IMPORTAÇÃO DE DADOS

ARIS Process Performance Manager 10

Outubro 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.

Índice

1. Geral 5
2. XML 6
2.1. O que é o XML? 6
2.2. Estrutura de um documento XML 6
3. Importação de dados XML 7
3.1. Formato gráfico 7
3.1.1. Tipos de objeto 9
3.1.2. Conexões 9
3.1.3. Relações (opcional) 10
3.1.4. Diretrizes para a estrutura do gráfico 10
3.1.5. Atributos 11
3.1.6. Gráfico XML de exemplo 11
3.2. Formato do evento de sistema 14
3.2.1. Definição de fragmentos de processo 15
3.2.2. Definição de mapeamento 17
3.2.2.1. Definição de mapeamento de fragmentos de processo 17
3.2.2.2. Definição do mapeamento de atributo 20
3.2.2.3. Unidades organizacionais 27
3.2.2.4. Caso especial de mapeamento de atributo 29
3.2.3. Criar definições de fragmento em ARIS 31
3.2.3.1. Modelar o processo geral 31
3.2.3.2. Modelar as definições de fragmento de processo 32
3.2.3.3. Formato do arquivo do evento de sistema 35
3.2.3.4. Executar o relatório de ARIS 36
3.2.4. Gerar o arquivo de saída XML 37
3.2.5. Sumário 38
3.3. Formatos da data 43
3.3.1. Caracteres especiais em documentos XML 44
3.4. Gerar os fragmentos de instância de processo 44
3.4.1. Expandir a configuração de atributos 45
3.4.1.1. Especifique o tipo de dados de atributos desconhecidos 46
3.4.2. Expandir a configuração de mapeamento 50
3.4.3. Atributos de evento de sistema de valores múltiplos 51
3.4.4. Importação direta de atributos de processo 52
3.4.5. Caso especial de sistema dimensionado 54
3.4.6. Arquivamento de arquivos de importação XML 56
3.5. programa de linha de comando runxmlimport 56
3.5.1. argumentos runxmlimport 58
3.6. Importar várias fontes de dados 60
3.7. Reimportar os mesmos dados 61
3.7.1. Formato gráfico 61
3.7.2. Formato do evento de sistema 61
4. Importação de dados independentes da instância de processo 63
4.1. Medidas independentes da instância de processo 63
4.1.1. Formatos de importação de dados 63
4.1.1.1. Formato XML 64
4.1.1.2. Formato CSV 67
4.1.1.3. Formato XLS 68
4.1.2. Reimportação de dados 70
4.1.3. Exportar valores da série de dados independentes da instância de processo 71
4.1.4. Excluir valores da série de dados independentes da instância de processo 71
4.1.5. programa de linha de comando runpikidata 72
4.2. Valores de dimensão 75

4.2.1. Formato XML 75
4.2.2. Formato CSV 76
4.2.3. Valores padrão e de substituição 77
4.2.4. Reimportação de dados 77
4.2.5. Excluir valores de dimensão 78
4.2.6. programa de linha de comando rundimdata 78
4.3. Análise de dados 80
5. Como tratar EPCs grandes 81
5.1. Importar EPCs grandes 81
5.2. Excluir EPCs grandes 81
6. Apêndice 82
6.1. Projeto de um armazém de processos 82
6.1.1. Gerar fragmentos de processo 84
6.1.2. Merge de fragmentos de processo 85
6.1.2.1. Copiar os atributos da instância de processo 88
6.1.2.2. Tornar as unidades organizacionais anônimas 88
6.1.3. Tipificar processos 88
6.1.4. Calcular medidas 89
6.1.5. Verificar valores planejados 89
6.2. Cenários de importação 90
6.2.1. Valores de parâmetros 91
7. Suporte e informações legais 92
7.1. Escopo da documentação 92
7.2. Suporte 92
7.3. Legal notices 93

1. Geral

O PPM usa o **XML** como um formato de dados universal, ou seja, todo o sistema PPM pode ser configurado usando arquivos XML.

Esta referência técnica descreve as interfaces XML do PPM que são usadas para importar do sistema de origem para o sistema PPM como arquivos XML. O PPM usa seus próprios formatos de dados para importar dados, ou seja, **formato de evento do sistema** e **formato gráfico**.

Os dados do sistema de origem são extraídos em primeiro lugar de praticamente qualquer tipo de sistema de origem (por exemplo, SAP, JDBC, CSV) usando os extratores de processo do PPM (formato de evento de sistema) ou outros adaptadores.

O capítulo **Projeto de um armazém de processos [82]** fornece uma visão baseada em projeto de como os dados do sistema de origem extraídos podem ser usados para gerar instâncias de processo que refletem as sequências dos processos do sistema de origem e estão disponíveis para análise de medidas adicionais no PPM.

2. XML

Este capítulo contém informações básicas sobre XML, necessárias para entender os capítulos seguintes.

2.1. O que é o XML?

A abreviatura XML significa e**X**tensible **M**arkup **L**anguage. XML é uma metalinguagem para a descrição de idiomas de exibição, como HTML. As metalinguagens fornecem as regras necessárias para a definição de tipos de documento. Os idiomas de exibição permitem que os documentos sejam exibidos corretamente.

2.2. Estrutura de um documento XML

Um documento XML é um arquivo de texto e é composto de dois tipos de caracteres: os dados reais e as chamadas tags ou marcações. Marcas são instruções XML, que descrevem a divisão do documento em unidades de armazenamento e sua estrutura lógica. A própria estrutura é salva em uma definição de tipo de documento (DTD).

As marcas são sempre escritas em pares e em parênteses angulares. Cada marca inicial sempre tem uma marca final correspondente.

Os atributos XML são usados dentro das marcas. Um atributo só pode ocorrer uma vez dentro de uma marca.

Os documentos XML são compostos por elementos. Um elemento é composto de duas marcas XML e o texto incluído. Elementos em branco são compostos por apenas uma marca e sempre terminam com uma barra (/) antes do parêntese final.

Pode criar documentos XML simples com um editor de texto. No exemplo a seguir, a DTD é especificada entre colchetes diretamente no arquivo XML:

```
<?xml version="1.0"?>
<!DOCTYPE memberlist
[
<!ELEMENT memberlist (no, name, age)>
<!ELEMENT no (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
]>
<memberlist>
<no>001</no>
<name>Doe, John</name>
<age>27</age>
</memberlist>
```

Se salvar este documento com o nome de sua escolha e a extensão **.xml**, o Internet Explorer pode exibir o documento em uma forma estruturada.

3. Importação de dados XML

Este capítulo descreve a importação baseada em XML dos dados da instância de processo.

Os dados da instância para os processos reais concluídos são extraídos do sistema de aplicativos operacional (sistema de origem) por software especial e salvos em arquivos de saída XML. Estes arquivos de saída são importados para o PPM usando a interface de importação XML. A estrutura interna dos arquivos XML é especificada por uma DTD (Definição de tipos de documento).

A interface de importação XML do PPM suporta dois formatos de importação diferentes, o formato gráfico do PPM e o formato de evento de sistema PPM.

Formato gráfico do PPM

O formato gráfico do PPM é usado para importar dados de processo já estruturados de sistemas de aplicativos orientados para processos (por exemplo, sistemas de fluxo de trabalho). O adaptador específico do aplicativo gera arquivos XML, nos quais as instâncias de processo, incluindo sua lógica do decurso, são descritas no formato gráfico do PPM. Ao contrário do formato de evento do sistema PPM, é possível importar instâncias de processo completas. Não é necessária uma operação de merge. Ao importar instâncias de processo completas para uma nova importação de dados da instância, as instâncias de processo completas devem sempre ser importadas.

O formato gráfico é usado no sistema PPM para a troca universal de dados baseados em EPC.

Formato do evento de sistema PPM

O formato de evento do sistema PPM é usado para todos os sistemas de aplicativos orientados para atividades, dos quais as informações que compõem o processo (lógica do decurso) não podem ser extraídas.

Ao importar dados no formato de evento do sistema, os eventos do sistema são registrados em um arquivo XML. Todos os tipos de eventos do sistema, que devem ser importados para o PPM, devem ser definidos em modelos de fragmento do processo antes da importação. As regras também são definidas para como esses modelos de fragmentos de processo são mesclados em um processo geral.

O PPM gera fragmentos de instância de processo mapeando os eventos de sistema para modelos de fragmentos de processo. Eles são então vinculados para formar instâncias de processo.

O formato de evento do sistema permite que instâncias de processo já importadas sejam estendidas e modificadas importando dados delta.

3.1. Formato gráfico

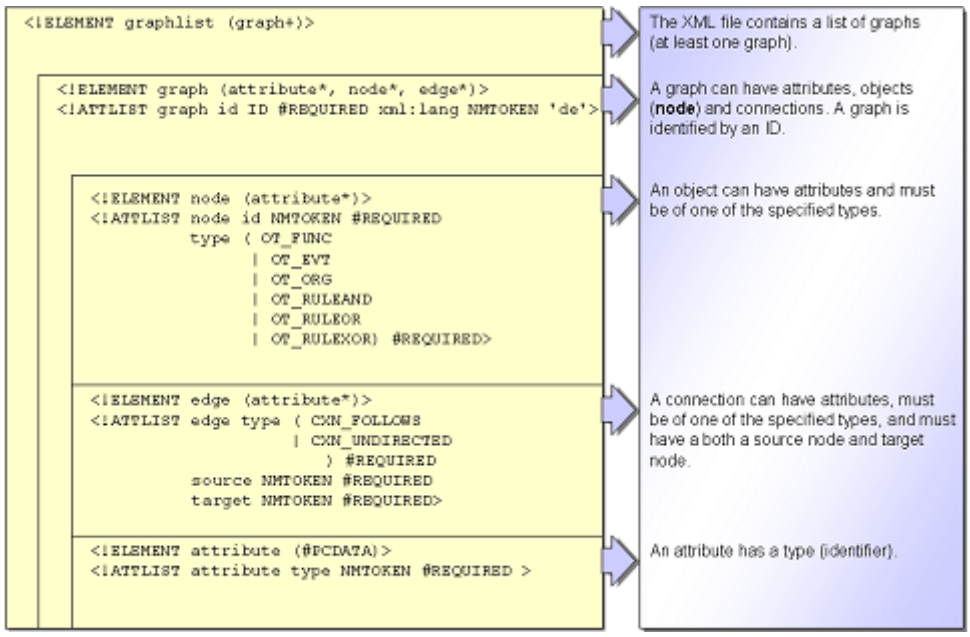
Um arquivo XML em formato gráfico do PPM contém uma lista de gráficos (EPCs). Cada gráfico representa uma instância de processo ou um fragmento de instância de processo. Um gráfico é composto de diferentes tipos de objetos, conexões e outras relações entre objetos. O gráfico e os objetos, conexões e relações podem ter atributos.

O arquivo XML abaixo contém um gráfico simples, que é composto de três objetos vinculados (evento - função - evento):

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

```
<graphlist>
<graph id="00093862" xml:lang="en">
<attribute type="AT_ID">XMLGraph-Job-00093862</attribute>
<attribute type="AT_EPK_KEY">00093862</attribute>
<attribute type="AT_PROCTYPE">Standard order</attribute>
<attribute type="AT_PROCTYPEGROUP">Order processing</attribute>

<node id="Start" type="OT_EVT">
<attribute type="AT_OBJNAME_INTERN">AUFTRAG_ANZU</attribute>
<attribute type="AT_OBJNAME">Customer order to be created</attribute>
</node>

<node id="Function" type="OT_FUNC">
<attribute type="AT_OBJNAME_INTERN">AUFTRAG</attribute>
<attribute type="AT_OBJNAME">Create customer order</attribute>
<attribute type="AT_START_TIME">14.2.2000 13:12:57</attribute>
<attribute type="AT_END_TIME">14.02.2000 13:22:57</attribute>
</node>

<node id="Processor" type="OT_ORG">
<attribute type="AT_OBJNAME">Ms. Smith</attribute>
</node>

<node id="End" type="OT_EVT">
<attribute type="AT_OBJNAME_INTERN">LIEFERUNG_ANZU</attribute>
<attribute type="AT_OBJNAME">Delivery to be created</attribute>
<attribute type="AT_ID">XMLGraph-Job-Evt2</attribute>
</node>

<edge type="CXN_FOLLOWS" source="Start" target="Function" />
<edge type="CXN_FOLLOWS" source="Function" target="End" />
<relationtype name="REL_CARRY_OUT">
<relation source="Processor" target="Function">
<attribute type="AT_KI_PK_R">7.5 EUR</attribute>
<attribute type="AT_KI_RNUM">1</attribute>
</relation>
</relationtype>
</graph>
</graphlist>
```

Quando importada para o PPM, a estrutura dos arquivos XML no formato gráfico é verificada na DTD a seguir:



3.1.1. Tipos de objeto

A tabela abaixo mostra todos os tipos de objetos usados no formato gráfico do PPM:

Tipo de objeto	Identificador	Descrição
Função	OT_FUNC	As funções descrevem atividades no processo. Os atributos de funções são usados para aplicar valores reais em cálculos de medidas.
Evento	OT_EVT	Eventos são estados de processo e descrevem o estado que aciona uma função e o resultado da execução de uma função. Os fragmentos de processo são mesclados em instâncias de processo usando eventos do mesmo tipo.
Unidade organizacional (opcional)	OT_ORG	Os processadores de uma função podem ser associados a unidades organizacionais se forem tornados anônimos. Os atributos das unidades organizacionais são a base para a contabilidade de custos do processo.
Regra AND (opcional)	OT_RULEAND	Separa ou consolida um fluxo de processo. Os dois caminhos de processo que seguem a regra AND são ambos executados.
Regra OR (opcional)	OT_RULEOR	Separa ou consolida um fluxo de processo. Pelo menos um dos caminhos de processo que seguem a regra OR é executado.
Regra XOR (opcional)	OT_RULEXOR	Separa ou consolida um fluxo de processo. Somente um dos caminhos de processo que seguem a regra OR é executado.

Em um gráfico, um objeto é identificado de forma única pelo seu **node id**. O **node id** encontra-se no atributo XML **id** para o elemento XML **node**. Os objetos com o mesmo **node id** não são permitidos e são combinados em um único objeto.

3.1.2. Conexões

A tabela abaixo mostra todas as conexões permitidas no PPM:

Tipo de conexão	Identificador	Descrição
Conexão de fluxo	CNX_FOLLOWS	Vincula objetos que formam a estrutura do processo (eventos, funções, regras) no gráfico.
Conexão de comentário	CXN_UNDIRECTED	Associa uma unidade organizacional a uma função que ela executa.

3.1.3. Relações (opcional)

A tabela abaixo mostra todos os tipos de relações permitidos no PPM:

Tipo de relação	Identificador	Descrição
executes	REL_CARRY_OUT	Cria uma relação entre uma função e a unidade organizacional em execução.
cooperates with (sem espaços)	REL_WORKS_ TOGETHER	Cria uma relação entre unidades organizacionais como o objeto de referência de origem e as unidades organizacionais como o objeto de referência de destino. Um cálculo de relação é realizado apenas entre instâncias de função que se seguem diretamente e nas quais são especificadas unidades organizacionais. Não deve haver outras instâncias de função sem unidades organizacionais entre elas.
coopera com (com espaços)	REL_WORKS_ TOGETHER_LONG_DISTANCE	Cria uma relação entre unidades organizacionais como o objeto de referência de origem e as unidades organizacionais como o objeto de referência de destino. Entre as instâncias de função nas quais as unidades organizacionais são especificadas podem ocorrer instâncias de função adicionais sem unidades organizacionais. Estas são ignoradas no cálculo da relação, ou seja, as instâncias de função com as unidades organizacionais não precisam se seguir diretamente.
Ping pong	REL_PING_PONG	Cria uma relação entre unidades organizacionais como o objeto de referência de origem e as unidades organizacionais como o objeto de referência de destino.

As ocorrências de relações reais têm os atributos de medidas que são usados como base para cálculo das medidas de relação definidas na configuração de medidas - se necessário, para cada tipo de relação individual.

Consulte a referência técnica **Personalização PPM** para detalhes sobre como as relações são definidas e calculadas.

3.1.4. Diretrizes para a estrutura do gráfico

As convenções gerais da EPC (sequência no fluxo de conexão **Evento - Função - Evento**) resultam nas seguintes diretrizes para criar o gráfico para uma instância de processo:

- Uma instância de processo deve começar e terminar com um ou mais eventos e conter pelo menos uma função.
- Como regras, as instâncias de processo podem conter regras AND exclusivas, uma vez que representam processos de trabalho reais.
- Um evento pode ser seguido por uma regra de ramificação (OR ou XOR).
- Uma função pode ser seguida por outra função. O evento de vinculação entre duas funções pode ser removido.
- Um evento apenas pode ser seguido por uma função ou por uma regra de junção.

3.1.5. Atributos

Os seguintes atributos devem ser especificados para gráficos de uma instância de processo e seus objetos:

Atributos de processo

Nome de atributo	Identificador	Data type	Descrição
ID do processo (opcional)	AT_EPK_KEY	TEXT	Identificador único da instância de processo. O valor do atributo deve corresponder ao ID gráfico (marca **id** do elemento XML **<graph>**). Este atributo identifica uma instância como completa.
Identificação do processo	AT_ID	TEXT	Identificação do processo. O valor do atributo é exibido na lista de instâncias do processo e na visualização EPC da respectiva instância de processo. O valor do atributo **AT_EPK_KEY** é um valor do atributo útil para instâncias completas.
Grupo de tipos de processo	AT_PROCTYPE GROUP	TEXT	Nome do grupo de tipos de processo ao qual a instância de processo pertence. Se o atributo não for especificado, será criado pelo tipificador.
Tipo de processo	AT_PROCTYPE	TEXT	Nome do tipo de processo da instância de processo. Se o atributo não for especificado, será criado pelo tipificador.
Instância	AT_IS_PROC INSTANCE	BOOLEAN	Especifica se é o gráfico para uma instância de processo individual (não especificado ou valor = TRUE) ou uma instância de processo agregada (valor = FALSE).

Atributos de objeto

Nome de atributo	Identificador	Data type	Descrição
Nome	AT_OBJNAME	TEXT	Nome do objeto. É usado para visualização EPC.
Nome interno	AT_OBJNAME_ INTERN	TEXT	Nome interno do objeto, independente do idioma. É usado para referenciar o objeto.
Hora de início	AT_START_ TIME	TIME STAMP	Especifica a hora de início para executar uma função. A especificação do atributo **End time** em uma função é opcional.
Hora de término	AT_END_TIME	TIME STAMP	Especifica a hora de término para executar uma função. A especificação do atributo **Start time** em uma função é opcional.

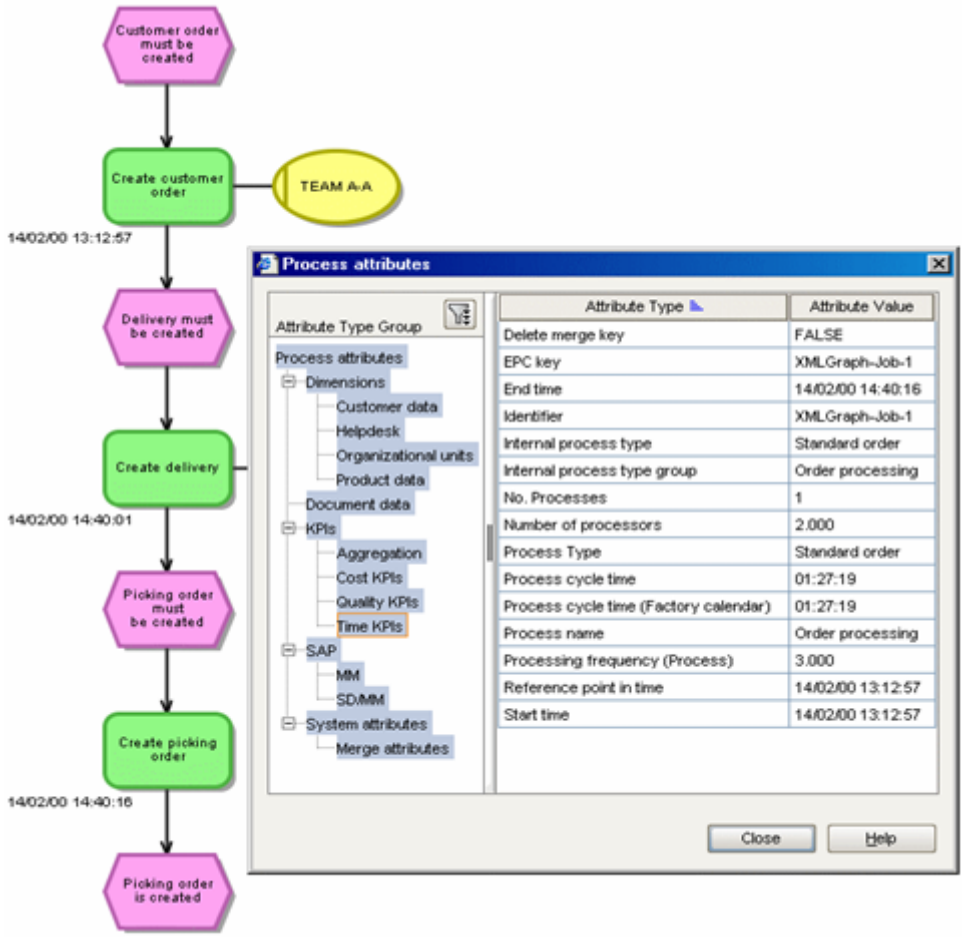
Atributos de objeto opcionais

A tabela a seguir fornece um exemplo de vários atributos de objeto que podem ser usados para calcular medidas.

Nome de atributo	Identificador	Tipo	Descrição
Número de execuções	AT_COUNT_ PROCESSINGS	LONG	Especifica a frequência de execução de uma função.
Padrão de desempenho	AT_LS	TIME SPAN	É usado para calcular a duração média de execução de uma instância.
Usuário em lote	AT_IS_BATCH USER	BOOLEAN	Especifica se uma unidade organizacional é um usuário em lote (programa).

3.1.6. Gráfico XML de exemplo

A ilustração abaixo mostra um gráfico com atributos de instância de processo criados (medidas calculadas e tipo de processo) como uma visualização EPC, com diálogo de atributo de processo ativo:



O arquivo XML associado é semelhante a isso (as linhas que representam objetos têm um fundo colorido, de acordo com os objetos EPC):

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE graphlist SYSTEM "graph.dtd">
<graphlist>
<graph id="XMLGraph-Job-1" xml:lang="en">
<attribute type="AT_EPK_KEY">XMLGraph-Job-1</attribute>
<attribute type="AT_TIME">14.2.2000 13:12:57</attribute>
<attribute type="AT_PROCTYPE">Standard order</attribute>
<attribute type="AT_PROCTYPEGROUP">Order processing</attribute>
<attribute type="AT_ID">XMLGraph-Job-1</attribute>
<node id="XMLGraph-Job-Evt1" type="OT_EVT">
<attribute type="AT_OBJNAME_INTERN">AUFTRAG_ANZU</attribute>
<attribute type="AT_OBJNAME">Customer order to be created</attribute>
</node>
<node id="XMLGraph-Job-Func1" type="OT_FUNC">
<attribute type="AT_TIME">14.2.2000 13:12:57</attribute>
<attribute type="AT_OBJNAME_INTERN">AUFTRAG</attribute>
<attribute type="AT_OBJNAME">Create customer order</attribute>
<attribute type="AT_END_TIME">14.02.2000 01:12:57 PM</attribute>
</node>
<node id="HDMXMLGraph-Job-Func1" type="OT_ORG">
<attribute type="AT_OBJNAME">HDM</attribute>
</node>
```

