror="CONTINUE"/>
</exists>
</not>
<then>
<addtimespan>
<min>
<attribute name="AT_TIME" nodetype="OT_FUNC" objectname="SAP.LIEF"
onerror="EXIT_NO_WARNING"/>
</min>
<constant>
<dataitem value="345600.0">
4,000
<datatype name="TIMESPAN">Time span
</datatype>
<scale name="DAY" factor="86400.0">Day(s)
</scale>
</dataitem>
</constant>
</addtimespan>
</then>
</if>
</calculation>
</calcattr>
...
<timedim name="CRIT_TIME_WAUS" dimtype="PROCESS"
precision="HOUR" attrname="AT_CRITICAL_WAUS_DATE"
calculated="TRUE" internal="yes" earlyalert="yes"
importmode="OPTIONAL">
<description name="kritischer Warenausgangstermin" language="de"/>
</timedim>
...
</keyindicatorconfig>
```

Die Überprüfung der Überschreitung kritischer Warenausgangstermine erfolgt in der Komponente **Frühwarnsystem** des **Instanz Controlling** im PPM-Frontend. Im dem Beispiel wird davon ausgegangen, dass die kritische Dimension **CRIT_TIME_WAUS** am Prozesstyp **Auftragsabwicklung\Terminauftrag** angemeldet ist.



Die zum Ausführungszeitpunkt aktuelle Abweichung vom kritischen Warenausgangstermin wird im Analysebereich in der Spalte **Absolute Abweichung [Tag(e)]** angegeben. Ausführungszeitpunkt der Frühwarnüberprüfung ist die aktuelle Systemzeit. Im dem Beispiel werden nach der Ausführung der Frühwarnüberprüfung für die Dimension **Kritischer Warenausgangstermin** alle Prozessinstanzen des Prozesstyps **Terminauftrag** in einer Prozessinstanztabelle angezeigt, für die eine Überschreitung des kritischen Warenausgangstermins ermittelt wurde. Für die in der Abbildung gewählte Prozessinstanz wurde eine aktuelle Überschreitung des kritischen Warenausgangstermins von 983,843 Tagen ermittelt. Das ist die zum aktuellen Ausführungszeitpunkt (im Beispiel der 10.08.05 15:31) absolute Abweichung in Tagen vom kritischen Warenausgangstermin.

Alternativ können Sie kritische Prozessinstanzen über das Kommandozeilenprogramm **runppmanalytics** mit der Option **-earlyalert** ermitteln (siehe Benutzerhandbuch **PPM Operation Guide**).

8.3.5. Zeitraumdimensionen

Zeitraumdimensionen sind spezielle Zeitdimensionen. Sie ermöglichen Zustände von Prozessen auf Basis eines vergangenen Zeitraums (Startzeitpunkt bis Endzeitpunkt) zu betrachten.

Es werden die folgenden drei Varianten unterschieden.

- Stichtagsbasierte Zeitraumdimension, bezogen auf den Startzeitpunkt
- Stichtagsbasierte Zeitraumdimension, bezogen auf den Endzeitpunkt
- Intervallbasierte Zeitraumdimension

Zeitraumdimensionen werden durch das XML-Element **timerangedim** definiert.

Beispiel 89. *[en-us]*

Eine Zeitraumdimension wird beispielhaft mit folgender Syntax konfiguriert.

```
<timerangedim name="RANGEDIM_KEYWORD" reference="END" dimtype="PROCESS">
<startattribute name="AT_START_TIME" calculated="TRUE"/>
<endattribute name="AT_END_TIME" calculated="TRUE"/>
<description name="Display name" language="en"></description>
</timerangedim>
```

XML-Tag	Beschreibung
timerangedim	
▪ name	Interner Name der Dimension, wird im Paramset angezeigt. Unter diesem Namen wird in der Datenbank eine Tabelle angelegt. Für den angegebenen Namen gelten die in Kapitel **Tabellenname** beschriebenen Richtlinien.
▪ dimtype	Dimensionstyp: Gültige Werte: **PROCESS** (Prozessdimension) **FUNCTION** (Funktionsdimension, veraltet, nur noch aus Kompatibilitätsgründen zu verwenden) **OT_FUNC** (Funktionsdimension) **RELATION** (Beziehungsdimension) **OT_ORG** (Organisationsdimension)
▪ reference	Definiert, ob es sich um eine stichtagsbasierte Zeitraumdimension mit Startzeitpunkt (**value = "START"**), mit Endzeitpunkt (**value="END"**) oder um eine intervallbasierte Zeitraumdimension (**value="RANGE"**) handelt. Angabe = optional, Standardwert = "**END**", Eine nachträgliche Änderung des Typs **START**, **END** oder **RANGE** ist nicht erlaubt und wird durch den Konfigurationsimport verhindert.
▪ internal	Kennzeichnung als interne Dimension mit **yes**. Standardwert:**nein**
▪ importmode	Ausgabe von Fehlermeldungen beim Berechnen von Dimensionswerten **OPTIONAL**: Berechnungsfehler werden nicht ausgegeben. **MANDATORY**: Berechnungsfehler werden ausgegeben Standardwert: **OPTIONAL**
▪ dimgroup	Name der Dimensionengruppe, der die Zeitdimension zugeordnet werden soll. Angabe: Optional
startattribute/ endattribute	
▪ name	Spezifizieren das EPK-Attribut, aus dem der Dimensionswert der Startzeit bzw. Endzeit berechnet werden soll Attributtyp = TIME, Angabe = obligatorisch
▪ calculated	**TRUE**: Attributwert muss berechnet werden. Standardwert: **FALSE**.
▪ location	Nur bei **dimtype="RELATION"** Gültige Werte: **SOURCE** (Attributplatzierung am Quellbezugsobjekt der Beziehung) **TARGET** (Attributplatzierung am Zielbezugsobjekt der Beziehung) **THIS** (Standardwert: Attribut wird an der Beziehung selbst platziert) Angabe: Optional
description	
▪ name	Sprachabhängiger Oberflächenname der Dimension im PPM-Frontend

XML-Tag	Beschreibung
▪ language	Sprache, in der der Oberflächenname angezeigt wird

8.3.6. Tageszeitdimensionen

Die Konfigurationsmerkmale einer Tageszeitdimension sind im XML-Element **hourdim** zusammengefasst.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<hourdim name="TIMEOFDAY" dimtype="PROCESS" attrname="AT_TIME_OF_DAY"
tablename="DAYTIME" precision="SECOND" calculated="TRUE" importmode="OPTIONAL">
<description language="de" name="Uhrzeit"/>
<description language="en" name="Time of day"/>
</hourdim>
...
</keyindicatorconfig>
```

Für jede Schrittweite einer Tageszeitdimension (Stunde, Minute, Sekunde) wird in der Datenbanktabelle der Dimension eine eigene Spalte angelegt.

XML-Tag	Beschreibung
name	Name der Dimension. Unter diesem Namen wird in der Datenbank eine Tabelle angelegt. Für den angegebenen Namen gelten die in Kapitel **Tabellenname** beschriebenen Richtlinien.
dimtype	Dimensionstyp: Gültige Werte: **PROCESS** (Prozessdimension) **FUNCTION** (Funktionsdimension, veraltet, nur noch aus Kompatibilitätsgründen zu verwenden) **OT_FUNC** (Funktionsdimension) **RELATION** (Beziehungsdimension) **OT_ORG** (Organisationsdimension)
location (optional)	Nur bei **dimtype="RELATION"** Gültige Werte: **SOURCE** (Attributplatzierung am Quellbezugsobjekt der Beziehung) **TARGET** (Attributplatzierung am Zielbezugsobjekt der Beziehung) **THIS** (Standardwert: Attribut wird an der Beziehung selbst platziert)
attrname	Name des referenzierten Attributs
precision	Feinste Schrittweite der Dimension (**HOUR**, **MINUTE**, **SECOND**). Standardwert: **HOUR**
calculated	**TRUE**: Attributwert muss berechnet werden. Standardwert: **FALSE**
importmode	Ausgabe von Fehlermeldungen beim Berechnen von Dimensionswerten **OPTIONAL**: Berechnungsfehler werden nicht ausgegeben. **MANDATORY**: Berechnungsfehler werden ausgegeben Standardwert: **OPTIONAL**

XML-Tag	Beschreibung
deleteon **compression**	Das interne Verdichtungsattribut **AT_INTERNAL_COMPRESSCRITERION** muss gepflegt sein (Konfiguration des internen Verdichtungsattributes [198]). Nur bei **dimtype="PROCESS"** **TRUE**: Gleiche und unterschiedliche Dimensionswerte werden bei der dauerhaften Verdichtung über Kommandozeile (**runppmcompress**) - (siehe Benutzerhandbuch **PPM Operation Guide**) gelöscht (Verdichtungsverhalten ändern [198]). **FALSE**: Bei der dauerhaften Verdichtung über Kommandozeile werden gleiche Dimensionswerte in die verdichtete EPK übernommen, unterschiedliche Werte dagegen gelöscht. Standardeinstellung: **FALSE**

8.3.7. Suchdimensionen

Mit diesem speziellen Dimensionstyp können Sie anhand bestimmter Werte eines Suchattributs nach Prozessinstanzen suchen. Die Suchdimension wirkt wie ein Filter auf die Menge der aktuell verfügbaren Prozessinstanzen. Der eingestellte Filterausdruck kann wie bei den anderen Dimensionstypen bearbeitet bzw. entfernt werden. Es können mehrere Suchdimensionsfilter in einer Analyse gleichzeitig verwendet werden.

Die Angabe eines Suchkriteriums erfolgt im Frontend mit dem Kontextmenü **Filter bearbeiten** der Suchdimension. Alternativ können Suchkriterien im Prozessinstanzsuche-Assistenten angegeben werden.

Ein Suchkriterium besteht aus einer Zeichenfolge. Optional kann am Ende der Zeichenfolge einer der Platzhalter **?**oder ***** angegeben werden. Das Zeichen ***** steht für eine beliebige Zeichenfolge, das Zeichen **?** steht für genau ein beliebiges Zeichen.

Bei der Konfiguration von Suchdimensionen ergeben sich gegenüber den anderen Dimensionstypen folgende Vereinfachungen:

- Suchdimensionen basieren ausschließlich auf Prozessattributen
- Suchattribute müssen vom Typ **TEXT** sein

Die Konfigurationsmerkmale einer Suchdimension sind im XML-Element **searchdim** zusammengefasst:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<searchdim name="BELEGNR">
<description language="de" name="Suche Belegnummer"/>
<description language="en" name="Searching for document number"/>
<dimitem attrname="AT_SAP_BELEGNR" colname="BELEGNR" calculated="TRUE">
<description language="de" name="Belegnummer"/>
<description language="en" name="Document number"/>
</dimitem>
</searchdim>
...
</keyindicatorconfig>
```

Für jede Suchdimension wird in der Datenbanktabelle eine eigene Spalte angelegt.

XML-Tag	Beschreibung
name	Interner Name der Dimension. Unter diesem Namen wird in der Datenbank eine Tabelle angelegt. Für den angegebenen Namen gelten die in Kapitel **Tabellenname** beschriebenen Richtlinien.
description **name**	Sprachabhängiger Oberflächenname der Dimension im PPM-Frontend
dimitem	Definition einer Spalte in der Datenbanktabelle. Jede Suchdimension besitzt genau ein **dimitem**-Element, in welchem die Dimension beschrieben wird.
attrname	Bezeichner des Attributes, das als Grundlage der Suchdimension verwendet wird
colname	Name der Datenspalte im Info-Cube. Fehlt die Angabe, wird der Dimensionsname verwendet. Für den angegebenen Namen gelten die in Kapitel **Tabellenname** beschriebenen Richtlinien.
calculated	**TRUE**: Attributwert muss berechnet werden. Standardwert: **FALSE**
compression **wert** (optionales Unterelement zu **dimitem**)	Das interne Verdichtungsattribut **AT_INTERNAL_COMPRESSCRITERION** muss gepflegt sein (Konfiguration des internen Verdichtungsattributes [198]). Nur bei **dimtype="PROCESS"** Gleiche und unterschiedliche Dimensionswerte werden bei der dauerhaften Verdichtung über Kommandozeile (**runppmcompress**, siehe Benutzerhandbuch **PPM Operation Guide**) gelöscht und durch den angegebenen Verdichtungswert ersetzt (Verdichtungsverhalten ändern [198]).

Suchdimensionen werden in den Info-Cube der Prozesstypgruppe geschrieben. Wenn Sie mehrere Suchdimensionen in einer Prozesstypgruppe verwenden, müssen Sie verschieden Namen für die Datenspalten vergeben.

8.3.8. Variantendimension

Mit PPM 10.1 wurde der neue Dimensionstyp **VARIANTDIM** eingeführt.

Die Varianten klassifizieren Prozessinstanzen nach ihrer Struktur. Die entsprechende Struktur ist die Reihenfolge der Funktionen in einer Prozessinstanz. **Variante** hat zwei Dimensionsstufen, **Kombinierte Variante** (grobe Schrittweite) und **Genaue Variante** (verfeinerte Schrittweite), und den **Namen** als Dimensionswert.

Weitere grundlegende Informationen zu Varianten, eine Liste mit betroffenen Funktionen und Informationen darüber, wie die Variantenfunktion zu PPM hinzugefügt wird, sind in der Dokumentation **PPM Customizing Toolkit** zu finden.

8.3.8.1. Attributekonfiguration

Die folgenden Attributtypen sind als Attribute für den Dimensionstyp **VARIANTEN** definiert. Sie sind vom Typ **Text** und Teil der Attributekonfiguration.

Attributtypen

<attributedefinition key="AT_INTERNAL_FUNCTION_FLOW_VARIANT" type="TEXT" group="AG_INTERNAL" />

<attributedefinition key="AT_INTERNAL_PRECISE_VARIANT" type="TEXT" group="AG_INTERNAL" />

Attributnamen

<attribute key="AT_INTERNAL_FUNCTION_FLOW_VARIANT" name="Internal combined variant" />

<attribute key="AT_INTERNAL_PRECISE_VARIANT" name="Internal precise variant" />

Aufgrund einer spezifischen Semantik ist es nicht empfehlenswert, Werte zu importieren oder einen Attributberechner zu definieren. Siehe Kapitel Verwendung von Variantenattributen beim Import [174].

8.3.8.2. Kennzahlenkonfiguration - Dimensionstyp

Das **variantdim** Element wird benutzt um die Variatendimension zu definieren. Es hat nur drei Attribute und ist geschachtelt mit **leveldesc** Elementen um die Attribute zu konfigurieren, die es gewohnt sind zwei Stufen zu speisen.

Attribute des Elements **variantdim** sind:

XML-Tag	Beschreibung
name	Passwort wie benutzt in paramset (erforderlich)
comment	Kommentar für Dimension (optional)
dimgroup	Name der Gruppe, welcher diese Dimension zugehörig ist (optional)

Jede Dimensionsstufe (leveldesc) hat genau ein **dimitem** als Wert und eine Beschreibung ist nicht für beide Dimensionsstufen erlaubt. Nur das **attrname**-Element ist für das **dimitem**-Element erforderlich.

XML-Tag	Beschreibung
attrname	Name des referenzierten Attributs, das die Dimensionsdaten enthält. Es ist nur der Datentyp **TEXT** zulässig.

In der Regel ist die Variantendimension folgendermaßen definiert.

```
<variantdim name="D_EPC_VARIANT">
<description language="en" name="Variant" />
<description language="de" name="Variante" />
<leveldesc>
<dimitem attrname="AT_INTERNAL_FUNCTION_FLOW_VARIANT">
<description language="en" name="Combined variant" />
<description language="de" name="Kombinierte Variante" />
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="AT_INTERNAL_PRECISE_VARIANT">
<description language="en" name="Precise variant" />
<description language="de" name="Präzise Variante" />
</dimitem>
</leveldesc>
</variantdim>
```

8.3.8.3. Prozessbaumkonfiguration

Die Variantendimensionen können – ähnlich wie alle anderen Dimensionstypen – durch Verwendung des Elements **<usedim>** Prozessbaumknoten zugeordnet werden.

<usedim name="D_EPC_VARIANT" />

Optional kann die <usedim>-Zuordnung ein Standard-Refinement enthalten. Ist dies nicht der Fall, wird die gröbste Stufe als Refinement verwendet, ähnlich wie bei allen anderen **Text**-Dimensionstypen.

8.3.8.4. Verwendung von Variantenattributen beim Import

Die Verwendung von Variantenattributen beim Merge und Attribut-Mapping ist zwar nicht verboten, es wird jedoch unbedingt davon abgeraten. Die Werte von Variantenprozessattributen werden von der Variantenberechnung überschrieben, die erst erfolgt, nachdem bereits alle anderen Berechnungen verarbeitet wurden. Also können weder die Ergebnisse der Variantenberechnung verwendet werden noch enthalten die vollständig verarbeiteten EPKs die Werte, die außerhalb der Variantenberechnung eingestellt wurden.

Es besteht eine spezielle Gefahr beim Befüllen der Dimensionsattribute durch Berechnungsregeln oder durch Mapping: Wenn Sie dies tun, während die Variantendimension nicht im Prozesstyp der EPK angemeldet ist, werden diese Werte nicht überschrieben. Wenn Sie die Dimension dann anmelden, den Analyseserver neu initialisieren und die EPK nicht neu berechnen, wird sie ungültige Dimensions-IDs enthalten, zum Beispiel würde in der GUI eine den Werten entsprechende Variante angezeigt werden, die jedoch keiner tatsächlichen Variante in der Datenbank entspräche. Um die Situation zu korrigieren, können Sie runppmimport mit **–keyindicator new** verwenden. (Durch Verwendung des Kommandozeilenparameters **–ps** zur Spezifizierung einer geeigneten Abfrage können Sie die Neuberechnung auf die EPKs des Prozesstyps beschränken.)

Detaillierte Informationen zur Verwendung von runppmimport finden sich im PPM Operation Guide.

8.3.9. Shared Function Dimension

Standardmäßig werden Shared Function durch den einmaligen Import eines Shared Fragment in das System übernommen. Durch Anwenden der Shared-Fragment-Regeln werden die in den eingelesenen Shared Fragments enthaltenen Shared Functions mit normalen Prozessinstanzfragmenten zusammengeführt, wobei alle Objekte des Shared Fragment in die Fragmentinstanz kopiert werden. Dadurch ist die Eindeutigkeit einer Shared Function automatisch sichergestellt und zur Unterscheidung von Shared Functions kann die Funktions-ID einer Shared Function verwendet werden.

Wenn Sie Shared Functions als normale Fragmentinstanzen mittels Event-Format direkt importieren, wird für jede importierte Funktion eine eindeutige ID erzeugt. Die Funktions-ID kann nicht als Erkennungsmerkmal einer Shared Function verwendet werden. In diesem Fall können Sie eine Shared-Function-Dimension definieren, wobei identische Dimensionswerte Funktionen zu Shared Functions zusammenfassen. Nicht gepflegte Dimensionswerte werden nicht berücksichtigt.

Eine Shared-Function-Dimension hat folgende Eigenschaften:

- Pro Mandant kann nur eine einzige Shared-Function-Dimension definiert werden.
- Die Shared-Function-Dimension muss an Prozessbaumwurzel angemeldet werden.
- Die Shared-Function-Dimension ist an der Oberfläche nicht sichtbar.
- Die Shared-Function-Dimension kann nicht mit PPM CTK angepasst werden.

Beispiel 90.

Das folgende Beispiel zeigt Auszüge aus den Kennzahlen- und Prozessbaumkonfigurationsdateien. Für Funktionen, die ein Attribut **AT_IS_SHARED_FUNCTION** mit Wert **true** gepflegt haben, konkateniert die Berechnungsvorschrift des für die Shared-Function- Dimension verwendeten Funktionsattributes **AT_SHARED_FUNCTION_ID** den internen Funktionsnamen mit dem Zeitstempel der Funktionsausführung.

Dateiauszug keyindicator.xml