```
<node id="XMLGraph-Job-Evt2" type="OT_EVT">
<attribute type="AT_OBJNAME_INTERN">LIEFERUNG_ANZU</attribute>
<attribute type="AT_OBJNAME">Delivery to be created</attribute>
<attribute type="AT_ID">XMLGraph-Job-Evt2</attribute>
</node>
<node id="XMLGraph-Job-Func2" type="OT_FUNC">
<attribute type="AT_TIME">14.2.2000 02:40:01 PM</attribute>
<attribute type="AT_OBJNAME_INTERN">LIEFERUNG</attribute>
<attribute type="AT_OBJNAME">Create delivery</attribute>
<attribute type="AT_END_TIME">14.02.2000 02:40:01 PM</attribute>
</node>
<node id="HDMXMLGraph-Job-Func2" type="OT_ORG">
<attribute type="AT_OBJNAME">HDM</attribute>
</node>
<node id="XMLGraph-Job-Evt3" type="OT_EVT">
<attribute type="AT_OBJNAME_INTERN">KOM_AUFTRAG_ANZU</attribute>
<attribute type="AT_OBJNAME">Pick order must be
created</attribute>
<attribute type="AT_ID">XMLGraph-Job-Evt3</attribute>
</node>
<node id="XMLGraph-Job-Func3" type="OT_FUNC">
<attribute type="AT_TIME">14.2.2000 02:40:16 PM</attribute>
<attribute type="AT_OBJNAME_INTERN">KOM_AUFTRAG</attribute>
<attribute type="AT_OBJNAME">Create pick order</attribute>
<attribute type="AT_END_TIME">14.02.2000 02:40:16 PM</attribute>
</node>
<node id="XMLGraph-Job-Evt4" type="OT_EVT">
<attribute type="AT_OBJNAME_INTERN">KOM_AUFTRAG_ALGT</attribute>
<attribute type="AT_OBJNAME">Pick order created</attribute>
</node>
<edge type="CXN_FOLLOWS" source="XMLGraph-Job-Evt1"
target="XMLGraph-Job-Func1"/>
<edge type="CXN_FOLLOWS" source="XMLGraph-Job-Func1"
target="XMLGraph-Job-Evt2" />
<edge type="CXN_UNDIRECTED" source="HDMXMLGraph-Job-Func1"
target="XMLGraph-Job-Func1">
<attribute type="AT_COUNT_PROCESSINGS">1</attribute>
</edge>
<edge type="CXN_FOLLOWS" source="XMLGraph-Job-Func2"
target="XMLGraph-Job-Evt3" />
<edge type="CXN_UNDIRECTED" source="HDMXMLGraph-Job-Func2"
target="XMLGraph-Job-Func2">
<attribute type="AT_COUNT_PROCESSINGS">1</attribute>
</edge>
<edge type="CXN_FOLLOWS" source="XMLGraph-Job-Func3"
target="XMLGraph-Job-Evt4" />
<edge type="CXN_FOLLOWS" source="XMLGraph-Job-Evt2"
target="XMLGraph-Job-Func2" />
<edge type="CXN_FOLLOWS" source="XMLGraph-Job-Evt3"
target="XMLGraph-Job-Func3" />
</graph>
</graphlist>
```

3.2. Formato do evento de sistema

Muitos sistemas de origem registram o processamento de uma operação documentando mudanças de estado ou estados específicos do sistema. Estes são salvos na forma de eventos do sistema (eventos do sistema de origem), com informações suplementares. Um exemplo típico é um sistema para processamento de pedidos, no qual a criação de um novo pedido ou o faturamento de um pedido são salvos. Outro exemplo é o módulo SAP R/3 SD, que documenta os estados e o progresso de um pedido em transações individuais.

A importação de fragmentos de instância de processo no formato de evento do sistema PPM tem três etapas:

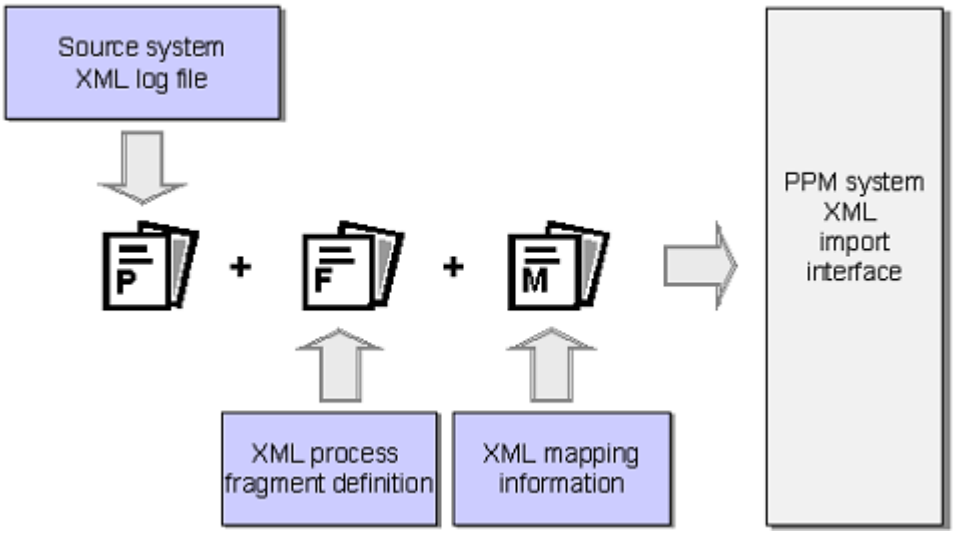
1. Define mapeamento e fragmentos de processos

   Todos os tipos de evento do sistema que ocorrem nos dados do sistema de origem e que devem ser importados para o PPM (por exemplo, Criar pedido, Faturar pedido), devem ser criados em primeiro lugar como definições de fragmento de processo, na forma de uma EPC, no arquivo de fragmento. Além disso, o arquivo de mapeamento deve especificar que tipos de evento do sistema devem ser atribuídos a que definições de fragmento quando importados e que atributos devem ser transferidos para o PPM. No arquivo de mapeamento, deve levar em conta atributos que permitem o cálculo das medidas e a mesclagem de fragmentos de instância de processo em uma instância de processo (por exemplo, carimbo de data/hora, número de evento sequencial do sistema, número de pedido).
2. Gerar um arquivo de saída XML do sistema de origem

   Eventos individuais do sistema com informações adicionais (atributos com dados reais) são extraídos do sistema de aplicativos para um arquivo de saída XML, de acordo com o fluxo de atividade.
3. Gerar os fragmentos de instância de processo

   Ao importar dados, é feita uma pesquisa para a definição de fragmento associada (nível de tipo) para cada evento do sistema no arquivo de saída XML, e isso é então copiado para o banco de dados do PPM. Atributos do evento do sistema com dados reais (por exemplo, hora de execução, processador, número de pedido e de cliente) são transferidos para objetos na cópia da definição de fragmento, gerando um fragmento de instância de processo (nível de instância) no banco de dados do PPM que corresponde a este evento do sistema.

A ilustração abaixo destaca o processo de geração de fragmentos de instância do processo:



3.2.1. Definição de fragmentos de processo

Para poder gerar instâncias de processo para o sistema PPM a partir dos eventos do sistema, cada evento do sistema a ser importado para o PPM deve ser vinculado a uma definição de fragmento de processo. Cada evento de sistema é associado a um tipo de evento do sistema. Tem de ser criada uma definição de fragmento para cada tipo de evento de sistema.

Cada tipo de evento de sistema é associado a um evento final em uma EPC separada. Supõe-se que no processo, uma função deve ter precedido e acionado o evento final. Adicionar um evento inicial antes da função resulta em um fragmento de processo completo, em conformidade com o sistema de modelagem. O evento inicial de um fragmento de processo pode corresponder ao evento final de outro fragmento de processo.

Todas as definições de fragmento de processo são salvas em um arquivo XML como uma lista de gráficos. O arquivo XML de definição de fragmento usa o formato gráfico DTD para a descrição de formato.

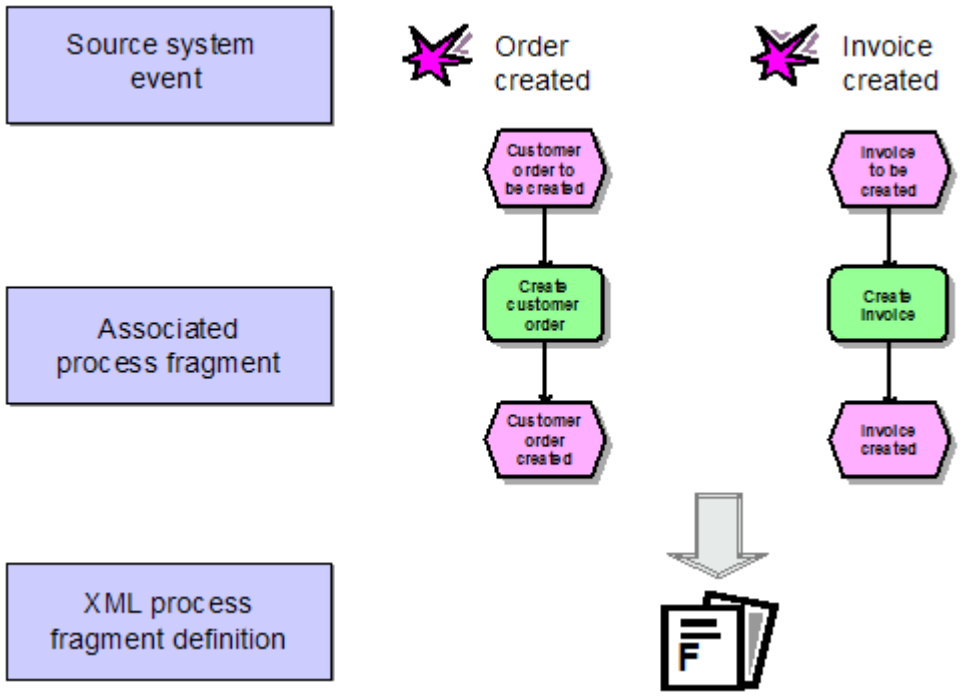
Fragmentos de instância de processo importados são vinculados para formar uma instância de processo por meio dos eventos de início e fim dos fragmentos individuais. Estes eventos são conhecidos como eventos de merge.

Não é absolutamente necessário que o evento de sistema e o evento final do fragmento da instância de processo sejam os mesmos, por exemplo, se o processo para a definição de fragmento terminar com dois eventos finais. O importante é que todas as descrições do sistema contidas no evento de sistema sejam representadas pelo fragmento da instância de processo criado.

Exemplo 1. *[en-us]*

O sistema de origem contém os dois eventos de sistema **Pedido criado** e **Fatura criada**. Cada um destes é transferido para uma EPC como um evento final e interpretado como o resultado da função **Criar pedido do cliente** ou **Criar fatura**. Deve usar todos os conhecimentos do sistema de origem para determinar os eventos que precedem essas duas funções. No exemplo, estes são os eventos **Pedido do cliente a ser criado** e **Fatura a ser criada**.

A ilustração abaixo mostra a descrição do fragmento de processo para os dois eventos do sistema.



O extrato de arquivo abaixo mostra um possível arquivo de definição de fragmento para os fragmentos de processo ilustrados:

```
<?xml version="1.0" encoding='ISO-8859-1'?>
<!DOCTYPE graphlist SYSTEM "graph.dtd">
<graphlist>
<graph id="FRG_ORD_CREATED">
<node id="EVT_ORD_TOBECREATED" type="OT_EVT">
<attribute type="AT_OBJNAME">Customer order to be created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_ORD_TOBECREATED</attribute>
</node>
<node id="FCT_CREATE_ORDER" type="OT_FUNC">
<attribute type="AT_OBJNAME">Create customer order</attribute>
<attribute type="AT_OBJNAME_INTERN">FCT_CREATE_ORDER</attribute>
</node>
<node id="EVT_ORD_CREATED" type="OT_EVT">
<attribute type="AT_OBJNAME">Customer order created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_ORD_CREATED</attribute>
</node>
<edge type="CXN_FOLLOWS" source="EVT_ORD_TOBECREATED"
target="FCT_CREATE_ORDER"/>
<edge type="CXN_FOLLOWS" source="FCT_CREATE_ORDER"

  target="EVT_ORD_CREATED"/>
</graph>
<graph id="FRG_INVOICED">
<node id="EVT_TOBE_INVOICED" type="OT_EVT">
<attribute type="AT_OBJNAME">Invoice to be created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_TOBE_INVOICED</attribute>
</node>
<node id="FCT_INVOICE" type="OT_FUNC">
<attribute type="AT_OBJNAME">Create invoice</attribute>
<attribute type="AT_OBJNAME_INTERN">FCT_INVOICE</attribute>
</node>
<node id="EVT_INVOICED" type="OT_EVT">
<attribute type="AT_OBJNAME">Invoice created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_INVOICED</attribute>
</node>
<edge type="CXN_FOLLOWS" source="EVT_TOBE_INVOICED"
target="FCT_INVOICE"/>
<edge type="CXN_FOLLOWS" source="FCT_INVOICE"
target="EVT_INVOICED"/>
</graph>
</graphlist>
```

Os gráficos de definição de fragmentos não podem conter qualquer atributo especificado. Por padrão, quando os fragmentos da instância de processo são mesclados subsequentemente, somente os atributos do objeto são considerados. Atributos da instância de processo a serem retidos quando a mesclagem pode ser especificada na configuração de merge.

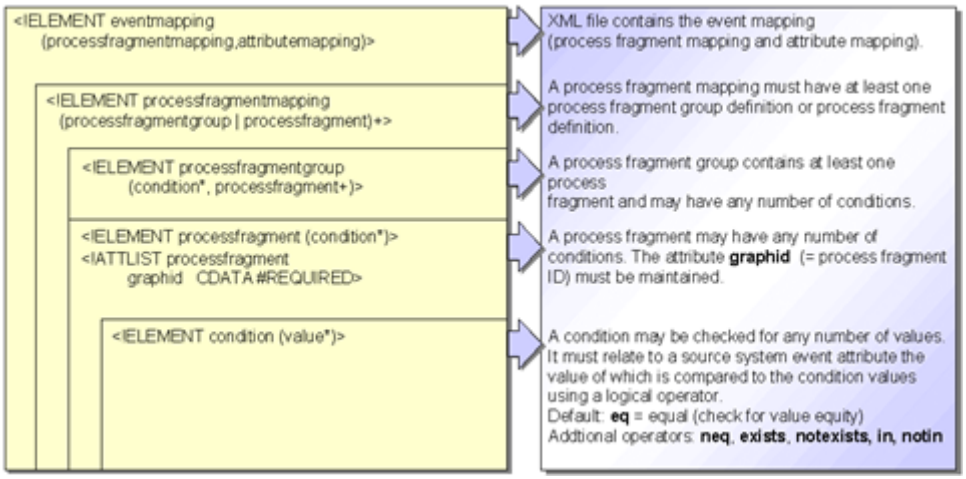
3.2.2. Definição de mapeamento

O arquivo de mapeamento contém a associação dos tipos de evento do sistema para processar definições de fragmentos e determina os atributos dos eventos do sistema, que são copiados para os fragmentos de processo para os quais uma instância é criada no sistema PPM.

3.2.2.1. Definição de mapeamento de fragmentos de processo

O mapeamento de fragmento de processo define quais definições de fragmento de processo são usadas para instanciar os tipos de evento do sistema. Pode ser controlado por qualquer número de condições (elemento XML **condition**) vinculadas entre si por regras AND.

As regras para a estrutura de mapeamento de fragmento de processo no arquivo de mapeamento XML são especificadas no seguinte extrato do arquivo **eventmapping.dtd**:




```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE eventmapping SYSTEM "eventmapping.dtd">
<eventmapping>
<processfragmentmapping>
<processfragment graphid="FRG_ORD_CREATED">
<condition eventattributetype="AUFTR_TYP">
<value>C</value>
</condition>
<condition eventattributetype="MAT_NR" logicaloperator="in">
<value>123456</value>
<value>56789</value>
<value>78901</value>
</condition>
</processfragment>
...
</processfragmentmapping>
<attributemapping>
...
</attributemapping>
</eventmapping>
```

O fragmento de processo **FRG_ORD_CREATED** é criado se as duas condições seguintes forem atendidas:

- O evento de sistema relevante representa um documento de pedido (o atributo **AUFTR_TYP** tem o valor **C**).
- O valor do atributo de evento de sistema **MAT_NR** corresponde a um dos números do material especificado.

Uma definição de fragmento de processo não pode conter uma condição. Nesse caso, a mesma definição de fragmento especificada é usada para cada evento de sistema importado. Os objetos no fragmento instanciado são especificados no mapeamento de atributo subsequente.

Definição de um grupo de fragmentos de processo

As definições de fragmento de processo podem ser resumidas em grupos. Isso resulta nas vantagens seguintes:

- Aumento de desempenho
- Criação simplificada de definições de mapeamento do fragmento de processo
- Maior clareza

As duas definições de fragmentos de processo abaixo estão contidas em um arquivo de mapeamento XML:

```
<processfragment graphid="AUFTRAG_ANLEGEN">
<condition eventattributetype="AUFTR_TYP">
<value>C</value>
</condition>
<condition eventattributetype="CHARGEN_PFL" logicaloperator="neq">
<value>X</value>
</condition>
</processfragment>
<processfragment graphid="CHPLICHT_AUFTRAG_ANLEGEN">
<condition eventattributetype="AUFTR_TYP">
<value>C</value>
</condition>
<condition eventattributetype="CHARGEN_PFL">
<value>X</value>
</condition>
</processfragment>
```

Exemplo 2.

O primeiro fragmento de processo **AUFTRAG_ANLEGEN** é criado para um evento de sistema do tipo **Documento de pedido** (o valor do atributo de **AUFTR_TYP** é **C**), que não está sujeito a gestão em lotes (o valor do atributo de **CHARGEN_PFL** não é igual a **X**).

O segundo fragmento de processo **CHPFLICH_AUFTRAG_ANLEGEN** é criado para um evento de sistema do tipo **Documento de pedido sujeito a gestão em lotes** (valor do atributo de **AUFTR_TYP** é **C** e o valor do atributo de **CHARGEN_PFL** é **X**).

Os dois fragmentos de processo exibidos podem ser resumidos em um grupo de fragmentos de processo:

```
<processfragmentgroup>
<condition eventattributetype="AUFTR_TYP">
```

```
<value>C</value>
</condition>

<processfragment graphid="AUFTRAG_ANLEGEN">
<condition eventattributetype="CHARGEN_PFL" logicaloperator="neq">
<value>X</value>
</condition>
</processfragment>

<processfragment graphid="CHPLICHT_AUFTRAG_ANLEGEN">
<condition eventattributetype="CHARGEN_PFL">
<value>X</value>
</condition>
</processfragment>

</processfragmentgroup>
```

Resumo em um grupo de fragmentos de processo significa que apenas uma verificação é feita se o atributo do sistema de origem **AUFTR_TYP** tem o valor **C** na altura da importação. Se não for esse o caso, nenhum dos dois fragmentos de processo no grupo de fragmentos de processo é instanciado.

Use grupos de fragmentos de processo para melhorar a clareza das definições de mapeamento do fragmento de processo e o desempenho do processo de importação.

Suprimir emissão de avisos

Se não quiser importar determinados eventos do sistema para a personalização e se não tiver definido nenhum mapeamento de processo para esses eventos do sistema, poderá suprimir a mensagem de erro esperada ao importar. Para isso, terá de especificar condições no elemento XML **ignoreevent** que suprimem a saída de uma mensagem de erro relativa a fragmentos específicos que não podem ser importados.

Para eventos de sistema especificados com mapeamento **ignoreevent**, a saída da mensagem de erro apenas é suprimida se o evento do sistema não puder ser importado. Isso significa que quando importa eventos do sistema com mapeamento de fragmento de processo e mapeamento **ignoreevent**, estes eventos do sistema são importados.

Exemplo 3.

Quer importar fragmentos de processo se o atributo de evento do sistema **EKKO_BSTYP** existir e o atributo de evento do sistema **MSEG_SHKZG** tiver o valor **S** para "Registrar entrada de mercadorias" ou **H** para "Cancelar entrada de mercadorias". Outros valores do atributo **MSEG_SHKZG** resultam na emissão de um aviso.

Se o atributo de evento do sistema **EKKO_BSTYP** não existir, nenhum fragmento de processo é importado e nenhum aviso é emitido.

O mapeamento de processo seguinte atende os requisitos acima:

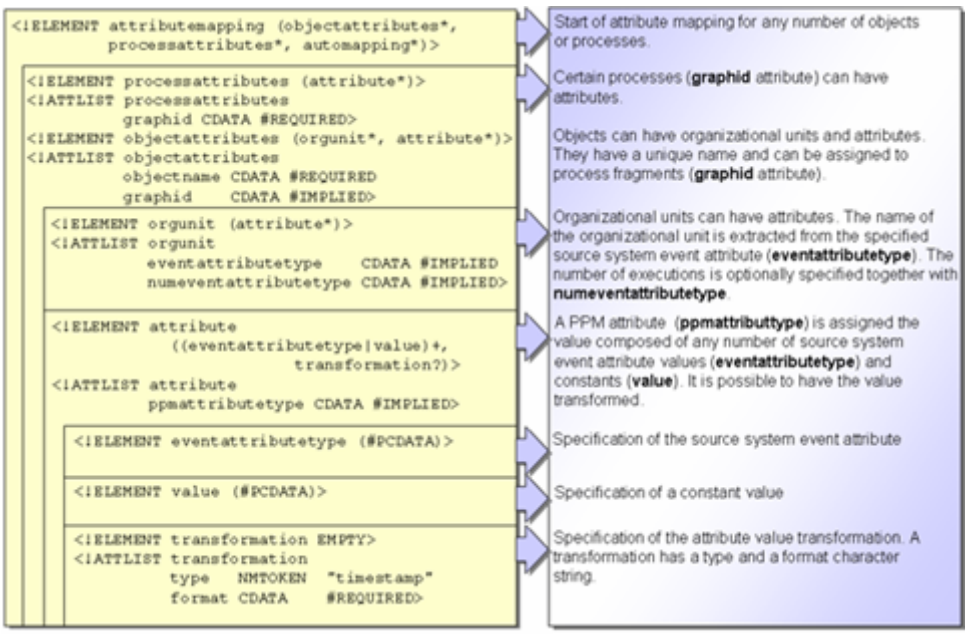
```
...
<processfragmentgroup>
<!-- Import goods receipts with MM predecessor documents only
-->
```

```
<condition eventattributetype="EKKO-BSTYP" logicaloperator="exists"/>
<processfragment graphid="GWEOF">
<!-- Post goods receipt
-->
<condition eventattributetype="MSEG-SHKZG" logicaloperator="eq">
<value>S</value>
</condition>
</processfragment>

<processfragment graphid="GWSOF">
<!-- Cancel goods receipt
-->
<condition eventattributetype="MSEG-SHKZG" logicaloperator="eq">
<value>H</value>
</condition>
</processfragment>
</processfragmentgroup>

<ignoreevent>
<!-- Do not output warning if no MM predecessor document exists
because these system events are not to be imported
-->
<condition eventattributetype="EKKO-BSTYP" logicaloperator="notexists"/>
</ignoreevent>
...
```

3.2.2.2. Definição do mapeamento de atributo

Este capítulo descreve a configuração do mapeamento de atributo. O mapeamento de atributo copia os atributos do sistema de origem para os atributos de objeto e processo da instância de fragmento (atributos do PPM).

As regras para a estrutura de mapeamento de atributo no arquivo de mapeamento XML são especificadas no seguinte extrato do arquivo **eventmapping.dtd**:



Os valores de atributo do PPM podem ser compostos por qualquer combinação de valores de atributo do evento de sistema e textos inalteráveis. A especificação opcional do nome de atributo interno do PPM permite que os atributos do sistema de origem sejam copiados para quaisquer atributos do PPM. Podem ser associados valores constantes aos atributos do PPM (elemento XML **value**).

```
...
<attribute>

  <eventattributetype>MATERIAL_CLASS</eventattributetype>

  </attribute>
...
```

Exemplo 4.

O valor do atributo de evento do sistema **MATERIAL_CLASS** é copiado para o atributo do PPM com o mesmo nome, mas incluindo o prefixo (por exemplo, **AT_**) especificado no atributo XML **attributeprefix** da fonte de dados, ou seja,**AT_MATERIAL_CLASS**.

```
...
<attribute ppmattributetype="AT_MATERIAL">

  <eventattributetype>MAT_NR</eventattributetype>

  </attribute>
...
```

Exemplo 5.

O valor do atributo de evento de sistema **MAT_NR** é copiado para o atributo do PPM **AT_MATERIAL**.

```
...
<attribute ppmattributetype="AT_IS_SHARED_FUNCTION">
```

```
  <value>TRUE</value>

  </attribute>
...
```

Exemplo 6.

O atributo do PPM **AT_IS_SHARED_FUNCTION** é associado à constante **TRUE**.

```
...
<attribute ppmattributetype="AT_ID">

  <eventattributetype>AUFTRAGS_SYSTEM</eventattributetype>

  <value>-</value>

  <eventattributetype>SYSTEM_NR</eventattributetype>

  <value>#</value>

  <eventattributetype>AUFTRAGS_NR</eventattributetype>

  <value>-</value>

  <eventattributetype>POSITIONS_NR</eventattributetype>

  <value>-</value>

  <eventattributetype>AUFTRAGS_TYP</eventattributetype>

  </attribute>
...
```

Exemplo 7.

O atributo do PPM **AT_ID** é associado ao valor **XYZ-401#4711-10-C**.

> **Atenção**
> Se um atributo de sistema de origem especificado não estiver disponível ao importar dados do arquivo de saída XML, o atributo do PPM correspondente não será criado.

Associar atributos a objetos

Os atributos do PPM criados são associados a objetos específicos na instância de fragmento usando o elemento XML **objectattributes**. O atributo XML **objname** especifica o identificador do objeto relevante (atributo de objeto **AT_OBJNAME_INTERN**). Opcionalmente, a especificação do objeto pode ser refinada especificando o ID gráfico (atributo XML **graphid**).

Exemplo 8.

O exemplo abaixo copia o valor do atributo do sistema de origem **END_TIME** para o atributo de objeto do PPM **AT_END_TIME**, para a função com o identificador **SAP.AUFT_ANLEG** na instância de fragmento **AUFTRAG_ANLEGEN** (ID gráfico da definição de fragmento):

```
...
<attributemapping>
<objectattributes objectname="SAP.AUFT_ANLEG" graphid="AUFTRAG_ANLEGEN">
<attribute ppmattributetype="AT_END_TIME">
<eventattributetype>END_TIME</eventattributetype>
</attribute>
</objectattributes>
</attributemapping>
...
```

Atribuir atributos a processos

Os atributos do PPM criados são associados a instâncias de fragmento específicas usando o elemento XML **processattributes**. É obrigatório especificar a ID do gráfico definição de fragmento no atributo XML **graphid**.

Para garantir que os atributos transferidos diretamente para o processo sejam retidos ao mesclar as instâncias de fragmento, terá de expandir a configuração de merge em conformidade. O esforço de cálculo necessário para comparar os atributos de processo resulta em uma pequena perda de desempenho.

O exemplo abaixo copia o valor do atributo do sistema de origem **PROCESSNAME** para o atributo de processo do PPM **AT_PROCTYPE**, para a instância de fragmento **AUFTRAG_ANLEGEN** (ID gráfico da definição de fragmento):

```
...
<attributemapping>
<processattributes graphid="AUFTRAG_ANLEGEN">
<attribute ppmattributetype="AT_PROCTYPE">
<eventattributetype>PROCESSNAME</eventattributetype>
</attribute>
</processattributes>
</attributemapping>
...
```

3.2.2.2.1. Transformações de atributos

Se o formato de um valor do atributo para um evento de sistema não corresponder ao formato requerido pelo sistema PPM, o valor gravado no atributo PPM terá de ser transformado.

3.2.2.2.2. Transformações do carimbo de data/hora

O PPM fornece várias transformações de atributos do tipo Transformação de carimbo de data/hora.

As transformações do carimbo de data/hora disponíveis no sistema PPM convertem valores do carimbo de data/hora do sistema de origem, em qualquer formato, para o formato interno válido do PPM, com o tipo de dados de destino adequado para o PPM. A tabela seguinte fornece uma visão geral das transformações do carimbo de data/hora disponíveis:

Transformação	Tipo de dados de destino do PPM
timestamp	TIME
timestamp_epoch	TIME
timeofday	TIMEOFDAY
timeofday_epoch	TIMEOFDAY
day	DAY
day_epoch	DAY
SAGDateTime	TIME

Transformações do carimbo de data/hora timestamp, timeofday, day

Os padrões para a configuração das transformações do carimbo de data/hora são especificados na definição do tipo de documento **eventmapping.dtd**:

```
...
<!ELEMENT transformation EMPTY>
<!ATTLIST transformation
type NMTOKEN "timestamp"
format CDATA #REQUIRED
>
...
```

No atributo XML **type** pode especificar quais das transformações do carimbo de data/hora disponíveis devem ser usadas. A transformação de atributos **timestamp** é padrão. O atributo XML **format** especifica o formato do carimbo de data/hora no atributo do sistema de origem.

Os dados no atributo **format** correspondem ao formato do carimbo de data/hora Java:

Símbolo	Descrição	Data type	Exemplo
G	Identificador Epoch	Text	AD
y	Ano	Figura	2003
MM	Mês do calendário	Figura	09
MMM	Mês do calendário	Text	Set
MMMM	Mês do calendário	Text	Setembro
d	Dia do calendário	Valor numérico	26
h	Hora, notação americana (1-12)	Valor numérico	12
H	Hora (0-23)	Valor numérico	14
m	Minuto	Valor numérico	42
s	Segundo	Valor numérico	57
S	Milissegundo	Valor numérico	978
E	Dia de semana	Text	Quinta-feira
F	Recorrência de weekday no mês	Valor numérico	2
D	Dia do ano	Valor numérico	189
w	Semana do calendário	Valor numérico	27
W	Semana do mês	Figura	2 (2. semana do mês)
a	Identificador da hora do dia	Text	PM

Símbolo	Descrição	Data type	Exemplo
k	Hora (1-24)	Figura	24
K	Hora, notação americana (0-11)	Figura	7
z	Fuso horário	Text	GMT
´	Caractere de escape para texto	Caracteres	'Example text'
´´	Apóstrofo simples	Caracteres	'Example' 'Text'

O valor **2002-12-24** (<Year>-<Month>-<Day>) para um atributo do sistema de origem é transformado no formato PPM usando a cadeia de caracteres de formato aaaa-MM-dd:

```
...
<transformation type="timestamp" format="yyyy-MM-dd"/>
...
```

Exemplo 9.

Valores de atributo do sistema de origem consolidado:

```
...
<attribute ppmattributetype="AT_END_TIME">
<eventattributetype>ERF_DAT</eventattributetype>
<eventattributetype>ERF_ZEIT</eventattributetype>
<transformation format="yyyyMMddHHmmss"/>
</attribute>
...
```

Exemplo 10.

No atributo do sistema de origem **ERF_DAT**, a data de criação está presente no formato **aaaaMMdd** e no atributo do sistema de origem **ERF_ZEIT**, a hora de criação está presente no formato **HHmmss**. Extrato do arquivo de saída XML:

```
<event>
...
<attribute type='ERF_DAT'>20011230</attribute>
...
</event>
<event>
...
<attribute type='ERF_ZEIT'>120730</attribute>
<attribute type='ERF_DAT'>20011101</attribute>
...
</event>
```

Exemplo 11.

No primeiro evento de sistema, o atributo **AT_END_TIME** não é criado porque o atributo **ERF_ZEIT** não está presente.

No segundo evento de sistema, ambos os atributos estão presentes. Os valores de atributo são unidos na sequência especificada para obter **20011101120730** e em seguida avaliados usando o formato

especificado **aaaaMMddHHmmss** e transformados no carimbo de data/hora compatível com o PPM **01.11.2001 12:07:30**.

No atributo **ERF_STD** apenas é registrada a hora na qual o evento de sistema foi criado. A hora de criação é sempre trinta minutos depois da hora.

```
...
<attribute ppmattributetype="AT_END_TIME">
<eventattributetype>ERF_DAT</eventattributetype>
<value>::</value>
<eventattributetype>ERF_STD</eventattributetype>
<value>30</value>
<transformation format="yyyyMMdd::HHmm"/>
</attribute>
...
```

Exemplo 12.

Evento de sistema associado do arquivo de saída:

```
...
<event>
...
<attribute type='ERF_STD'>12</attribute>
<attribute type='ERF_DAT'>20011001</attribute>
...
</event>
...
```

Exemplo 13.

Os valores de atributo e as cadeias de caracteres de constantes são combinados na cadeia de caracteres **20011001::1230** e transformados usando o formato **aaaaMMdd::hhmm** para obter o carimbo de data/hora compatível com o PPM **01.10.2001 12:30:00**, o qual é gravado no atributo do PPM **AT_END_TIME**.

Transformações do carimbo de data/hora timestamp_epoch, timeofday_epoch e day_epoch

A transformação do carimbo de data/hora **timestamp_epoch** transforma um valor inteiro indicando os segundos ou milissegundos que passaram desde janeiro 1 1970 de no formato interno do PPM. No atributo XML **format**, é especificado se o valor inteiro indica o número de segundos (**SECOND**) ou milissegundos (**MILLISECOND**) que passaram desde 01.01.1970 0:00:00GMT. O fuso horário atual do sistema é considerado no cálculo.

Use as transformações de atributos **timeofday_epoch** e **day_epoch** da mesma forma.

Exemplo 14.

No atributo do sistema de origem **WORK_ITEM-END_TIME**, é indicado o número de segundos desde 1 de Janeiro de 1970.

```
...
<attribute type="WORK_ITEM-END_TIME">1221482578</attribute>
...
```

Exemplo 15.

Use a seguinte regra de mapeamento para associar este valor do atributo ao valor do atributo **AT_END_TIME** do PPM:

```
...
<attribute ppmattributetype="AT_END_TIME">
<eventattributetype>

  WORK_ITEM-END_TIME
</eventattributetype>
<transformation type="timestamp_epoch"
format="SECOND"/>
</attribute>
...
```

3.2.2.2.3. Transformação do número de ponto flutuante

Para o tipo de dados do PPM **DOUBLE**, o PPM espera valores sem separador de milhares e com ponto como separador decimal. Se o valor do atributo do sistema de origem não corresponder a esse formato, ele precisará ser transformado em conformidade.

Ao importar números de ponto flutuante, pode especificar no atributo **format** os separadores decimal e de milhares a serem usados para analisar um valor duplo se estiver a usar a transformação de atributos **double**.

<transformation type="double" format=","/>

<transformation type="double" format="."/>

Se apenas um caractere for especificado, terá de ser o separador decimal. Se dois caracteres forem especificados, o primeiro deve ser o separador de milhares e o segundo o separador decimal.

Exemplo 16.

Valores de atributo com separador de milhares (,) e separador decimal (.)

1.000,00

2.324.213,42

Definição de elementos de atributos

<attribute ppmattributetype="AT_END_TIME">

<eventattributetype>TIMESTAMP_FIELD</eventattributetype>

<transformation type="double" format="."/>

</attribute>

3.2.2.3. Unidades organizacionais

As unidades organizacionais são criadas dinamicamente para funções na instância de fragmento.

Definir uma unidade organizacional

O elemento XML **orgunit** no arquivo de mapeamento define uma unidade organizacional. O valor do atributo XML **eventattributetype** especifica o nome do atributo do sistema de origem, o qual contém o nome da unidade organizacional. Este nome é associado aos atributos de objeto **AT_OBJNAME** e **AT_OBJNAME_INTERN** para a unidade organizacional.

A extração seguinte do arquivo de mapeamento cria duas unidades organizacionais, cujos nomes são extraídos dos atributos do sistema de origem **PROCESSOR_1** e **PROCESSOR_2**. O número de execuções das funções é lido dos atributos do sistema de origem **NUM_OF_PROCESSINGS_1** e **NUM_OF_PROCESSINGS_2** especificados no atributo XML **numeventattributtype**.

```
...
<attributmapping>
...
<orgunit eventattributetype="PROCESSOR_1"
numeventattributetype="NUM_OF_PROCESSINGS_1"/>
<orgunit eventattributetype="PROCESSOR_2"
numeventattributetype="NUM_OF_PROCESSINGS_2"/>
...
</attributmapping>
...
```

O mapeamento de atributo adequado permite que atributos adicionais sejam criados para unidades organizacionais ou que atributos existentes sejam substituídos.

O extrato de arquivo abaixo associa o valor **anonymous processor** ao nome da unidade organizacional exibida na interface do usuário do PPM substituindo o atributo de objeto **AT_OBJNAME**, independentemente do valor do atributo do sistema de origem.

```
...
<attributmapping>
...
<orgunit eventattributetype="PROCESSOR_1">
<attribute ppmattributetype="AT_OBJNAME">
<value>anonymous processor</value>
</attribute>
</orgunit>
...
</attributmapping>
...
```

Associar unidades organizacionais

As unidades organizacionais são associadas a funções específicas na instância de fragmento usando o elemento XML **objectattributes**. O atributo XML **objectname** especifica o identificador da função relevante (atributo de objeto **AT_OBJNAME_INTERN**). Opcionalmente, a especificação do objeto pode ser refinada especificando o ID gráfico (atributo XML **graphid**).

>O exemplo abaixo cria uma unidade organizacional para a função com o identificador **SAP.AUFT_ANLEG** da instância de fragmento **AUFTRAG_ANLEGEN** (ID gráfico da definição de fragmento), do atributo do sistema de origem **VBAP-ERNAM**:

```
<attributemapping>
<objectattributes objectname="SAP.AUFT_ANLEG"
```

```
graphid="AUFTRAG_ANLEGEN">
<orgunit eventattributetype="VBAP-ERNAM"/>
</objectattributes>
</attributemapping>
```

Unidades organizacionais e agregação

Durante o processo de agregação temporária, as unidades organizacionais criadas durante a anonimização são retidas, enquanto os usuários são excluídos.

Para transferir um usuário real sem anonimização, terá de criar o atributo **AT_ISUSERGROUP** com o valor **TRUE** no mapeamento de atributo para a unidade organizacional correspondente.

No exemplo abaixo, uma unidade organizacional cujo conteúdo é extraído do atributo do sistema de origem **UNIT_GROUP_LABEL** é criada para todas as funções, com o nome de objeto interno **FCT_A**, no fragmento da instância de processo com o ID gráfico **FRG_B**.

```
<eventmapping>
<processfragmentmapping>
...
</processfragmentmapping>
<attributemapping>
...
<objectattributes objectname="FCT_A" graphid="FRG_B">
<orgunit eventattributetype="UNIT_GROUP_LABEL">
<attribute ppmattributetype="AT_ISUSERGROUP">
<value>TRUE</value>
</attribute>
</orgunit>
...
</objectattributes>
...
</attributemapping>
</eventmapping>
```

3.2.2.4. Caso especial de mapeamento de atributo

Se vários objetos idênticos existirem em um gráfico de definição de fragmento (valor do atributo idêntico para **AT_OBJNAME_INTERN**), terá de especificar um atributo de objeto único **AT_NODE_ID** na definição de fragmento para fazer a diferenciação entre os objetos. No arquivo de mapeamento de atributo, em vez do valor do atributo de objeto **AT_OBJNAME_INTERN**, deverá especificar o valor do atributo de objeto **AT_NODE_ID**.

Exemplo 17.

Definição de fragmento:

A definição de fragmento contém dois eventos finais **O pedido é criado**.

```
...
<graph id="CHPFLICHT_AUFTRAG_ANLEGEN">
<node id="4" type="OT_EVT">
```

```
<attribute type="AT_OBJNAME">Order to be created</attribute>
<attribute type="AT_OBJNAME_INTERN">SAP.AUFT_ANZU</attribute>
</node>
<node id="5" type="OT_FUNC">
<attribute type="AT_OBJNAME">Create order</attribute>
<attribute type="AT_OBJNAME_INTERN">SAP.AUFT_ANLEG</attribute>
</node>
<node id="42" type="OT_RULEOR"/>
<node id="6" type="OT_EVT">
<attribute type="AT_OBJNAME">Order is created</attribute>
<attribute type="AT_OBJNAME_INTERN">SAP.AUFT_ANGELEGT</attribute>
<attribute type="AT_NODE_ID">End event 1</attribute>
</node>
<node id="7" type="OT_EVT">
<attribute type="AT_OBJNAME">Order is created</attribute>
<attribute type="AT_OBJNAME_INTERN">SAP.AUFT_ANGELEGT</attribute>
<attribute type="AT_NODE_ID">End event 2</attribute>
</node>
<edge type="CXN_FOLLOWS" source="4" target="5"/>
<edge type="CXN_FOLLOWS" source="5" target="42"/>
<edge type="CXN_FOLLOWS" source="42" target="6"/>
<edge type="CXN_FOLLOWS" source="42" target="7"/>
</graph>
...
```

Exemplo 18.

Mapeamento de atributo:

Nos dois eventos finais **O pedido é criado**, o mapeamento de atributo gera um atributo do PPM **AT_ID** com um valor diferente:

```
...
<attributemapping>
<objectattributes objectname="End event 1" graphid="AUFTRAG_ANLEGEN">
<attribute ppmattributetype="AT_ID">
<value>This is the first end event</value>
</attribute>
</objectattributes>
<objectattributes objectname="End event 2" graphid="AUFTRAG_ANLEGEN">
<attribute ppmattributetype="AT_ID">
<value>This is the second end event</value>
</attribute>
</objectattributes>
</attributemapping>
...
```

3.2.3. Criar definições de fragmento em ARIS

Para importar corretamente os dados, não importa como o arquivo de definição de fragmento e o arquivo de mapeamento foram criados. Criar o arquivo de definição de fragmento com ARIS ou o arquivo de mapeamento com PPM Customizing Toolkit é mais fácil do que editar diretamente os arquivos XML.

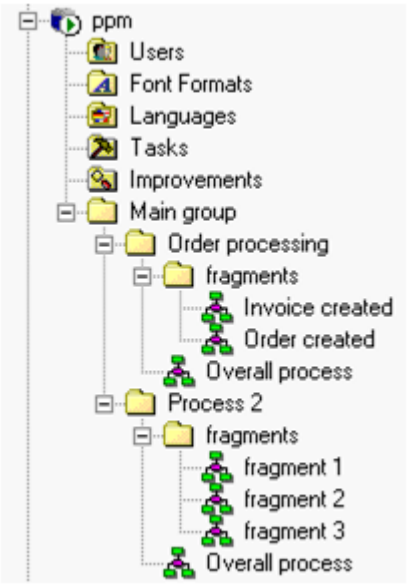
As definições de fragmento de processo são modeladas como EPCs. Um filtro metodológico especial em ARIS está disponível para modelagem. Muitas vezes, o processo geral a ser analisado pelo PPM já é modelado usando o ARIS e pode ser usado como base para a criação dos fragmentos.

As EPCs para as definições de fragmento de processo também podem ser modeladas diretamente. No entanto, na prática, a criação de um processo geral revelou-se mais eficaz por razões de clareza.

3.2.3.1. Modelar o processo geral

Em primeiro lugar, todo o processo a ser monitorado pelo PPM é modelado como uma EPC no sistema de aplicativos usando o fluxo de atividade real. Ao criar o processo, certifique-se de que todos os eventos de sistema que estão ocorrendo são modelados na EPC como eventos de uma função anterior. As unidades organizacionais devem ser especificadas nas funções.

Para cada processo global, crie um grupo separado com um subgrupo para as definições de fragmento de processo. Modele uma EPC para o processo global e uma EPC adicional para cada fragmento no subgrupo.



Diretrizes de modelagem para o processo geral

- Um processo começa com um evento e termina com pelo menos um evento.
- Os eventos não podem preceder regras de ramificação OR e XOR.
- Especifique os atributos de ARIS **Name** e **Identifier** para todos os eventos e funções. Quando o relatório ARIS é executado, o nome é transferido para a definição de fragmento como o atributo do PPM específico do idioma **AT_OBJNAME**, e o identificador como o atributo do PPM independente do idioma **AT_OBJNAME_INTERN**.

3.2.3.2. Modelar as definições de fragmento de processo

Em primeiro lugar, gere uma associação de modelo para todas as funções do processo geral no grupo, a qual deve conter a definição de fragmento. O identificador do modelo associado (definição do fragmento de processo) especifica o nome da definição de fragmento. Em seguida copie todos os eventos, que estão junto a uma função na lógica do decurso, para o modelo que é associado à função.

Respeite as diretrizes a seguir ao modelar os fragmentos, independentemente de estar usando um processo geral ou criando os fragmentos sem um processo geral.

Linhas diretrizes de modelagem para definição de fragmento

Use o filtro metodológico especial do PPM **FragmentXML_ARISToolSet_Filter.amc**, no diretório **<PPM installation directory>\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-<version>-runnable.zip\ppm\ctk\ARIS**. Isso limita o número de elementos de modelagem ao objeto permitido e aos tipos de conexão, facilitando assim a modelagem.

- Cada fragmento de processo deve ser modelado em um modelo separado do tipo **EPC**. O identificador do modelo é transferido para o arquivo de definição de fragmento XML como **graph id** e especifica o nome do fragmento de processo.

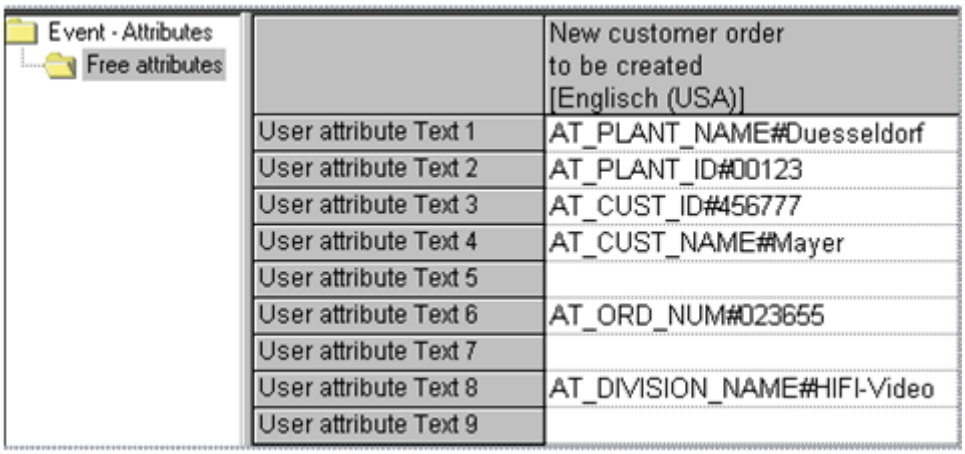
Se você estiver usando um processo geral, ao modelar os fragmentos, irá gerar cópias de ocorrência dos objetos no processo geral.

- Evite caminhos de processos cíclicos (loops). Uma definição de fragmento que contém ciclos é identificada e ignorada pelo relatório de modelo e uma mensagem correspondente é exibida na janela de saída do ARIS.
- Todos os objetos (eventos, funções, regras) do modelo de fragmento devem ter um identificador único. Este é transferido para o PPM como o atributo independente do idioma **AT_OBJNAME_INTERN**.
- As unidades organizacionais não precisam de ser modeladas. Elas são ignoradas ao gerar o arquivo de definição de fragmento usando o ARIS Report. As unidades organizacionais são dinamicamente associadas às funções correspondentes pelo mapeamento de atributo (elemento XML **orgunit** - ver Unidades organizacionais [27]).

Também pode especificar **atributos livres** para todos os objetos e conexões. Eles são gravados nos objetos ou conexões correspondentes no arquivo de definição de fragmento como combinações de atributos/valores fixos do PPM, do tipo **TEXT**. A lista deve ser especificada em ordem crescente, começando com **User attribute text 1** e é avaliada até o primeiro atributo não especificado. Formato: **<Chave de atributo>#<Valor de atributo>**. O separador **#** não deve aparecer na chave ou no valor do atributo.

Exemplo 19.

Para o evento **Novo pedido do cliente a ser criado**, são especificados os seguintes atributos livres:



	New customer order to be created [Englisch (USA)]
User attribute Text 1	AT_PLANT_NAME#Duesseldorf
User attribute Text 2	AT_PLANT_ID#00123
User attribute Text 3	AT_CUST_ID#456777
User attribute Text 4	AT_CUST_NAME#Mayer
User attribute Text 5	
User attribute Text 6	AT_ORD_NUM#023655
User attribute Text 7	
User attribute Text 8	AT_DIVISION_NAME#HIFI-Video
User attribute Text 9	

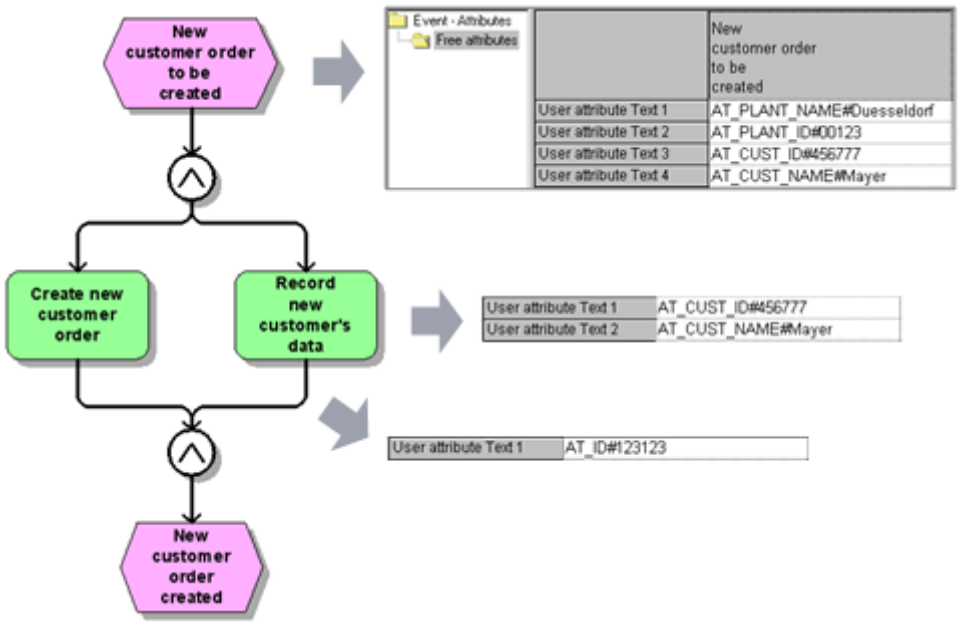
No extrato de arquivo abaixo do arquivo de definição de fragmento criado, os atributos livres transferidos são exibidos em **negrito**.

```
...
<node id="EVT_NEWCUSTORD2BE_CREATED" type="OT_EVT">
<attribute type="AT_OBJNAME">New customer order to be created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_NEWCUSTORD2BE_CREATED</attribute>
<attribute type="AT_PLANT_NAME">Duesseldorf</attribute>
<attribute type="AT_PLANT_ID">00123</attribute>
<attribute type="AT_CUST_ID">456777</attribute>
<attribute type="AT_CUST_NAME">Mayer</attribute>
</node>
...
```

- Os seguintes tipos de conexão ARIS são incluídos na avaliação de relatório e transferidos para o tipo de conexão do PPM **CXN_FOLLOWS**:

Objeto de origem	Objeto de destino	Tipo de conexão ARIS
Evento	Função	**ativa**
Evento	Regra	**é avaliado por**
Função	Evento	**cria**
Função	Regra	**conduz para**
Função	Função	**é predecessor de**
Regra	Evento	**conduz para**
Regra	Função	ativa
Regra	Regra	vinculado

A ilustração abaixo mostra um fragmento de processo corretamente modelado com atributos livres adicionais especificados para objetos e uma conexão:



O **Relatório ARIS** gera o seguinte arquivo de definição de fragmento:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE graphlist SYSTEM "graph.dtd">
<graphlist>
<graph id="FRG_NEWCUST_ORD">
<node id="EVT_NEWCUSTORD2BE_CREATED" type="OT_EVT">
<attribute type="AT_OBJNAME">New customer order to be created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_NEWCUSTORD2BE_CREATED</attribute>
<attribute type="AT_PLANT_NAME">Duesseldorf</attribute>
<attribute type="AT_PLANT_ID">00123</attribute>
<attribute type="AT_CUST_ID">456777</attribute>
<attribute type="AT_CUST_NAME">Mayer</attribute>
</node>
<node id="FCT_CREATE_NEWCUSTORD" type="OT_FUNC">
<attribute type="AT_OBJNAME">Create new customer order</attribute>
<attribute type="AT_OBJNAME_INTERN">FCT_CREATE_NEWCUSTORD</attribute>
</node>
<node id="OT_RULE_AND" type="OT_RULEAND">
<attribute type="AT_OBJNAME_INTERN">OT_RULE_AND</attribute>
<attribute type="AT_SAP_ID">455455</attribute>
</node>
<node id="FCT_REC_NEWCUSTDAT" type="OT_FUNC">
<attribute type="AT_OBJNAME">Enter new customer data</attribute>
<attribute type="AT_OBJNAME_INTERN">FCT_REC_NEWCUSTDAT</attribute>
<attribute type="AT_CUST_ID">456777</attribute>
<attribute type="AT_CUST_NAME">Mayer</attribute>
</node>
<node id="OT_RULE_AND" type="OT_RULEAND">
<attribute type="AT_OBJNAME_INTERN">OT_RULE_AND</attribute>
</node>
<node id="EVT_NEWCUSTORD_CREATED" type="OT_EVT">
<attribute type="AT_OBJNAME">New customer order created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_NEWCUSTORD_CREATED</attribute>
</node>
```

```
<edge type="CXN_FOLLOWS" source="FCT_REC_NEWCUSTDAT" target="OT_RULE_AND">
<attribute type="AT_ID">123123</attribute>
</edge>
<edge type="CXN_FOLLOWS" source="EVT_NEWCUSTORD2BE_CREATED"
target="OT_RULE_AND"/>
<edge type="CXN_FOLLOWS" source="OT_RULE_AND" target="FCT_REC_NEWCUSTDAT"/>
<edge type="CXN_FOLLOWS" source="OT_RULE_AND"
target="EVT_NEWCUSTORD_CREATED"/>
<edge type="CXN_FOLLOWS" source="FCT_CREATE_NEWCUSTORD"
target="OT_RULE_AND"/>
<edge type="CXN_FOLLOWS" source="OT_RULE_AND"
target="FCT_CREATE_NEWCUSTORD"/>
</graph>
</graphlist>
```

Visão geral das diretrizes de modelagem de fragmento de processo

Elemento ARIS	Atributo ARIS	Definição de fragmento do PPM
EPC (modelo de fragmento)	Nome	**graph id** do gráfico de definição criado
Função	Nome	AT_OBJNAME
	Identificador	AT_OBJNAME_INTERN
Evento	Nome	AT_OBJNAME
	Identificador	AT_OBJNAME_INTERN
Regra	Identificador	AT_OBJNAME_INTERN
Atributos livres	Atributo de usuário Texto <x> x = Número inteiro de 1 - 37	Atributo PPM do tipo **TEXT**, incluindo o valor para objetos ou conexões)

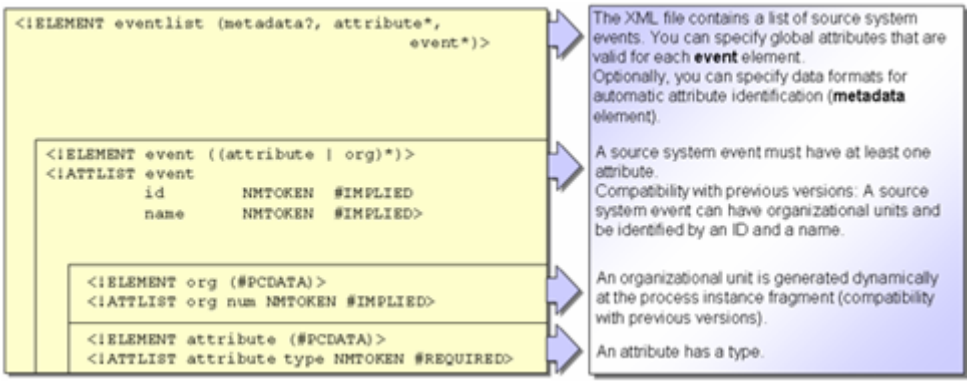
3.2.3.3. Formato do arquivo do evento de sistema

O arquivo de eventos de sistema contém as informações reais da instância sobre o curso real dos processos. Um extrato do arquivo XML para o exemplo acima poderia ter a seguinte aparência:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE eventlist SYSTEM "event.dtd">
<eventlist>
...
<event>
<attribute type="PPM_EVENT_ID">A-Nr 4711</attribute>
<attribute type="PPM_EVENT_NAME">AUFTRAG_ANGELEGT</attribute>
<attribute type="AUFTRAGNR">4711</attribute>
<attribute type="AUFTRAGART">Rush order</attribute>
<attribute type="START_TIME">11.11.2002 11:11:11</attribute>
<attribute type="PPM_ORG_NAME">Mr Miller</attribute>
<attribute type="PPM_ORG_NUM">1</attribute>
</event>
...
<event>
```