```
...
<calcattr name="AT_SHARED_FUNCTION_ID" type="OT_FUNC" delete="yes">
<calculation>
<if>
<and mode="PPM4">
<exists mode="PPM4">
<filteredattribute name="AT_IS_SHARED_FUNCTION" nodetype="OT_FUNC"
objectname="this" onerror="CONTINUE" filter="EARLY" />
</exists>
<eq mode="PPM4">
<filteredattribute name="AT_IS_SHARED_FUNCTION" nodetype="OT_FUNC"
objectname="this" onerror="CONTINUE" filter="EARLY" />
<constant>
<dataitem value="TRUE">
TRUE
<datatype name="BOOLEAN">Logical value</datatype>
</dataitem>
</constant>
</eq>
</and>
<then>
<concat mode="PPM4">
<set mode="PPM4">
<filteredattribute name="AT_OBJNAME_INTERN" nodetype="OT_FUNC"
objectname="this" onerror="EXIT_WARNING" filter="EARLY" />
<convert datatype="TEXT">
<filteredattribute name="AT_END_TIME" nodetype="OT_FUNC" objectname="this"
onerror="EXIT_WARNING" filter="EARLY" />
</convert>
</set>
</concat>
</then>
</if>
</calculation>
</calcattr>
...
<sharedfunctiondim name="SHARED_FUNCTION">
<description name="Shared Function" language="de" />
<description name="Shared Function" language="en" />
<dimitem attrname="AT_SHARED_FUNCTION_ID" colname="SHARED_FUNCTION"
calculated="TRUE">
<description language="de" name="SHARED_FUNCTION" />
<description language="en" name="SHARED_FUNCTION" />
</dimitem>
</sharedfunctiondim>
...
```

Dateiauszug processtree.xml

```
...
<usesfdim name="SHARED_FUNCTTION" />
<processtypegroup name="Standard order" dbtablename="CUBE1">
<processtype name="Order processing" autovisible="FALSE" />
```

```
</processtypegroup>
...
```

8.3.10. Verwenden von Organisationseinheiten als Dimensionen

Das Verwenden einer Organisationseinheit als Dimension stellt keinen eigenen Dimensionstyp dar. Durch Angabe einer speziellen Berechnungsvorschrift wird der Name der Organisationseinheit als Attribut an die jeweiligen Funktionen der Prozessinstanz kopiert.

Beispiel 91.

Folgende Dateiauszüge der Kennzahlenkonfiguration veranschaulichen das Erstellen einer Dimension aus Organisationseinheiten:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE keyindicatorconfig SYSTEM 'KeyindicatorConfiguration.dtd'>
<keyindicatorconfig>
...
<calcattr name="AT_ORGUNIT" type="OT_FUNC">
<calcclass name=
"com.idsscheer.ppm.server.keyindicator.attributecalculator.ZAttributeCalculatorO
riginator"/>
</calcattr>
...
<oneleveldim name="ORGUNIT" dimtype="FUNCTION" internal="no">
<description language="de" name="Processor"/>
<leveldesc>
<dimitem attrname="AT_ORGUNIT" colname="FIRST_ID" calculated="TRUE">
<description language="de" name="Processor"/>
</dimitem>
</leveldesc>
</oneleveldim>
...
</keyindicatorconfig>
```

Beispiel 92.

An jeder Funktionsinstanz wird das Attribut **AT_ORGUNIT** erzeugt, dem als Wert der Name der Organisationseinheit zugewiesen wird. Dieses Attribut wird zur Bildung der einstufigen Funktionsdimension **ORGUNIT** verwendet.

Das Attribut **AT_ORGUNIT** gehört zu den Standardattributen des PPM-Systems und braucht nicht definiert zu werden.

8.4. Definition von Datenzugriffsdimensionen

Mit der Konfiguration von Datenzugriffsdimensionen können Sie Datenzugriffsrechte vergeben, die zusätzlich zu Prozesszugriffsrechten die Möglichkeit bieten, den Zugriff auf PPM-Daten zu steuern.

Datenzugriffsrechte werden für Benutzergruppen vergeben und an die jeweils zugeordneten Benutzer vererbt. Der Administrator (PPM-Benutzer mit Funktionsrecht **Benutzerverwaltung**) definiert die

Datenzugriffsrechte durch Festlegen bestimmter Filter auf Dimensionen, die vom Benutzer nicht bearbeitet werden können. Diese Dimensionen werden Datenzugriffsdimensionen genannt und in der Konfiguration des Prozessbaumes durch das XML-Element **roledim** angegeben. Das Element **roledim** muss eine bereits konfigurierte Textdimension (Kapitel Textdimensionen [154]) referenzieren, die an der Wurzel des Prozessbaumes angemeldet sein muss. Dadurch ist sichergestellt, dass Datenzugriffsdimensionen im gesamten Prozessbaum verwendet werden können. Dabei sind für das Element **roledim** nur ein- und zweistufige Textdimensionen erlaubt.

Wenn Sie möchten, dass eine Datenzugriffsdimension nicht an der PPM-Oberfläche angezeigt wird, geben Sie in der Definition der Dimension das XML-Attribut **internal="yes"** an.

Beispiel 93.

In der Konfigurationsdatei des Prozessbaums werden die Datenzugriffsdimensionen **Auftraggeber** und **Verkaufsorganisation** wie folgt angegeben:

```
...
<roledim name="VKORG"/>
<roledim name="PRINCIPAL" refinement="BY_LEVEL1"/>
...
<usedim name="VKORG"/>
<usedim name="PRINCIPAL" refinement="BY_LEVEL1" scale="LEVEL1SCALE"/>
...
```

Die beiden Dimensionen stehen in der Rechteverwaltung als Datenzugriffsdimensionen zur Verfügung.

Ein PPM-Benutzer erbt die Datenzugriffsrechte aller Benutzergruppen, denen er zugeordnet ist. Die Datenzugriffsrechte werden dabei wie folgt verknüpft:

- Unterschiedliche Datenzugriffsrechte für dieselbe Dimensionen werden durch ODER verknüpft.
- Datenzugriffsrechte für unterschiedliche Dimensionen werden durch UND verknüpft. Ist der Benutzer einer Benutzergruppe zugeordnet, die das Datenzugriffsrecht Keine hat, wird Keine nicht berücksichtigt.
- Wenn ein Benutzer mindestens einer Benutzergruppe angehört, die das Datenzugriffsrecht Alle hat, wird dieses Datenzugriffsrecht nicht durch Datenzugriffsrechte anderer Gruppen, denen er angehört, eingeschränkt.

Ein Benutzer, der keiner Benutzergruppe zugeordnet ist, hat keine Datenzugriffsrechte.

Sonderfall

Wenn Sie Datenzugriffsrechte für unterschiedliche Dimensionen durch ODER verknüpfen möchten, fassen sie die Werte dieser Dimensionen mit Hilfe des Attributberechners zu einer neuen unsichtbaren Dimension zusammen und geben Sie die berechnete Dimension als Datenzugriffsdimension an.

Beispiel 94.

Sie möchten Datenzugriffsrechte für die beiden Dimensionen **Standort 1** und **Standort 2** so vergeben, dass ein Benutzer Daten sieht, wenn das Werk **München** in einer der Dimensionen **Standort 1** oder **Standort 2** vorkommt.

Alle Dimensionswerte der beiden Dimensionen werden in der zu berechnenden Dimension **Standort 3** zusammengefasst. Diese wird in der Konfiguration des Prozessbaumes als Datenzugriffsdimension angegeben.

Standort 1	Standort 2	Standort 3
München	Berlin	München_Berlin
Stuttgart	Leipzig	Stuttgart_Leipzig
Hamburg	München	Hamburg_München
Saarbrücken	Hamburg	Saarbrücken_Hamburg

Durch Verwenden des Filterausdrucks ***München*** wird die gewünschte Datenzugriffsberechtigung erstellt.

8.4.1. Verwenden von Datenzugriffsdimensionen

Ein PPM-Benutzer, der sich unter Verwendung von einschränkenden Datenzugriffsrechten anmeldet, sieht nur Daten, die für ihn freigegeben sind. Aus Sicht des Gesamtsystems wirkt sich die Verwendung von Datenzugriffsdimensionen wie folgt aus:

Prozesszugriffsrechte

Für einen Benutzer festgelegte Prozesszugriffsrechte werden unabhängig von Datenzugriffsrechten ausgewertet. Innerhalb der für ihn freigegebenen Prozesstypen kann ein Benutzer nur die Daten analysieren, die seinen Datenzugriffsrechten entsprechen.

Datenanalyse

Jede Analyseanfrage wird automatisch um die Filter der für den angemeldeten Benutzer gültigen Zugriffsdimensionen erweitert. Die Filter mehrerer Datenzugriffsdimensionen werden durch UND verknüpft.

Filterdialoge

Wenn für einen Benutzer die Datenzugriffsrechte eingeschränkt sind, werden im Filterdialog der entsprechenden Datenzugriffsdimension nur die für diesen Benutzer freigegebenen Dimensionswerte zur Auswahl angezeigt.

Planwerte

Ein Benutzer kann Planwerte nur für Daten erstellen, für die er Datenzugriffsrechte hat. Wie bei der Datenanalyse wird automatisch der Filter der für den angemeldeten Benutzer gültigen Zugriffsdimensionen zur Planwertdefinition hinzugefügt.

Sind Planwerte definiert, für die der für den angemeldeten Benutzer gültige Filter der Datenzugriffsdimension nur ein Teil des für den Planwert gültigen Filters ist, so wird der Planwert angezeigt, kann vom Benutzer jedoch nicht bearbeitet werden.

Verdichten und Löschen

Ein Benutzer kann nur die Daten verdichten und löschen, für die er Datenzugriffsrechte hat. Wie bei der Datenanalyse wird automatisch der Filter der für den angemeldeten Benutzer gültigen Zugriffsdimensionen verwendet.

Beim persistenten Verdichten wird automatisch über die Datenzugriffsdimensionen iteriert, damit die Zuordnung der verdichteten Prozessinstanzen zu den Datenzugriffsdimensionen erhalten blieb.

Datenimport

Der Import von Daten erfolgt uneingeschränkt. Wenn ein Benutzer das Recht Datenimport hat, kann er auch Daten importieren, für die er nach Abschluss des Imports keine Zugriffsberechtigung hat.

Prozessinstanzunabhängige Daten

Das Verhalten von Datenzugriffsrechten gilt uneingeschränkt auch für prozessinstanzunabhängige Kennzahlen und Dimensionen. Verwenden Sie hierfür beliebige Dimensionen der importierten prozessinstanzunabhängigen Daten als Datenzugriffsdimensionen.

8.5. Definition des Prozessbaums

Die Definition des Prozessbaumes ist in einer XML-Datei angegeben. Diese Datei hat folgende Struktur:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE processtree SYSTEM "KeyindicatorProcesstree.dtd">
<processtree name="...">
<!-- ROOT - Definition -->
<processparamset>
<paramset>
...
</paramset>
</processparamset>
<functionparamset>
<paramset>
...
</paramset>
</functionparamset>
<!-- Standard measures -->
<useki ... >
<usepidim ... />
</useki>
<!-- Default dimensions -->
<usedim ... "/>
<!-- Start of process tree definition -->
<processtypegroup name="Auftragsabwicklung">
<processparamset>
...
</processparamset>
<functionparamset>
...
</functionparamset>
<useki ... >
<usepidim ... />
</useki>
<usedim ... "/>
<processtype name="Other orders">
</processtype>
...
</processtypegroup>
<processtypegroup name="...">
<processparamset>
...
</processparamset>
<useki ... />
<usedim ... "/>
<processtype name="..." autovisible="...">
</processtype>
...
```

```
</processtypegroup>
...
</processtree>
```

XML-Element	Beschreibung
processtree	Name des Prozessbaumes. Wird als Wurzel angezeigt. Die Definition des Prozessbaumes beinhaltet folgende Angaben: - Standardabfrage für Prozesse und Funktionen - im gesamten Baum verfügbare Kennzahlen und Dimensionen - mindestens eine Prozesstypgruppe
processparamset	Spezifiziert jeweils eine Standardabfrage (Standard-Paramset) für die Wurzel des Prozessbaumes, jede Prozesstypgruppe und jeden Prozesstyp. Die Standardabfrage wird beim Start der Analysekomponente des PPM-Frontend dargestellt. Die Standardabfrage lässt sich jederzeit über das Kontextmenü **Standardabfrage anzeigen** des Prozessbaumes aufrufen.
functionparamset	Spezifiziert jeweils eine Standardabfrage (Standard-Paramset) für jeden Funktionstyp.
paramset	Beschreibt die Darstellung der Standardabfragen als Analyse in XML-Notation.
useki	Ordnet die angegebene Kennzahl den gewünschten Elementen des Prozessbaums zu (XML-Elemente **processtree**, **processtypegroup**, **processtype**).
usedim	Ordnet die angegebene Dimension den gewünschten Elementen des Prozessbaums zu (XML-Elemente **processtree**, **processtypegroup**, **processtype**).
userelki	Ordnet die angegebene Beziehungskennzahl den gewünschten Elementen des Prozessbaums zu (XML-Elemente **processtree**, **processtypegroup**, **processtype**).
usereldim	Ordnet die angegebene Beziehungsdimension den gewünschten Elementen des Prozessbaums zu (XML-Elemente **processtree**, **processtypegroup**, **processtype**).
usepidim **(optional)**	Meldet die angegebene Dimension am selben Prozessbaumelement an, an dem die mit **useki** angegebene proressinstanzunabhängige Kennzahl angemeldet ist. Diese Angabe ist nur dann erforderlich, wenn die Dimension für das Prozessbaumelement nicht bereits durch Anmeldung oder Vererbung verfügbar ist.
processtypegroup	Definiert eine Prozesstypgruppe. Die Definition einer Prozesstypgruppe beinhaltet folgende Angaben: - Name der Prozesstypgruppe - Standardabfrage für Prozesse und Funktionen der Prozesstypgruppe - Kennzahlen und Dimensionen, die der Prozesstypgruppe zusätzlich zu den globalen Kennzahlen und Dimensionen zugeordnet sind - mindestens einen Prozesstyp
processtype	Definiert einen Prozesstyp. Die Definition eines Prozesstyps beinhaltet folgende Angaben: - Name des Prozesstyps - Standardabfrage für Prozesse und Funktionen des Prozesstyps - Kennzahlen und Dimensionen, die dem Prozesstyp zusätzlich zu denen der Prozesstypgruppe zugeordnet sind Durch die optionale Angabe **autovisible="TRUE"** haben neu angelegte PPM-Benutzer automatisch Zugriffsrechte für diesen Prozesstyp. Die Voreinstellung ist **FALSE**, das heißt, neu angelegte PPM-Benutzer haben zunächst kein Zugriffsrecht für diesen Prozesstyp.

Die Namen von Prozesstypen und Prozesstypgruppen des Prozessbaums müssen eindeutig sein und den in der Prozesstypdefinition verwendeten Namen entsprechen. Deshalb werden sie nur einsprachig in der Sprache des Quellsystems angegeben.

8.5.1. Anmeldung von Kennzahlen und Dimensionen am PPM-System

Die in der mandantenspezifischen Kennzahlenkonfiguration (**KeyindicatorConfiguration.xml**) definierten Kennzahlen und Dimensionen müssen dem PPM-System durch Anmeldung in der Prozessbaumkonfiguration (**ProcessTree.xml**) bekannt gemacht werden. Erst dadurch sind sie nach erfolgreichem Konfigurationsimport für Analysen und Berechnungen an der PPM-Oberfläche verfügbar.

Kennzahlen und Dimensionen werden einzelnen Funktions- oderProzesstypen bzw.einzelnen Prozesstypgruppen oder auch allen Prozesstypgruppen (globale Kennzahlen und Dimensionen) zugeordnet.

Kennzahlen und Dimensionen einer Prozesstypgruppe werden automatisch an untergeordnete Prozesstypen vererbt. Bei Wahl der Prozessbaumwurzel werden nur die Kennzahlen und Dimensionen angezeigt, die allen Prozesstypgruppen zugeordnet sind.

8.5.1.1. Kennzahl anmelden

Eine Kennzahl wird in der Konfigurationsdatei des Prozessbaumes durch folgendes XML-Element am PPM-System angemeldet:

```
<useki name="..." assessment="..." scale="..." refinement="..."/>
```

XML-Tag	Beschreibung
name	Interner Name der Kennzahl. Die Kennzahl selbst wird im XML-Element **kidef** der Kennzahlkonfiguration definiert.
assessment	Bewertung der Kennzahl. **POS** gibt an, dass große Kennzahlwerte positiv bewertet werden. **NEG** gibt an, dass niedrige Kennzahlwerte positiv bewertet werden.
scale (optional)	Standardskalierung der Kennzahl. Angegeben wird eine Einheit des Attributdatentyps, auf dem die Kennzahl basiert (z.B. Einheit **HOUR** bei Verwendung des Datentyps **TIMESPAN**). Die Skalierung kann in der Analyse geändert werden. Fehlt die Angabe, ermittelt PPM automatisch einen Wert zur optimalen Darstellung.
refinement (optional)	Standardschrittweite der Kennzahl bei Verwendung als Iteration. Der Wert muss mit Einheit angegeben werden (z.B. 2.5 **PER_DAY**). Die Schrittweite kann über das Kontextmenü der Kennzahl in der Analyse geändert werden.

8.5.1.1.1. Beziehungskennzahl anmelden

Eine Beziehungskennzahl wird in der Konfigurationsdatei des Prozessbaumes durch folgendes XML-Element am PPM-System angemeldet:

```
<userelki name="..." relname="..." assessment="..." scale="..."
refinement="..."/>
```

Dabei wird ein und dieselbe Beziehungskennzahl gegebenenfalls für jede Beziehung einzeln angemeldet.

XML-Tag	Beschreibung
name	Interner Name der Beziehungskennzahl. Die Kennzahl selbst wird im XML-Element **kidef** der Kennzahlkonfiguration definiert.
relname	Interner Name der Beziehung, für die die Beziehungskennzahl verfügbar sein soll
assessment	Kapitel Kennzahl anmelden [181]

XML-Tag	Beschreibung
scale (optional)	Kapitel Kennzahl anmelden [181]
refinement (optional)	Kapitel Kennzahl anmelden [181]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE processtree SYSTEM "KeyindicatorProcesstree.dtd">
<processtree name="Processes">
<processparamset>
...
</processparamset>
<functionparamset>
...
</functionparamset>
...
<processtypegroup name="Customer Services" dbtablename="CUBE6">
<processparamset>
...
</processparamset>
<userelki name="RNUMA" relname="REL_CARRY_OUT" assessment="POS"/>
<userelki name="RNUMA" relname="REL_PING_PONG" assessment="POS"/>
...
<processtype name="..." autovisible="TRUE">
<typifierrule function="..." priority="..."/>
...
<userelki name="..." relname="..."/>
...
</processtype>
...
</processtypegroup>
...
</processtree>
```

8.5.1.1.2. Kennzahlen und Dimensionen prozessinstanzunabhängiger Datenreihen anmelden

Prozessinstanzunabhängige Kennzahlen und referenzierte Dimensionen prozessinstanzunabhängiger Datenreihen müssen am Prozessbaum angemeldet werden, bevor sie in der Analyse verwendet werden können.

Prozessinstanzunabhängige Kennzahlen werden wie prozessinstanzabhängige Kennzahlen am Prozessbaum durch das XML-Element **useki** angemeldet.

Eine prozessinstanzunabhängige Kennzahl kann an mehreren Prozesstypgruppen bzw.Prozesstypen angemeldet werden.

Beispiel 95. ***[en-us]***

Die prozessinstanzunabhängige Kennzahl **OVERHEAD_COSTS** wird im Prozessbaum an der Prozesstypgruppe **Auslieferung** angemeldet. Basiseinheit der Kennzahl ist **EUR**.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE processtree SYSTEM "KeyindicatorProcesstree.dtd">
<processtree name="Processes">
...
<processtypegroup name="Shipping">
...
<useki name="OVERHEAD_COSTS" scale="EUR" assessment="NEG"/>
...
</processtypegroup>
...
</processtree>
```

8.5.1.1.3. Sonderfall: Referenzierte Dimensionen anmelden

Wenn eine prozessinstanzunabhängige Kennzahl sich auf Dimensionen bezieht, die nicht am selben Prozessbaumelement wie die prozessinstanzunabhängige Kennzahl verfügbar sind, müssen diese Dimensionen mit dem XML-Element **usepidim** innerhalb des **useki**-Elements am Prozessbaumelement der prozessinstanzunabhängigen Kennzahl angemeldet werden.

Beispiel 96. ***[en-us]***

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE processtree SYSTEM "KeyindicatorProcesstree.dtd">
<processtree name="Processes">
...
<processtypegroup name="Shipping">
...
<useki name="OVERHEAD_COSTS" scale="EUR" assessment="NEG">
<usepidim name="PRINCIPAL"/>
</useki>
<usedim name="D_COLOR" />
<usedim name="D_EQUIPMENT" />
<usedim name="D_PRODUCT" />
<processtype name="...">
...
</processtype>
</processtypegroup>
...
</processtree>
```

Die Dimension Auftraggeber (**PRINCIPAL**) wird zusammen mit der Anmeldung der prozessinstanzunabhängigen Kennzahl **OVERHEAD_COSTS** an der Prozesstypgruppe **Auslieferung** angemeldet, an der die Dimension selbst nicht mittels **usedim**-Element angemeldet ist.

Die so angemeldete, von der prozessinstanzunabhängigen Datenreihe referenzierte Dimension dient ausschließlich zu Analysezwecken in Zusammenhang mit der prozessinstanzunabhängigen Kennzahl **OVERHEAD_COSTS**.

Mit dem XML-Element **usepidim** angemeldete Dimensionen können nicht zur Kennzahlenberechnung auf Prozessinstanzebene verwendet werden.

8.5.1.2. Dimension anmelden

Eine Dimension wird in der Konfigurationsdatei des Prozessbaumes durch folgendes XML-Element am PPM-System angemeldet:

```
<usedim name="..." scale="..." refinement="..." variance="..."/>
```

XML-Tag	Beschreibung
name	Interner Name der Dimension. Die Dimension selbst wird in einem der XML-Elemente **oneleveldim**, **twoleveldim**, **floatingdim**, **timedim**, **hourdim** oder **searchdim** der Kennzahlkonfiguration definiert.
scale (optional)	Standardskalierung der Dimension. Die Skalierung kann über das Kontextmenü der Dimension in der Analyse geändert werden. Fehlt die Angabe, ermittelt PPM automatisch einen Wert zur optimalen Darstellung.
refinement (optional)	Standardschrittweite der Dimension. Die möglichen Angaben sind vom Attributdatentyp, auf dem die Dimension basiert, abhängig: - **numerisches Attribut**: Wertangabe mit Einheit - **alphanumerisches Attribut** (nochmals datentypabhängig): - **Text**: Keine Angaben möglich. Iterationsschritte sind durch die unterschiedlichen Attributwerte fest vorgegeben. - **Textpaar BY_LEVEL1**, **BY_LEVEL2** (grob, detailliert): Für jede Dimensionsstufe sind die Iterationsschritte durch die Attributwerte fest vorgegeben. - **Zeit**: **BY_YEAR**, **BY_QUARTER**, **BY_MONTH**, **BY_WEEK**, **BY_DAY**, **BY_HOUR**, **BY_MINUTE** Fehlt die Angabe, ermittelt PPM automatisch einen Wert zur optimalen Darstellung.
Variance (optional)	Wurde für künftige Erweiterungen definiert und wird z. Zt. nicht verwendet.

8.5.1.2.1. Bezugsdimension anmelden

Die für die Definition von Beziehungen verwendeten Bezugsdimensionen werden an der Prozessbaumwurzel angemeldet, damit sie in allen Prozesstypgruppen und Prozesstypen verfügbar sind. Die Anmeldung erfolgt dabei über das XML-Element **usereldim** für jede Beziehung getrennt.

XML-Tag	Beschreibung
name	Interner Name der Bezugsdimension. Die Dimension selbst wird in einem der XML-Elemente **oneleveldim** bzw. **twoleveldim** der Kennzahlenkonfiguration definiert.
relname	Interner Name der Beziehung (Kapitel Definition von Beziehungen [189]), für die die Bezugsdimension verfügbar sein soll
scale (optional)	Kapitel Dimension anmelden [184]
refinement (optional)	Kapitel Dimension anmelden [184]
Variance (optional)	Kapitel Dimension anmelden [184]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE processtree SYSTEM "KeyindicatorProcesstree.dtd">
<processtree name="Processes">
<processparamset>
...
</processparamset>
<functionparamset>
...
</functionparamset>
<roledim name="..."/>
...
<useki name="..." assessment="..."/>
...
<usedim name="..." refinement="..." scale="..."/>
...
<usereldim name="FROMORG" relname="REL_CARRY_OUT"/>
<usereldim name="FUNCTION" relname="REL_CARRY_OUT"/>
<usereldim name="FROMORG" relname="REL_WORKS_TOGETHER"/>
<usereldim name="TOORG" relname="REL_WORKS_TOGETHER"/>
<usereldim name="FROMORG" relname="REL_PING_PONG"/>
<usereldim name="TOORG" relname="REL_PING_PONG"/>
...
</processtree>
```

8.5.1.2.2. Beziehungsdimension anmelden

Die Anmeldung von Beziehungsdimensionen erfolgt in der gleichen Weise wie die Anmeldung von Bezugsdimensionen (Kapitel **Bezugsdimension anmelden [184]**) mit dem Unterschied, dass Beziehungsdimensionen an verschiedenen Prozessbaumelementen (Prozessbaumwurzel, Prozesstypgruppe oder Prozesstyp) angemeldet werden können.
Bei der Anmeldung an übergeordneten Prozessbaumelementen gelten dieselben Vererbungsmechanismen wie in der Einleitung zur Prozessbaumkonfiguration beschrieben (Kapitel Anmeldung von Kennzahlen und Dimensionen am PPM-System [181]).

Gegebenenfalls wird ein und dieselbe Beziehungsdimension für jede Beziehung, für die sie verfügbar sein soll, einzeln am entsprechenden Prozessbaumelement angemeldet.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE processtree SYSTEM "KeyindicatorProcesstree.dtd">
<processtree name="Processes">
<processparamset>
...
</processparamset>
<functionparamset>
...
</functionparamset>
...
<processtypegroup name="Customer Services" dbtablename="CUBE6">
<processparamset>
...
</processparamset>
```

```
<usereldim name="SOURCEFUNC" relname ="REL_WORKS_TOGETHER"/>
<usereldim name="TARGETFUNC" relname ="REL_WORKS_TOGETHER"/>
...
<processtype name="..." autovisible="TRUE">
<typifierrule function="..." priority="..."/>
...
<usereldim name="..." relname="..."/>
...
</processtype>
...
</processtypegroup>
...
</processtree>
```

8.5.2. Automatische Prozessbaumerweiterung

Wenn während des Datenimports Prozessinstanzen bearbeitet werden, deren Prozesstyp und Prozesstypgruppe nicht im Prozessbaum existieren, wird der Prozessbaum automatisch um die fehlenden Elemente erweitert. Hierbei werden folgende Fälle unterschieden:

- Neuer Prozesstyp und neue Prozesstypgruppe

  Die neue Prozesstypgruppe und der neue Prozesstyp werden unter der Wurzel des Prozessbaumes angelegt. Sie erben alle Kennzahlen und Dimensionen der Baumwurzel.
- Neuer Prozesstyp in existierender Prozesstypgruppe

  Der neue Prozesstyp wird unter der existierenden Prozesstypgruppe angelegt. Der Prozesstyp erbt alle Kennzahlen und Dimensionen der Prozesstypgruppe.

> **Warnung**
> Ein einzelner Prozesstyp kann nicht mehreren Prozesstypgruppen zugeordnet werden. In einem solchen Fall wäre nicht definiert, welche Kennzahlen und Dimensionen diesem Prozesstyp in der neuen Gruppe zugeordnet sind. Dieser Fall kann auftreten, wenn Sie Prozesstypisierungsinformationen direkt aus dem Quellsystem in die Prozessinstanzattribute **AT_PROCTYPEGROUP** und **AT_PROCTYPE** übernehmen.

Automatisch erzeugte Prozesstypgruppen und Prozesstypen erben die den übergeordneten Elementen zugeordneten Kennzahlen und Dimensionen. Ihnen können keine weiteren Kennzahlen und Dimensionen zugeordnet werden.

8.5.3. Manuelle Prozessbaumerweiterung

Im Laufe des operativen Betriebs Ihres PPM-Systems kann es durch den fortschreitenden Datenimport zur automatischen Erweiterung des bestehenden Prozessbaumes kommen (siehe Kapitel Automatische Prozessbaumerweiterung [186]).

Wenn Sie die Prozessbaumkonfiguration manuell erweitern möchten, sollten Sie zunächst den aktuellen Prozessbaum mit den automatisch erzeugten Erweiterungen in eine Konfigurationsdatei sichern.

Exportieren Sie dazu den aktuellen Prozessbaum mittels **runppmconfig** (siehe Benutzerhandbuch **PPM Operation Guide**) lokal in eine XML-Datei.

Machen Sie die gewünschten Erweiterungen in der exportierten XML-Datei.

Importieren Sie die Datei mit den Erweiterungen wiederum mit **runppmconfig** ins PPM-System.

8.5.4. Definition des Prozessbaumes in Multibyte-Zeichensätzen

Der folgende Auszug aus der Prozessbaumkonfigurationsdatei zeigt beispielhaft die Definitionsmöglichkeiten von Prozesstypgruppen bzw. Prozesstypen unter Verwendung eines Multibyte-Zeichensatzes:

Definition von Prozesstypgruppe und Prozesstyp mit Standardabfrage:

```
...
<processtypegroup name="διεκπεραίωση εντολής" dbtablename="CUBE10">
<processparamset>
...
</processparamset>
<functionparamset>
...
</functionparamset>
<processtype name="πώληση τοις μετρητοίς" autovisible="FALSE">
<processparamset>
<paramset>
...
<kiquery showzero="auto">
<keyindicator>
<criterion name="PNUM">
Number of processes

  </criterion>
</keyindicator>
<iteration nullvalue="no">
<criterion name="PROCESSTYPE">Process type

  </criterion>
<refinement name="BY_LEVEL1">Rough

  </refinement>
</iteration>
<filter>
<criterion name="PROCESSTYPE">Process type

  </criterion>
<filteritem operator="or">
<dataitem>
διεκπεραίωση εντολής\πώληση

  τοις μετρητοίς
<datatype name="TEXTPAIR">
Text pair
```

```
  </datatype>
<scale name="LEVEL2SCALE" factor="2.0">

  Detailed

  </scale>
</dataitem>
</filteritem>
</filter>
</kiquery>
...
</paramset>
</processparamset>
<functionparamset>
<paramset>
...
<kiquery showzero="auto">
...
<filter>
<criterion name="PROCESSTYPE">Process type

  </criterion>
<filteritem operator="or">
<dataitem>
διεκπεραίωση εντολής\πώληση

  τοις μετρητοίς
<datatype name="TEXTPAIR">Text pair

  </datatype>
<scale name="LEVEL2SCALE" factor="2.0">

  Detailed

  </scale>
</dataitem>
</filteritem>
</filter>
...
</kiquery>
</paramset>
</functionparamset>
</processtype>
</processtypegroup>
...
```

Im Beispiel wird die Prozesstypgruppe **διεκπεραίωση εντολής** definiert, welcher der Prozesstyp **πώληση τοις μετρητοίς** zugeordnet wird.

8.6. Beziehungen

Im Modul **Interaktionsanalyse** können Beziehungen zwischen verschiedenen Objekten auf Prozessinstanzebene analysiert werden.

Eine Beziehung ist eine Verknüpfung zwischen zwei Objektinstanzen in einer Prozessinstanz.

Als Bezugsobjekte können dabei Organisationseinheiten und Funktionen verwendet werden. Der Berechner (Kapitel Definition von Beziehungsberechnungen [191]) einer Beziehung entscheidet, zwischen welchen Objektinstanzen die Beziehung existiert.

Diese Beziehungen (Beziehungsausprägungen) kann man sich wie eine unsichtbare Kante zwischen den Objektinstanzen in der Prozessinstanz vorstellen. Für jede Beziehung können spezielle Kennzahlen und Dimensionen (Kapitel Definition von Beziehungskennzahlen [195] und Definition von Beziehungs- und Organisationsdimensionen [196]) definiert werden, die für jede Beziehungsausprägung auf Prozessinstanzebene berechnet werden.

Konfigurationsbeispiele zum Thema **Interaktionsanalyse** finden Sie in Ihrer PPM Customizing Toolkit-Installation im Verzeichnis **<PPM-Installationsverzeichnis\ppm\server\bin\agentLocalRepo\.unpacked\<Installationszeit>_ppm-client-run-prod-<Version>-runnable.zip\ppm\ctk\ctk\examples\custom\organalysis**.

8.6.1. Definition von Beziehungen

Beziehungen bilden die Grundlage für die Definition von Beziehungskennzahlen und Beziehungsdimensionen. Sie werden in der Kennzahlenkonfiguration definiert.

Eine Beziehung besteht zwischen einer Quellbezugsdimension und einer Zielbezugsdimension (Kapitel Bezugsdimensionen [190]), hat einen Namen und einen ihr mittels XML-Element **calcrel** zugeordneten Beziehungsberechner.

XML-Tag	Beschreibung
relation	Beziehungsdefinition
name	Eindeutiges Schlüsselwort der Beziehung zur internen Referenzierung
id	Eindeutige Ganzzahl zwischen 0 und 999, unter der die entsprechende Datenbanktabelle erzeugt wird
description	Sprachabhängiger Oberflächenname. Die Angabe in der Standardsprache muss gepflegt sein.
sourcedim	Quellbezugsdimension. Nur ein- bzw. zweistufige Dimensionen vom Typ **OT_FUNC** oder **OT_ORG** (Kapitel Bezugsdimensionen [190])
targetdim	Zielbezugsdimension. Nur ein- bzw. zweistufige Dimensionen vom Typ **OT_FUNC** oder **OT_ORG** (Kapitel Bezugsdimensionen [190])
refki	Mindestens eine Kennzahl vom Typ **RELATION**
refdim	Dimension vom Typ **RELATION** bzw. **PROCESS**.
(optional)	Jede referenzierte Dimension kann nur im Kontext der angegebenen Beziehung ausgewertet werden.

Beispiel 97. ***[en-us]***

(Auszug aus **Keyindicator.xml**)

```
...
<relation name="REL_CARRY_OUT" id="0">
```

```
<description name="executes" language="de" />
<sourcedim name="FROMORG" />
<targetdim name="FUNCTION" />
<refki name="REL_CO_CORATE" />
<refki name="RNUMA" />
<refki name="REL_CO_DLZ" />
<refki name="ORGCAPA" />
<refki name="REL_CO_COST" />
<refdim name="REL_CO_TIME" />
</relation>
...
```

Warnung
Verwenden Sie als Quell- und Zielbezugsdimension einer Beziehung nicht ein und dieselbe Dimension. Dies führt beim Import der Kennzahlenkonfiguration zum Abbruch mit entsprechender Fehlermeldung.

Sie können im XML-Element **refki** keine Kardinalitätskennzahlen referenzieren.

8.6.1.1. Bezugsdimensionen

In PPM besteht eine Beziehung immer zwischen einem Quellobjekt und einem Zielobjekt, den so genannten Bezugsobjekten. Für die Spezifizierung der Bezugsobjekte einer Beziehung müssen Sie Bezugsdimensionen (**sourcedim**, **targetdim**) zu diesen Objekten in der Kennzahlenkonfiguration definieren. Für jede Beziehung definieren Sie eine Quellbezugsdimension und eine Zielbezugsdimension vom Typ **TEXT** (ein-, zwei- und n-stufige Dimensionen). Über die einzelnen Dimensionswerte werden bestimmte Organisationseinheiten bzw. Funktionen referenziert. Das erforderliche Attribut-Mapping erfolgt in der Konfigurationsdatei der Organisationseinheiten und bzw. oder in der Mapping-Informationsdatei.

Über das XML-Attribut **dimtype** wird angegeben, für welchen Objekttyp die Bezugsdimension definiert wird. Zulässige Werte sind **OT_FUNC** für Funktionsdimensionen und **OT_ORG** für Organisationsdimensionen.

Beispiel 98. ***[en-us]***

(Auszug aus **Keyindicator.xml**)

```
...
<twoleveldim name="FROMORG" dimtype="OT_ORG" dimgroup="DIM_GROUP_CRITERIA"
internal="no" importmode="OPTIONAL">
<description language="de" name="Organisationseinheit (Start)"/>
<leveldesc>
<dimitem attrname="AT_ORGGRP" colname="GRP" calculated="FALSE">
<description language="de" name="Group" />
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="AT_OBJNAME" colname="NAME" calculated="FALSE">
<description language="de" name="Name" />
</dimitem>
</leveldesc>
```

```
</twoleveldim>
...
```

Warnung
Die verwendeten Quellbezugs- und Zielbezugsdimensionen müssen Sie in der Prozessbaumkonfiguration an der Prozessbaumwurzel anmelden (siehe Kapitel Bezugsdimension anmelden [184]). Andernfalls wird beim Import der Prozessbaumkonfiguration eine Fehlermeldung ausgegeben.

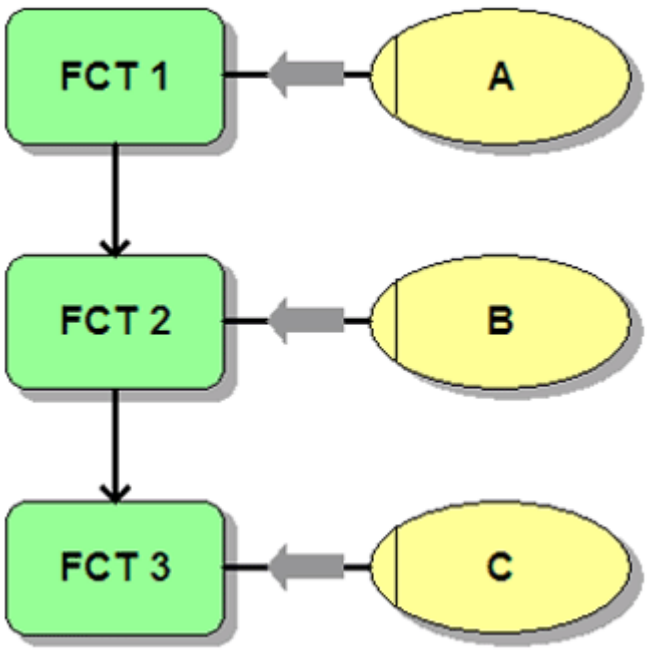
8.6.2. Definition von Beziehungsberechnungen

Die definierten Beziehungen werden über Beziehungsberechner in der Prozessinstanz erzeugt. Für jede Beziehung wird dabei in der Kennzahlenkonfiguration im XML-Element **calcrel** die entsprechende Berechnungsklasse angegeben. Standardmäßig sind dies die folgenden vier Klassen (der feststehende Teil des Klassennamens wird jeweils weggelassen):

- ZRelationCalculatorCarriesOut für die Beziehung **Führt aus**

Zieht in den entsprechenden Prozessinstanzen, d. h., für deren Prozesstyp oder Prozesstypgruppe die betreffenden Beziehungskennzahlen bzw. Beziehungsdimensionen angemeldet sind, eine Beziehung von jeder Instanz einer Organisationseinheit zur zugehörigen Funktionsinstanz, die der Organisationseinheit über die Kante **führt aus** (**CXN_UNDIRECTED**) zugeordnet ist.