```
<attribute type="PPM_EVENT_ID">A-Nr 4711</attribute>
<attribute type="PPM_EVENT_NAME">RECHNUNG_ERSTELLT</attribute>
<attribute type="AUFTRAGNR">4711</attribute>
<attribute type="AUFTRAGART">Rush order</attribute>
<attribute type="START_TIME">12.11.2002 14:21:15</attribute>
<attribute type="PPM_ORG_NAME">Ms. Smith</attribute>
<attribute type="PPM_ORG_NUM">1</attribute>
</event>
...
</eventlist>
```

O formato do arquivo de saída XML é especificado pela DTD para o formato do evento de sistema PPM:



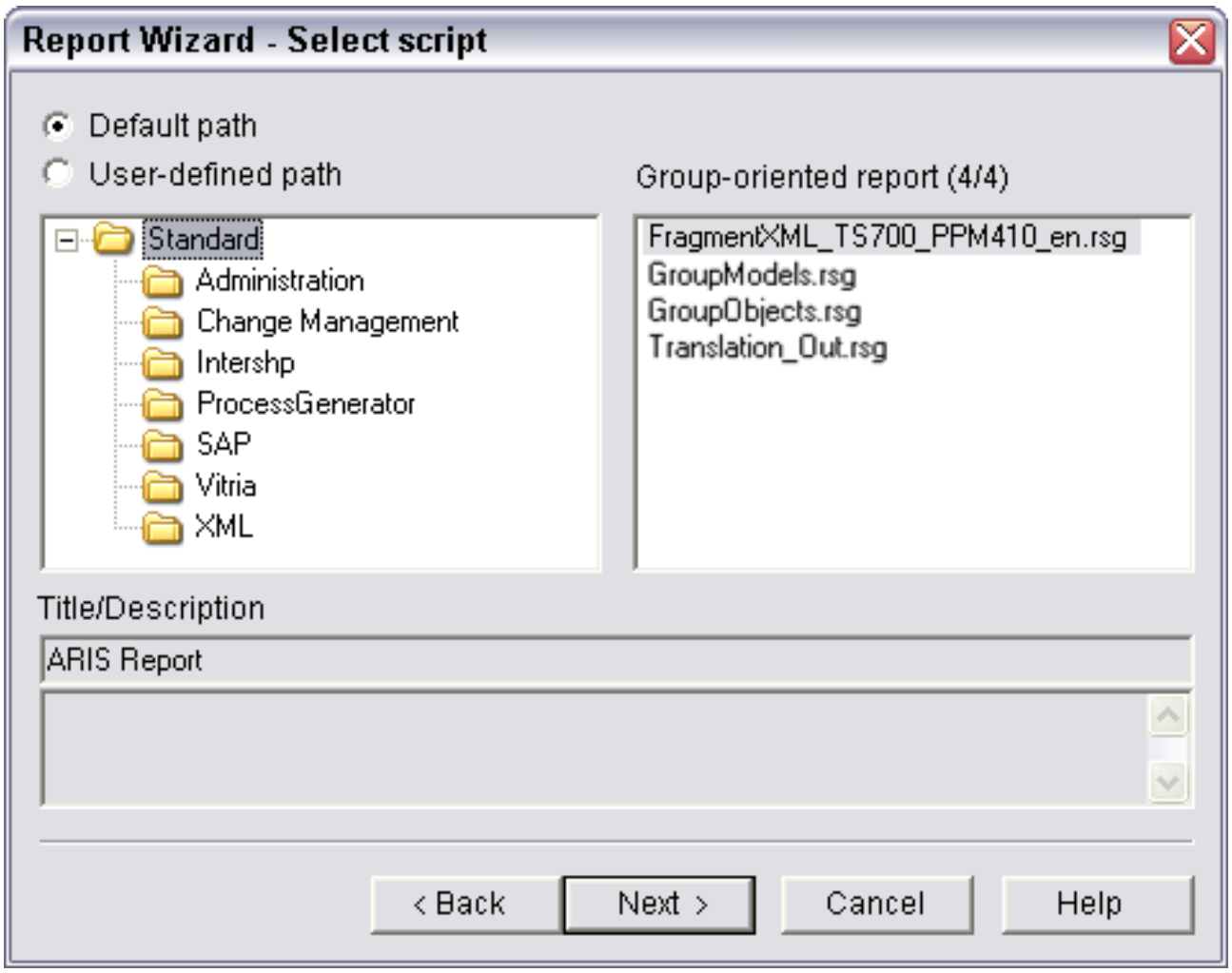
Os atributos XML **id** e **name** dos elementos XML **event** e **org** permitem continuar a usar arquivos de evento do sistema gerados para as versões 1.x e 2.x do PPM, sem alterações.

A partir da versão 3.0 do PPM, eles são substituídos pelos atributos de evento do sistema **PPM_EVENT_ID** e **PPM_EVENT_NAME** e pelo elemento XML **orgunit**.

3.2.3.4. Executar o relatório de ARIS

O relatório do ARIS para gerar o arquivo XML de definição de fragmento é projetado como um relatório de grupo.

Copie o script do relatório **FragmentXML_TS700_PPM410_en.rsg** localizado em <PPM installation directory>\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-<version>-runnable.zip\ppm\ctk\ARIS para o subdiretório **script\report\en** da instalação de ARIS (diretório padrão de ARIS para scripts de relatório). Para iniciar o relatório, selecione o menu de contexto **Avaliar/ Relatório** para o grupo que contém as definições de fragmento.



Se estiver a usar o ARIS Architect importe o script **FragmentXML_BA700_PPM410.arx**. Para importar, recomendamos que crie uma categoria separada chamada **PPM**. Para executar o relatório, use a entrada **FragmentXML**.

Depois que o relatório é executado com sucesso, o arquivo de definição de fragmento XML pode ser encontrado no diretório de saída selecionado. O diretório padrão é o diretório **script\report\out** na instalação de ARIS. O nome de arquivo é composto pelo prefixo **Frag_** e um carimbo de data/hora.

- Certifique-se de que o relatório é executado sem avisos ou mensagens de erro. A mensagem **Acesso a um atributo que não está especificado** indica que não foi inserido um atributo obrigatório, por exemplo, o atributo **Identifier**. Neste caso, os arquivos XML gerados não podem ser usados.
- Se não tiver acesso a um script de relatório atualizado para criar arquivos de definição de fragmento, deverá converter o script **FragmentXML.rsg** do diretório **\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-<version>-runnable.zip\ppm\ctk\ARIS** da instalação do PPM para um formato válido usando o **Conversor de script do ARIS**.

3.2.4. Gerar o arquivo de saída XML

Para gerar o arquivo de saída, os possíveis eventos do sistema devem ser identificados e nomeados. O exemplo abaixo usa os eventos de sistema **Pedido criado** e **Fatura criada**.

A ocorrência desses eventos de sistema é gravada em um arquivo de saída XML com informações adicionais (por exemplo, número do pedido, tempos de execução, valores base para o cálculo de medida e criação de dimensões).

3.2.5. Sumário

O exemplo simples abaixo destina-se a destacar a importação no formato do evento de sistema descrito.

Para importar os arquivos de saída tem de criar o arquivo de definição de fragmentos **frag.xml** e o arquivo de mapeamento **map.xml**:

Definições de fragmentos no arquivo **frag.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE graphlist SYSTEM "Graph.dtd">
<graphlist>
<graph id="FRG_ORD_CREATED">
<node id="EVT_ORD_TOBECREATED" type="OT_EVT">
<attribute type="AT_OBJNAME">Customer order to be created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_ORD_TOBECREATED</attribute>
</node>
<node id="FCT_CREATE_ORDER" type="OT_FUNC">
<attribute type="AT_OBJNAME">Create customer order</attribute>
<attribute type="AT_OBJNAME_INTERN">FCT_CREATE_ORDER</attribute>
</node>
<node id="EVT_ORD_CREATED" type="OT_EVT">
<attribute type="AT_OBJNAME">Customer order created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_ORD_CREATED</attribute>
</node>
<edge type="CXN_FOLLOWS" source="EVT_ORD_TOBECREATED"
target="FCT_CREATE_ORDER"/>
<edge type="CXN_FOLLOWS" source="FCT_CREATE_ORDER" target="EVT_ORD_CREATED"/>
</graph>
<graph id="FRG_INVOICED">
<node id="EVT_TOBE_INVOICED" type="OT_EVT">
<attribute type="AT_OBJNAME">Invoice to be created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_TOBE_INVOICED</attribute>
</node>
<node id="FCT_INVOICE" type="OT_FUNC">
<attribute type="AT_OBJNAME">Create invoice</attribute>
<attribute type="AT_OBJNAME_INTERN">FCT_INVOICE</attribute>
</node>
<node id="EVT_INVOICED" type="OT_EVT">
<attribute type="AT_OBJNAME">Invoice created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_INVOICED</attribute>
</node>
```

```
<edge type="CXN_FOLLOWS" source="EVT_TOBE_INVOICED" target="FCT_INVOICE"/>
<edge type="CXN_FOLLOWS" source="FCT_INVOICE" target="EVT_INVOICED"/>
</graph>
</graphlist>
```

Definições de mapeamento no arquivo **map.xml**:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE eventmapping SYSTEM "eventmapping.dtd">
<eventmapping>
<processfragmentmapping>
<processfragment graphid="FRG_ORD_CREATED">
<condition eventattributetype="PPM_EVENT_NAME">
<value>FRG_ORD_CREATED</value>
</condition>
</processfragment>
<processfragment graphid="FRG_INVOICED">
<condition eventattributetype="PPM_EVENT_NAME">
<value>FRG_INVOICED</value>
</condition>
</processfragment>
</processfragmentmapping>
<attributemapping>
<objectattributes objectname="EVT_ORD_TOBECREATED" graphid="FRG_ORD_CREATED">
<attribute ppmattributetype="AT_MERGE_KEY_1">
<eventattributetype>EVT_PRED</eventattributetype>
</attribute>
</objectattributes>
<objectattributes objectname="FCT_CREATE_ORDER" graphid="FRG_ORD_CREATED">
<orgunit eventattributetype="PPM_ORG_NAME"/>
<attribute ppmattributetype="AT_MERGE_KEY_2">
<eventattributetype>START_TIME</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_KI_FBZ">
<eventattributetype>PROCESSINGTIME</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_SAPCLIENT">
<eventattributetype>KUNDENNR</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_SAP_BSTYP">
<eventattributetype>AUFTRAGAT</eventattributetype>
</attribute>
</objectattributes>
<objectattributes objectname="EVT_ORD_CREATED" graphid="FRG_ORD_CREATED">
<attribute ppmattributetype="AT_ID">
<eventattributetype>AUFTRAGNR</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_SAPCLIENT">
<eventattributetype>KUNDENNR</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_MERGE_KEY_1">
<eventattributetype>EVT_NR</eventattributetype>
</attribute>
```

```
</objectattributes>
<objectattributes objectname="EVT_TOBE_INVOICED" graphid="FRG_INVOICED">
<attribute ppmattributetype="AT_MERGE_KEY_1">
<eventattributetype>EVT_PRED</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_ID">
<eventattributetype>AUFTRAGNR</eventattributetype>
</attribute>
</objectattributes>
<objectattributes objectname="FCT_INVOICE" graphid="FRG_INVOICED">
<orgunit eventattributetype="PPM_ORG_NAME"/>
<attribute ppmattributetype="AT_MERGE_KEY_2">
<eventattributetype>START_TIME</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_KI_FBZ">
<eventattributetype>PROCESSINGTIME</eventattributetype>
</attribute>
</objectattributes>
<objectattributes objectname="EVT_INVOICED" graphid="FRG_INVOICED">
<attribute ppmattributetype="AT_ID">
<eventattributetype>AUFTRAGNR</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_SAPCLIENT">
<eventattributetype>KUNDENNR</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_MERGE_KEY_1">
<eventattributetype>EVT_NR</eventattributetype>
</attribute>
</objectattributes>
</attributemapping>
</eventmapping>
```

A extração de dados do sistema de aplicativos criou o arquivo de saída XML **events.xml**:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE eventlist SYSTEM "event.dtd">
<eventlist>
<event>
<attribute type="EVT_PRED">0</attribute>
<attribute type="PPM_EVENT_ID">4711</attribute>
<attribute type="PPM_EVENT_NAME">FRG_ORD_CREATED</attribute>
<attribute type="PPM_ORG_NAME">Mr Miller</attribute>
<attribute type="AUFTRAGNR">4711</attribute>
<attribute type="AUFTRAGART">Rush order</attribute>
<attribute type="START_TIME">11.11.2002 11:11:11</attribute>
<attribute type="KUNDENNR">5711</attribute>
<attribute type="PROCESSINGTIME">7 MINUTE</attribute>
<attribute type="EVT_NR">1</attribute>
</event>
<event>
<attribute type="EVT_PRED">1</attribute>
<attribute type="PPM_EVENT_ID">4711</attribute>
<attribute type="PPM_EVENT_NAME">FRG_INVOICED</attribute>
```

```
<attribute type="PPM_ORG_NAME">Ms. Smith</attribute>
<attribute type="AUFTRAGNR">4711</attribute>
<attribute type="KUNDENNR">5711</attribute>
<attribute type="START_TIME">12.11.2002 14:21:15</attribute>
<attribute type="PROCESSINGTIME">14 MINUTE</attribute>
<attribute type="EVT_NR">2</attribute>
</event>
</eventlist>
```

Executar a linha de comando

```
runxmlimport -user system -password manager -f frag.xml -m
map.xml -i event.xml
```

gera fragmentos de instância de processo no banco de dados do PPM. A lista abaixo no formato de gráfico do PPM ilustra os fragmentos de instância de processo gerados no banco de dados do PPM. Na lista, as informações do fragmento da instância de processo, que foram adicionadas à definição do fragmento, são gravadas em negrito:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE graphlist SYSTEM "graph.dtd">
<graphlist>
<graph id="FRG_ORD_CREATED" xml:lang="en">
<node id="FRG_ORD_CREATED-EVT_ORD_TOBECREATED-1-8835472025300230616"
type="OT_EVT">
<attribute type="AT_OBJNAME">Customer order to be created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_ORD_TOBECREATED</attribute>
<attribute type="AT_EPK_FRAGMENT_ID">FRG_ORD_CREATED</attribute>
<attribute type="AT_INTERNAL_OBJECT_KEY">EVT_ORD_TOBECREATED0</attribute>
<attribute type="AT_MERGE_KEY_1">0</attribute>
<attribute type="AT_ORIG_EPK_ID">1</attribute>
</node>
<node id="FRG_ORD_CREATED-FCT_CREATE_ORDER-2-8344758599463564840"
type="OT_FUNC">
<attribute type="AT_OBJNAME">Create customer order</attribute>
<attribute type="AT_OBJNAME_INTERN">FCT_CREATE_ORDER</attribute>
<attribute type="AT_EPK_FRAGMENT_ID">FRG_ORD_CREATED</attribute>
<attribute type="AT_INTERNAL_OBJECT_KEY">
FCT_CREATE_ORDER11.11.2002 11:11:11</attribute>
<attribute type="AT_KI_FBZ">7 MINUTE</attribute>
<attribute type="AT_MERGE_KEY_2">11.11.2002 11:11:11</attribute>
<attribute type="AT_ORIG_EPK_ID">1</attribute>
<attribute type="AT_SAPCLIENT">5711</attribute>
</node>
<node id="FRG_ORD_CREATED-EVT_ORD_CREATED-3-7299716715746582786"
type="OT_EVT">
<attribute type="AT_OBJNAME">Customer order created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_ORD_CREATED</attribute>
<attribute type="AT_EPK_FRAGMENT_ID">FRG_ORD_CREATED</attribute>
<attribute type="AT_ID">4711</attribute>
<attribute type="AT_INTERNAL_OBJECT_KEY">EVT_ORD_CREATED1</attribute>
```

```
<attribute type="AT_MERGE_KEY_1">1</attribute>
<attribute type="AT_ORIG_EPK_ID">1</attribute>
<attribute type="AT_SAPCLIENT">5711</attribute>
</node>
<node id="FRG_ORD_CREATED-1--2--7661491465378853387" type="OT_ORG">
<attribute type="AT_INTERNAL_OBJECT_KEY">
FRG_ORD_CREATED-1--2-7661491465378853387</attribute>
<attribute type="AT_OBJNAME">Mr Miller</attribute>
</node>
<edge type="CXN_FOLLOWS"
source="FRG_ORD_CREATED-EVT_ORD_TOBECREATED-1-8835472025300230616"
target="FRG_ORD_CREATED-FCT_CREATE_ORDER-2-8344758599463564840" />
<edge type="CXN_FOLLOWS"
source="FRG_ORD_CREATED-FCT_CREATE_ORDER-2-8344758599463564840"
target="FRG_ORD_CREATED-EVT_ORD_CREATED-3-7299716715746582786" />
<edge type="CXN_UNDIRECTED"
source="FRG_ORD_CREATED-1--2-7661491465378853387"
target="FRG_ORD_CREATED-FCT_CREATE_ORDER-2-8344758599463564840">
<attribute type="AT_COUNT_PROCESSINGS">1</attribute>
</edge>
</graph>
<graph id="FRG_INVOICED" xml:lang="en">
<node id="FRG_INVOICED-EVT_TOBE_INVOICED-1-246376083169224336" type="OT_EVT">
<attribute type="AT_OBJNAME">Invoice to be created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_TOBE_INVOICED</attribute>
<attribute type="AT_EPK_FRAGMENT_ID">FRG_INVOICED</attribute>
<attribute type="AT_ID">4711</attribute>
<attribute type="AT_INTERNAL_OBJECT_KEY">EVT_TOBE_INVOICED1</attribute>
<attribute type="AT_MERGE_KEY_1">1</attribute>
<attribute type="AT_ORIG_EPK_ID">2</attribute>
</node>
<node id="FRG_INVOICED-FCT_INVOICE-2--1285282699037363371" type="OT_FUNC">
<attribute type="AT_OBJNAME">Create invoice</attribute>
<attribute type="AT_OBJNAME_INTERN">FCT_INVOICE</attribute>
<attribute type="AT_EPK_FRAGMENT_ID">FRG_INVOICED</attribute>
<attribute type="AT_INTERNAL_OBJECT_KEY">FCT_INVOICE12.11.2002
14:21:15</attribute>
<attribute type="AT_KI_FBZ">14 MINUTE</attribute>
<attribute type="AT_MERGE_KEY_2">12.11.2002 14:21:15</attribute>
<attribute type="AT_ORIG_EPK_ID">2</attribute>
</node>
<node id="FRG_INVOICED-EVT_INVOICED-3-1541227247726154405" type="OT_EVT">
<attribute type="AT_OBJNAME">Invoice created</attribute>
<attribute type="AT_OBJNAME_INTERN">EVT_INVOICED</attribute>
<attribute type="AT_EPK_FRAGMENT_ID">FRG_INVOICED</attribute>
<attribute type="AT_ID">4711</attribute>
<attribute type="AT_INTERNAL_OBJECT_KEY">EVT_INVOICED2</attribute>
<attribute type="AT_MERGE_KEY_1">2</attribute>
<attribute type="AT_ORIG_EPK_ID">2</attribute>
<attribute type="AT_SAPCLIENT">5711</attribute>
</node>
<node id="FRG_INVOICED-1--2--8231301810541650135" type="OT_ORG">
<attribute type="AT_INTERNAL_OBJECT_KEY">FRG_INVOICED-1--2-
8231301810541650135</attribute>
<attribute type="AT_OBJNAME">Ms. Smith</attribute>
```

```
</node>
<edge type="CXN_FOLLOWS"
source="FRG_INVOICED-EVT_TOBE_INVOICED-1-246376083169224336"
target="FRG_INVOICED-FCT_INVOICE-2--1285282699037363371" />
<edge type="CXN_FOLLOWS"
source="FRG_INVOICED-FCT_INVOICE-2-1285282699037363371"
target="FRG_INVOICED-EVT_INVOICED-3-1541227247726154405" />
<edge type="CXN_UNDIRECTED"
source="FRG_INVOICED-1--2--8231301810541650135"
target="FRG_INVOICED-FCT_INVOICE-2--1285282699037363371">
<attribute type="AT_COUNT_PROCESSINGS">1</attribute>
</edge>
</graph>
</graphlist>
```

3.3. Formatos da data

Este capítulo descreve os formatos dos dados do sistema de origem, que são esperados na forma de valores de atributo no arquivo de saída XML.

O mapeamento de atributo associa um tipo de atributo do PPM aos atributos do sistema de origem. Os atributos do sistema de origem são portadores puros de informações na forma de uma cadeia de caracteres. O tipo de dados do atributo do PPM especifica como o texto do atributo do sistema de origem será interpretado.

Dados numéricos

Os dados numéricos são essencialmente interpretados como informação no sistema decimal. Números de ponto flutuante usam pontos como separadores entre locais, antes e depois do ponto decimal, independentemente da configuração do sistema operacional.

Os valores de atributo são importados com uma unidade, por exemplo,**57 USD**. Se nenhuma unidade for especificada, será assumida a unidade base do tipo de dados no qual o tipo de atributo do PPM se baseia.

As percentagens são especificadas como **58 PERCENT** ou como um fator na forma de **0.58 VALUE_ONLY**.

Dados baseados no tempo

O PPM distingue entre três categorias de dados de referência baseados no tempo:

Data:

É especificado o dia do calendário, por exemplo, **26.06.2003**.

Tempo:

É especificada a hora do dia em formato de 24 horas, por exemplo, **14:29**.

Carimbo de data/hora:

A especificação de um carimbo de data/hora é composta pelas informações da data e da hora, por exemplo, **26.06.2003 14:29**.

Para analisar a cadeia de caracteres de dados é usada a cadeia de caracteres de formato especificada no arquivo **AdapterConfig_settings.properties** na pasta de configuração do cliente para a categoria relevante.

Atenção
Para importação em formato gráfico são usadas as informações da cadeia de caracteres de formato do arquivo **AdapterConfig_settings.properties**. A partir da versão 3.0 do PPM passou a ser fornecido o atributo XML **format**, no mapeamento de atributo, para permitir a importação de dados de hora diferentes do formato padrão do PPM.

Como extensão do padrão Java (ver Definição do mapeamento de atributo [20], seção **Transformações**), o PPM fornece a cadeia de caracteres de formato **Q.yyyy**, que permite a importação de dados de hora como trimestre, por exemplo, **2.2003** para o segundo trimestre de 2003.

3.3.1. Caracteres especiais em documentos XML

Caracteres especiais como **&**, **>**, **<** e **"** são caracteres de controle (metacaracteres) do documento XML. Se esses caracteres ocorrerem diretamente nos dados de origem, poderão ocorrer interpretações incorretas durante a importação do arquivo XML. Para poder usar esses caracteres nos dados de referência, terá de substituí-los pelas seguintes entidades (cadeias de caracteres):

Caracteres	Entidade	Caracteres	Entidade
&	&	"	"
<	<	>	>

A tabela abaixo mostra a representação de caracteres especiais em arquivos XML:

Caracteres	Entidade	Caracteres	Entidade	Caracteres	Entidade
¡	¡	»	»	£	£
¥	¥	|	¦	§	§
©	©	ª	ª	«	«
®	®	°	°	±	±
1	¹	2	²	3	³
¼	¼	½	½	¾	¾
Ä	Ä	Ö	Ö	Ü	Ü
ä	ä	ö	ö	ü	ü
ß	ß	x	×	÷	&divide
È	È	É	É	Ê	Ê
Ë	Ë	è	È	é	É
ê	Ê	ë	Ë	Ø	&Oslach;

Pode usar qualquer caractere, indicando a sua codificação da fonte. Para isso, especifique o código ASCII do caractere relevante na notação decimal da seguinte forma: **&#<Decimal code of character>;**.

Por exemplo, para usar o caractere @, insira **@** no documento XML.

3.4. Gerar os fragmentos de instância de processo

Formato gráfico

A importação para o sistema PPM envolve a importação dos dados da instância de fragmento contidos no arquivo de saída XML e a geração de fragmentos da instância de processo.

Formato de evento

A importação para o sistema PPM envolve o mapeamento de dados de instância para os eventos de sistema contidos no arquivo de saída XML, para as definições de fragmento de processo no arquivo de fragmento especificado e a geração de fragmentos de instância de processo. Os atributos especificados no arquivo de mapeamento são então copiados para os fragmentos de instância de processo gerados.

Independentemente do formato usado para importar a instância de fragmento, antes que um fragmento de instância de processo seja finalmente salvo no banco de dados do PPM, as chaves de processo (ver Merge de fragmentos de processo [85]) são calculadas para permitir o merge de fragmentos da instância de processo em uma instância de processo.

Atenção
Pelo menos uma chave de processo deve ser calculada para cada fragmento da instância de processo importada. Os fragmentos de instância de processo para os quais nenhuma chave de processo pode ser calculada não são salvos no banco de dados do PPM. Estes fragmentos não fazem parte de um processo geral.

Durante a importação, as mensagens de estado relevantes são exibidas no prompt de comando e, opcionalmente, no arquivo de log especificado.

3.4.1. Expandir a configuração de atributos

Pode identificar automaticamente os atributos contidos nos arquivos de importação XML que não são conhecidos no sistema PPM. Opcionalmente, pode atribuir um tipo de dados aos novos atributos adicionados e transferi-los automaticamente para a configuração de atributos do cliente PPM.

A identificação automática de atributo suporta ambos os formatos de importação de dados.

Pode usar as seguintes opções para ativar a extensão automática da configuração de atributos:

Argumentos de linha de comando

Quando chamar o **runxmlimport**, especifique os argumentos adicionais **-autoextendattributes** e **-extractattributes** na linha de comando.

A especificação desses argumentos tem o seguinte efeito ao importar:

Argumento	Efeito
-extractattributes <File name>	Atributos desconhecidos são salvos no arquivo especificado. Nenhum arquivo de fragmento XML é importado.
-autoextendattributes	Atributos desconhecidos são transferidos para a configuração de atributos. Se a opção **automapping** estiver habilitada, a configuração de mapeamento de atributo é expandida em conformidade. Os dados de fragmento XML são então importados.
São especificados ambos os argumentos	As ações descritas em -autoextendattributes e -extractattributes são executadas consecutivamente.

Arquivo da fonte de dados

Associe o valor **true** ao atributo **autoextendattributes** do elemento XML **attributesettings** no arquivo usado **datasource**.

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

<datasource name="Events" type="EVENT">
...
<attributesettings autoextendattributes="true">
...
</attributesettings>
</datasource>
```

Certos atributos podem ser excluídos da identificação automática de atributos usando identificação de padrões. Para isso, especifique o padrão do nome dos atributos que deseja excluir da extensão automática no elemento XML **excludepattern**. Pode usar os espaços reservados **?** (apenas um caractere) e ***** (qualquer conjunto de caracteres).

No exemplo abaixo, todos os atributos de evento do sistema cujo nome começa com **TEST** ou que tenham o nome **USER** são excluídos da identificação automática de atributos.

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

<datasource name="Events" type="EVENT">
...
<attributesettings autoextendattributes="true">
<excludepatterns>
<excludepattern>TEST*</excludepattern>
<excludepattern>USER</excludepattern>
</excludepatterns>
...
</attributesettings>
</datasource>
```

3.4.1.1. Especifique o tipo de dados de atributos desconhecidos

Pode especificar o tipo de dados a ser associado a um atributo desconhecido das seguintes formas:

- Metadados especificados no arquivo de evento do sistema (apenas para formato de importação de evento)
- Identificação de padrões no nome de atributo, por exemplo, **TEXT_***
- Análise do valor do atributo

A sequência mostrada corresponde à priorização da identificação automática do tipo de dados.

> **Atenção**
> O tipo de dados de atributos PPM existentes não pode ser modificado.

Metadados

Nos arquivos de evento do sistema a serem importados, pode especificar um tipo de dados do PPM e, opcionalmente, um formato (por exemplo, para identificar carimbos de data/hora) e descrição de atributo para cada atributo de um evento de sistema. Os atributos que ainda não são conhecidos no sistema PPM são gerados automaticamente com o tipo de dados especificado.

Os metadados são especificados no elemento XML **metadata**:

```
<eventlist>
<metadata>
<attr_desc type="...">
<ppmdatatype>...</ppmdatatype>
<format>...</format>
<description>...</description>
</attr_desc>
...
</metadata>

<event>
...
</event>
...
</eventlist>
```

As marcas XML para o elemento XML **metadata** têm o seguinte significado:

Marca XML	Descrição
attr_desc	Definição individual de metadados de um atributo de evento de sistema. Podem ser especificadas diversas definições.
type	Nome do atributo do evento de sistema
ppmdatatype	Tipo de dados do PPM a ser associado ao atributo do PPM gerado.
format (opcional)	Formato a ser usado para análise do valor do atributo. Se também tiver habilitado o mapeamento automático, o formato também será usado para transformar o valor em formato de hora do PPM.
description (opcional)	Descrição do atributo do PPM gerado. A descrição especificada é transferida na configuração de atributos, no idioma padrão do cliente PPM.

```
<eventlist>
<metadata>
<attr_desc type="SWWWIHEAD-WI_ID">
<ppmdatatype>TEXT</ppmdatatype>
<description>Work item ID</description>
</attr_desc>
<attr_desc type="SWWWIRET-WI_AED">
<ppmdatatype>DAY</ppmdatatype>
<format>yyyyMMdd</format>
<description>End date of work item</description>
</attr_desc>
</metadata>

<event>
<attribute type="SWWWIHEAD-WI_ID">000000525723</attribute>
<attribute type="SWWWIRET-WI_AED">20011211</attribute>
</event>
</eventlist>
```

Os extratores disponíveis para o PPM (PPM Process Extractor SAP-2-PPM e PPM Process Extractor JDBC-2-PPM) geram automaticamente o elemento XML **metadata**, permitindo importar diretamente os arquivos de evento de sistema gerados, sem que seja necessária qualquer ação. Os atributos que ainda não são conhecidos no sistema PPM são gerados automaticamente com o tipo de dados correto.

Identificação de padrões

O elemento XML **datatypekeydetectionsettings** do arquivo da fonte de dados pode ser usado para especificar uma lista de padrões (elementos XML **datatypedetectionpattern**) a ser usada para associação do tipo de dados. Os nomes de todos os novos atributos identificados são comparados com os padrões. O primeiro padrão identificado na lista especifica o tipo de dados do atributo.

Exemplo 20.

O extrato seguinte do arquivo de fonte de dados configura o seguinte padrão de nomeação para identificar o tipo de dados de novos atributos identificados:

- Um evento de sistema cujo nome começa com **L_** ou termina em **L** é associado ao tipo de dados **LONG**.
- Um evento de sistema cujo nome começa e termina em **D** é associado ao tipo de dados **DOUBLE**.