Beispiel 99.

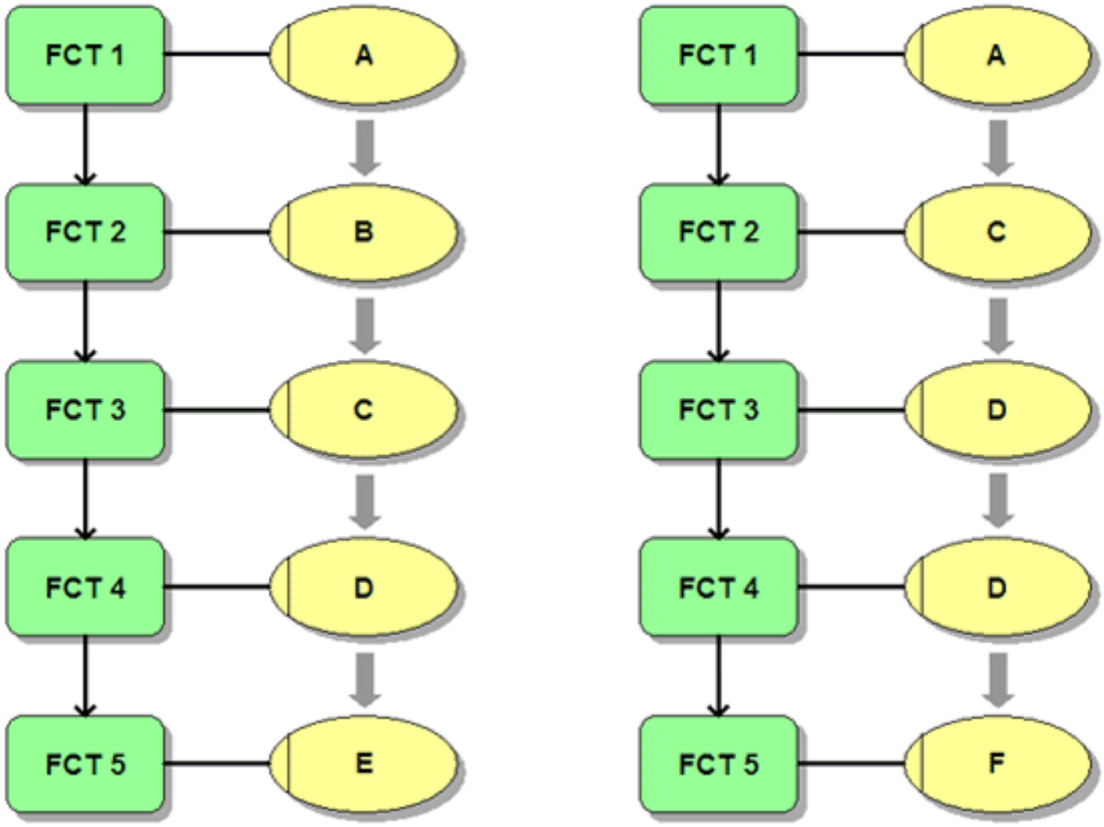


In der Prozessinstanz wird vom Beziehungsberechner zwischen jeder Organisationseinheit und Funktion jeweils die Beziehung **Führt aus** (graue Pfeile) erzeugt. Quellobjekt ist dabei die Organisationseinheit, Zielobjekt die Funktion.

- ZRelationCalculatorWorksTogether für die Beziehung **Arbeitet zusammen mit (ohne Brüche)**

Beispiel 100.

Zieht in den entsprechenden Prozessinstanzen eine Beziehung von der Instanz einer Organisationseinheit zu jeder Instanz der Organisationseinheit, die die direkt nachfolgende Funktionsinstanz ausführt. Dabei darf es sich auch um gleiche Organisationseinheiten handeln, d.h. um Organisationseinheiten gleichen Namens (**AT_OBJNAME**). Mit dieser Beziehung können durch Wahl geeigneter Filter auch organisatorische Strukturen innerhalb einer Organisationseinheit ausgewertet werden.

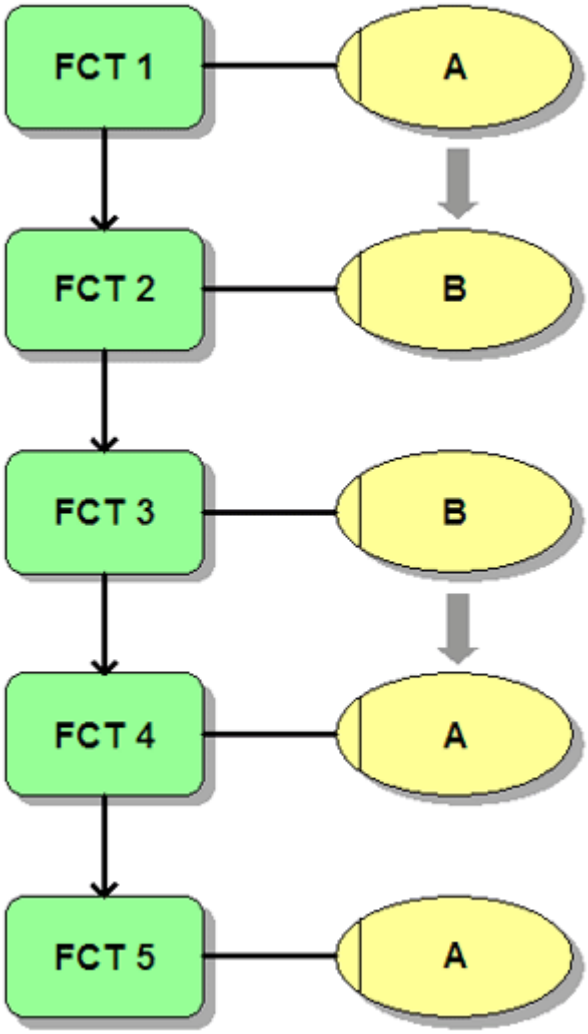


In beiden Prozessinstanzen wird vom Beziehungsberechner jeweils zwischen einer Organisationseinheit und der Organisationseinheit, die die direkt nachfolgende Funktion ausführt, die Beziehung **Arbeitet zusammen mit** (graue Pfeile) erzeugt. Die Grafik zeigt, dass beispielsweise die Organisationseinheit **D** mit den Organisationseinheiten **D**, **E** bzw. **F** zusammenarbeitet.

- ZRelationCalculatorWorksTogetherLongDistance für die Beziehung **Arbeitet zusammen mit (mit Brüchen)**, wobei die Funktionsinstanzen mit den Organisationseinheiten im Gegensatz zur zuvor beschriebenen Beziehung **Arbeitet zusammen mit (ohne Brüche)** nicht direkt aufeinanderfolgen müssen.
- ZRelationCalculatorOrgBreak für die Beziehung **Organisationsbruch**

Verhält sich genau so wie die Beziehung Arbeitet zusammen mit, mit dem Unterschied, dass die Beziehung nur zwischen verschiedenen Organisationseinheiten erzeugt wird, d.h., zwischen Organisationseinheiten mit verschiedenen Namen (**AT_OBJNAME**). Mit dieser Beziehung werden organisatorische Strukturen zwischen unterschiedlichen Organisationseinheiten ausgewertet.

Beispiel 101.

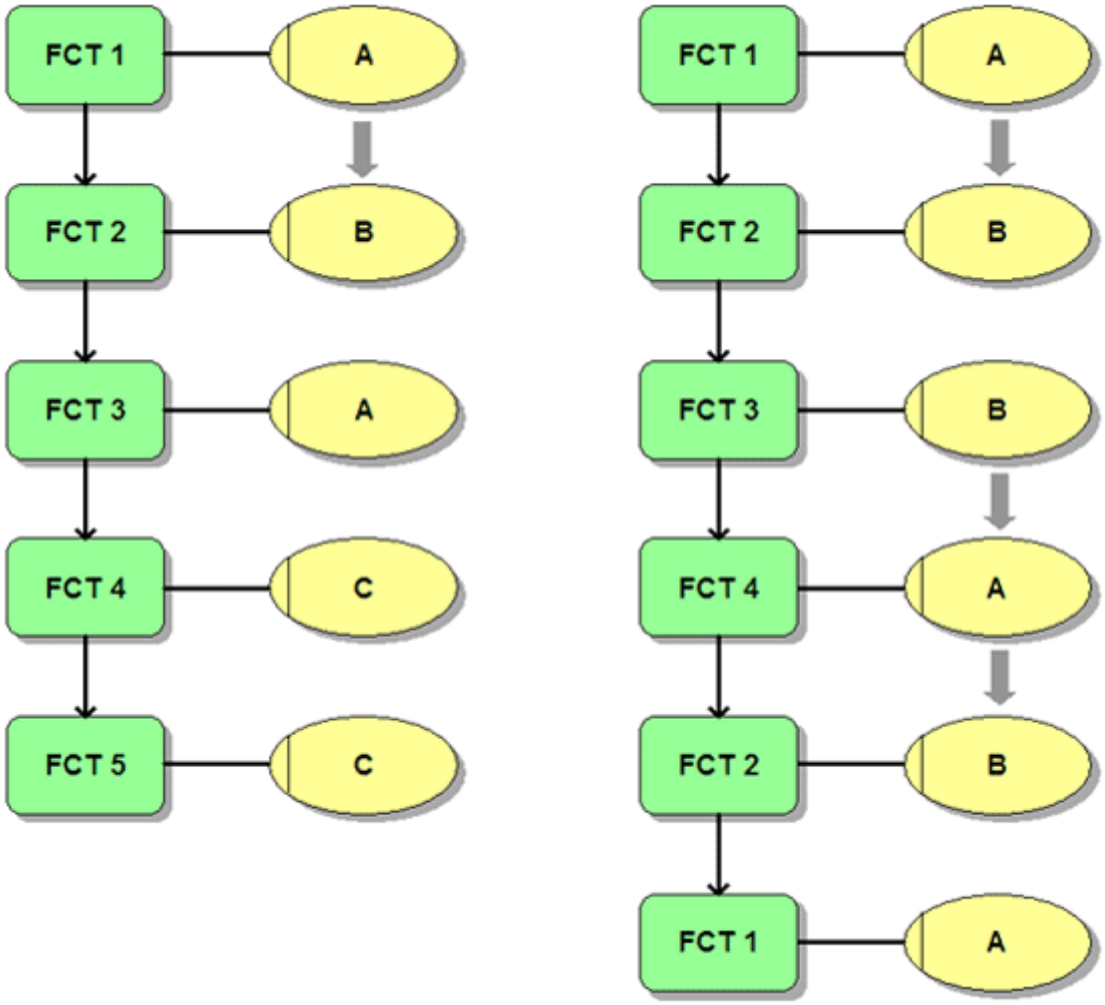


In der Prozessinstanz wird vom Beziehungsberechner jeweils zwischen zwei Organisationseinheiten, bei denen es bei der Ausführung der direkt nachfolgenden Funktion zu einem organisatorischen Wechsel kommt, die Beziehung **Organisationsbruch** (graue Pfeile) erzeugt.

- ZRelationCalculatorPingPong für die Beziehung **Ping Pong**

Beispiel 102.

Zieht in den entsprechenden Prozessinstanzen eine Beziehung zwischen zwei Organisationseinheiten mit verschiedenen Namen, die sich in der Ausführung der direkt nachfolgenden Funktion im weiteren Prozessablauf mindestens einmal wieder direkt abwechseln, ohne dass eine weitere Organisationseinheit beteiligt ist. Dabei kann es sich um die Ausführung sowohl unterschiedlicher als auch gleicher Funktionen (**AT_OBJNAME**) handeln.



In beiden Prozessinstanzen wird vom Beziehungsberechner zwischen der Organisationseinheit **A** und der Organisationseinheit **B** die Beziehung **Ping Pong** (graue Pfeile) erzeugt.

(Auszug aus **Keyindicator.xml**)

```
...
<calcrel name="REL_CARRY_OUT">
<calcclass
name="com.idsscheer.ppm.server.keyindicator.relation.calculator.ZRelationCalcula
torCarriesOut"/>
<calcparam key="..." value="..."/>
</calcrel>
...
```

XML-Tag	Beschreibung
calcrel	Beziehungsberechner
name	Interner Name der zu berechnenden Beziehung
calcclass	Name der Berechnungsklasse. Über beliebige, optionale XML-Elemente **calcparam** werden bei Klassenaufruf Berechnungsparameter übergeben (Kapitel Definition von Attributberechnungen [50]).

XML-Tag	Beschreibung
depends (optional)	Name und Typ eines Attributs (**PROCESS**, **OT_FUNC**, **OT_EVT**, **OT_ORG** oder **RELATION**), das existieren muss, damit die Berechnung ausgeführt wird. Falls es sich bei dem angegebenen Attribut um ein zu berechnendes Attribut handelt, wird dieses zunächst berechnet. Im Attribut **relname** wird die Beziehung angegeben, von der eine Abhängigkeit besteht (nur bei **type="RELATION"**). Es können gleichzeitig mehrere Elemente **depends** angegeben werden. Nicht zusammen mit **dependsrel** zu verwenden.
dependsrel (optional)	Name der Beziehung, von der eine Abhängigkeit besteht. Es können gleichzeitig mehrere Elemente **dependsrel** angegeben werden. Nicht zusammen mit **depends** zu verwenden.

8.6.3. Definition von Beziehungskennzahlen

Beziehungskennzahlen werden einer bestimmten Beziehung in der Kennzahlkonfiguration über das XML-Element **refki** zugewiesen. Beziehungskennzahlen können nur im Modul **Interaktionsanalyse** mit der entsprechenden Beziehung ausgewertet werden. Beziehungskennzahlen werden an der Oberfläche mit gelben Symbolen gekennzeichnet.

Beziehungskennzahlen werden in der Kennzahlenkonfiguration durch folgende XML-Elemente und XML-Attribute konfiguriert (siehe Kapitel Definition von Standardkennzahlen [132]):

XML-Tag	Beschreibung
name	Eindeutiges Schlüsselwort der Kennzahl. Wird im XML-Element **refki** der Beziehungsdefinition (Kapitel Definition von Beziehungen [189]) referenziert. Empfohlenes Präfix: **REL_**
type	**RELATION** (Beziehungskennzahl)
location	Nur bei **type="RELATION"**
(optional)	Gültige Werte: **SOURCE** (Das Attribut, aus dem der Kennzahlwert genommen wird, wird am Quellbezugsobjekt der Beziehung gesucht.)
	TARGET (Das Attribut, aus dem der Kennzahlwert genommen wird, wird am Zielbezugsobjekt der Beziehung gesucht.)
	THIS (Standardwert: Suche nach dem Attribut, aus dem der Kennzahlwert genommen wird, wird an der Beziehungskante selbst durchgeführt)

Beispiel 103.

Kennzahldefinition:

```
...
<kidef name="REL_CO_DLZ" type="RELATION"
attrname="AT_APX_PROCESSINGTIME" calculated="TRUE"
location="TARGET" distribution="TRUE"
standarddeviation="TRUE" sharedfunctionki="FALSE"
functionspanki="FALSE" retrievertype="KEYINDICATOR"
dimreferring="LOOSE" importmode="OPTIONAL">
<description language="de" name="Average working time" />
</kidef>
...
```

Beispiel 104.

Zugehörige Berechnungsvorschrift:

```
...
<calcattr name="AT_APX_PROCESSINGTIME" type="OT_FUNC" delete="no">
<calculation>
<max>
<set>
<constant>
<dataitem value="0.0">
0.000
<datatype name="DOUBLE">Floating point number
</datatype>
</dataitem>
</constant>
<max>
<attribute name="AT_APX_PROCESSINGTIME" nodetype="OT_FUNC" onerror="CONTINUE" />
</max>
</set>
</max>
</calculation>
</calcattr>
...
```

8.6.4. Definition von Beziehungs- und Organisationsdimensionen

Alle Dimensionstypen (außer Suchdimensionen) können über das XML-Attribut **dimtype** als Beziehungsdimensionen bzw. Organisationsdimensionen definiert werden (siehe Kapitel Definition von Dimensionen [152]). Diese Dimensionstypen sind nur für Auswertungen im Modul **Interaktionsanalyse** verfügbar. An der Oberfläche werden diese Dimensionen durch gelbe Symbole gekennzeichnet. Für Beziehungsdimensionen wird zusätzlich im Attribut **location** angegeben, an welchem Objekt der Beziehung die entsprechenden Dimensionswerte (Attribute) gesucht werden. Standardmäßig geschieht dies an der Beziehung selbst.

Beziehungs- und Organisationsdimensionen werden in der Kennzahlenkonfiguration durch folgende XML-Elemente und XML-Attribute konfiguriert:

XML-Tag	Beschreibung
name	Eindeutiges Schlüsselwort der Dimension. Wird im XML-Element **refdim** der Beziehungsdefinition referenziert (Kapitel Definition von Beziehungen [189]). Empfohlenes Präfix: **REL_**
dimtype	**RELATION** (Beziehungsdimension) **OT_ORG** (Organisationsdimension)
location (optional)	Nur bei **type="RELATION"** Gültige Werte: **SOURCE** (Das Attribut, aus dem der Dimensionswert genommen wird, wird am Quellbezugsobjekt der Beziehung gesucht.) **TARGET** (Das Attribut, aus dem der Dimensionswert genommen wird, wird am Zielbezugsobjekt der Beziehung gesucht.) **THIS** (Standardwert: Das Attribut, aus dem der Dimensionswert genommen wird, wird der Beziehung selbst gesucht.)

Beispiel 105.

Zeitdimension als Beziehungsdimension:

```
...
<timedim name="REL_CO_TIME" dimtype="RELATION"
```

```
attrname="AT_END_TIME" location="TARGET"
tablename="FUNC_ENDTIME" precision="HOUR"
dimgroup="DIM_GROUP_TIME" calculated="FALSE" internal="no"
earlyalert="no" importmode="OPTIONAL">
<description language="de" name="Time" />
</timedim>
...
```

Beispiel 106.

Zweistufige Dimension als Organisationsdimension:

```
...
<twoleveldim name="FROMORG" dimtype="OT_ORG" dimgroup="DIM_GROUP_CRITERIA"
internal="no" importmode="OPTIONAL">
<description language="de" name="Organisationseinheit (Start)"/>
<leveldesc>
<dimitem attrname="AT_ORGGRP" colname="GRP" calculated="FALSE">
<description language="de" name="Group" />
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="AT_OBJNAME" colname="NAME" calculated="FALSE">
<description language="de" name="Name" />
</dimitem>
</leveldesc>
</twoleveldim>
...
```

Warnung
Suchdimensionen (XML-Element **searchdim**) können weder als Beziehungsdimensionen noch als Organisationsdimensionen definiert werden.

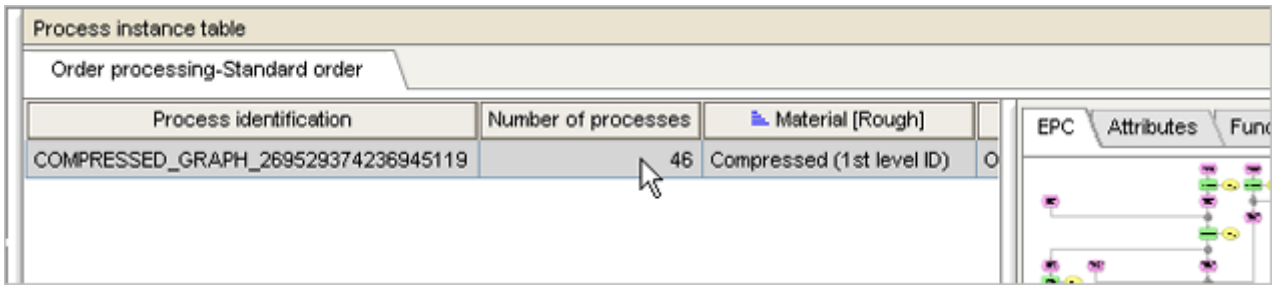
9. Verdichtungsverhalten ändern

Standardmäßig werden bei der dauerhaften Verdichtung von Prozessinstanzen mittels PPM-Kommandozeilenverdichter **runppmcompress** (siehe Benutzerhandbuch **PPM Operation Guide**) gleiche Werte einer Prozessdimension als Dimensionswert in die verdichtete EPK übernommen, wohingegen unterschiedliche Werte gelöscht werden, so dass die Dimension in diesem Fall in der verdichteten EPK keinen Wert besitzt. Es ist dann nicht mehr nachverfolgbar, ob Dimensionswerte gepflegt waren oder nicht. Dies gilt nur für Dimensionen, die keine Verdichtungskriterien sind, d. h.die nicht als Iterationen im Verdichter-ParamSet enthalten sind.

Sie können das Verhalten des Kommandozeilenverdichters durch spezielle Konfigurationseinstellungen Ihres PPM-Systems dahingehend ändern, dass gleiche und unterschiedliche Dimensionswerte einheitlich während des Verdichtungsvorgangs gelöscht werden, wenn im Verdichtungsparamset nicht über diese Dimension iteriert wird. In der verdichteten Prozessinstanz werden die gelöschten Werte dann mittels eines einheitlichen Verdichtungswertes sichtbar gemacht. Verdichtungswerte geben Sie in der Kennzahlenkonfiguration für die betreffende Prozessdimension (Textdimensionen und Suchdimensionen) an.

Für Zeitdimensionen (**timedim** und **hourdim**) können Sie lediglich angeben, dass gleiche und unterschiedliche Dimensionswerte einheitlich während des Verdichtungsvorgangs gelöscht werden. In der Analyse ist die Dimension in der verdichteten Prozessinstanz aufgrund der gelöschten Werte nicht mehr darstellbar.

Beispiel 107.



Alle Werte der Prozessdimension **Material** in den gewählten sechsundvierzig Prozessinstanzen wurden im permanenten Verdichtungsvorgang über Kommandozeile gelöscht und durch den in der Kennzahlenkonfiguration angegebenen Verdichtungswert ersetzt (**verdichtet (Schlüssel der 1. Stufe)**). Dabei spielte es keine Rolle, ob die Dimensionswerte gleich oder verschieden waren.

9.1. Konfiguration des internen Verdichtungsattributes

Stellen Sie sicher, dass das interne Verdichtungsattribut **AT_INTERNAL_COMPRESSCRITERION** in der Attributkonfiguration Ihres PPM-Systems gepflegt ist (siehe Definition von Attributtypen und Attributtypgruppen [19]).

attributenames.xml

```
...
<attribute key="AT_INTERNAL_COMPRESSCRITERION" name="Compression criteria"/>
...
```

attributetypes.xml

```
...
<attributedefinition key="AT_INTERNAL_COMPRESSCRITERION" type="TEXT"
group="AG_KPI_COMPRESS"/>
...
```

In der verdichteten Prozessinstanz werden im Prozessattribut **AT_INTERNAL_COMPRESSCRITERION** alle Verdichtungskriterien aufgelistet, d.h. alle Prozessdimensionen, die als Iterationen im verwendeten Verdichterparamset enthalten sind, sowie alle Datenzugriffsdimensionen (siehe Definition von Datenzugriffsdimensionen [176]). Hinter dem internen Bezeichner der Dimension wird in runden Klammern die evtl. eingestellte Verfeinerungsstufe angegeben.

Die Prozessdimension **PROCESSTYPE** ist immer im internen Verdichtungsattribut enthalten, da sie fester Bestandteil eines jeden Verdichtungsparamsets ist. Die Prozessdimension **TIME** dagegen nur, wenn sie als Iteration im Paramset enthalten ist. Ist lediglich ein Zeitfilter im Verdichtungsparamset enthalten, wird die Dimension **Zeit** (wie auch andere Zeitdimensionen vom Typ **timedim** bzw. **hourdim**) nicht als Verdichtungskriterium aufgeführt.

9.2. Verdichtungswerte vergeben

Verdichtungswerte, die in einer verdichteten EPK angezeigt werden, um gelöschte Dimensionswerte sichtbar zu machen, werden in der Kennzahlenkonfiguration (siehe Definition von Dimensionen [152]) angegeben.

Für die folgenden Prozessdimensionstypen können Sie Verdichtungswerte vergeben

- Ein-, zwei- bzw. n-stufige Dimensionen
  (oneleveldim [159], twoleveldim [160], nleveldim [156])
- Suchdimensionen
  (searchdim [171])
- Zeitdimensionen
  (timedim [164], hourdim [170])

Beispiel 108.

Beispiel 1 (Verdichtungswerte für eine zweistufige Dimension) Sie möchten nur für die erste Stufe der zweistufigen Prozessdimension **Material** Verdichtungswerte angeben:

```
...
<twoleveldim name="MATERIAL" dimtype="PROCESS" dimgroup="DIM_GROUP_CRITERIA">
<description name="Material" language="en"/>
<leveldesc>
<dimitem attrname="AT_MATERIAL_KIND" colname="FIRST_ID" calculated="FALSE">
<description language="en" name="Material type"/>
<compressionvalue>

  Compressed (1st level ID)

  </compressionvalue>
</dimitem>
<dimitem attrname="AT_MATERIALKIND_NAME" colname="FIRST_DESC"
calculated="FALSE">
```

```
<description language="en" name="Material type name"/>
<compressionvalue>

  Compressed (1st level description)

  </compressionvalue>
</dimitem>
</leveldesc>
<leveldesc>
<dimitem attrname="AT_MATERIAL" colname="SECOND_ID" calculated="FALSE">

  <description language="en" name="Material"/>
</dimitem>
<dimitem attrname="AT_MATERIAL_NAME" colname="SECOND_DESC" calculated="FALSE">
<description language="en" name="Material type name"/>
</dimitem>
</leveldesc>
</twoleveldim>
...
```

Für Schlüssel und Beschreibung der ersten Stufe der Dimension ist jeweils mit dem XML-Element **compressionvalue** ein Verdichtungswert angegeben. Bei der dauerhaften Verdichtung mittels Kommandozeilenverdichter werden alle gleichen und unterschiedlichen Werte der angegebenen Prozessdimension gelöscht. In der verdichteten Prozessinstanz werden als Wert der Dimension die angegebenen Verdichtungswerte angezeigt, sofern die Dimension nicht im Verdichterparamset enthalten ist. Als Wert für die verdichtete zweite Stufe wird standardmäßig **nicht gepflegt** angezeigt, da für die Stufenelemente (**dimitem**) der zweiten Stufe keine Verdichtungswerte angegeben sind.

Process instance table

Order processing

Process identification	Time ...	Nu...	Material [Detailed]
COMPRESSED_GRAPH_8263291222575049880	Jan 07	34	Compressed (1st level ID) (Compressed (1st level description))-Not maintained

Geben Sie die Verdichtungswerte der gewünschten Stufe immer paarweise (für Schlüssel und Beschreibung) an, ansonsten bricht der Import der Kennzahlenkonfiguration mit einer Fehlermeldung ab.

Beispiel 109.

Beispiel 2 (Zeitdimension: Dimensionswerte bei der Verdichtung löschen) Sie möchten sicherstellen, dass in dauerhaft verdichteten Prozessinstanzen keine Werte mehr für die Zeitdimension **Prozessendzeit** angezeigt werden, unabhängig davon, ob die Dimensionswerte der zu verdichtenden Prozessinstanzen gleich oder verschieden sind.

```
...
<timedim name="PROZESSENDZEIT" dimtype="PROCESS" ...

  attrname="AT_END_TIME" ...

  calculated="TRUE" ...

  deleteoncompression="TRUE" ... >
<description name="Process end time" language="en" />
```

```
</timedim>
...
```

Durch die Angabe **deleteoncompression="TRUE"** für die Prozessdimension **Prozessendzeit** werden gleiche wie unterschiedliche Dimensionswerte der zu verdichtenden Prozessinstanzen im permanenten Verdichtungsvorgang über Kommandozeile gelöscht, wenn die Dimension **Prozessendzeit** nicht als Iteration im Verdichtungsparamset enthalten ist. In der verdichteten EPK sind dann keine Dimensionswerte mehr vorhanden.

Warnung
Wenn Sie die Dimension **Prozessendzeit** in der PPM-Analyse der angezeigten verdichteten Prozessinstanz hinzufügen, können aufgrund der gelöschten Dimensionswerte keine Daten mehr dargestellt werden.

Für die Prozessdimension **Prozesstyp** (**PROCESSTYPE**) können Sie keine Verdichtungswerte angeben, da diese Dimension automatisch als Iteration in jedem Verdichtungsparamset enthalten ist. Ebenso können Sie für die Prozessdimension **Zeit** (**TIME**) nicht angeben, dass die Dimensionswerte bei der Verdichtung gelöscht werden sollen.

Geben Sie Verdichtungswerte für die gewünschte Prozessdimension bequem in PPM Customizing Toolkit in der Komponente **Dimensionen** des Moduls **Kennzahlen und Dimensionen** an. Dort können Sie auch für Zeitdimensionen mit der Verwendung **PROCESS** angeben, ob die Dimensionswerte bei der Verdichtung gelöscht werden sollen. Voraussetzung dafür ist aber, dass das Attribut **AT_INTERNAL_COMPRESSCRITERION** gepflegt, d. h. angelegt ist.

10. Systemanbindungen

10.1. SAP ausführbare Einheiten

Von einer in der Prozessinstanztabelle gewählten Prozessinstanz können Sie bei entsprechender Konfiguration mittels Kontextmenüaufruf über einen Anmeldedialog eine ausführbare Einheit in der SAP-Oberfläche starten, in welcher Daten zur gewählten Prozessinstanz angezeigt werden.

10.1.1. Software-Voraussetzungen

Das SAP Logon muss auf demselben Rechner installiert sein wie das PPM-Frontend.

- Der SAP Java Connector (JCo) muss auf dem Client-Rechner und auf dem SAP-Server mit der gleichen Versionsnnummer installiert sein.
- Hinweise zur Installation des SAP Java Connector erhalten Sie im PPM-Installationshandbuch.

10.1.2. Berechtigungen im SAP-System

Der die ausführbare Einheit aufrufende PPM-Benutzer benötigt einen SAP-Benutzer mit mindestens folgenden Berechtigungen:

- Anmeldeberechtigung über SAP GUI
- RFC-Berechtigung
- Berechtigung zum Ausführen des Remote Function Call ABAP4_CALL_TRANSACTION
- Berechtigungen zum Aufrufen der in der Konfiguration festgelegten SAP ausführbare Einheiten

10.1.3. Transaktionsaufruf

Sollten im Transaktionsaufruf übergebene Parameter fehlerhaft sein, findet die entsprechende Fehlerbehandlung im SAP-System selbst statt (siehe SAP Batch-Programmierung).

Der Aufruf einer Transaktion erfolgt in einem eigenen Prozess. Deshalb können auch mehrere Transaktionen gleichzeitig geöffnet werden.

10.1.4. Konfiguration

Die SAPausführbaren Einheiten werden in einer eigenen XML-Datei konfiguriert, die mit dem Kommandozeilenprogramm **runppmconfig** importiert bzw. exportiert werden kann (siehe Benutzerhandbuch **PPM Operation Guide**). Die sprachabhängigen Beschreibungen (XML-Elemente **description**) müssen zumindest in der Standardsprache gepflegt sein.