```
...
<attributesettings autoextendattributes="true">
<datatypedetectionsettings>
<datatypekeydetectionsettings >
<datatypedetectionpattern datatype=”LONG”>*L
</datatypedetectionpattern>
<datatypedetectionpattern datatype=”LONG”>L_*
</datatypedetectionpattern>
<datatypedetectionpattern datatype=”DOUBLE”>D*D
</datatypedetectionpattern>
</datatypekeydetectionsettings>
</datatypedetectionsettings>
</attributesettings>
...
```

Análise do valor do atributo

Se a identificação de padrões não puder identificar um tipo de dados, será feita uma tentativa para identificar o tipo de dados com base no valor do atributo. Os seguintes tipos de dados são identificados: **BOOLEAN**, **LONG**, **DOUBLE**, **TIME**, **TIMEOFDAY**, **DAY** e **TEXT**. A enumeração corresponde à prioridade para identificação do tipo de dados.

Tipo de dados BOOLEAN

Os valores de atributo **true** e **false** são associados ao tipo de dados **BOOLEAN** independentemente da capitalização. Outros valores, por exemplo, **0** e **1** não são identificados como **BOOLEAN**.

Tipos de dados LONG e DOUBLE

Se o tipo de dados **BOOLEAN** não for identificado, o sistema tentará identificar valores numéricos. Ao identificar tipos de dados numéricos, é feita uma distinção entre números inteiros e números de ponto flutuante. No atributo **doubleonly** para o elemento XML **datatypevaluedetectionsettings**, pode usar o valor **TRUE** (valor padrão) para especificar que os valores de atributo inteiros devem ser associados ao tipo de dados **DOUBLE**.

A identificação do tipo de dados **DOUBLE** não requer a especificação de um separador de milhares. O ponto é o único separador decimal identificado.

Tipos de dados TIME, TIMEOFDAY e DAY

Se nenhum tipo de dados numérico for identificado, o sistema tenta identificar o valor do atributo como um carimbo de data/hora (**TIME**), hora do dia (**TIMEOFDAY**) ou data (**DAY**). A enumeração corresponde à prioridade para identificação do tipo de dados. Para cada um dos tipos de dados do carimbo de data/hora, hora do dia e data, pode usar opcionalmente os elementos XML **timeformat**, **timeofdayformat** e **dayformat** para especificar cadeias de caracteres de formato descrevendo os formatos dos valores de atributo especificados. Se não especificar cadeias de caracteres de formato, os seguintes formatos padrão serão aplicados:

Data type	Formato padrão
TIME	dd.MM.yyyy HH:mm:ss
TIMEOFDAY	dd.MM.yyyy
DAY	HH:mm:ss

Tipo de dados TEXT

Se nenhum tipo de dados for identificado através da análise do valor do atributo, o atributo será associado ao tipo de dados **TEXT**.

Comportamento com tipos de dados ambíguos

Pode usar o atributo **numberofvaluestocheck** para o elemento XML **datatypevaluedetectionsettings** para especificar com que frequência o tipo de dados deve ser recuperado ao analisar o valor do atributo. Se o atributo XML não for especificado, é usado o valor padrão **100**. Se diferentes tipos de dados forem identificados para um atributo, o tipo de dados identificado anteriormente é convertido em um tipo de dados geral. A tabela seguinte se aplica:

Data type	Tipo de dados geral
BOOLEAN	TEXT
LONG	DOUBLE, TEXT
DOUBLE	TEXT
TIME	TEXT
TIMEOFDAY	TEXT
DAY	TEXT
TEXT	TEXT

Exemplo 21.

O extrato de arquivo abaixo mostra uma configuração completa para identificação automática de tipo de dados, analisando o valor do atributo.

```
...
<attributesettings autoextendattributes="true">
<excludepatterns>
<excludepattern>TEST*</excludepattern>
<excludepattern>USER</excludepattern>
</excludepatterns>
<datatypedetectionsettings>
<datatypekeydetectionsettings>
<datatypedetectionpattern datatype = "LONG">LG_*
</datatypedetectionpattern>
```

```
<datatypedetectionpattern datatype = "LONG">*LONG*
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "LONG">*_lng
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "DOUBLE">DOUBLE_*
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "DOUBLE">*_dbl
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "DAY">DAY_*
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "DAY">*_day
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "TIMEOFDAY">TIMEOFDAY_*
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "TIMEOFDAY">*_tod
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "TIME">TIME_*
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "TIME">*_tm
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "TEXT">TEXT_*
</datatypedetectionpattern>
<datatypedetectionpattern datatype = "TEXT">*_txt
</datatypedetectionpattern>
</datatypekeydetectionsettings>
<datatypevaluedetectionsettings
doubleonly = "FALSE"
numberofvaluestocheck = "100">
<timeformat>dd.MM.yyyy HH:mm:ss</timeformat>
<timeofdayformat>HH:mm:ss</timeofdayformat>
<dayformat>dd.MM.yyyy</dayformat>
</datatypevaluedetectionsettings>
</datatypedetectionsettings>
<attributeprefix>AT_</attributeprefix>
</attributesettings>
...
```

3.4.2. Expandir a configuração de mapeamento

Se estiver usando o formato de evento para importar, poderá ativar o mapeamento automático, ou seja, todos os atributos de um evento de sistema serão transferidos para os tipos de objetos especificados na definição de fragmento associada.

A expansão da configuração de mapeamento é configurada usando atributos do elemento XML **automapping**.

Atributos XML	Descrição
nodetype	determina o tipo de objeto para o qual este **automapping** é válido. Valores válidos: **OT_FUNC** para funções, **OT_EVT** para eventos e **PROCESS** para processos.
graphid	determina o gráfico de definição de fragmento para cujos objetos este **automapping** é válido. O valor especificado corresponde ao atributo **id** do elemento XML **graph** na definição de fragmento.

Atributos XML	Descrição
addmergeattributes (opcional)	Para o tipo de objeto **PROCESS**, determina se os atributos são adicionados à configuração de merge, o que significa que serão retidos como atributos de processo ao mesclar fragmentos de instância (valor **TRUE**) ou não (valor **FALSE**). Esta entrada é ignorada para os tipos de objetos **OT_FUNC** e **OT_EVT**. Para o tipo de objeto **PROCESS**, terá de especificar **addmergeattributes**.

Se precisar de especificar a extensão de mapeamento automático para vários tipos de objeto ou gráficos de definição de fragmento, é possível especificar um elemento XML **automapping** separado para cada tipo de objeto ou gráfico de definição de fragmento obrigatório. O mapeamento automático leva em conta o prefixo especificado no elemento XML **attributeprefix**, por exemplo, **AT_**. Se especificou um mapeamento explícito para atributos específicos, isso substitui os atributos transferidos anteriormente pelo mapeamento automático.

Exemplo 22. *[en-us]*

No extrato de arquivo abaixo, a extensão de mapeamento automático é configurada para funções do gráfico de definição de fragmento **FRG_CATCH_ALL**.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE eventmapping SYSTEM "eventmapping.dtd">
<eventmapping>
<processfragmentmapping>
<!--FRG_CATCH_ALL-->
<processfragment graphid="FRG_CATCH_ALL">
...
</processfragment>
...
</processfragmentmapping>
<attributemapping>
...
<automapping
nodetype="OT_FUNC"
graphid="FRG_CATCH_ALL"
/>
<attributemapping>
</eventmapping>
```

3.4.3. Atributos de evento de sistema de valores múltiplos

Se um atributo de evento do sistema ocorrer várias vezes em um evento do sistema, não há garantia de que o último valor lido será transferido. Para transferir todos os valores para atributos que ocorrem várias vezes em um evento de sistema, pode especificar o elemento XML **multieventattributetype** em vez do elemento XML **attribute** na configuração de mapeamento. Todos os valores dos atributos de evento do sistema identificados dessa maneira são concatenados com um separador (ponto e vírgula por padrão). Use o atributo XML opcional **delimiter** para especificar um separador diferente do padrão.

Usar vários valores de atributo de evento do sistema facilita o merge de caminhos de processos paralelos se gravar todas as chaves de merge do fragmento anterior no arquivo de evento do sistema como atributos **multieventattributetype**.

Exemplo 23.

Extrato do arquivo do evento de sistema

```
...
<event>
<attribute type="EVENTTYP">Change customer order</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>
</event>
...
```

Exemplo 24.

Extrato do arquivo de mapeamento

```
...
<attributemapping>
...
<!-- Mapping Event Start BEGIN -->
<objectattributes objectname="EVT_START" graphid="FRG_CATCH_ALL">
<attribute ppmattributetype="AT_OBJNAME">
<eventattributetype>EVENTTYP</eventattributetype>
<value> to be done</value>
</attribute>
<attribute ppmattributetype="AT_ID">
<eventattributetype>AT_PRCNO</eventattributetype>
</attribute>
<!-- Mulivalue Mapping -->
<attribute ppmattributetype="AT_KEY">
<multieventattributetype delimiter=";">PREV_KEY</multieventattributetype>
</attribute>
...
</objectattributes>

...
```

O atributo **AT_KEY** gerado para o evento **EVT_START** é associado ao valor **1;2**. Se o merge for configurado em conformidade, serão geradas duas chaves de merge para este evento.

3.4.4. Importação direta de atributos de processo

Os atributos necessários para calcular medidas e dimensões no nível da instância de processo são copiados normalmente para a instância de processo dos objetos, nas instâncias de fragmento importadas. Esta operação pode ser configurada usando regras de cópia de atributos ou uma regra de cálculo correspondente na calculadora da medida.

Como alternativa, pode importar diretamente os atributos da instância de processo (formato gráfico) ou gerá-los mapeando os atributos de evento de sistema para a definição de fragmento (formato de evento).

Para substituir atributos de processo de instâncias de processo existentes, pode importar diretamente instâncias de fragmentos sem objetos e conexões que contenham exclusivamente atributos de processo (formato gráfico) ou gerá-los mapeando atributos de evento de sistema para uma definição de fragmento sem objetos (formato de evento).

Formato gráfico

Exemplo 25.

Gráfico sem objetos e conexões

```
...
<graph id="FRG_EMPTY">
<attribute type="AT_ID">1</attribute>
<attribute type="AT_SAP_BELEGNR">Document 2</attribute>
<attribute type="AT_SAPCLIENT">R3</attribute>
</graph>
...
```

Formato de evento

Ao usar o formato de evento, os atributos de evento de sistema são transferidos para os gráficos instanciados em uma definição de fragmento sem objetos.

Exemplo 26.

Extrato de definição de fragmento sem objetos:

```
...
<graph id="FRG_EMPTY">
<attribute type="AT_ID">1</attribute>
</graph>
...
```

Exemplo 27.

Extrato de regra de mapeamento:

```
...
<processattributes graphid="FRG_EMPTY">
<attribute ppmattributetype="AT_SAP_BELEGNR">
<value>Document </value>
<eventattributetype>SAP_BELEGNR</eventattributetype>
</attribute>
<attribute ppmattributetype="AT_SAPCLIENT">
<eventattributetype>SAPCLIENT</eventattributetype>
</attribute>
</processattributes>
...
```

Atenção
Para poder associar fragmentos a serem importados sem objetos para instâncias de processo, deve garantir que as chaves de processo válidas podem ser calculadas com base nos atributos de processo.

Para garantir que os atributos de processo importados diretamente são retidos na instância, deve especificar todos os atributos importados diretamente no elemento XML **mergeattributes** para a configuração de merge.

```
...
<processmerge>
```

```
<mergeattributes>
<attribute key = "AT_SAP_BELEGNR"/>
<attribute key = "AT_SAPCLIENT"/>
</mergeattributes>
</processmerge>
...
```

3.4.5. Caso especial de sistema dimensionado

Em um sistema dimensionado, os dados são importados para o subservidor. Todos os subservidores têm de ter uma configuração idêntica. As opções **autoextendattributes** e **addmergeattributes** mudariam a configuração de um subservidor individual e a configuração uniforme de todos os subservidores seria perdida. Por isso, não é possível usar estas duas opções em um sistema dimensionado.

Se ainda quiser usar a extensão automática da configuração de atributos, precisará exportar as extensões possíveis para cada subservidor e importá-las manualmente no servidor mestre, que poderá então distribuir as extensões para todos os subservidores no sistema.

Antes de importar dados com novos atributos para o subservidor, execute os passos a seguir:

1. Identifique novos atributos de evento de sistema e extraia-os usando o parâmetro -**extractattributes <File name>** no programa de linha de comando **xmlimport**.
2. Importe os novos atributos para o servidor mestre.
3. Identifique os novos atributos que deseja reter ao mesclar fragmentos de processo e extraia-os usando o parâmetro **-extractmergeattributes <File name>** no programa de linha de comando **xmlimport**.
4. Expanda manualmente a configuração de merge no servidor mestre, tal como é descrito na seção **Adicionar os atributos de merge** abaixo.
5. Efetue uma importação de dados XML para o subservidor, com a opção **automapping**.

Adicionar os atributos de merge

Se especificou a extensão de mapeamento automático para processos (**<automapping nodetype="PROCESS" ...**), pode extrair os novos atributos adicionados que serão retidos ao mesclar os fragmentos de processo, especificando o parâmetro **-extractmergeattributes <File name>** na linha de comando. Nenhuns dados XML são importados. O arquivo gerado contém apenas um elemento XML **mergeattributes** com uma lista de todos os novos atributos e tem a seguinte estrutura:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mergeattributes>
<attribute key="..." />
...
</mergeattributes>
```

Use o parâmetro **-export -merger <File name>** para exportar a configuração de merge para o servidor mestre. Edite a configuração de merge adicionando os atributos (elementos XML **attribute** para elemento XML **mergeattributes**) a todos os atributos de merge da configuração de merge exportada anteriormente para o servidor mestre exportado, para cada subservidor usando **-exctractmergeattributes <File name>**.

Por fim, use o parâmetro **-import -merger <File name>** para importar a configuração de merge do servidor mestre.

Exemplo 28.

Configuração de merge exportada para o servidor mestre

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mergerconfig SYSTEM "mergerconfig.dtd">
<mergerconfig>
<mergehandling>
<processmerge>
<mergeattributes>
<attribute key = "AT_SAPSYSTEM"/>
<attribute key = "AT_SAP_BELEGNR"/>
</mergeattributes>
</processmerge>
<eventmerge priority="1">
<mode>
<keymerge/>
</mode>
</eventmerge>
</mergehandling>
</mergerconfig>
```

Exemplo 29.

Atributos de merge exportados para o subservidor 1

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mergeattributes>
<attribute key = "AT_SAPCLIENT"/>
</mergeattributes>
```

Atributos de merge exportados para o subservidor 2

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mergeattributes>
<attribute key="AT_SAP_BSTYP" />
<attribute key="AT_SAP_BSTYP" />
</mergeattributes>
```

Exemplo 30.

Configuração de merge consolidada do servidor mestre

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mergerconfig SYSTEM "mergerconfig.dtd">
<mergerconfig>
<mergehandling>
<processmerge>
<mergeattributes>
<attribute key = "AT_SAPSYSTEM"/>
<attribute key = "AT_SAP_BELEGNR"/>
```

```
<!-- Merge attributes sub-server 1 -->
<attribute key = "AT_SAPCLIENT"/>
<!-- Merge attributes sub-server 2 -->
<attribute key="AT_SAP_BSTYP" />
<attribute key="AT_SAP_BSTYP" />
</mergeattributes>
</processmerge>
<eventmerge priority="1">
<mode>
<keymerge/>
</mode>
</eventmerge>
</mergehandling>
</mergerconfig>
```

3.4.6. Arquivamento de arquivos de importação XML

Se quiser impedir que os arquivos de importação XML já importados sejam importados novamente, especifique que os arquivos importados serão renomeados ou movidos para um diretório diferente.

A opção de arquivamento de arquivos de importação é configurada usando o elemento XML **archive** no arquivo da fonte de dados usada:

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

<datasource name="Events" type="EVENT">
...
<archive>
<directory>...</directory>
<prefix>...</prefix>
</archive>
</datasource>
```

Elemento XML	Descrição
directory	Especifica o diretório para onde os arquivos XML importados serão movidos. Todos os diretórios no caminho especificado que ainda não existam serão criados automaticamente.
prefix	Especifica o prefixo dado ao nome dos arquivos de importação XML importados.

Se especificar ambos os atributos XML, a avaliação do atributo **directory** passa a ter prioridade e o atributo **prefix** é ignorado.

3.5. programa de linha de comando runxmlimport

Os arquivos de importação XML são importados usando o programa de linha de comando **runxmlimport**. Durante uma importação, os fragmentos da instância de processo são importados para o buffer interno do sistema PPM e, inicialmente, não são mais processados.

Atenção
Certifique-se de que o servidor PPM foi iniciado antes da importação XML. Apenas pode estar ativo um programa de linha de comando.

Se ocorrer um erro durante a operação de importação, será exibida uma mensagem de erro na console e o valor de retorno do programa não será igual a **0**.

Chamar o programa sem parâmetros ou com **-h** ou **-?** gera a ajuda on-line no console. A ajuda descreve todas as opções disponíveis:

Importar instâncias de processo em formato XML

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

[-datasource <file>]

[-datasourcelist <file>]

[-i <file1>[,<file2>...]]

[-f <fragment file> -m <mapping file>]

[-extractattributes <attribute file>] [-autoextendattributes]

[-extractmergeattributes <mergeattribute file>]

[protocoloptions] [-language <ISO code>] [-version]

-user <user name>	Designação do usuário
-password <password>	Senha do usuário
-client <name>	Nome do cliente. Se nenhum cliente for especificado, o processo de importação para o cliente padrão será executado.
-version	Número de versão
-datasource <file>	Fonte de dados a ser usada para a importação XML.
-datasourcelist <file>	Lista das fontes de dados a serem usadas para a importação XML.
-i <file1> [,<file2>...]	Importar arquivos. Os arquivos ZIP são suportados.

Se a importação for executada no formato de evento, -f e -m têm de ser especificados, se nenhuma fonte de dados ou lista de fontes de dados tiverem sido especificadas.

[-f <fragment file>]	Arquivo de fragmento
[-m <mapping file>]	Arquivo de mapeamento
[-extractattributes <attribute file>]	Gera um arquivo XML com todos os tipos de atributos que ocorrem no(s) arquivo(s) de entrada, mas que ainda não foram definidos.
[-autoextendattributes]	Antes da importação XML, são importados todos os tipos de atributos que ocorrem no(s) arquivo(s) de entrada, mas que ainda não foram definidos.
[-extractmergeattributes <mergeattribute file>]	Gera um extrato do arquivo de configuração de merge com todos os novos atributos que devem ser retidos como atributos de processo ao mesclar instâncias de fragmento.

A opção **protocoloptions** pode ser composta pelas seguintes instruções:

-protocolfile <file name>	Registrar no arquivo <file name>
-information {yes|no|default}	Registrar informação
-warning {yes|no|default}	Registrar avisos
-error {yes|no|default}	Registrar erros
-language <ISO code>	Idioma da saída de log

3.5.1. argumentos runxmlimport

-version

A versão de software do PPM e o esquema do banco de dados são emitidos para a console. Outros argumentos são ignorados.

-user <user name> -password <password>

Especifique o nome de usuário e a senha de usuário do PPM que está a executar a importação. O usuário deve ter o direito de função **Importação de dados**.

-client <client name>

Especifique o cliente PPM em cujo esquema do banco de dados os fragmentos de processo importados devem ser salvos. Se não inserir nada aqui, é usado o cliente padrão (**Default**).

-i <Name of XML files>

Especifica qualquer número de arquivos a ser importado. Os nomes podem conter espaços reservados. **?** é um espaço reservado para um único caractere, ***** para qualquer número de caracteres. O nome do arquivo tem de começar com um sinal de subtração.

Exemplo: O argumento **-i <Path>\?ata*.xml** lê todos os arquivos do diretório **<Path>** com a extensão **<.xml** que começam com qualquer caractere seguido pela cadeia de caracteres **<ata** e depois por qualquer outra cadeia de caracteres.

Se os arquivos especificados forem um arquivo ZIP, todos os arquivos XML no arquivo ZIP serão lidos, independentemente da estrutura de pastas do arquivo.

-m <name of mapping file> -f <name of fragment file>

Especifique o tipo de importação. Se especificar os argumentos **-f** e **-m**, o formato de evento do sistema PPM é usado automaticamente. Se não especificar estes argumentos, é usado o formato gráfico.

-datasource <file name>

Os elementos XML **data**, **fragments** e **mapping** indicados no arquivo especificam os arquivos XML a serem usados e substituem os parâmetros da linha de comando **-i**, **-f** e **-m**. O formato do arquivo XML é descrito na definição do tipo de documento **datasource.dtd**. O PPM Customizing Toolkit usa este arquivo para configuração e para extrair fontes de dados.

O arquivo está localizado em
<installation directory>\ppm\server\bin\agentLocalRepo\.unpacked\<installation_time>_ppm-client-run-prod-<version>-runnable.zip\ppm\dtd

Os exemplos seguintes ilustram o conteúdo de um arquivo de fonte de dados para importação de dados no formato de evento gráfico e de sistema:

Arquivo da fonte de dados **datasource.xml** para formato gráfico

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE datasource SYSTEM "datasource.dtd">
<datasource name="Sales" type="GRAPH">
<description name="default_description"
language="de">Demo-Datenbankinhalt</description>
<description name="default_description"
language="en">Demo database content</description>
<data>C:\ARIS\ppm\server\bin\work\data_ppm\custom\umg_en\data\umgsales_umg_en.zi
p</data>
<eventattributetypes />
</datasource>
```

Arquivo da fonte de dados **datasource.xml** para formato de evento

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE datasource SYSTEM "datasource.dtd">
<datasource name="CustomerService" type="EVENT">
<description name="default_description"
language="de">Demo-Datenbankinhalt</description>
<description name="default_description"
language="en">Demo database content</description>
<data>C:\ARIS\ppm\server\bin\work\data_ppm\custom\umg_en\data\CS\csdemodata.zip<
/data>
<fragments>C:\ARIS\ppm\server\bin\work\data_ppm\custom\umg_en\xml\CS\CustomerSer
vice_Fragments.xml</fragments>
<mapping>C:\ARIS\ppm\server\bin\work\data_ppm\custom\umg_en\xml\CS\CustomerServi
ce_Mapping.xml</mapping>
<eventattributetypes />
</datasource>
```

Se especificar o parâmetro **-datasource** na linha de comando, os parâmetros **-i**, **-f** e **-m** não serão considerados.

Para informações mais detalhadas sobre configuração de fontes de dados, consulte a referência técnica **Extratores de processo do PPM**.

-datasourcelist <file name>

Com o argumento **-datasourcelist**, pode importar várias fontes de dados simultaneamente, exceto fontes de dados do tipo **GRAPH**. A importação corresponde a várias importações usando o argumento **-datasource**.

Ver o capítulo Importar várias fontes de dados [60].

-autoextendattributes

Se especificar essa opção na linha de comando, os novos atributos serão identificados e a configuração do atributo no sistema PPM será expandida em conformidade. Se tiver habilitado a opção **automapping** na configuração da fonte de dados usada, os novos atributos identificados serão transferidos para os objetos especificados na definição de fragmento.

-extractattributes <File name>

Os atributos contidos nos arquivos de importação XML que não são conhecidos no sistema PPM são identificados e gravados no arquivo especificado. O arquivo gerado é compatível com a DTD e pode ser importado usando o programa de linha de comando **runppmconfig** para expandir a configuração de atributo do PPM. Se usar o argumento **-extractattributes** sem a opção **-autoexentattributes**, nenhuns dados de importação XML serão importados.

-extractmergeattributes <File name>

Esse argumento grava atributos transferidos para instâncias de processo usando a opção **automapping** no arquivo especificado. Nenhum arquivo de importação XML é importado.

Opções de log

Esses argumentos podem ser usados para limitar a saída de log. As mensagens de erro que resultam em cancelamento do programa sempre serão exibidas na console.

3.6. Importar várias fontes de dados

Podem ser importadas várias fontes de dados simultaneamente usando o argumento **-datasourcelist <file>** (ver o capítulo programa de linha de comando runxmlimport [56]). Para importar várias fontes de dados, está disponível um arquivo de configuração, no qual pode especificar uma lista de fontes de dados. Durante uma importação XML, os dados das fontes de dados especificadas no arquivo de configuração são importados consecutivamente, como se a importação XML fosse chamada várias vezes consecutivamente usando o argumento -**datasource <file>**. A sequência de importação da fonte de dados é especificada no arquivo de configuração.

O arquivo de configuração deve corresponder à **datasourcelist.dtd**, que terá a seguinte aparência.

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

Tem de especificar um ID para cada fonte de dados, ou seja, o nome da fonte de dados também usada no PPM CTK, o tipo de fonte de dados e o caminho para o arquivo da fonte de dados.

Um arquivo XML pode ter a seguinte aparência.

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

Comportamento de erro

Se a importação XML for chamada por meio de um arquivo de configuração válido que não contenha nenhuma fonte de dados, a importação será terminada sem exibir uma mensagem de erro.

Se a importação XML for chamada por meio de um arquivo de configuração contendo várias fontes de dados, e se ocorrer um erro durante a importação de uma fonte de dados que leve ao cancelamento

dessa importação, a importação continuará com o arquivo de fonte de dados seguinte. Isso significa que o cancelamento da importação de uma fonte de dados não leva ao cancelamento de toda a importação.

Se ocorrer um erro durante a importação de pelo menos uma fonte de dados de um arquivo de configuração, que até agora conduzia a um estado de erro de saída (ou seja, "-1") durante a importação de fontes de dados individuais, a importação que use esse arquivo de configuração também retornará este estado de erro de saída.

3.7. Reimportar os mesmos dados

No sistema PPM, a importação repetida dos mesmos dados de origem sempre leva ao mesmo resultado exclusivo.

3.7.1. Formato gráfico

Ao reimportar os dados da instância usando o formato gráfico, as instâncias do processo são identificadas exclusivamente pelo atributo **AT_EPK_KEY**. Instâncias de processo existentes são substituídas por instâncias de processo importadas com o mesmo valor do atributo.

3.7.2. Formato do evento de sistema

Ao reimportar os dados da instância usando o formato de evento de sistema, os objetos idênticos são substituídos automaticamente. Objetos idênticos são identificados por uma chave de objeto interna idêntica, que é calculada durante a importação e é armazenada no atributo de objeto **AT_INTERNAL_OBJECT_KEY**. As regras para cálculo das chaves de objeto são especificadas no elemento XML **internalobjectkeyrules**, no arquivo **keyrules.xml**. Caso os objetos sejam idênticos, o último objeto importado é transferido para a instância de processo.

Exemplo

O extrato de arquivo abaixo define regras para o cálculo da chave de objeto para funções e eventos.
Os eventos são identificados como idênticos se os valores dos atributos **AT_OBJNAME_INTERN** e **AT_MERGE_KEY_1** forem correspondentes. As funções são identificadas como idênticas se os valores dos atributos **AT_OBJNAME_INTERN** e **AT_END_TIME** forem correspondentes.

```
...
<internalobjectkeyrule>
<refobjects>
<refobject objecttype="OT_EVT"/>
</refobjects>
<keyparts>
<keypart attributetype="AT_OBJNAME_INTERN"/>
<keypart attributetype="AT_MERGE_KEY_1"/>
</keyparts>
</internalobjectkeyrule>
<internalobjectkeyrule>
<refobjects>
<refobject objecttype="OT_FUNC"/>
</refobjects>
```

```
<keyparts>
<keypart attributetype="AT_OBJNAME_INTERN"/>
<keypart attributetype="AT_END_TIME"/>
</keyparts>
</internalobjectkeyrule>
...
```

Atenção
Defina regras chave de objetos internos, que garantem que chaves de objeto idênticas sejam calculadas para objetos idênticos na instância de processo, para que esses objetos sejam substituídos se forem reimportados.