Die XML-Konfiguration enthält Angaben zum Kontextmenü, den Verbindungsdaten der verfügbaren SAP-Systeme und den Konfigurationen der ausführbaren Einheiten:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE r3transactionconf SYSTEM "mysaptransaction.dtd">
<r3transactionconf>
<submenu>
<description language="..." name="..."/>
</submenu>
<logoutMenuEntry>
<description language="..." name="..."/>
</logoutMenuEntry>
<r3system systemid="..." client="...">
<description language="..." name="...">
Description of the SAP system
</description>
<locales>
<defaultlocale value="..."/>
<locale value="..."/>
</locales>
<applicationserver appserver="..." systemnumber="..."/>
</r3system>
<transaction systemid="..." transactionid="..."
transactioncode="..." skipfirstscreen="..."
proctypegroup="..." mode="..." update="...">
<description language="..." name="..."/>
<batchinputline ... />
...
<batchinputlist ... />
...
</transaction>
<transaction ...>
...
</transaction>
...
</r3transactionconf>
```

Detailliertere Angaben zur Konfiguration von Systemzugängen finden Sie in der Technischen Referenz PPM Process Extractors.

Für einen funktionstüchtigen Transaktionsaufruf muss neben korrekten Verbindungsdaten mindestens eine Transaktion (XML-Element **transaction**) konfiguriert sein.

Erstellen Sie Konfigurationen ausführbarer Einheiten unter Zuhilfenahme des SAP-ausführbare Einheiten-Rekorders, indem Sie die entsprechende ausführbare Einheit im ABAP-Batch-Input-Format aufzeichnen. Bezüglich Funktionsweise des Rekorders und Syntax der ABAP-Batch-Input-Skripte sei auf die SAP-Dokumentation verwiesen.

Warnung
Die folgenden Anleitungen zum Erstellen von Konfigurationen können die SAP-Dokumentation nicht ersetzen, insbesondere nicht die Behebung von Skriptfehlern. Grundwissen über Batch-Input-Skripte ist zum Teil erforderliche Voraussetzung, um eine Konfiguration ausführbarer Einheiten zu erstellen.

Es können zwei grundsätzlich verschiedene Arten von Aufrufen ausführbarer Einheiten konfiguriert werden:

- Aufruf ist immer nur auf einer selektierten Prozessinstanz möglich (Single Select)
- Aufruf ist auf einer oder mehreren selektierten Prozessinstanzen möglich (Multi Select)

10.1.4.1. Konfigurationsbeispiele

Single Select-Transaktion VA03 im SAP-Frontend aufzeichnen

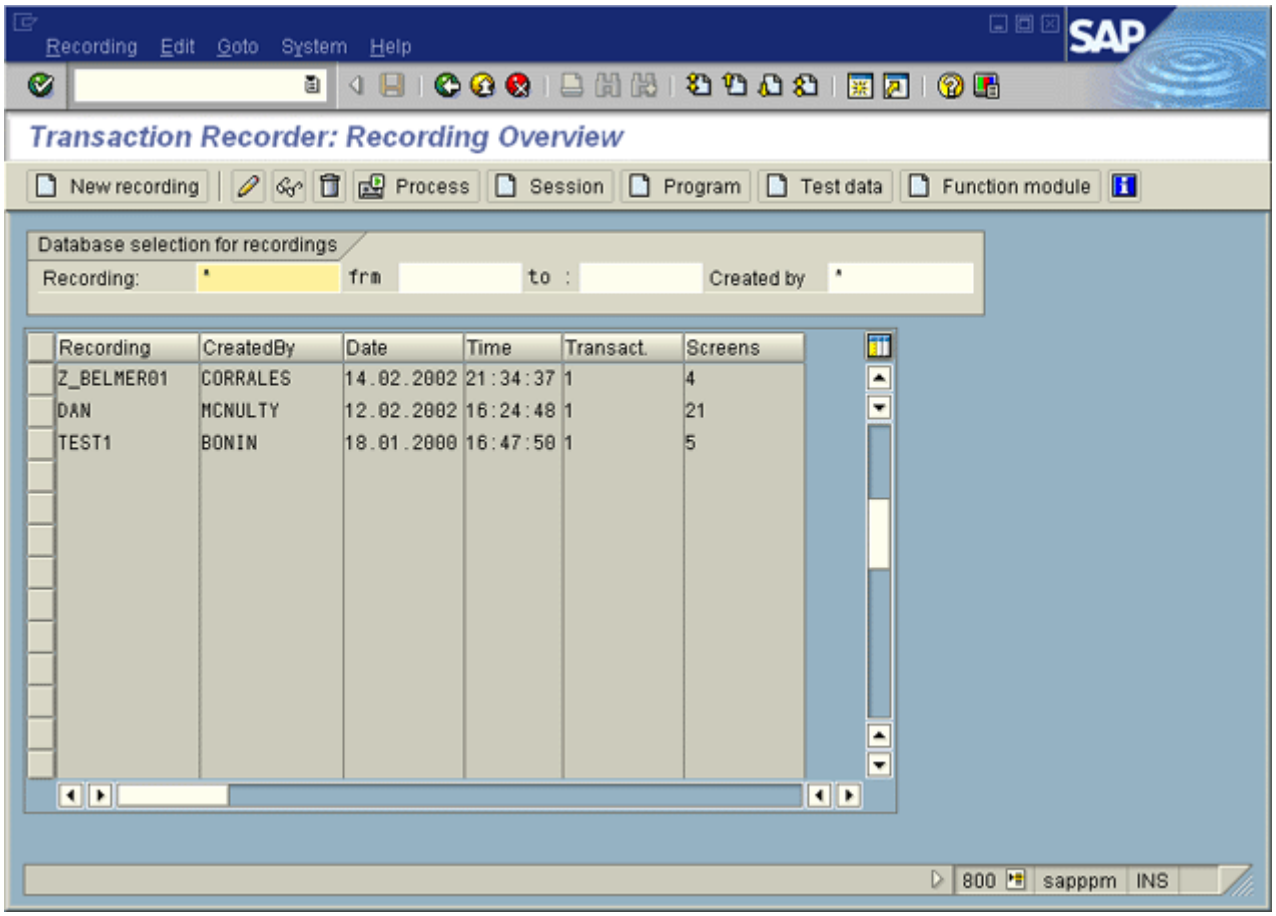
Anforderungen

Bei Einfachselektion einer Prozessinstanz der Prozesstypgruppe Auftragsabwicklung in PPM soll die Transaktion **VA03 (Auftrag anzeigen)** in dem SAP-System mit der ID **sapppm** aufgerufen werden können.

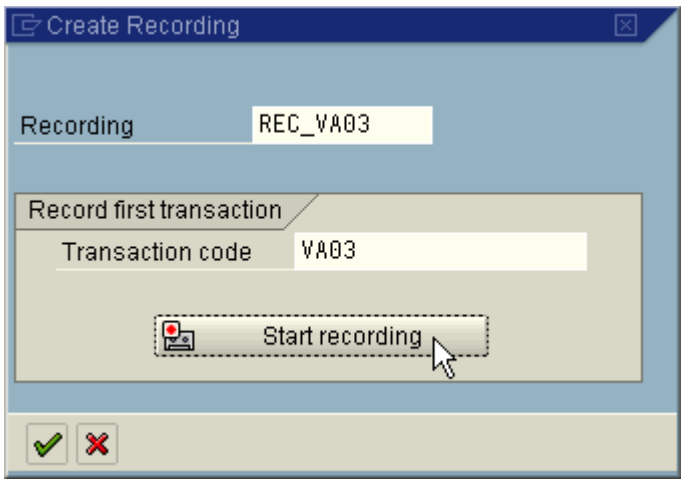
Der Transaktion soll das PPM-Prozessattribut **AT_SALES_ORDER_NUMBER** (Auftragsnummer der gewählten Prozessinstanz) übergeben werden.

Im Folgenden ist schrittweise beschrieben, wie Sie mittels SAP-Transaktionsrekorder die Transaktion **VA03 (Auftrag anzeigen)** unter Berücksichtigung der gegebenen Anforderungen aufzeichnen.

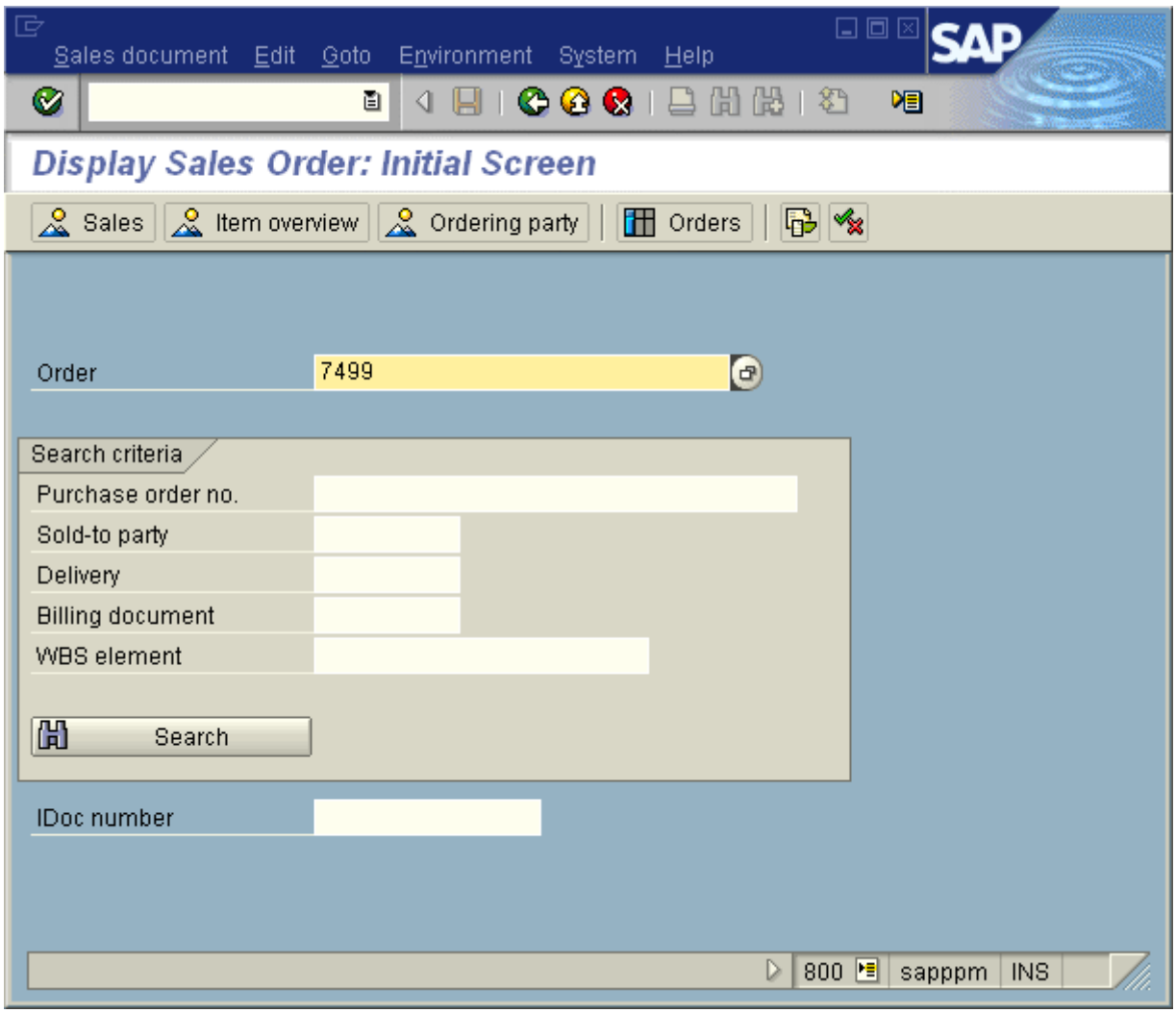
Starten Sie im SAP-Frontend den Transaktionsrekorder (Transaktion SHDB). Sie erhalten folgende Ansicht:



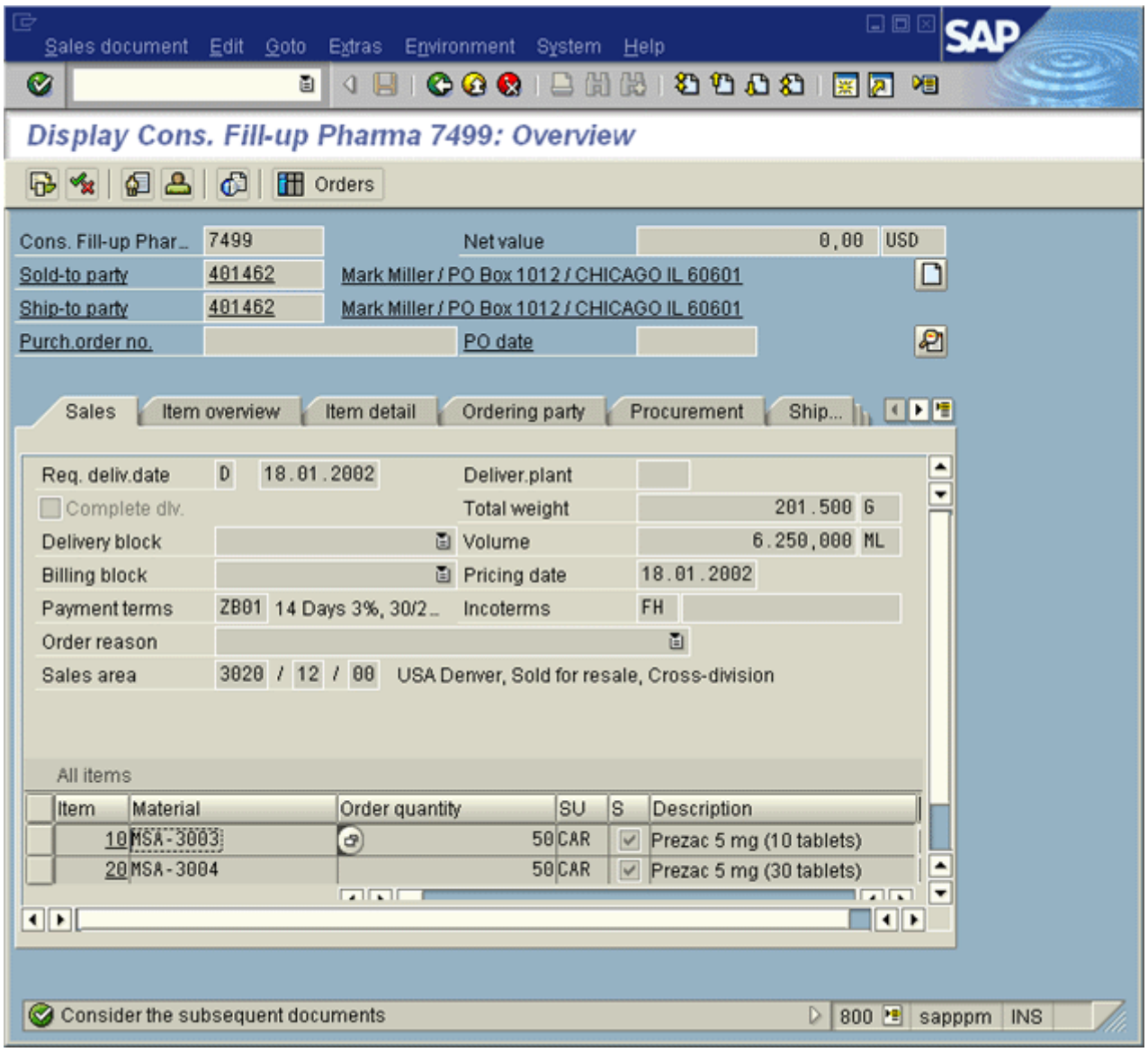
Legen Sie eine neue Aufzeichnung an und geben Sie im Feld Aufzeichnung einen beliebigen Namen ein, unter dem Sie die Aufzeichnung speichern möchten. Im Feld Transaktionscode geben Sie den Namen der aufzuzeichnenden Transaktion an:



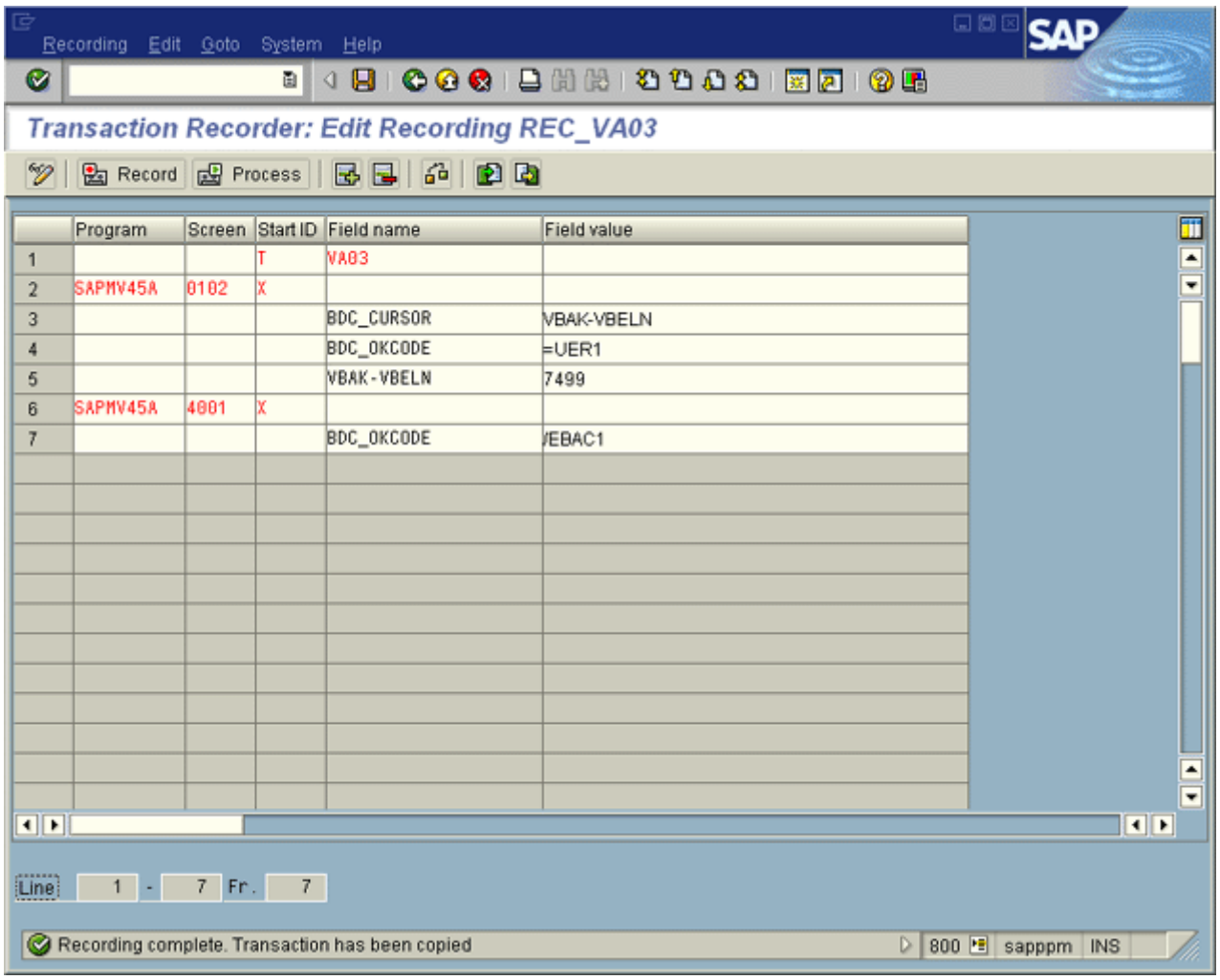
Starten Sie nun die Aufzeichnung und machen Sie in der folgenden Maske die erforderlichen Angaben, dasheißt, in den Feldern, die mit PPM-Prozessattributen gefüllt werden sollen, geben Sie entsprechende Werte ein und füllen auch die Felder aus, die beim Aufruf der Transaktion mit festen Werten belegt werden sollen. Im vorliegenden Beispiel gibt es keine Werte, die vorbelegt werden sollen, sondern nur die Auftragsnummer, die von der PPM-Prozessinstanz übernommen werden soll. Geben Sie dazu im Feld Auftrag eine in Ihrem System vorhandene Auftragsnummer ein:



Bestätigen Sie Ihre Eingaben mit der Taste **F5** und lassen Sie sich die Daten des Auftrags **7499** anzeigen:



Verlassen Sie nun die Transaktion mittels der Schaltfläche **Zurück** (F3). Sie sind jetzt wieder im Transaktionsrekorder und sehen Ihre Angaben im ABAP-Batch-Input-Format:



Die erste Zeile des Batch-Input-Skripts entspricht dem Transaktionsaufruf. Die beiden letzten Zeilen stellen die Auftragsanzeige und das Betätigen der Schaltfläche **Zurück** dar. Diese Zeilen sind in der anschließenden Erstellung der XML-Transaktionskonfiguration zu ignorieren.

Die Angaben aller anderen Zeilen übertragen Sie gemäß folgender Matrix in das XML-Format der batchinputline-Elemente.

Spaltenname im ABAP-Batch-Input-Format	XML-Attribut
Programm	program
Dynpro	dynpro
Startkennzeichen	dynprobegin
Feldname	fieldname
Feldwert	fieldvalue

Felder ohne Wert brauchen Sie nicht anzugeben, da dies dem Vorgabewert von fieldvalue entspricht. Das XML-Format des Beispiels sieht wie folgt aus:

```
...
<batchinputline program="SAPMV45A" dynpro="0102" dynprobegin="X"/>
<batchinputline fieldname="BDC_CURSOR" fieldvalue="VBAK-VBELN"/>
```

```
<batchinputline fieldname="BDC_OKCODE" fieldvalue="=UER1"/>
<batchinputline fieldname="VBAK-VBELN" fieldvalue="7499"/>
...
```

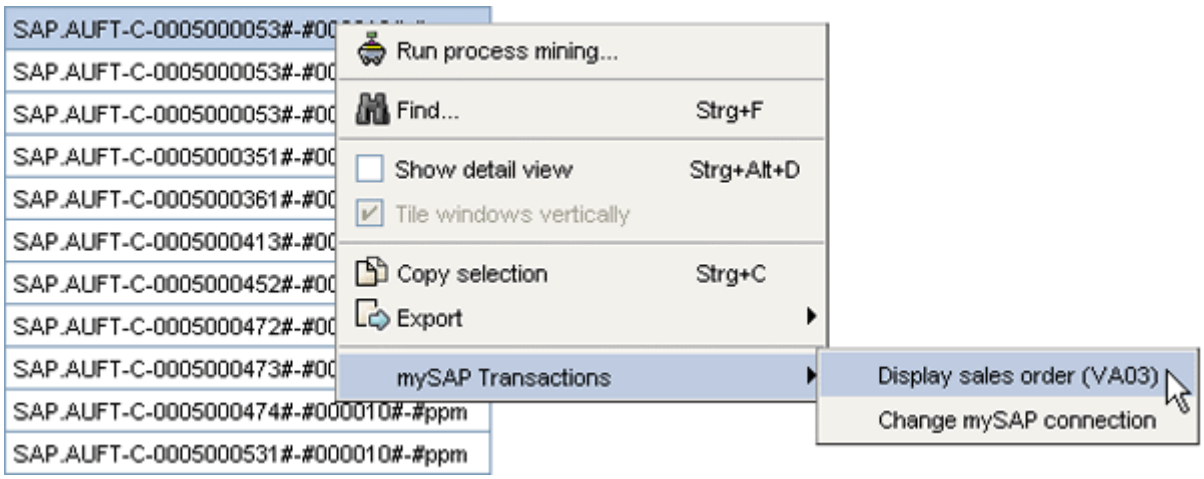
Damit nicht stets der Auftrag **7499** angezeigt wird, ganz gleich von welcher Prozessinstanz innerhalb der Prozesstypgruppe Auftragsabwicklung Sie die Transaktion **VA03** aufrufen, ersetzen Sie im letzten Element batchinputline die statische Wertangabe für das Feld mit der Auftragsnummer (hier: VBAK-VBELN) durch das entsprechende PPM-Prozessattribut, das in Ihrer Prozessinstanz die Auftragsnummer enthält, z. B. AT_SALES_ORDER_NUMBER.

Zusammen mit (beispielhaften) Verbindungsdaten sieht die Transaktionskonfiguration nun wie folgt aus (ABAP-Batch-Input-Daten in Fettschrift):

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE r3transactionconf SYSTEM "mysaptransaction.dtd">
<r3transactionconf>
<submenu>
<description language="de" name="SAP-Transaktionen"/>
<description language="en" name="SAP transactions"/>
</submenu>
<logoutMenuEntry>
<description language="de" name="SAP-Verbindung ändern"/>
<description language="en" name="Change SAP connection"/>
</logoutMenuEntry>
<r3system systemid="sapppm" client="800">
<description language="de" name="SAP-System 'sapppm' "/>
<description language="en" name="SAP system 'sapppm' "/>
<locales>
<defaultlocale value="de"/>
<locale value="en"/>
</locales>
<applicationserver appserver="sapppm" systemnumber="00"/>
</r3system>
<transaction systemid="sapppm" transactioncode="VA03" transactionid="VA03"
proctypegroup="Order processing">
<description language="de" name="Auftrag anzeigen (VA03)"/>
<description language="en" name="Display sales order (VA03)"/>
<batchinputline program="SAPMV45A" dynpro="0102" dynprobegin="X"/>
<batchinputline fieldname="BDC_CURSOR" fieldvalue="VBAK-VBELN"/>
<batchinputline fieldname="BDC_OKCODE" fieldvalue="=UER1"/>
<batchinputline fieldname="VBAK-VBELN" attributname="AT_SALES_ORDER_NUMBER"/>
</transaction>
</r3transactionconf>
```

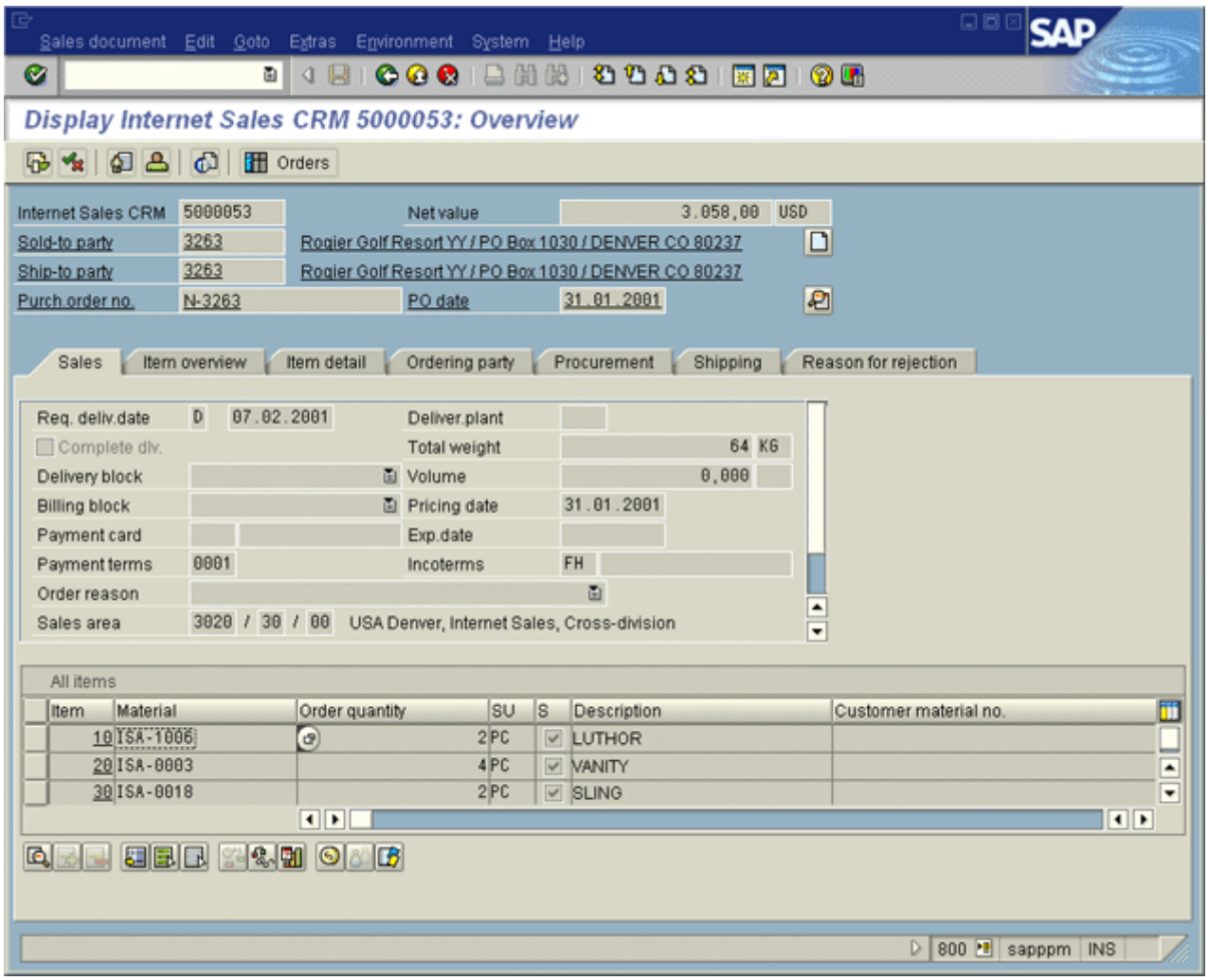
Anzeige in PPM

Das Kontextmenü zum Transaktionsaufruf auf einer gewählten Prozessinstanz des Prozesstyps **Auftragsabwicklung\sonstige Aufträge** sieht gemäß obiger Konfiguration folgendermaßen aus:



Der anschließend erscheinende SAP-Anmeldedialog sieht gemäß Konfiguration folgendermaßen aus:

Nach erfolgreicher Authentifizierung des Benutzers im SAP-System wird die Transaktion **VA03** im SAP-Frontend aufgerufen und es werden die Daten zur Auftragsnummer der gewählten Prozessinstanz (hier: 5000053) angezeigt:



Multi Select-Transaktion ME5F aufzeichnen

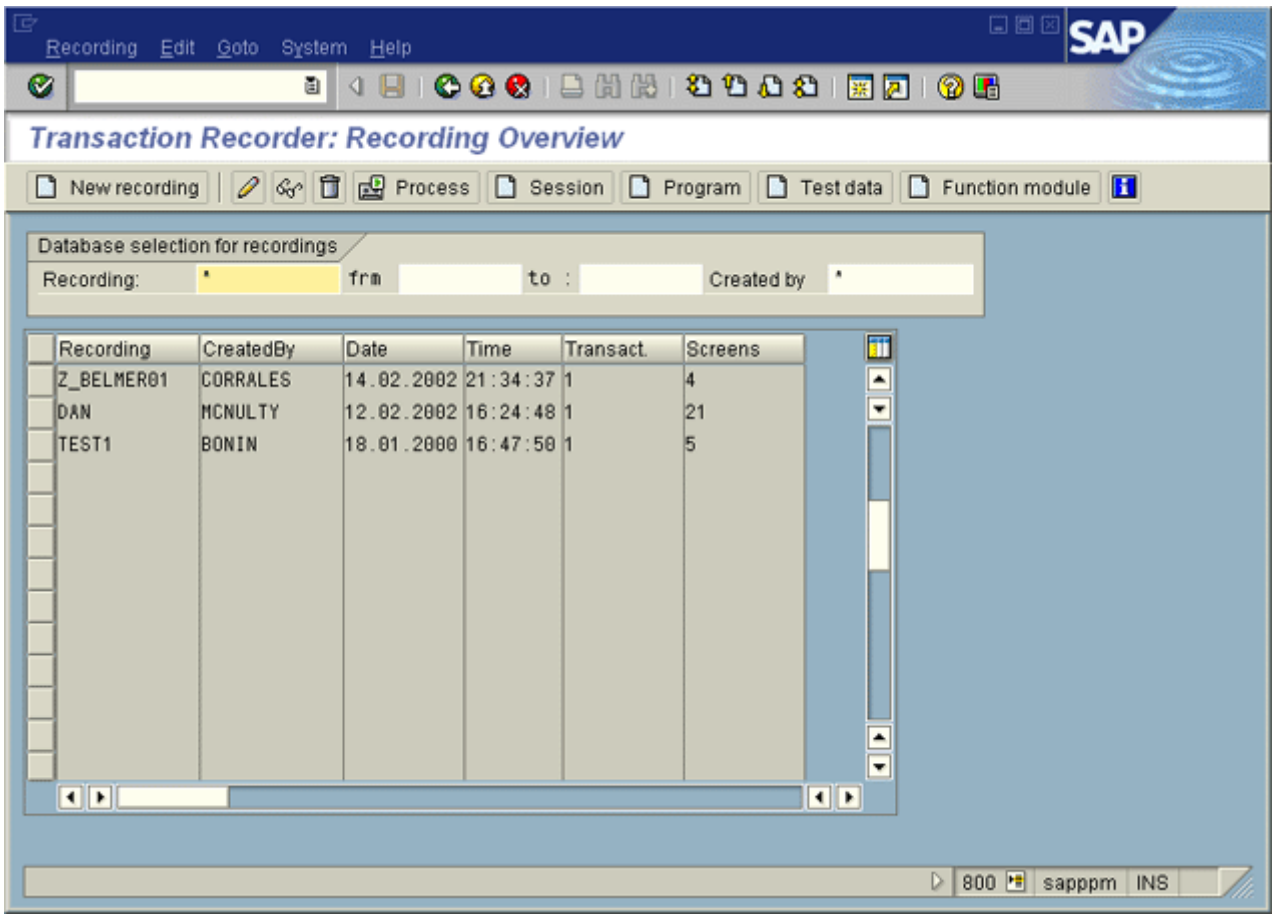
Anforderungen

Wenn mehrere Prozessinstanzen der Prozesstypgruppe Bestellanforderungen in der Prozessinstanztabelle selektiert sind, soll die Transaktion **ME5F** (Freigabeerinnerung: Bestellanforderungen) in dem SAP-System mit der ID **sapppm** aufgerufen werden können. Bei jedem Aufruf der Transaktion **ME5F** sollen als Freigabecode **KY** und als Freigabegruppe **01** verwendet werden.

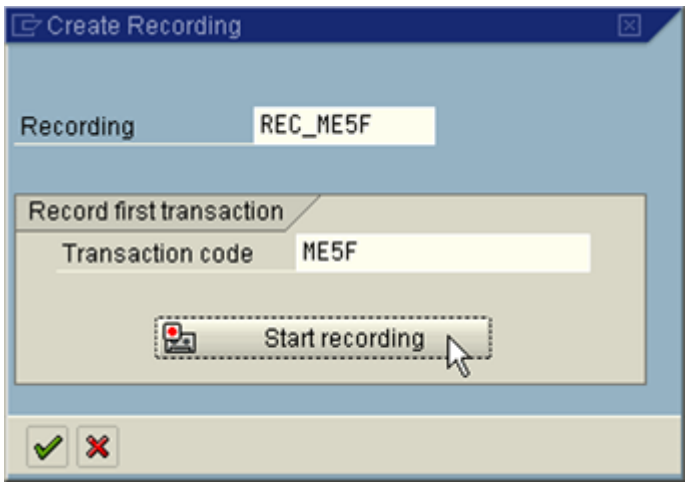
Der Transaktion sollen die Werte des PPM-Prozessattributs **AT_BANF_NUMBER** (Bestellanforderungsnummer) der gewählten Prozessinstanzen übergeben werden.

Im Folgenden ist schrittweise beschrieben, wie Sie mittels SAP-Transaktionsrekorder die Transaktion **ME5F** (Freigabeerinnerung BANF) unter Berücksichtigung der gegebenen Anforderungen aufzeichnen.

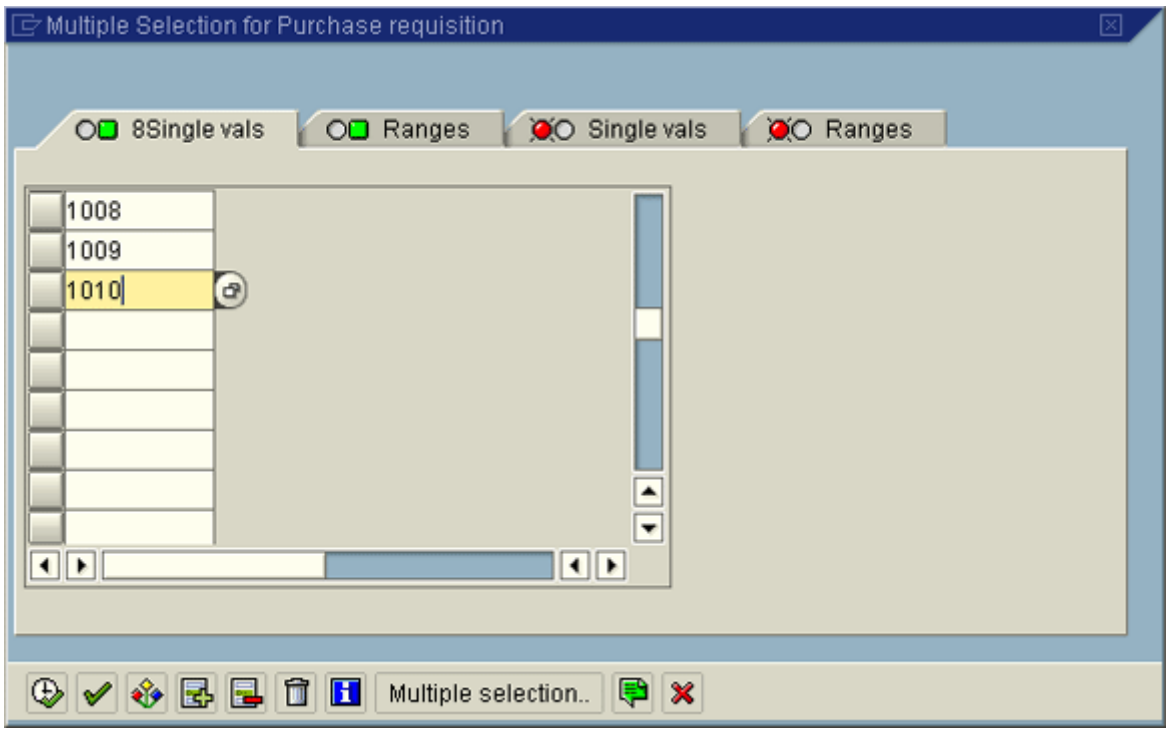
Starten Sie im SAP-Frontend den Transaktionsrekorder (Transaktion SHDB). Sie erhalten folgende Ansicht:



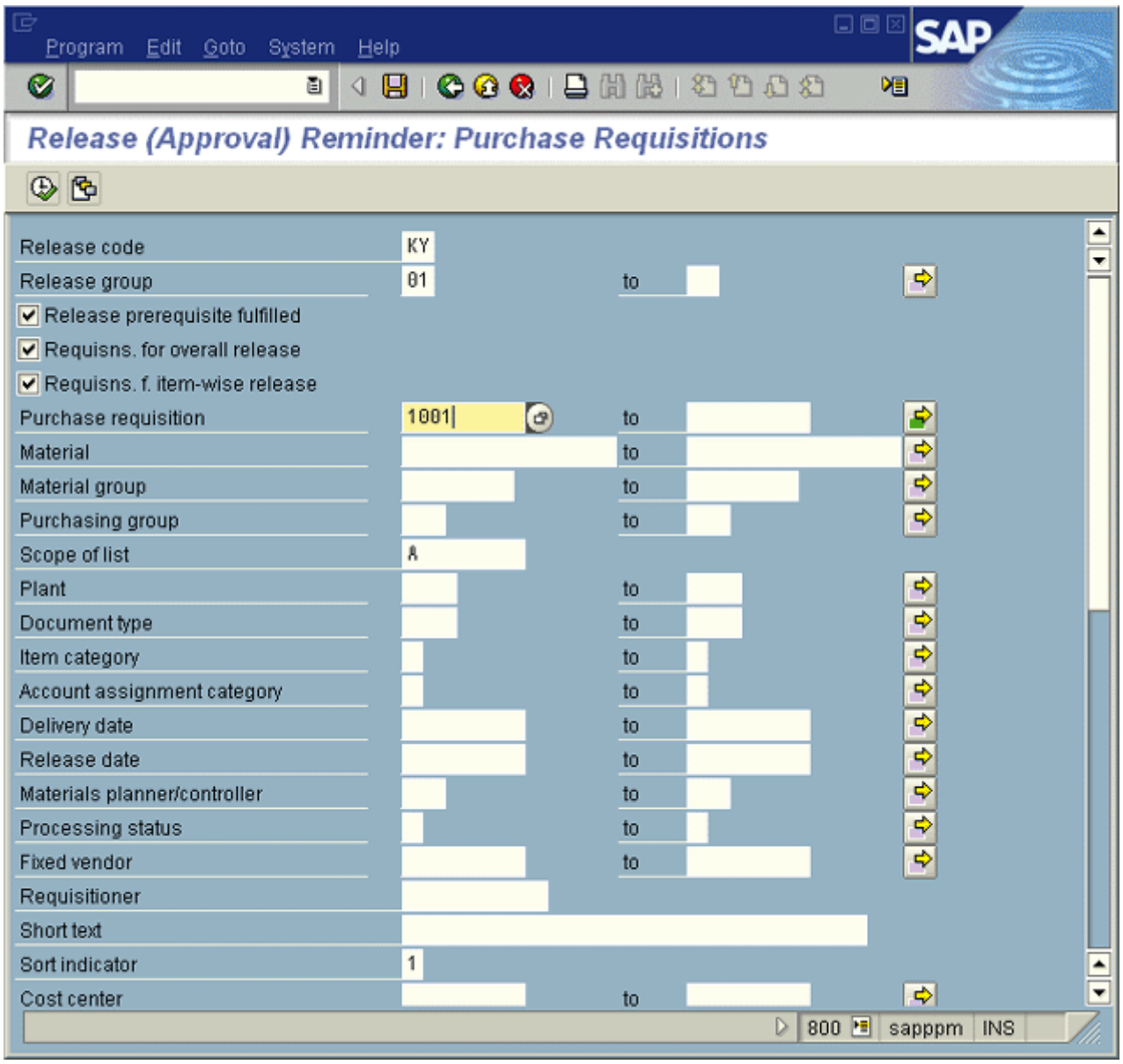
Legen Sie eine neue Aufzeichnung an und geben Sie im Feld Aufzeichnung einen beliebigen Namen ein, unter dem Sie die Aufzeichnung speichern möchten. Im Feld **Transaktionscode** geben Sie den Namen der aufzuzeichnenden Transaktion an:



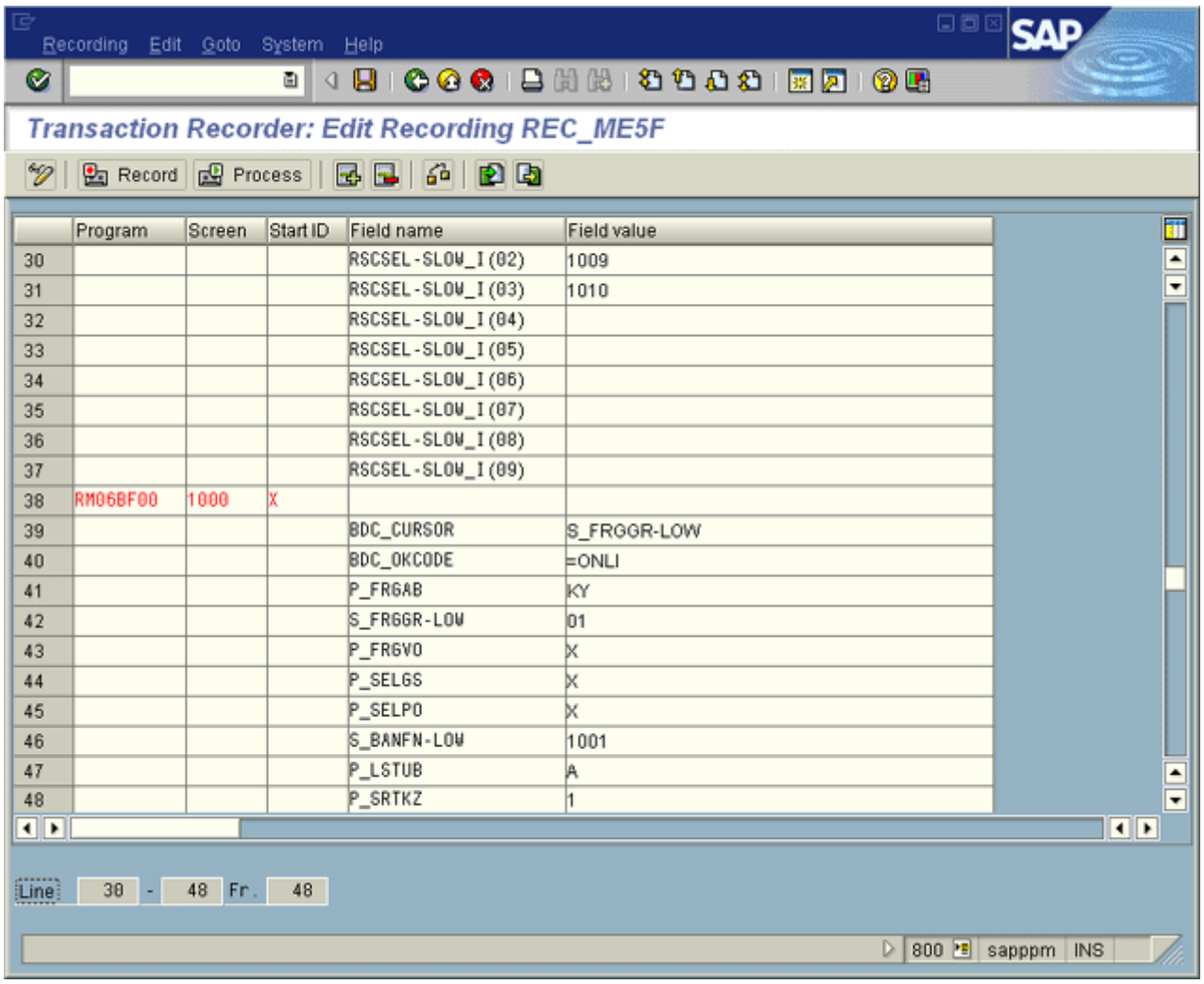
Starten Sie nun die Aufzeichnung und machen Sie in der folgenden Maske die erforderlichen Angaben. Als Bestellanforderungsnummern geben Sie die Nummern **1001–1010** als Einzelwerte an. Navigieren Sie im Dialog mit der **Bild nach unten**-Taste, um nicht funktionierenden OK-Code im ABAP-Batch-Skript mittels vertikaler Bildlaufleiste zu vermeiden.