4. Importação de dados independentes da instância de processo

Este capítulo descreve a interface de importação para importar a medida independente da instância de processo e valores de dimensão.

Os dados independentes de instância de processo levam em consideração aspectos que não são orientados para processos, por exemplo, custos comerciais fixos, satisfação do cliente ou medidas financeiras, os quais caracterizam a visão financeira de uma empresa.

Medidas independentes da instância de processo

Os valores de medidas independentes da instância de processo (ver o capítulo Medidas independentes da instância de processo [63]) são valores de medida que não são calculados com base em dados da instância de processo. São importados diretamente com os valores de dimensão referenciados e sem referência aos dados da instância de processo.

As medidas independentes da instância de processo podem ser usadas como base para medidas definidas pelo usuário e, portanto, combinadas com medidas dependentes da instância de processo.

Dados de dimensão independentes da instância de processo

Pode importar com antecedência para um sistema PPM valores de chaves e descrições para dimensões de texto de um nível, dois níveis e nível n, ou seja, antes da importação de dados real (ver o capítulo Valores de dimensão [75]).

4.1. Medidas independentes da instância de processo

Os capítulos seguintes descrevem a configuração de formatos de dados de importação (XML, CSV, XLS) para medidas independentes da instância de processo, bem como a importação e exportação de valores de medida independente da instância de processo.

Um pré-requisito para isso é a existência das medidas independentes da instância de processo correspondentes no sistema PPM. Se quiser definir e registrar medidas independentes da instância de processo, consulte a referência técnica **Personalização do PPM** para mais instruções.

4.1.1. Formatos de importação de dados

Pode importar dados de importação de medidas independentes da instância de processo nos seguintes formatos:

- XML (padrão)
- CSV

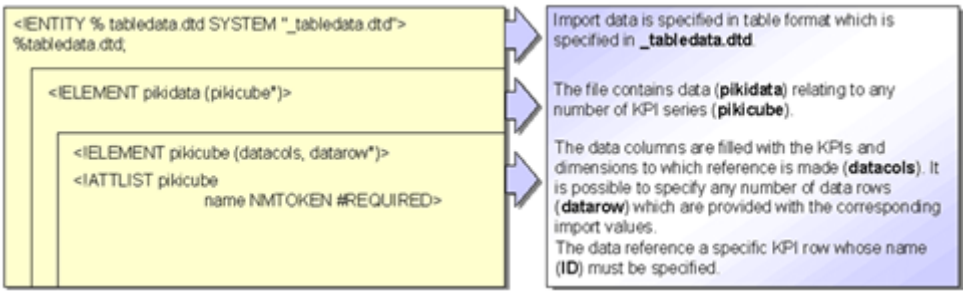
- XLS

Independentemente do formato de dados de importação, deve sempre especificar todas as dimensões chave (atributo **iskeydimension="TRUE"** do elemento **refdim** na definição da série de dados, ver a referência técnica **Personalização do PPM**) e pelo menos uma das medidas independentes da instância de processo da série de dados. Opcionalmente, pode especificar outras medidas independentes da instância de processo ou dimensões não-chave do cubo PIKI relevante na estrutura de dados de importação.

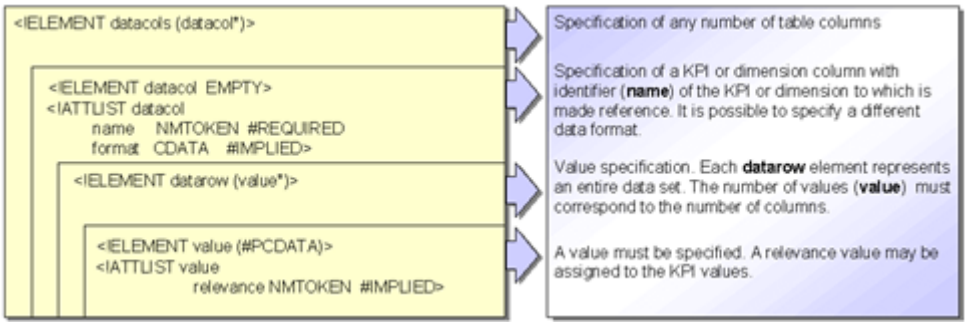
4.1.1.1. Formato XML

O formato de importação de dados XML para medidas independentes da instância de processo é predefinido pelas seguintes definições do tipo de documento:

DTD **pikidata.dtd** (referência de séries de dados como dados de importação)



DTD **_tabledata.dtd** (formato de tabela para importar dados independentes da instância de processo):



Assim, os dados de importação de medidas independentes da instância de processo no formato XML têm a seguinte estrutura geral:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE pikidata SYSTEM "pikidata.dtd">
<pikidata>
  <pikicube name="...">
    <datacols>
      <datacol name="..."/>
      ...
    </datacols>
    <datarow>
      <value relevance="...">...</value>
       ...
    </datarow>
```

```
    ...
  </pkicube>
  ...
</pikidata>
```

pikidata ELEMENT e ATTLIST	Descrição
pikidata	Lista de dados de importação em qualquer número de séries de dados independentes da instância de processo (cubo PIKI)
pikicube	Série de dados para a qual os dados especificados têm de ser importados
name	Identificador da série de dados. Deve corresponder ao nome do cubo PIKI (atributo XML **pikicube name**) especificado na configuração de medidas.

datacols ELEMENT e ATTLIST	Descrição
datacols	Estrutura de dados de importação (colunas da tabela da série de dados)
datacol	Especificação de uma coluna da tabela. É especificada uma coluna para cada medida independente da instância de processo e para cada dimensão referenciada relevante.
name	Identificador da coluna da tabela. Deve corresponder aos nomes das medidas independentes da instância de processo (**pikidef name**) especificadas na configuração de medidas e aos nomes das dimensões referenciadas (**refdim name**).
format	Formato de importação opcional para valores da coluna. Os formatos suportados são números de ponto flutuante, valores de hora e hora do dia.
datarow	Dados de importação específicos para a série de dados na forma de uma linha de dados. A sequência de valores de importação (elementos **value**) deve corresponder à sequência da coluna (**datacols**) para que cada critério da série de dados (medida independente da instância de processo ou dimensão referenciada) possa receber um valor exclusivo.
value	Valor da coluna da linha de dados a ser importado (valor de uma dimensão referenciada ou de uma medida independente da instância de processo)
relevância (opcional)	Valor de relevância referente ao valor correspondente de uma medida independente da instância de processo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE pikidata SYSTEM "pikidata.dtd">
<pikidata>
<pikicube name="PIKICUBE_COSTS">
<datacols>
<datacol name="OVERHEAD_COSTS"/>
<datacol name="PROCESSTYPE"/>
<datacol name="TIME" format="MMM yyyy"/>
<datacol name="MATERIAL"/>
</datacols>
<datarow>
<value relevance="100">1000 EUR</value>
<value>Order processing\Car industry</value>
<value>June 2000</value>
<value>NaviSet B6{Navigation system}</value>
</datarow>
<datarow>
...
</datarow>
```

```
...
</pikicube>
...
</pikidata>
```

Especificação da estrutura de dados de importação (elementos XML datacols e datacol)

O elemento XML **datacols** especifica os identificadores das medidas independentes da instância de processo e as dimensões referenciadas (**datacol name="..."**) a que a importação de dados se refere para as linhas de dados relevantes. A estrutura de dados de importação é especificada em um formato de tabela simples no qual cada critério da série de dados (medida independente da instância de processo ou dimensão referenciada) representa uma coluna da tabela.

Importação de dados

Os valores de importação reais são extraídos dos elementos XML **datarow**. Cada um desses elementos representa uma linha de dados que é importada. Cada elemento **value** especificado corresponde ao elemento **datacol** correspondente da estrutura de dados de importação especificada.

A sequência dos valores (elemento XML **value**) em uma linha de dados (elemento XML **datarow**) deve corresponder à sequência de colunas (no elemento XML **datacols**) na estrutura de dados de importação.

Exemplo 31.

No atributo XML **format="MMM yyyy"**, é especificado o formato de dados para o valor a ser importado para a dimensão referenciada **TIME**. A informação do formato é opcional.

Especificar um texto descritivo entre parênteses **{ }** significa que os valores para a dimensão de um nível **MATERIAL** são esperados no formato **<Identifier>{<Descriptive text>}**. A especificação de um texto descritivo é opcional, pois somente o identificador é crucial para a referência da dimensão.

O conjunto de dados de exemplo tem um valor de relevância para a medida independente da instância de processo **OVERHEAD_COSTS**. Se não especificar um valor de relevância, **relevance="1"** passa a ser o valor padrão.

Por padrão, se nenhuma unidade for especificada para um valor de dados, será usada a unidade base para o tipo de dados de atributo na qual a medida independente da instância de processo ou a dimensão referenciada se baseia.

Exemplo 32.

Existem dois formatos diferentes para importar valores de dimensões de dois níveis:

- Definição do valor em uma linha:

Exemplo 33.

Os valores são separados por uma barra invertida (****).

```
  <datacols>
    ...
    <datacol name="PRINCIPAL"/>
    ...
  </datacols>
```

```
  ...
  <datarow>
    ...
    <value>DE{Germany}\0000000001{Becker}</value>
    ...
  </datarow>
  ...
```

- Definição do valor em duas linhas:

Exemplo 34.

Os valores são especificados em dois elementos XML que se seguem diretamente.

```
...
  <datacols>
    ...
    <datacol name="PRINCIPAL"/>
    <datacol name="PRINCIPAL"/>
    ...
  </datacols>
  <datarow>
    ...
    <value>DE{Germany}</value>
    <value>0000000001{Becker}</value>
    ...
  </datarow>
  ...
```

4.1.1.2. Formato CSV

Os seguintes recursos especiais aplicam-se para importação no formato CSV:

- Um arquivo CSV apenas pode conter dados para uma série de dados.
- Os valores especificados em um **Arquivo CSV** não podem conter o separador de dados especificado. O separador de dados é especificado usando a opção **-csvchar "<Character>"** no programa de linha de comando **runpikidata**.

Exemplo 35.

Este exemplo mostra a importação do formato **CSV** com números de linhas.

```
1 PIKICUBE_COSTS
2 OVERHEAD_COSTS(.,);OVERHEAD_COSTSNUM;TIME(MM.yyyy);
```

```
1 PIKICUBE_COSTS
2 OVERHEAD_COSTS(.,);OVERHEAD_COSTSNUM;TIME(MM.yyyy);PROCESSTYPE;PROCESSTYPE
3 1.000,00 EUR;100;05.2000;Order processing;Car industry
4 1.020,00 EUR;;06.2000;Order processing;Car industry
...
```

Motivo

- Linha 1: Nome da linha de dados (por exemplo, **PIKICUBE_COSTS**)
- Linha 2: Especificação da estrutura de dados de importação

  As colunas individuais da tabela da série de dados são separadas por ponto e vírgula. As informações de formato podem ser especificadas após o nome de coluna entre parênteses.

  A especificação opcional da relevância deve seguir diretamente a medida relevante. O nome da coluna de relevância é composto pelo nome da medida complementado pela cadeia de caracteres **NUM** (por exemplo, **OVERHEAD_COSTSNUM**).
- A partir da linha 3: Importar dados

Cada linha representa uma série de dados a ser importada. Os valores individuais são separados pelo separador especificado.

Para os valores de importação da medida independente da instância de processo **OVERHEAD_COSTS**, um valor de relevância (**OVERHEAD_COSTSNUM**) de 100 (**1.000,00 EUR;100**) é especificado na primeira linha de dados (linha 3), a segunda linha de dados (linha 4) não tem valor de relevância (**1.020,00 EUR;;**). Nesse caso, o valor de relevância é automaticamente definido para **1**.

Os valores das dimensões de dois níveis (no exemplo: **PROCESSTYPE**) são importados usando duas colunas separadas e sucessivas, com o mesmo nome. A primeira coluna contém o valor aproximado, a segunda o valor detalhado.

O formato opcional **MM.aaaa** para a dimensão referenciada **Time** consiste em um número de mês inteiro seguido por um número de ano com quatro dígitos. Os valores de campo individuais são separados por um ponto (ver **Formatos de dados [43]**).

4.1.1.3. Formato XLS

Pode importar dados para linhas de dados independentes da instância de processo de arquivos, do Excel para o sistema PPM. Os seguintes recursos especiais aplicam-se para importação no formato XLS:

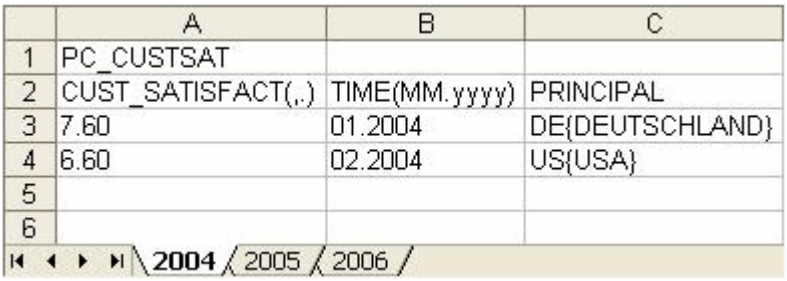
- Um arquivo de Excel apenas pode conter dados para uma série de dados em cada planilha. São importadas todas as planilhas do arquivo Excel.
- Quando importar uma planilha, toda a área é sempre importada. As linhas e colunas sem conteúdo são ignoradas.
- Os nomes das planilhas a serem importadas podem ser especificadas com o parâmetro da linha de comando opcional **-sheet**. Se o nome contém espaços, terá de colocar o nome entre aspas. Para importar várias planilhas de um arquivo, insira os nomes separados por espaços.
- O formato de célula para as colunas de dimensão tem de ter o tipo de dados **Text**. O formato das células nas colunas de medida tem de ser do tipo **Text** ou **Number**. As regras de formatação (por exemplo, separadores para decimais e milhares) só podem ser especificadas para valores de medida em formato de texto.

Posicionamento implícito

Em princípio, o formato dos dados corresponde ao formato CSV. As informações da planilha são esperadas na seguinte ordem:

- Célula A1: Nome da série de dados
- Célula A2 e células seguintes (apenas linha 2): Definição da estrutura de dados

- Célula A3 e células seguintes (todas as linhas): Valores de dados



	A	B	C
1	PC_CUSTSAT		
2	CUST_SATISFACT(,)	TIME(MM.yyyy)	PRINCIPAL
3	7.60	01.2004	DE{DEUTSCHLAND}
4	6.60	02.2004	US{USA}
5			
6			

2004 / 2005 / 2006 /

Posicionamento explícito

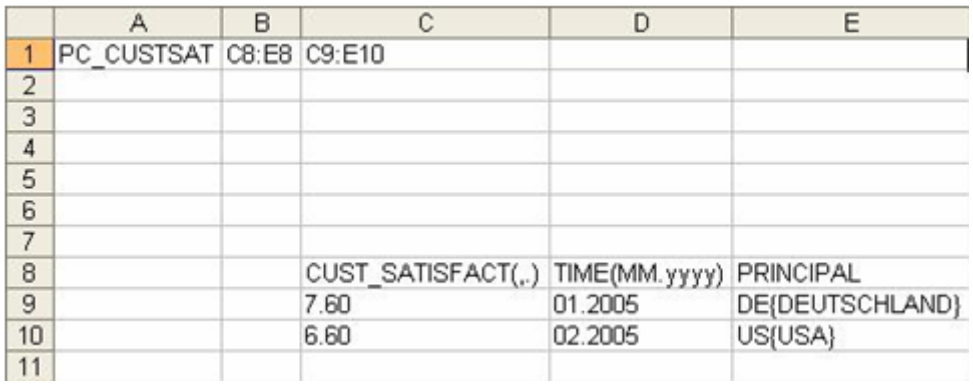
O nome da série de dados é especificado na célula A1. Na célula B1, é especificada a área que contém as informações da estrutura de dados de importação e a célula C1 especifica a área que contém os dados de importação reais. O posicionamento tem o formato geral

<Position of start cell>[:<Position of end cell>]

Se apenas especificar a posição da célula inicial, por exemplo, C8, também serão importados os dados de todas as colunas subsequentes. Neste caso, a especificação corresponde a uma área infinita da tabela.

Se especificar o mesmo número de linha ao especificar a posição do intervalo de dados de importação a serem extraídos, por exemplo,C9:E9, as colunas especificadas são extraídas até ao final da planilha e um número ilimitado de linhas de dados é, portanto, importado. Para importar um intervalo limitado de valores, deve especificar a primeira e a última linha de dados, por exemplo, C9:E10.

Exemplo 36.



	A	B	C	D	E
1	PC_CUSTSAT	C8:E8	C9:E10		
2					
3					
4					
5					
6					
7					
8			CUST_SATISFACT(,)	TIME(MM.yyyy)	PRINCIPAL
9			7.60	01.2005	DE{DEUTSCHLAND}
10			6.60	02.2005	US{USA}
11					

A estrutura de dados de importação da série de dados independente da instância do processo **PC_CUSTSAT** é especificada nas células de C8 a E8 e os valores reais a serem importados são especificados nas células de C9 a E10.

Pode usar a área restante da planilha do Excel para comentários, etc. Essa área é ignorada durante a importação.

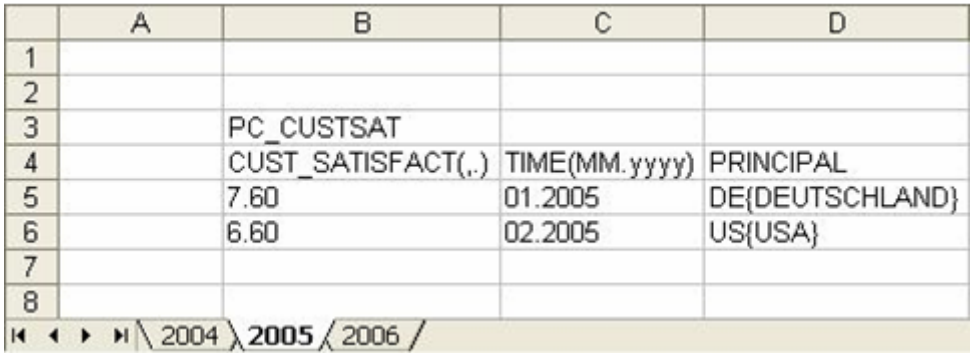
> **Atenção**
> Para usar o posicionamento explícito, tem de respeitar a sequência básica: Deve especificar a linha para a estrutura de dados de importação antes das linhas que contêm os valores de dados.

Ponto inicial relativo

O nome da série de dados é especificado em uma posição diferente da posição implícita (célula A1). A posição da célula na planilha é especificada pelo parâmetro **sheet** e pelo nome da planilha. A sintaxe é

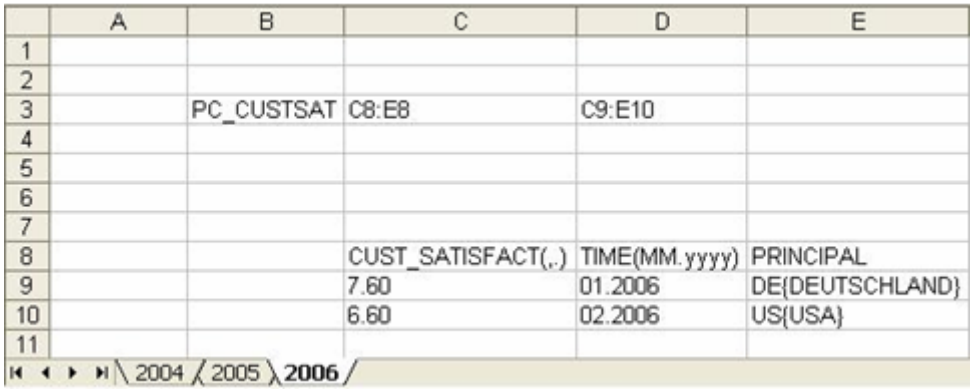
-sheet <Worksheet name>:<Cell>.

Exemplo 37.



	A	B	C	D
1				
2				
3		PC_CUSTSAT		
4		CUST_SATISFACT(,.)	TIME(MM.yyyy)	PRINCIPAL
5		7.60	01.2005	DE{DEUTSCHLAND}
6		6.60	02.2005	US{USA}
7				
8				

2004 **2005** 2006



	A	B	C	D	E
1					
2					
3		PC_CUSTSAT	C8:E8	C9:E10	
4					
5					
6					
7					
8			CUST_SATISFACT(,.)	TIME(MM.yyyy)	PRINCIPAL
9			7.60	01.2006	DE{DEUTSCHLAND}
10			6.60	02.2006	US{USA}
11					

2004 2005 **2006**

Em ambos os exemplos, os dados são importados usando o comando seguinte:

```
runpikidata -user <username> -password <password> -mode import -format XLS
-file excelpikidata.xls -sheet 2006:B3
```

4.1.2. Reimportação de dados

A importação de dados para séries de dados independentes da instância de processo **adiciona** novas linhas de dados e **substitui** linhas de dados existentes alterados.

Se a importação de dados for repetida, as linhas de dados existentes serão substituídas por valores alterados e atualizados de medidas independentes da instância de processo e dimensões referenciadas da série de dados, e novas linhas de dados serão adicionadas à série de dados relevante. O sistema PPM reconhece linhas de dados existentes por meio de valores de dimensão chave de uma série de dados.

As dimensões chave são determinadas pelo atributo **iskeydimension="TRUE"** do elemento **refdim** na definição da série de dados. Para obter mais informações, consulte o documento **Personalização do PPM**.

Exemplo 38.

A linha de dados seguinte já foi importada para uma série de dados existente:

Exemplo 39.	Exemplo 40.	Exemplo 41.	Exemplo 42.	Exemplo 43.	Exemplo 44.
D_COUNTRY*	D_PLANT*	D_DEPARTMENT*	D_RECORDED BY	SALES	COSTS
Exemplo 45.	**Exemplo 46.**	**Exemplo 47.**	**Exemplo 48.**	**Exemplo 49.**	
Alemanha	Hamburg	42	Silva	400000	

Exemplo 50.

As dimensões de referência marcadas com * são as dimensões chave da série de dados. Assim, a combinação de valores **Germany; Hamburg; 42** é o identificador da linha de dados exibida. As duas medidas independentes da instância de processo **SALES** e **COSTS** estão configuradas para a série de dados.

Se a importação de dados for repetida, por exemplo, com os valores **Germany; Hamburg; 42; Huber;;280000**, o sistema PPM reconhece que a linha de dados já existe e substitui o valor da dimensão **D_RECORDED BY** (**Schmidt**) pelo valor alterado (**Huber**), adicionando um valor para a medida independente da instância de processo **COSTS**.

Exemplo 51.	Exemplo 52.	Exemplo 53.	Exemplo 54.	Exemplo 55.	Exemplo 56.
D_COUNTRY*	D_PLANT*	D_DEPARTMENT*	D_RECORDED BY	SALES	COSTS
Exemplo 57.	**Exemplo 58.**	**Exemplo 59.**	**Exemplo 60.**	**Exemplo 61.**	**Exemplo 62.**
Alemanha	Hamburg	42	Huber	400000	280000

Exemplo 63.

O valor da medida independente da instância de processo **SALES** permanece inalterado e é retido porque nenhum novo valor foi importado durante a importação de dados repetidos.

4.1.3. Exportar valores da série de dados independentes da instância de processo

Os valores das medidas independentes da instância de processo podem ser exportados em formato XML usando o programa de linha de comando **runpikidata**. Para isso, use o argumento da linha de comando **-mode export** (ver o capítulo programa de linha de comando runpikidata [72]).

Os dados em todas as séries de dados do cliente especificado são gravados no arquivo especificado usando o argumento **-file**. Para exportar séries de dados específicas, especifique os nomes das séries de dados que deseja exportar, separados por vírgulas, usando o argumento **-pikicube**.

O usuário do PPM que realiza a operação requer o direito de função **Importação de dados**.

```
runpikidata -user system -password manager -client umg_en -mode export -file
pikidata -pikicube
PC_SALES_REVENUES
```

4.1.4. Excluir valores da série de dados independentes da instância de processo

Pode excluir séries de dados independentes da instância de processo no sistema PPM usando o programa de linha de comando **runpikidata**. Para isso, use os argumentos de linha de comando **-mode delete** e **-pikicube <cube name>[,<cube name>,...]** para especificar as séries de dados das quais deseja excluir valores.

Por padrão, todos os valores dos cubos PIKI especificados são excluídos.

Pode limitar as linhas de dados de uma série de dados a serem excluídas, especificando um conjunto que contenha um filtro em uma dimensão referenciada da série de dados, por exemplo. Para isso, insira o nome do arquivo do conjunto de parâmetros usando o parâmetro **-ps**.

Se o conjunto de parâmetros de exclusão contém filtros de dimensão que não são incluídos como dimensões referenciadas na definição da série de dados especificada, é emitida uma mensagem de erro o programa é executado. O usuário do PPM que realiza a operação requer o direito de função **Importação de dados**.

Exemplo 64. *[en-us]*

Quer excluir os valores de importação da série de dados com o identificador interno **PC_SALES** (cubo PIKI Vendas) durante o primeiro trimestre de 2009.

Na análise, defina um conjunto de parâmetros de exclusão com o filtro de tempo correspondente e exporte-o para um arquivo XML local. Em seguida, chame o programa de linha de comando **runpikidata** da seguinte forma:

```
runpikidata -client <ppmclient> -user <username> -password <password> -pikicube
PC_SALES -mode delete -ps <deleteparamset>.xml
```

4.1.5. programa de linha de comando runpikidata

A importação de dados ou exportação de séries de dados independentes da instância de processo é executada usando o programa de linha de comando **runpikidata**:

```
runpikidata -user <user name> -password <password>

  [-client <name>]
-mode (import|export|delete)
[-compatible4]
-file <file1>[,<file2>...]
[-format {XML|CSV|XLS} [[-csvchar "<character>"] -encoding "<encoding name>"]]
-pikicube <cubename>[,<cubename>...]
-ps <filename>
[-sheet <name>[:<cell>] [<name>[:<cell>]]]
[-version]
[-language <ISO code>][ protocoloptions ]
[-recoveryfile {yes|no}]

-user <user name> Name of user
-password <password> User password
-client <name> Name of client If

  no client is specified
the server of the

  default client will be used.
-language <ISO code> Import language
-mode (import|export|delete) Import, export,
or deletion of data
-compatible4 Import of dimension values as
```

```
  in PPM 4.x without checking the

  refinement level
-file <file1>[,<file2>...] Import files (-mode import)

  or export files

  (-mode export).

  ZIP files are supported.
-format {XML|CSV|XLS} Data format for the import

  (default: XML)
-csvchar "<character>" Data separator for the

  "CSV" import format

  (Default: comma)
-encoding "<encodingname>" Encoding of the CSV file
-pikicube <cubename> Data series from which data

  [,<cubename>...] is to be deleted or

  exported (-mode delete|export)
-ps <filename> Paramset specifying the

  data to be deleted (-mode delete)
-sheet <name>[:<cell>]> Specifies Excel sheets for the

  "XLS" import format. The

  cell specifies the position from

  which the data is to be

  imported.
-recoveryfile {yes|no} Creates after successful

  import the new

  process analysis-relevant

  analysis server recovery files

  Default: yes)

-version Version number of the application

  and the database database schema

protocoloptions can consist of the following instructions:
-protocolfile <file name> Log to file

  <file name<
-information {yes|no} Logging of
```

```
  information
-warning {yes|no} Logging of warnings
-error {yes|no} Logging of errors
```

-version

A versão de software do PPM e o esquema do banco de dados são emitidos para a console. Outros argumentos são ignorados.

-user <user name> -password <password>

Com este parâmetro, pode especificar o nome do usuário e a senha do usuário do PPM que está a executar a importação. O usuário deve ter o direito de função **Importação de dados**.

-client <client name>

Com este parâmetro, especifica o cliente PPM em cujo esquema do banco de dados os fragmentos de processo importados devem ser salvos. Se não usar esta opção, será usado o cliente padrão.