Geben Sie im Feld **Freigabecode** den Wert **KY** und im Feld **Freigabegruppe** den Wert **01** ein. Übernehmen Sie Ihre Eingaben mit der Taste **F8**. Ihre Transaktion sieht nun folgendermaßen aus:



Führen Sie nun durch erneutes Betätigen der Taste **F8** die Transaktion mit den angegebenen Werten aus und lassen Sie sich die entsprechenden Bestellanforderungen im ABAP-Batch-Skript-Format anzeigen:



Anstatt die Werte direkt aus dem SAP-Frontend zu übernehmen, können Sie das Skript auch mittels Umsch+F8 als DAT-Datei exportieren und sich den Inhalt mit eingeblendeten Zeilennummern im Editor anzeigen lassen:

```
 1      0000    T   ME5F
 2  RM06BF00    1000    X
 3      0000        BDC_CURSOR  S_FRGGR-LOW
 4      0000        BDC_OKCODE  =%005
 5      0000        P_FRGAB KY
 6      0000        S_FRGGR-LOW 01
 7      0000        P_FRGVO X
 8      0000        P_SELGS X
 9      0000        P_SELPO X
10      0000        P_LSTUB A
11      0000        P_SRTKZ 1
12  SAPLALDB    3000    X
13      0000        BDC_OKCODE  =P+
14      0000        BDC_SUBSCR  SAPLALDB                                3010SCREEN_HEADER
15      0000        BDC_CURSOR  RSCSEL-SLOW_I(08)
16      0000        RSCSEL-SLOW_I(01)   1001
17      0000        RSCSEL-SLOW_I(02)   1002
18      0000        RSCSEL-SLOW_I(03)   1003
19      0000        RSCSEL-SLOW_I(04)   1004
20      0000        RSCSEL-SLOW_I(05)   1005
21      0000        RSCSEL-SLOW_I(06)   1006
22      0000        RSCSEL-SLOW_I(07)   1007
23      0000        RSCSEL-SLOW_I(08)   1008
24      0000        RSCSEL-SLOW_I(09)   _________
25  SAPLALDB    3000    X
26      0000        BDC_OKCODE  =ACPT
27      0000        BDC_SUBSCR  SAPLALDB                                3010SCREEN_HEADER
28      0000        BDC_CURSOR  RSCSEL-SLOW_I(03)
29      0000        RSCSEL-SLOW_I(01)   1008
30      0000        RSCSEL-SLOW_I(02)   1009
31      0000        RSCSEL-SLOW_I(03)   1010
32      0000        RSCSEL-SLOW_I(04)
33      0000        RSCSEL-SLOW_I(05)
34      0000        RSCSEL-SLOW_I(06)
35      0000        RSCSEL-SLOW_I(07)
36      0000        RSCSEL-SLOW_I(08)
37      0000        RSCSEL-SLOW_I(09)
38  RM06BF00    1000    X
39      0000        BDC_CURSOR  S_FRGGR-LOW
40      0000        BDC_OKCODE  =ONLI
41      0000        P_FRGAB KY
42      0000        S_FRGGR-LOW 01
43      0000        P_FRGVO X
44      0000        P_SELGS X
45      0000        P_SELPO X
46      0000        S_BANFN-LOW 1001
47      0000        P_LSTUB A
48      0000        P_SRTKZ 1
```

Die erste Zeile des Batch-Input-Skripts entspricht dem Transaktionsaufruf. Ignorieren Sie diese Zeile zusammen mit den Zeilen **41-48**, da es sich um Eingabewiederholungen aus den Zeilen **5-11** bzw. wie in Zeile **46** um einen automatischen Übertrag der unteren Grenze aus dem Mehrfachselektionsdialog handelt. Auch die Zeilen mit den Feldnamen **BDC_SUBSCR** (bspw. Zeile **14**) sind in der anschließenden Erstellung der XML-Transaktionskonfiguration zu ignorieren.

Die Angaben aller anderen Zeilen übertragen Sie gemäß folgender Matrix in das XML-Format der batchinputline-Elemente.

Spaltenname im ABAP-Batch-Input-Format	DAT-Datei Zeilennummer	XML-Attribut
Programm	2, 12, 25, 38	program
Dynpro	2, 12, 25, 38	dynpro
Startkennzeichen	2, 12, 25, 38	dynprobegin
Feldname	z.B. 3-11 (1. Maske)	fieldname
Feldwert	z.B. 3-11 (1. Maske)	fieldvalue

Felder ohne Wert brauchen Sie nicht anzugeben, da dies dem Vorgabewert entspricht.

```
...
<!-- 1st screen -->
<batchinputline program="RM06BF00" dynpro="1000" dynprobegin="X"/>
<batchinputline fieldname="BDC_CURSOR" fieldvalue="S_FRGGR-LOW"/>
<batchinputline fieldname="BDC_OKCODE" fieldvalue="=%005"/>
<batchinputline fieldname="P_FRGAB" fieldvalue="KY" />
<batchinputline fieldname="S_FRGGR-LOW" fieldvalue="01" />
<batchinputline fieldname="P_FRGVO" fieldvalue="X" />
<batchinputline fieldname="P_SELGS" fieldvalue="X" />
<batchinputline fieldname="P_SELPO" fieldvalue="X" />
<batchinputline fieldname="P_LSTUB" fieldvalue="A" />
<batchinputline fieldname="P_SRTKZ" fieldvalue="1" />

<!-- 2nd screen -->
<batchinputline program="SAPLALDB" dynpro="3000" dynprobegin="X"/>
<batchinputline fieldname="BDC_OKCODE" fieldvalue="=P+" />
<batchinputline fieldname="BDC_CURSOR" fieldvalue="RSCSEL-SLOW_I(08)"/>
<batchinputline fieldname="RSCSEL-SLOW_I(01)" fieldvalue="1001"/>
<batchinputline fieldname="RSCSEL-SLOW_I(02)" fieldvalue="1002"/>
<batchinputline fieldname="RSCSEL-SLOW_I(03)" fieldvalue="1003"/>
<batchinputline fieldname="RSCSEL-SLOW_I(04)" fieldvalue="1004"/>
<batchinputline fieldname="RSCSEL-SLOW_I(05)" fieldvalue="1005"/>
<batchinputline fieldname="RSCSEL-SLOW_I(06)" fieldvalue="1006"/>
<batchinputline fieldname="RSCSEL-SLOW_I(07)" fieldvalue="1007"/>
<batchinputline fieldname="RSCSEL-SLOW_I(08)" fieldvalue="1008"/>
<batchinputline fieldname="RSCSEL-SLOW_I(09)" fieldvalue="__________"/>

<!-- 3rd screen -->
<batchinputline program="SAPLALDB" dynpro="3000" dynprobegin="X"/>
<batchinputline fieldname="BDC_OKCODE" fieldvalue="=ACPT"/>
<batchinputline fieldname="BDC_CURSOR" fieldvalue="RSCSEL-SLOW_I(03)"/>
<batchinputline fieldname="RSCSEL-SLOW_I(01)" fieldvalue="1008"/>
<batchinputline fieldname="RSCSEL-SLOW_I(02)" fieldvalue="1009"/>
<batchinputline fieldname="RSCSEL-SLOW_I(03)" fieldvalue="1010"/>
<batchinputline fieldname="RSCSEL-SLOW_I(04)" fieldvalue=""/>
<batchinputline fieldname="RSCSEL-SLOW_I(05)" fieldvalue=""/>
<batchinputline fieldname="RSCSEL-SLOW_I(06)" fieldvalue=""/>
<batchinputline fieldname="RSCSEL-SLOW_I(07)" fieldvalue=""/>
<batchinputline fieldname="RSCSEL-SLOW_I(08)" fieldvalue=""/>
<batchinputline fieldname="RSCSEL-SLOW_I(09)" fieldvalue=""/>

<!-- 4th screen -->
<batchinputline program="RM06BF00" dynpro="1000" dynprobegin="X"/>
<batchinputline fieldname="BDC_OKCODE" fieldvalue="=ONLI"/>
...
```

Diese Transaktionskonfiguration würde zwar zusammen mit den Verbindungsdaten aus dem vorhergehenden Beispiel funktionieren, hätte aber zur Folge, dass, ganz gleich welche Prozessinstanzen in der Prozesstypgruppe **Bestellanforderungen** gewählt sind, stets eine Transaktion zu einer Bestellanforderung mit der Nummer **1001–1010** angezeigt würde.

Um zu jeder gewählten Prozessinstanz die tatsächlich gehörende Bestellanforderung anzeigen zu lassen, ersetzen Sie in der bisher erstellten Konfiguration die beiden Bildschirmseiten der Mehrfachselektion

(alle XML-Elemente batchinputline der 2. und 3. Bildschirmseite) durch jeweils ein einziges XML-Element batchinputlist. Hierzu sind folgende Angaben notwendig:

XML-Attribut	Wert (Beschreibung)
program	SAPLALDB (Name des Programmes)
dynpro	3000 (Dynpro-Name)
okcodefieldname	BDC_OKCODE (Name des Dynpro-Felds, das den OK-Code enthält)
okcodepagedown	=P+ (Wert des OK-Codes, um eine Seite nach unten zu blättern)
okcodeaccept	=ACPT (Wert des OK-Codes, um die Eingabe in der Mehrfachselektion zu akzeptieren)
fieldname	RSCSEL-SLOW_I (Name des Feldes, dem die Werte des PPM-Attributs zugewiesen werden sollen. Das Feld ist ohne die Zeilenindizierung [(01), (02) usw.] anzugeben)
attributname	AT_BANF_NUMBER (Interner Bezeichner des PPM-Prozessattributs, dessen Werte der aufgerufenen Transaktion übergeben werden)
linesperpage	9 (Anzahl der Eingabezeilen auf jeder Bildschirmseite der Mehrfachselektion)

Das batchinputlist-Element sieht auf Basis dieser Angaben wie folgt aus:

```
<batchinputlist program="SAPLALDB" dynpro="3000"
okcodefieldname="BDC_OKCODE" okcodepagedown="=P+"
okcodeaccept="=ACPT" fieldname="RSCSEL-SLOW_I"
attributname="AT_BANF_NUMBER" linesperpage="9" />
```

Zusammen mit den Verbindungsdaten aus dem Single Select-Beispiel sieht die Transaktionskonfiguration nun wie folgt aus:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE r3transactionconf SYSTEM "mysaptransaction.dtd">
<r3transactionconf>
<submenu>
<description language="de" name="SAP-Transaktionen"/>
<description language="en" name="SAP transactions"/>
</submenu>
<logoutMenuEntry>
<description language="de" name="SAP-Verbindung ändern"/>
<description language="en" name="Change SAP connection"/>
</logoutMenuEntry>
<r3system systemid="sapppm" client="800">
<description language="de" name="SAP-System 'sapppm' "/>
<description language="en" name="SAP system 'sapppm' "/>
<locales>
<defaultlocale value="de"/>
<locale value="en"/>
</locales>
<applicationserver appserver="sapppm" systemnumber="00"/>
</r3system>
<transaction
systemid="sapppm" transactioncode="ME5F" transactionid="ME5F">
<description language="de" name="Freigabeerinnerung BANF (ME5F)"/>
<description language="en" name="Release (approval) reminder: Purchase
```

```
Requisitions (ME5F)"/>
<!-- 1st screen -->
<batchinputline program="RM06BF00" dynpro="1000" dynprobegin="X"/>
<batchinputline fieldname="BDC_CURSOR" fieldvalue="S_FRGGR-LOW"/>
<batchinputline fieldname="BDC_OKCODE" fieldvalue="=%005"/>
<batchinputline fieldname="P_FRGAB" fieldvalue="KY" />
<batchinputline fieldname="S_FRGGR-LOW" fieldvalue="01"/>
<batchinputline fieldname="P_FRGVO" fieldvalue="X" />
<batchinputline fieldname="P_SELGS" fieldvalue="X" />
<batchinputline fieldname="P_SELPO" fieldvalue="X" />
<batchinputline fieldname="P_LSTUB" fieldvalue="A" />
<batchinputline fieldname="P_SRTKZ" fieldvalue="1" />
<!-- Multiple selection screens -->
<batchinputlist program="SAPLALDB" dynpro="3000"
okcodefieldname="BDC_OKCODE" okcodepagedown="=P+"
okcodeaccept="=ACPT" fieldname="RSCSEL-SLOW_I"
attributname="AT_BANF_NUMBER" linesperpage="9" />
<!-- 1st screen -->
<batchinputline program="RM06BF00" dynpro="1000" dynprobegin="X"/>
<batchinputline fieldname="BDC_OKCODE" fieldvalue="=ONLI"/>
</transaction>
</r3transactionconf>
```

10.1.4.2. Erläuterungen zur DTD

Die Konfigurationsmöglichkeiten sind durch die Datei **mysaptransaction.dtd** vorgegeben:

Konfiguration des Kontextmenüs

XML-Tag	Beschreibung
submenu (optional)	Name des Untermenüs. Standardwert: **SAP-Transaktionen**. Ist nur ein Untermenüeintrag vorhanden, wird dieser direkt im Kontextmenü ohne Untermenüeintrag angezeigt (z.B. initiale Anmeldung bei nur einer konfigurierten Transaktion)
logoutMenuEntry (optional)	Menüeintrag zum Zurücksetzen aller Verbindungsparameter nach mindestens einer erfolgreichen Anmeldung im SAP-System. Standardwert: **Zurücksetzen aller Verbindungsparameter** der **SAP-Systeme**

Konfiguration von Systemzugängen

XML-Tag	Beschreibung
r3system	Beschreibung und Verbindungsdaten des verfügbaren SAP-Systems. Es können die Daten beliebig vieler SAP-Systeme angegeben werden. Die Konfiguration von Systemzugängen ist ausführlich in der Technischen Referenz **PPM Process Extractors** beschrieben.
description	Mindestens die Beschreibung in der Standardsprache muss gepflegt sein.
name	Name des SAP-Systems im SAP-Anmeldedialog im PPM-Frontend
systemid	Eindeutiger Name eines SAP-Systems. Wird von der jeweiligen Transaktion referenziert.
client	Name des SAP-Mandanten.
locales	Verfügbare Sprachen des SAP-Systems
defaultlocale	Standardsprache. Wird im SAP-Anmeldedialog vorausgewählt

locale (optional)	Weitere gepflegte Sprache(n) des SAP-Systems
appserver	Rechnername oder IP-Adresse des SAP-Quellsystems
systemnumber	SAP-Systemnummer
mshost	Name des SAP Message Host
r3name	R/3-Systemname
group	Name der Application-Server-Gruppe
appserver	Name des Applikationsservers
systemnumber	SAP-Systemnummer
gwhost	Rechnername des R/3-Gateways
gwserv	Service-Nummer des R/3-Gateways

Konfiguration von Transaktionen

XML-Tag	Beschreibung
transaction	Konfiguration von Transaktionen
systemid	ID des SAP-Systems, in dem die Transaktion aufgerufen werden soll. Muss mit dem Wert einer **systemid** der angegebenen SAP-Systeme übereinstimmen (XML-Elemente **r3system**).
transactionid	ID der Transaktion
transactioncode	Der Transaktionscode der aufzurufenden Transaktion (siehe SAP-Dokumentation)
skipfirstscreen (optional)	Überspringen der Einstiegsseite der Transaktion, wenn alle Mussfelder ausgefüllt sind (siehe SAP-Dokumentation zum Funktionsbaustein CALL_TRANSACTION). Zulässige Werte: **yes** | **no** Standardwert: **yes**
mode (optional)	Ausführungsart des ABAP-Batch-Input (siehe SAP-Dokumentation zum Funktionsbaustein CALL_TRANSACTION). Zulässige Werte: **SHOW_DYNPROS** (Dynpros werden während der Ausführung angezeigt) **SHOW_DYNPROS_ONLY_ON_ERRORS** (Dynpros werden nur angezeigt, wenn ein Fehler auftritt, bzw. das Ende des Batch-Skripts erreicht wurde) **DONT_SHOW_DYNPROS** (Dynpros werden nicht angezeigt) Standardwert: **SHOW_DYNPROS_ONLY_ON_ERRORS**
update (optional)	Verbuchungsart im SAP-System (siehe SAP-Dokumentation zum Funktionsbaustein CALL_TRANSACTION). Zulässige Werte: **SYNCHRONOUS** (synchrone Verbuchung) **ASYNCHRONOUS** (asynchrone Verbuchung) **LOCAL** (lokale Verbuchung) Standardwert: **ASYNCHRONOUS**
proctypegroup (optional)	Prozesstypgruppe, in der die Transaktion verfügbar ist. Die Transaktion ist automatisch auch in allen Prozesstypen der angegebenen Prozesstypgruppe verfügbar. Fehlt die Angabe, ist die Transaktion im gesamten Prozessbaum verfügbar.

description	Sprachabhängiger Oberflächennamen der Transaktion
name	Kontextmenüeintrag der Transaktion
batchinputline	Eine Zeile im ABAP-Batch-Input-Format. Sind in einer Transaktionskonfiguration nur XML-Elemente **batchinputline** enthalten, kann die Transaktion i.d.R. nur bei Einfachselektion aufgerufen werden. Mehrfachselektion ist nur dann möglich, wenn das mit **attributname** angegebene PPM-Prozessattribut in allen gewählten Prozessinstanzen denselben Wert hat.
program	Name des Programmes
dynpro	Dynpro-Name
dynprobegin	Start eines Dynpros
fieldname	Name des Dynpro-Felds
attributname (optional)	Interner Bezeichner eines PPM-Prozessattributs, dessen Wert von der gewählten Prozessinstanz ermittelt werden soll. Der Wert wird dem Dynpro-Feld zugewiesen, **fieldvalue** wird dann nicht beachtet.
fieldvalue (optional)	Ein konstanter Wert, der dem Dynpro-Feld zugewiesen werden soll. Ist **attributname** gepflegt, wird **fieldvalue** ignoriert.
batchinputlist	Mehrere Zeilen im ABAP-Batch-Input-Format. Ist in einer Transaktionskonfiguration mindestens ein XML-Element **batchinputlist** enthalten, kann die Transaktion sowohl bei Einfach- als auch Mehrfachselektion aufgerufen werden.
program	Name des Programmes
dynpro	Dynpro-Name
okcodefieldname (optional)	Name des Dynpro-Felds, das den OK-Code enthält. Standardwert: **BDC_OKCODE**
okcodepagedown (optional)	Wert des OK-Codes, um eine Seite nach unten zu blättern. Standardwert: **=P+**
okcodeaccept (optional)	Wert des OK-Codes, um die Eingabe zu akzeptieren. Standardwert: **=ACPT**
fieldname	Name des Dynpro-Felds
attributname	Interner Bezeichner eines PPM-Prozessattributs, dessen Wert von der gewählten Prozessinstanz ermittelt werden soll. Der Wert wird dem Dynpro-Feld zugewiesen, **fieldvalue** wird dann nicht beachtet.
linesperpage (optional)	Anzahl der sichtbaren Wertezeilen auf dem Dynpro. Nach dieser Anzahl von Zeilen wird eine Zeile nach unten geblättert. Standardwert: **9**

11. Support und rechtliche Hinweise

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

11.1. Dokumentationsumfang

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

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

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

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

11.2. Support

Produktsupport

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

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

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

Community

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

Dokument

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

Wenn Sie zu Dokumentationsinhalten Feedback geben möchten, schreiben Sie an documentation@aris.com.

11.3. Legal notices

This document applies to ARIS Process Performance Manager Version 10 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2000-2026 SAG Aris GmbH, Saarbrücken, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name ARIS and all SAG Aris GmbH product names are either trademarks or registered trademarks of SAG Aris GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

The Software may contain or be distributed with certain commercial third party components (“Third Party Components”), which may be subject to special license terms and conditions (“Third Party Terms”) located at https://aris.com/terms.

Third Party Terms include important licensing and warranty information and disclaimers of third party licensors. For the avoidance of doubt, if Customer uses the Software in accordance with the Documentation, the Third Party Terms do not restrict the rights granted under this Agreement. In the event that a third party manufacturer no longer provides active support for any Third Party Component, Supplier will use commercially reasonable efforts to respond to any issues with the Software related to the Third Party Components.

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with SAG Aris GmbH and/or its subsidiaries and/or its affiliates and/or their licensors.