-mode import|export|delete)

Por exemplo, com este parâmetro especifica que quer importar um arquivo de origem (ou vários arquivos de origem) (**-mode import -file <file1>[,<file2>...]**), incluindo dados do formato de importação selecionado (**-format {XML|CSV|XLS}**) ou exportar (**-mode export -pikicube <cubename> -file <filename>**) dados de uma série de dados existente no sistema PPM para um arquivo XML, ou que quer excluir totalmente (**-mode delete -pikicube <cubename>[,<cubename>...]**), ou em parte, (**-mode delete -pikicube <cubename> -ps <filename>**) dados da série de dados especificada.

-compatible4

Com o parâmetro opcional, especifica que quando importa valores de dimensões de texto referenciadas, de vários níveis (**refdim**), o nível de refinamento (**refinement**) dos dados de importação não será considerado, tal como acontece nas versões 4.x do PPM. Isso é necessário se existirem valores de dimensão de dados de importação da medida independente da instância de processo para o PPM a partir da versão 5, com um nível de refinamento diferente daquele especificado na definição da série de dados (consulte a referência técnica **Personalização do PPM**).

-file <file1>[,<file>,…]

Com este parâmetro, especifica o(s) arquivo(s) de importação e o caminho para o(s) arquivo(s) de importação. O arquivo de origem pode ser um arquivo ZIP, que contém um ou mais arquivos de importação com o mesmo formato de dados.

-format {XML|CSV|XLS}

Com este parâmetro pode especificar o formato de importação de dados usado.

-csvchar <character>

Com esse parâmetro, especifica o separador para valores do campo de dados no formato de importação CSV (o valor padrão é a vírgula).

-encoding "<encodingname>"

Com este parâmetro, especifica o nome da codificação CSV que quer usar. O parâmetro **-encoding** é obrigatório para o formato CSV.

-sheet <name>[:<cell>]>

Com esse parâmetro, especifica o nome da planilha de Excel a partir da qual quer importar os dados. Pode especificar os nomes de várias planilhas, separadas por espaços. O argumento **<cell>** pode ser usado para cada planilha para especificar a posição inicial relativa (célula contendo o nome da série de dados) da qual os dados serão importados.

-**recoveryfile {yes|no}**

Com esse parâmetro, especifica se os arquivos de recuperação do servidor de análise relevantes para a análise de processo devem ser criados após a importação ou a exclusão de valores de séries de dados independentes da instância de processo. O valor padrão é **yes**.

4.2. Valores de dimensão

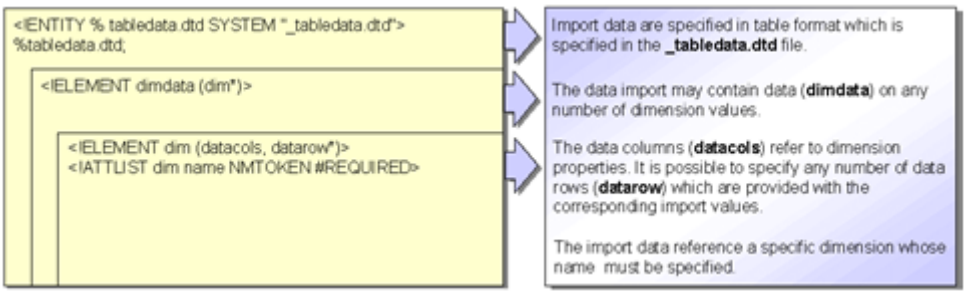
Para dimensões de texto de um, dois e nível n, é possível importar valores de dimensão (especialmente descrições de nível abrangentes) para o sistema PPM antes da importação real para o PPM. Os valores de dimensão consistem em um ID obrigatório e uma descrição opcional.

Vantagens:

- Pode criar definições de valor planejado no PPM antes que as instâncias de processo sejam importadas.
- Pode reduzir significativamente o volume de dados das instâncias de processo a serem importadas durante a importação real do PPM, não especificando as descrições de nível já importadas.

4.2.1. Formato XML

O formato do arquivo XML é especificado pela seguinte DTD:



Atenção
O nome das colunas de dados para a chave e descrição de um nível de dimensão deve começar com o prefixo **LEVEL** e terminar com o sufixo ID (chave) ou DESC (descrição). Além disso, deve usar números consecutivos começando com **1** para especificar o nível. Caso contrário, a importação será cancelada e uma mensagem de erro será exibida.

A tabela a seguir ilustra a associação de colunas de dados aos valores de dimensão (**<n>** representa a numeração consecutiva do nível):

configuração dimdata	Configuração da dimensão	Exemplo
LEVEL<n>_ID	ID do nível de dimensão n-th	**LEVEL5_ID**

configuração dimdata	Configuração da dimensão	Exemplo
LEVEL<n>_DESC	Descrição do nível de dimensão n-th	**LEVEL5_DESC**

Exemplo 65.

Para a dimensão de dois níveis **Sold-to party** (**PRINCIPAL**), são importados três valores:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE dimdata SYSTEM "dimdata.dtd">
<dimdata>
<dim name="PRINCIPAL">
<datacols>
<datacol name="LEVEL1_ID"/>
<datacol name="LEVEL1_DESC"/>
<datacol name="LEVEL2_ID"/>
<datacol name="LEVEL2_DESC"/>
</datacols>
<datarow>
<value>DE</value>
<value>Germany</value>
<value>0000000003</value>
<value>Becker</value>
</datarow>
<datarow>
<value>FR</value>
<value>France</value>
<value>0000000092</value>
<value>Leclerc</value>
</datarow>
<datarow>
<value>EN</value>
<value>United Kingdom</value>
<value>0000000027</value>
<value>Crichton</value>
</datarow>
</dim>
</dimdata>
```

Atenção
Quando importar valores para dimensões de texto de vários níveis, tem sempre de indicar a chave do primeiro nível da dimensão (**LEVEL1_ID**).

4.2.2. Formato CSV

Os seguintes recursos especiais aplicam-se para importação no formato CSV:

- Um arquivo CSV apenas pode conter dados para uma dimensão.

- Os valores de dimensão especificados em um **Arquivo CSV** não podem conter o separador de dados especificado. O separador de dados é especificado usando a opção **-csvchar "<Character>"** no programa de linha de comando **rundimdata**.

Exemplo 66.

Este exemplo mostra a importação do formato **CSV** com números de linhas.

```
1 PRINCIPAL
2 LEVEL1_ID;LEVEL1_DESC;LEVEL2_ID;LEVEL2_DESC
3 DE;Germany;0000000003;Becker
4 FR;France;0000000092;Leclerc
5 UK;United Kingdom;0000000027;Crichton
...1 PRINCIPAL
```

Motivo

- Linha 1: Identificadores de dimensão
- Linha 2: Definição da estrutura de dados

  As colunas individuais são separadas por um ponto e vírgula. Os nomes das colunas são palavras-chave que não podem ser alteradas.
- A partir da linha 3: Valores de dados

Cada linha contém um registro. Os valores individuais são separados pelo separador especificado. Os valores especificados devem corresponder aos tipos de dados especificados na configuração do tipo de atributo. Da esquerda para a direita, as entradas de coluna representam: Código de país, Nome de país, ID principal, Nome principal

4.2.3. Valores padrão e de substituição

Se você especificou valores de substituição ou padrão para a chave ou descrição de um nível de dimensão na configuração de medidas e quer usar esses valores para importar com **rundimdata**, terá de deixar vazios os elementos **value** correspondentes no arquivo de importação XML ou valores de coluna no arquivo CSV.

Para obter mais informações sobre o uso de valores padrão e de substituição em dimensões de texto, consulte a referência técnica **Personalização PPM**.

4.2.4. Reimportação de dados

Em princípio, a importação de valores de dimensão independentes da instância de processo (**rundimdata -mode import**) é aditiva, ou seja, os valores de dimensão que não existem são adicionados ao sistema PPM (uma importação aditiva só não é possível para a dimensão de texto de dois níveis **PROCESSTYPE**).

Se especificar, além disso, a opção de linha de comando **-overwrite**, as descrições existentes dos níveis de dimensão são substituídas por descrições importadas e alteradas.

Se quiser excluir as descrições de níveis de dimensão específicos, importe descrições vazias para esses níveis de dimensão no modo de substituição usando **-mode import -overwrite**.

4.2.5. Excluir valores de dimensão

Para excluir valores de dimensão, use a opção **-replace** no programa de linha de comando **rundimdata**. Isso marca para exclusão todos os valores de dimensão que não estão incluídos no arquivo de importação. Para excluir os dados permanentemente, o servidor de análise deve ser completamente reinicializado. Portanto, os arquivos de recuperação do servidor de análise relevantes para a análise de processo são recriados por padrão após a importação. Após a importação, o programa de linha de comando gera uma mensagem correspondente.

Se quiser excluir todos os dados de dimensão de uma dimensão, importe um arquivo semanticamente vazio.

A opção **-replace**, para substituir ou excluir valores de dimensão, implica a opção **-overwrite**. Por isso, não pode especificar a opção **-overwrite** na linha de comando quando usar a opção **-replace**.

4.2.6. programa de linha de comando rundimdata

Os valores de dimensão independentes da instância de processo no formato XML e CSV são importados usando o programa de linha de comando **rundimdata**:

```
rundimdata -user <username> -password <password> [-client <name>]
-mode import
-file <file1>[,<file2>...]
[-format {XML|CSV} [[-csvchar "<character>"] -encoding "<encoding name>"]]
[-overwrite |-replace]
[-recoveryfile {yes|no}]
[-version]
[-language <ISO code>][ protocoloptions ]

-user <user name> Name of user
-password <password> User password
-client <name> Name of client If no client is specified,
the server of the default client will be used.
-language <ISO code> Import language
-mode import Import of data
-recoveryfile {yes|no} After import into the analysis server
the recovery files relevant for
process analysis are recreated.
-file <file1>[,<file2>...] Import files. ZIP files are supported.
-format {XML|CSV} Data format for the import (default: XML)
-csvchar "<character>" Data separator for "CSV" import format
(default: comma)
-overwrite Overwriting dimension descriptions
-replace Delete existing dimension values that
have been changed by a data import for dimensions
or were supplemented.
-encoding "<encodingname>" Encoding of the CSV file
-version Version number of the application and the
database schema

protocoloptions can consist of the following instructions:
-protocolfile <file name> Logging in the file <file name>
-information {yes|no|default} Logging of information
```

```
-warning {yes|no|default} Logging of warnings
-error {yes|no|default} Logging of errors
```

-version

A versão de software do PPM e o esquema do banco de dados são emitidos para a console. Outros argumentos são ignorados.

-user <user name> -password <password>

Com este parâmetro, pode especificar o nome do usuário e a senha do usuário do PPM que está a executar a importação. O usuário deve ter o direito de função **Importação de dados**.

-client <client name>

Com esse parâmetro, especifica o cliente PPM para o qual deseja salvar os valores de dimensão importados. Se não usar esta opção, será usado o cliente padrão.

-mode import

Com esse parâmetro, especifica que os arquivos de origem devem ser importados com dados independentes da instância de processo.

-file <file1>[,<file2>...]

Com esse parâmetro especifica o(s) arquivo(s) de importação. Os arquivos de origem podem ser arquivos ZIP, que contêm um ou mais arquivo(s) XML com o mesmo formato de dados.

-encoding "<encodingname>"

Com este parâmetro, especifica o nome da codificação CSV que quer usar. O parâmetro **-encoding** é obrigatório para o formato CSV.

-recoveryfile {yes|no}

Com esse parâmetro, especifica se os arquivos de recuperação do servidor de análise relevantes para a análise de processo devem ser criados após a importação de valores de dimensão independentes da instância de processo. O valor padrão é **yes**.

Pode aumentar o desempenho de várias importações de dados de dimensão sequenciais, suprimindo a criação dos arquivos de recuperação do servidor de análise para todas as importações anteriores. Para isso, especifique a opção **-recoveryfile no**. Apenas a última importação cria os arquivos de recuperação do servidor de análise relevantes para a análise do processo se a opção estiver ausente ou se **-recoveryfile yes** foi especificado. Certifique-se de que os arquivos de recuperação do servidor de análise relevantes para a análise do processo são criados em qualquer caso, uma vez que a importação esteja concluída. Pode forçar a criação de todos os arquivos de recuperação usando a chamada de linha de comando **runppmadmin** com a opção **-recoveryfile force**.

-format {XML|CSV}

Com este parâmetro pode especificar o formato de dados usado.

-csvchar "<character>"

Com esse parâmetro, especifica o separador para valores do campo de dados no formato de importação CSV (o valor padrão é a vírgula).

-overwrite

Com esse parâmetro, especifica que as **descrições** existentes dos níveis de dimensão do texto serão substituídos com valores alterados (também vazios). As **chaves** (IDs) de dimensão de texto não podem ser substituídas.

-replace

Com esse parâmetro, pode substituir dados de dimensão já importados. As dimensões não incluídas no arquivo de importação são excluídas. Esse parâmetro implica o parâmetro **-overwrite**. Após a importação, os arquivos de recuperação do servidor de análise relevantes para a análise do processo são recriados. (comportamento padrão)

4.3. Análise de dados

Para além das análises do processo, função e interação, você também pode agora avaliar dados independentes do processo usando a Análise de dados. A análise de dados permite analisar dados abrangentes em formato de tabela, consistindo em várias tabelas vinculadas. Na análise de dados, os critérios de análise, que são as dimensões e medidas, são baseados na estrutura da tabela da base de dados, em que cada coluna da tabela representa um critério de análise.

A documentação técnica **Análise de dados do PPM** fornece informações sobre a importação de dados de análise de dados.

5. Como tratar EPCs grandes

5.1. Importar EPCs grandes

Se o número máximo de funções permitidas por EPC for excedido durante a importação de EPCs grandes, será exibida uma mensagem de erro. No entanto, a importação não é cancelada.

O limite para o número máximo de funções permitidas em um EPC é controlado pelo parâmetro de configuração **KI_EPC_FUNCTION_COUNT_THRESHOLD** definido no arquivo **EpkImport_settings**. O valor padrão para este parâmetro é **500**. Aplica-se se o parâmetro estiver em falta ou tiver um valor de **<= 0**.
Se o parâmetro retornar ao valor padrão, um aviso correspondente será gravado no log.

Assim que existe uma EPC grande na importação XML do banco de dados, a importação do PPM e a importação de processos são bloqueadas no modo **DEFAULT**. Uma mensagem de erro correspondente é emitida e a importação é cancelada. A importação de processos em modo **RECOVERIMPORT** não é afetada.

Para resolver esse problema, pode definir o parâmetro de configuração para um valor adequado. No entanto, isso pode levar o sistema a consumir mais memória principal. Como alternativa, pode excluir a EPC que causa o problema (consulte a seção a seguir).

5.2. Excluir EPCs grandes

Pode excluir uma EPC grande usando o programa de linha de comando **runppmdelete**. Isso se aplica somente se foi definida uma medida de processo para o número de funções em uma instância de processo. Se isso foi feito, é definido um conjunto de parâmetros para o programa de linha de comando.

- O conjunto de parâmetros contém o número de funções **only** como um filtro
- O conjunto de parâmetros contém o número de processos como uma medida.
- O conjunto de parâmetros contém um filtro de tempo.
- O conjunto de parâmetros contém um filtro de tipo de processo ou um filtro de grupo de tipos de processo.

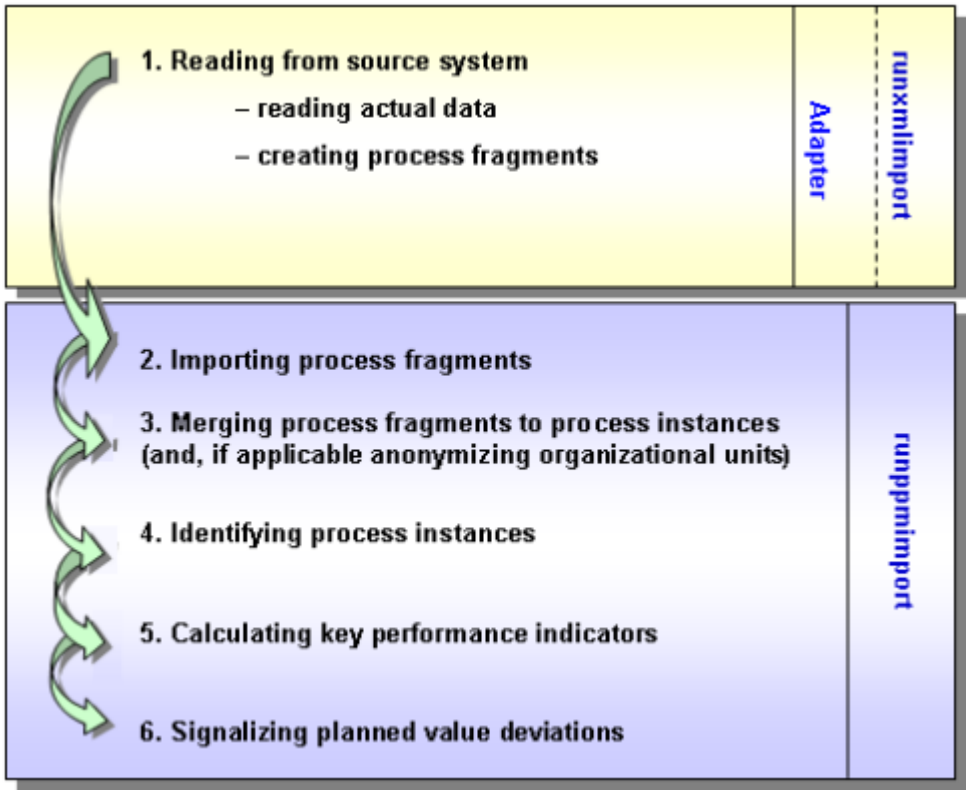
A medida também pode ser definida posteriormente (isto é, quando as importações já estão bloqueadas). Nesse caso, as EPCs têm de ser recalculadas usando o programa de linha de comando **runppmimport –keyindicator new**.

6. Apêndice

6.1. Projeto de um armazém de processos

Este capítulo fornece uma visão geral do projeto de um armazém de processos no PPM e, em seguida, fornece uma breve explicação de cada etapa individual. Se parte do princípio que a instalação e configuração necessárias do sistema PPM foram concluídas.

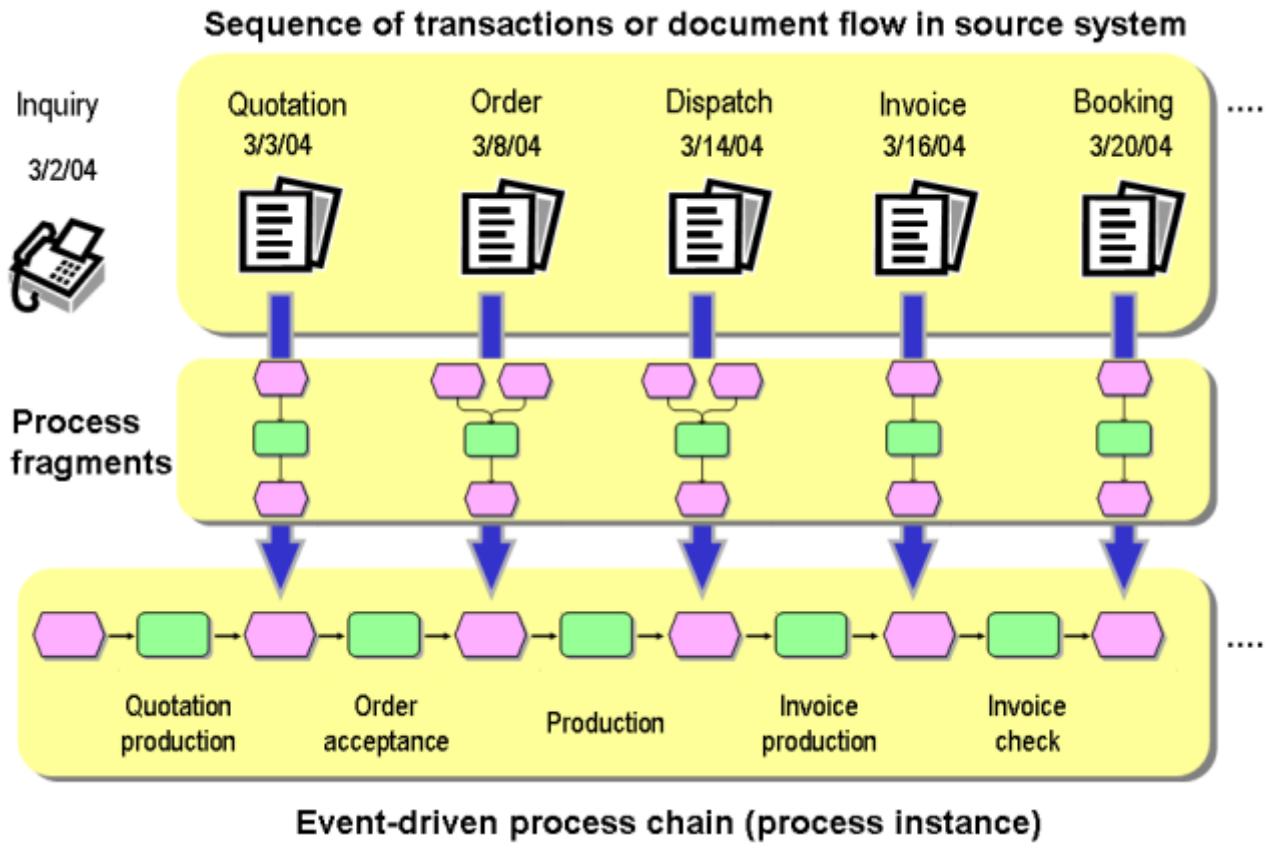
O projeto de um armazém de processos é composto pelas seguintes etapas:



1. Em primeiro lugar, os dados reais são extraídos do sistema de origem e fornecidos na forma de fragmentos de processo para importação.
2. Os fragmentos de processo são importados.
3. A etapa seguinte envolve pesquisar todos os dados importados pelos fragmentos pertencentes a cada evento de negócios e juntá-los em uma instância de processo. Os atributos de objeto são copiados para a instância de processo. Ao mesclar os fragmentos do processo, as informações sobre o usuário real podem ser tornadas anônimas.
4. As instâncias de processo geradas são então classificadas: instâncias de processo do mesmo tipo são associadas a tipos de processos, que por sua vez são resumidos em grupos de tipos de processos.
5. Para cada instância do processo, as medidas definidas são calculadas e armazenadas em cubos de informações.
6. As operações que excedam os valores planejados serão verificadas e, se necessário, sinalizadas.

Projetar um armazém de processos envolve, portanto, a geração de fragmentos de processo a partir de sequências de transações e fluxos de documentos do sistema de origem e a sua mesclagem em

instâncias de processo. Essas instâncias de processo são usadas como objetos de referência para análises e avaliações de desempenho do processo. Podem ser usadas como modelos de processos no PPM.



A cadeia de processos controlada por eventos (EPC), que consiste em uma cadeia de objetos, é usada para representar uma instância de processo. Ambos os objetos e a própria instância do processo podem ter atributos, nos quais os dados da instância são salvos. Esses dados são usados para classificar as instâncias de processo em tipos e para calcular as medidas. As medidas calculadas são, por sua vez, salvas em atributos dos objetos e instâncias de processo. Os atributos são assim os portadores reais de informação do sistema PPM. Os atributos possíveis são definidos na configuração do PPM. A configuração é composta por atributos padrão, como **Process identifier**, **End time** e atributos específicos do sistema, que podem ser definidos livremente.

Informações adicionais sobre EPCs:

Uma cadeia de processos controlada por eventos (EPC) é um tipo de modelo desenvolvido pelo Prof. Scheer para descrever graficamente a sequência cronológica de um processo de desempenho de entrega. É baseado nas seguintes suposições:

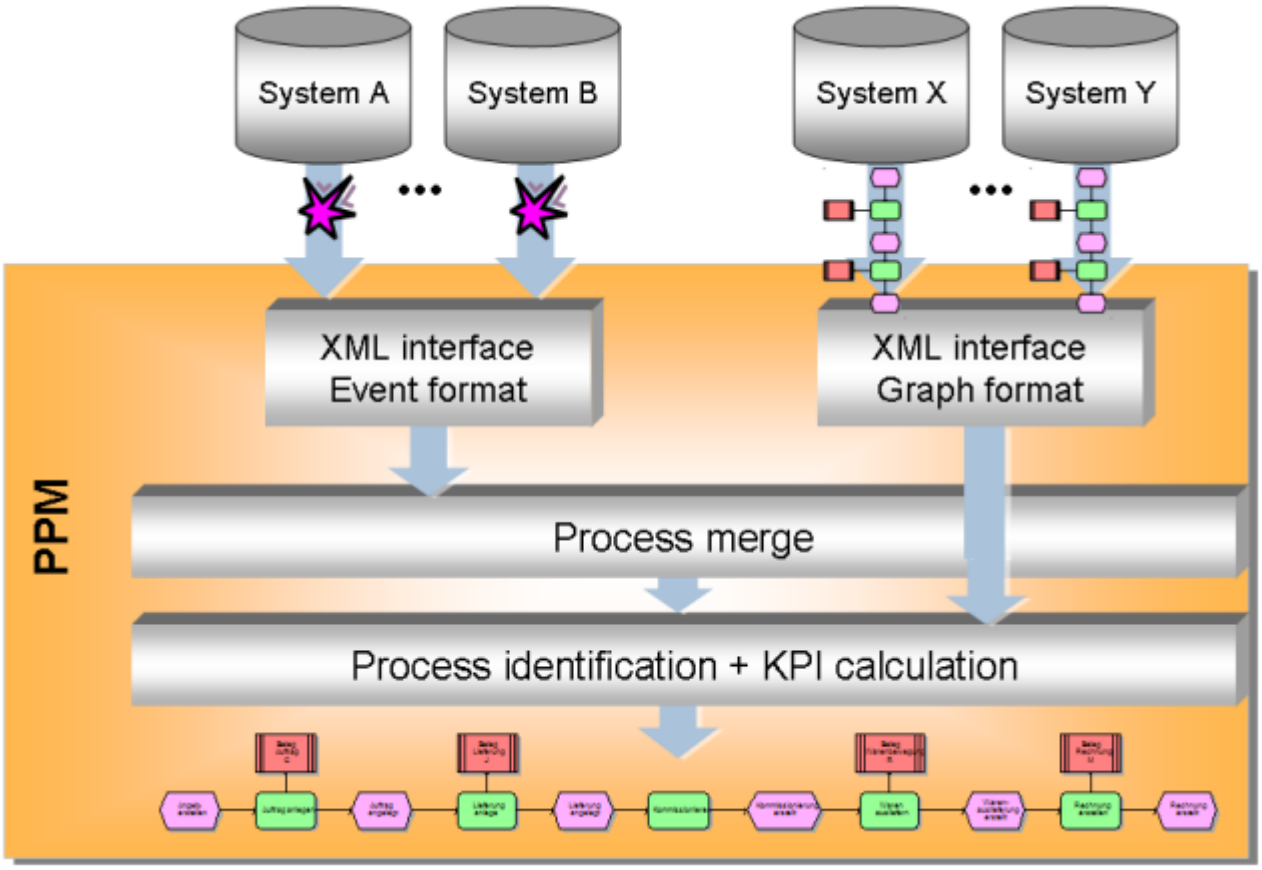
- Cada atividade dentro de um processo é desencadeada por uma alteração do estado comercialmente relevante de um objeto de informação. Cada atividade pode resultar numa alteração do estado comercialmente relevante de um objeto de informação.
- O estado de um objeto de informação comercialmente relevante é definido graficamente por um objeto do tipo **Event**.
- Os objetos do tipo **Função** são usados para a representação gráfica de atividades. Vincular eventos e funções em série e conectar esses objetos com conexões direcionadas representa graficamente o fluxo de controle do processo.

- Como um evento pode desencadear várias funções e, inversamente, uma função pode ter vários eventos como resultado, são inseridas conexões (regras) AND, OR ou OR exclusivas nessas ramificações. Elas ilustram o relacionamento lógico que existe entre os objetos sequenciados.
- As unidades organizacionais descrevem os grupos de usuários que executam uma função.

6.1.1. Gerar fragmentos de processo

Os dados de processo podem ser recuperados dos sistemas de aplicativos de vários modos.

Exemplo 67.



Com o SAP R/3, os adaptadores especiais acessam os dados operacionais do documento R/3 online e transformam os fluxos de documentos SAP em descrições de processo. Esses adaptadores apenas são mencionados aqui para fins de integridade.

Com todos os outros sistemas de aplicativos, os dados do processo são importados para o sistema PPM offline por meio de uma interface de importação XML generalizada, usando um arquivo.

A interface de importação XML pode processar dois tipos diferentes de arquivos XML: Arquivos XML em formato gráfico do PPM e em formato do evento de sistema PPM.

Formato gráfico

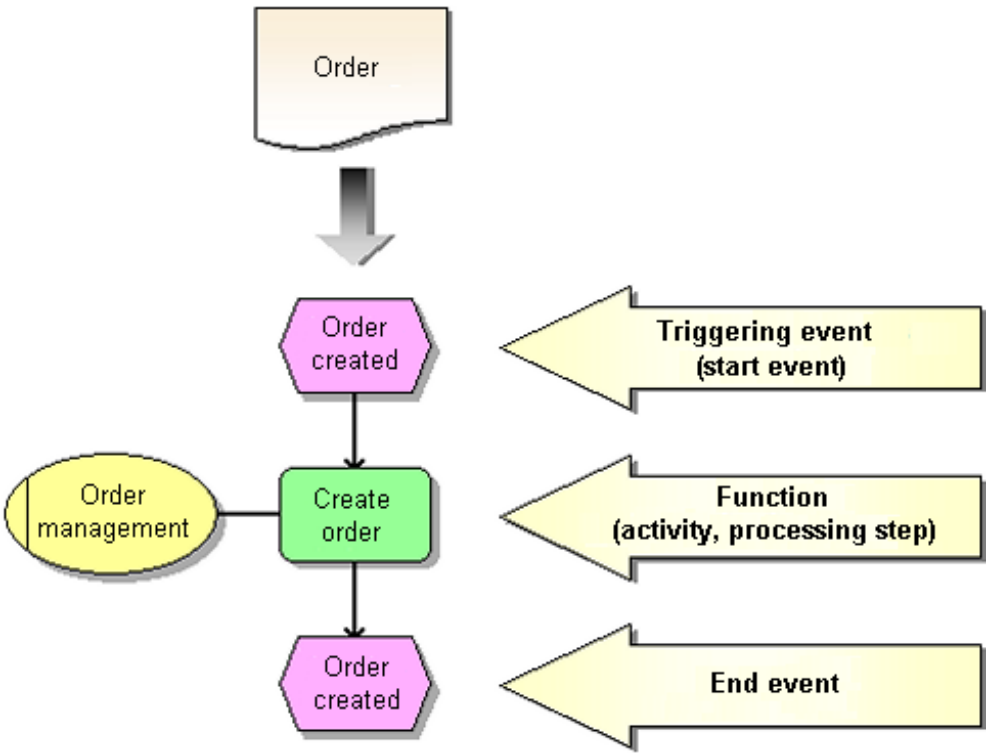
O formato gráfico é usado para transferir dados de processo já estruturados de sistemas de aplicativos orientados para processos (por exemplo, sistemas de fluxo de trabalho). O adaptador específico do

aplicativo gera arquivos XML, nos quais as instâncias de processo, incluindo sua lógica do decurso, são descritas no formato gráfico do PPM.

Formato do evento de sistema

O formato de evento do sistema é usado para todos os sistemas de aplicativos orientados para atividades, dos quais as informações que compõem o processo (lógica do decurso) não podem ser extraídas. Os eventos de sistema são interpretados como fragmentos de processo.

Exemplo 68.



Um fragmento de processo descreve uma parte de um processo global. Contém pelo menos uma função com seus eventos desencadeadores e resultantes. Um fragmento de processo pode ser interpretado como uma operação individual, uma atividade ou uma transação de um processo geral. Além do fluxo cronológico, um fragmento de processo também pode conter informações sobre o processador de uma função, na forma de unidades organizacionais.

Ao extrair dos sistemas de origem, é associado a cada evento do sistema um fragmento de processo, usando a definição de mapeamento. Os dados da instância para o evento do sistema são gravados como atributos nos objetos do fragmento de processo.

Usando os valores de atributo dos eventos de fragmento, os fragmentos de processo individuais são então mesclados em instâncias de processo.

Os fragmentos de processo apenas são importados para o banco de dados do PPM se puderem ser associados a instâncias de processo específicas.

6.1.2. Merge de fragmentos de processo

Este capítulo descreve a mesclagem de fragmentos de processo importados em instâncias de processo. Fragmentos de processos mesclados correspondem a processos de negócios que foram realmente

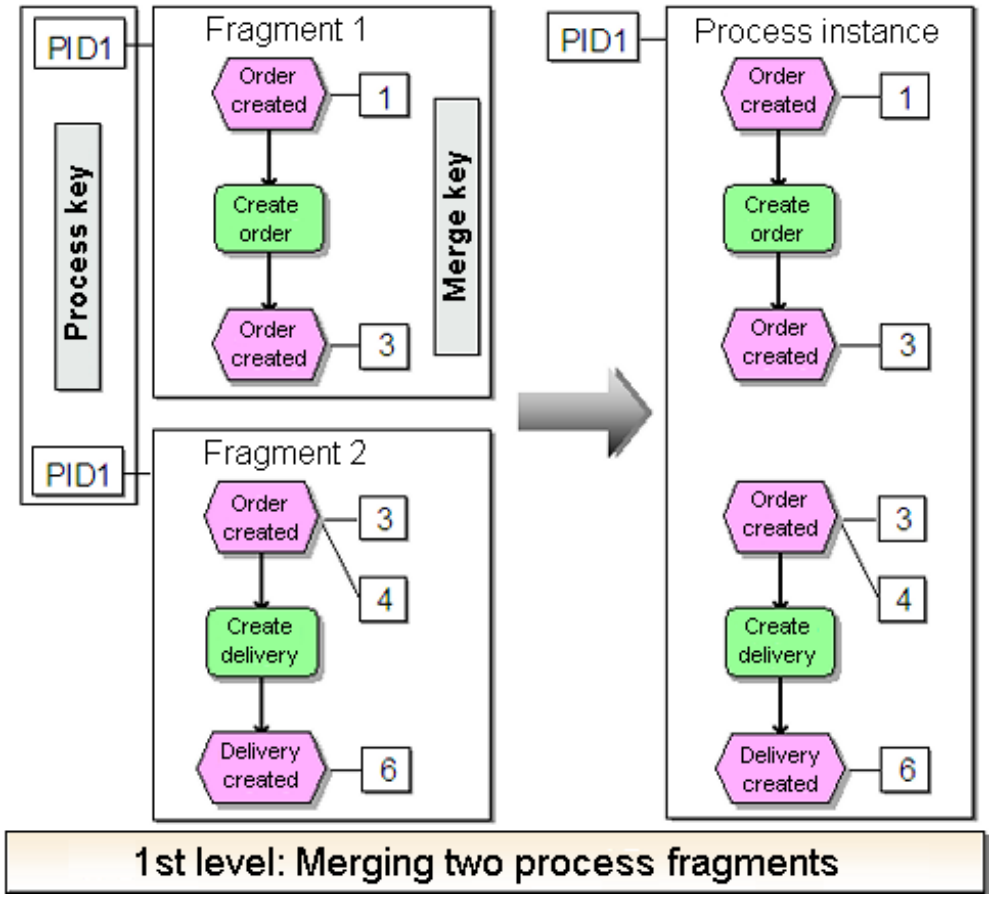
executados. Eles são conhecidos como instâncias de processo e, da mesma forma que os fragmentos, são representados na familiar notação EPC de ARIS.

Durante o merge, os atributos são copiados para as instâncias de processo e quaisquer dados específicos do processador relacionados com determinada pessoa são tornados anônimos.

O processo de merge decorre em duas etapas:

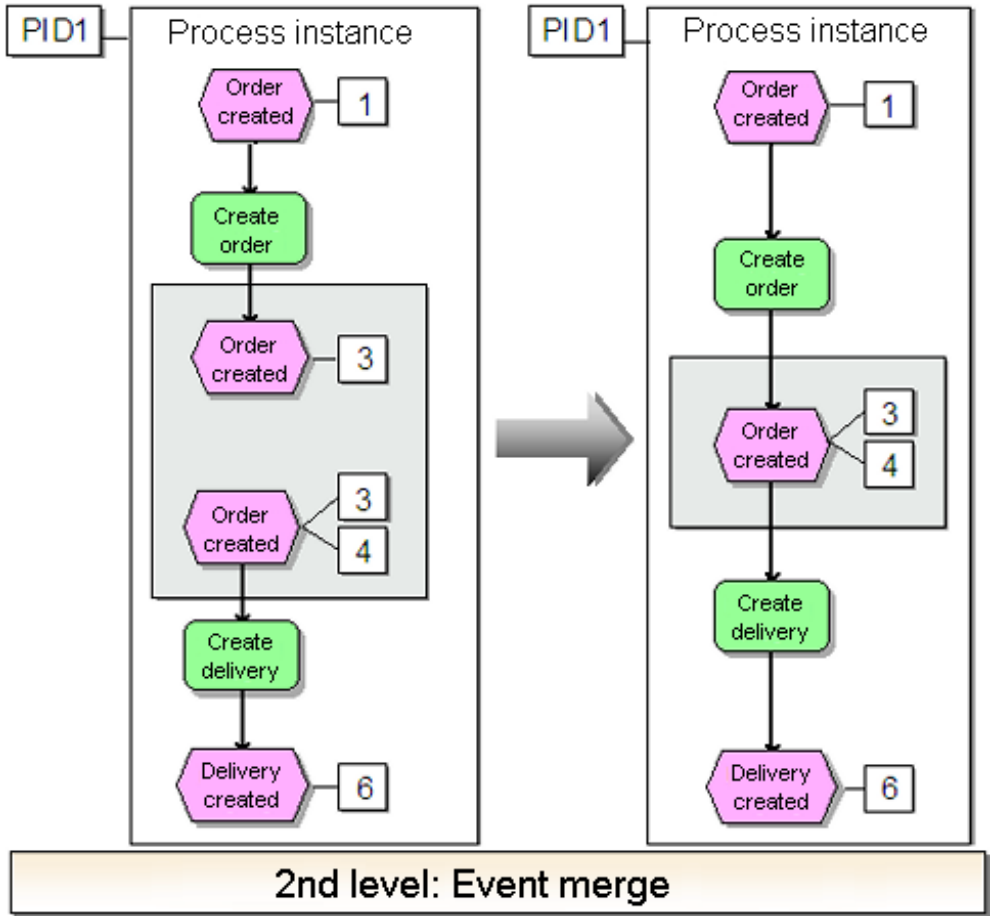
Na primeira etapa, os fragmentos de processo pertencentes à mesma instância de processo são identificados e copiados para uma instância de processo, usando chaves de processo.

Exemplo 69.



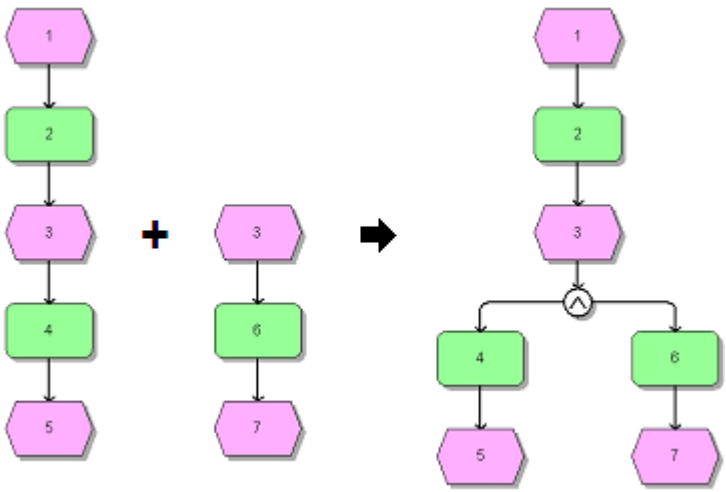
Na segunda etapa, os fragmentos de processo desconectados são vinculados ao mesclar os eventos de merge. Os eventos de merge são eventos para os quais as chaves de merge foram calculadas.

Exemplo 70.



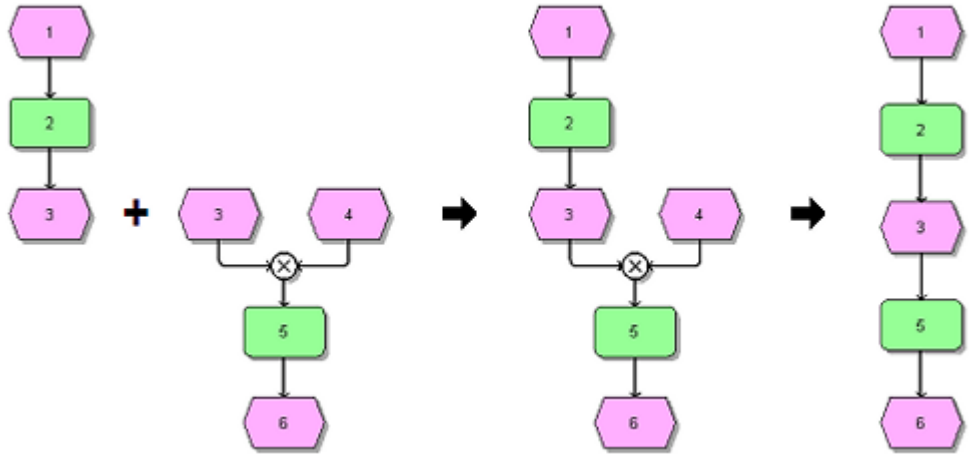
Quaisquer ramificações resultantes do processo de merge são expandidas usando regras e quaisquer regras desnecessárias são excluídas.

Exemplo 71.



No exemplo abaixo, uma regra de junção XOR é excluída durante o merge, pois as instâncias de processo representam transações comerciais reais e não podem conter nenhuma regra XOR. A regra XOR e o evento que não tem função anterior são excluídos (evento 4).

Exemplo 72.



Como não é possível distinguir entre os sistemas de origem individuais ao mesclar os dados importados, o PPM permite que os processos sejam visualizados nos sistemas e em toda a empresa.

A maneira como as chaves de processo e de mesclagem são calculadas é definida individualmente usando regras na configuração XML para cada cliente.

6.1.2.1. Copiar os atributos da instância de processo

Ao mesclar fragmentos de processo em instâncias de processo, os atributos de objetos são copiados para as instâncias de processo. Os atributos da instância de processo formam a base para o cálculo de medidas individuais, dependendo das dimensões.

Ao importar instâncias de processo completas no formato gráfico, não existe processo de merge. Essas instâncias de processo já têm atributos de instância de processo.

6.1.2.2. Tornar as unidades organizacionais anônimas

Em um nível de instância, os processadores reais de uma função são conhecidos. Como os usuários geralmente não podem ser mostrados por razões de proteção de dados e não são importantes para o cálculo de medidas, é possível tornar os processadores anônimos. Para isso, todos os funcionários têm de ser associados a uma unidade organizacional no PPM. Quando os dados são importados, os funcionários são substituídos pela unidade organizacional associada durante o processo de merge. As informações sobre o usuário real são irremediavelmente perdidas.

6.1.3. Tipificar processos

Após o merge de fragmentos de processo importados em instâncias de processo, as instâncias de processo precisam ser classificadas para permitir análises de medidas significativas. Para isso, elas são organizadas em uma hierarquia de dois níveis, definida livremente: As instâncias de processo são associadas a tipos de processo, que por sua vez são resumidos em grupos de tipos de processo. Um grupo de tipos de processo pode, portanto, conter vários tipos de processo, que por sua vez podem conter várias instâncias de processo. No entanto, uma instância de processo só pode ser associada a um único tipo de processo e, portanto, também a apenas um grupo de tipos de processo.

As regras para identificação são definidas livremente para cada sistema de origem específico. As informações de tipificação são salvas em atributos da instância de processo.

A associação de tipos de processo e grupos de tipos de processo é representada na árvore de processos no PPM. A configuração da árvore de processos também especifica quais as medidas e dimensões que estão disponíveis para os tipos de processo individuais.

Instâncias de processo são eventos de negócios reais que ocorreram e são compostos por fragmentos da instância de processo importados. Instâncias de processo semelhantes são resumidas em tipos de processo, que por sua vez são associados a um grupo de tipos de processo. O termo "processo", conforme é conhecido da modelagem de ARIS, não é especificado como parte do PPM e, portanto, não deve ser usado.

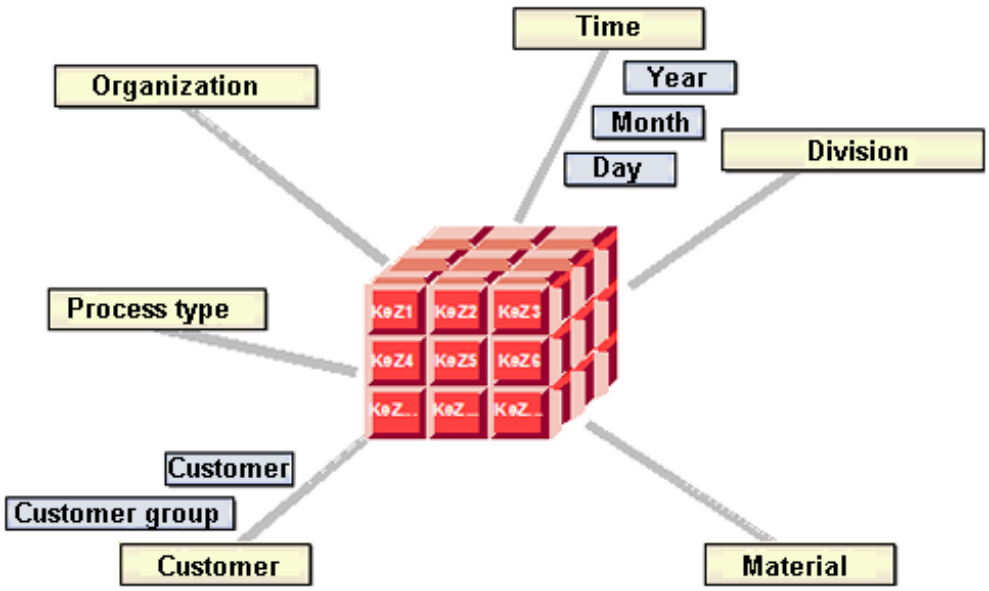
6.1.4. Calcular medidas

Uma instância de processo é analisada com base em medidas calculadas. As medidas são as propriedades de um processo ou de uma função calculada a partir de variáveis medidas. É feita uma distinção entre funções e medidas de processo. Juntamente com medidas padrão predefinidas, incluindo os indicadores **Número de processos** e **Tempo de ciclo do processo**, pode ser definido qualquer número de medidas na configuração usando a regra de cálculo associada.

Dimensões são critérios de acordo com os quais as medidas de instâncias de processos e funções podem ser diferenciadas, por exemplo, pelo tipo de processo ou pela localização.

Para os tipos de processo contidos na árvore de processos, as medidas especificadas são calculadas em função das dimensões. O resultado é armazenado permanentemente no banco de dados do PPM. Os valores são armazenados nos chamados **cubos de dados** para garantir que é possível realizar consultas rápidas.

Exemplo 73.



6.1.5. Verificar valores planejados

Para fins de monitoramento de processo, o sistema PPM permite definir valores planejados e valores de alarme. Valores planejados referem-se a um conjunto de instâncias de processo, valores de alarme

referem-se a uma instância de processo individual. Infrações críticas ao valor de alarme, para cima ou para baixo, são possíveis para instâncias de processo individuais, embora os valores planejados para o conjunto associado de instâncias de processo ainda estejam sendo atendidos.

A verificação específica do tipo de processo para infrações, para cima ou para baixo, aos valores de destino para medidas individuais conclui a importação de dados do sistema PPM. As medidas calculadas são comparadas com os valores planejados definidos na interface do usuário do PPM e as ações especificadas, como enviar um e-mail para o gerente de processo (mensagem de valor planejado), são executadas.

6.2. Cenários de importação

O parâmetro **IMPORT_SCENARIO** está disponível a partir da versão 10.2 do PPM. O parâmetro especifica um cenário de importação usado pelos programas de importação **runxmlimport**, **runppmimport**, **runpikidata** e **runppmcompress**.

O parâmetro está disponível no arquivo de configuração do parâmetro de memória **_vmparam.bat** e no arquivo de configuração do parâmetro de importação **EpkImport_settings.properties**. Ambos os arquivos devem conter sempre o mesmo valor **IMPORT_PARAMETER**.

- **_vmparam.bat** está localizado em **<PPM installation>\server\bin\work\data_ppm\config\<client name>\bin**
- **EpkImport_settings.properties** está localizado em **<PPM installation>\server\bin\work\data_ppm\config\<client name>**

Na configuração do cliente PPM CTK, poderá selecionar um cenário de importação predefinido que está definido em ambos os arquivos. Os cenários **Pequeno**, **Médio** e **Grande** estão disponíveis. Se nenhum cenário for selecionado, o cenário **Pequeno** será definido por padrão. Se a entrada estiver ausente nas configurações ou se um valor diferente dos valores permitidos for especificado, o cenário **Pequeno** é definido como padrão.

Cada cenário contém um conjunto específico de valores de parâmetros. Os valores de parâmetros são otimizados para o ambiente de sistema.

- **Pequeno** - Configuração otimizada para sistemas de servidor com pelo menos 1 GB de memória principal disponível para importar programas.
- **Médio** - Configuração otimizada para sistemas de servidor com pelo menos 4 GB de memória principal disponível para importar programas.
- **Grande** - Configuração otimizada para sistemas de servidor com pelo menos 8 GB de memória principal disponível para importar programas.

Se os cenários não corresponderem aos seus requisitos, poderá alterar os valores em um cenário de importação existente. Por exemplo, se forem necessários mais de 8 GB de memória principal para programas de importação.

Exemplo 74.

É definido IMPORT_SCENARIO=LARGE. Para aumentar a taxa de leitura de EPC para 1.500.000 neste cenário, defina LARGE_READ_RATE_EPC=1500000 em EPKImport_settings.properties. Defina também LARGE_LOCAL_PPM5_VM_MAX_MEM_RT_PPMIMPORT em _vmparam.bat para um valor mais alto (por exemplo, LARGE_LOCAL_PPM5_VM_MAX_MEM_RT_PPMIMPORT=10756) porque a importação do PPM passa a requerer mais memória principal.

6.2.1. Valores de parâmetros

O parâmetro **IMPORT_SCENARIO** está disponível nos arquivos **_vmparam.bat** e **EpkImport_settings.properties**. O valor para IMPORT_SCENARIO deve ser igual em ambos os arquivos (garantido pelo PPM CTK).

Os seguintes valores de parâmetros do arquivo **_vmparam.bat** são associados em função do cenário de importação selecionado. Os parâmetros são gravados com o cenário, mais um sublinhado.

- <IMPORT_SCENARIO>_LOCAL_PPM5_VM_MAX_MEM_RT_XMLIMPORT
- <IMPORT_SCENARIO>_LOCAL_PPM5_VM_MAX_MEM_RT_PPMIMPORT
- <IMPORT_SCENARIO>_LOCAL_PPM5_VM_MAX_MEM_RT_PIKIDATA
- <IMPORT_SCENARIO>_LOCAL_PPM5_VM_MAX_MEM_RT_PPMCOMPRESS
- <IMPORT_SCENARIO>_LOCAL_PPM5_VM_MIN_MEM_RT_XMLIMPORT
- <IMPORT_SCENARIO>_LOCAL_PPM5_VM_MIN_MEM_RT_PPMIMPORT
- <IMPORT_SCENARIO>_LOCAL_PPM5_VM_MIN_MEM_RT_PIKIDATA
- <IMPORT_SCENARIO>_LOCAL_PPM5_VM_MIN_MEM_RT_PPMCOMPRESS

Exemplo 75.

- SMALL_LOCAL_PPM5_VM_MAX_MEM_RT_XMLIMPORT=1024 é usado para IMPORT_SCENARIO=SMALL
- MEDIUM_LOCAL_PPM5_VM_MAX_MEM_RT_XMLIMPORT=4096 é usado para IMPORT_SCENARIO=MEDIUM
- LARGE_LOCAL_PPM5_VM_MAX_MEM_RT_XMLIMPORT=8192 é usado para IMPORT_SCENARIO=LARGE

Exceto para as exceções listadas abaixo, todos os valores de parâmetros no arquivo **EpkImport_settings.properties** são associados, em função do cenário de importação selecionado. Os parâmetros são gravados com o cenário, mais um sublinhado (por exemplo, <IMPORT_SCENARIO>_READ_RATE_EPC).

Exemplo 76.

- SMALL_READ_RATE_EPC=100000 é usado para IMPORT_SCENARIO=SMALL
- MEDIUM_READ_RATE_EPC=500000 é usado para IMPORT_SCENARIO=MEDIUM.
- LARGE_READ_RATE_EPC=1000000 é usado para IMPORT_SCENARIO=LARGE

7. Suporte e informações legais

Esta seção fornece algumas informações gerais sobre suporte ao produto e aspectos legais.

7.1. Escopo da documentação

As informações fornecidas descrevem as configurações e os recursos como estavam no momento da publicação. Como a documentação e o software estão sujeitos a ciclos de produção diferentes, a descrição das configurações e dos recursos pode ser diferente das configurações e recursos reais. Informações sobre discrepâncias são fornecidas nas Notas de versão que acompanham o produto. Leia as Notas de versão e considere as informações ao instalar, configurar e usar o produto.

Se você deseja instalar funcionalidades técnicas e/ou de sistema de negócios sem os serviços de consultoria fornecidos pelo SAG ARIS GmbH, precisa ter um amplo conhecimento do sistema a ser instalado, sua finalidade, os sistemas de destino e suas várias dependências. Devido ao número de plataformas e configurações de hardware e software interdependentes, só podemos descrever instalações específicas. Não é possível documentar todas as configurações e dependências.

Ao combinar várias tecnologias, observe as instruções dos fabricantes, especialmente os anúncios referentes a lançamentos em suas páginas da Internet. Não podemos garantir o funcionamento e a instalação adequados de sistemas de terceiros aprovados e não os suportamos. Siga sempre as instruções fornecidas nos manuais de instalação dos fabricantes relevantes. Se encontrar dificuldades, entre em contato com o fabricante relevante.

Se precisar de ajuda para instalar sistemas de terceiros, entre em contato com a sua organização local de vendas do ARIS. Observe que esse tipo de personalização específica do fabricante ou do cliente não é coberto pelo contrato de manutenção de software padrão do ARIS e pode ser realizada somente mediante solicitação e autorização especiais.

7.2. Suporte

Suporte aos Produtos

Fornecemos suporte para produtos ARIS a todos os clientes com um contrato de suporte válido.

Informações sobre disponibilidade e suporte da versão do produto, bem como datas de suporte sustentadas, estão disponíveis em ARIS Community.

Entre em contato com nossos serviços de suporte global na ARIS Community para criar e atualizar incidentes de suporte e postar ideias e solicitações de recursos.

Comunidade

Registre-se na ARIS Community para baixar produtos, atualizações e correções, encontrar informações especializadas e interagir com outros usuários.

Documentos

Para obter informações sobre como usar e ativar produtos e para outras questões de suporte técnico, visite o site de documentação do ARIS.

Se você quiser dar feedback sobre qualquer tópico de documentação, escreva para documentation@aris.com.

7.3. 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.