



SAP Exchange Infrastructure - Grafische Tools zur Definition performanter Mappings

Dr. Joachim Orb
SAP NetWeaver RIG XI, SAP AG

In diesem Vortrag werden folgende Themen behandelt:

- **Übersicht über die Mapping Programme der SAP Exchange Infrastructure**
- **Einführung in das grafische Message Mapping**
- **Beispiel eines Message Mapping**

Übersicht Mapping Programme

- Einführung
- Java Mapping
- XSLT Mapping
- ABAP Mapping

Das grafische Message Mapping

- Konzepte
- Standard Funktionen
- Werte-Mapping
- Benutzerdefinierte Funktionen

Zusammenfassung

Übersicht Mapping-Programme

- Einführung
- Java Mapping
- XSLT Mapping
- ABAP Mapping

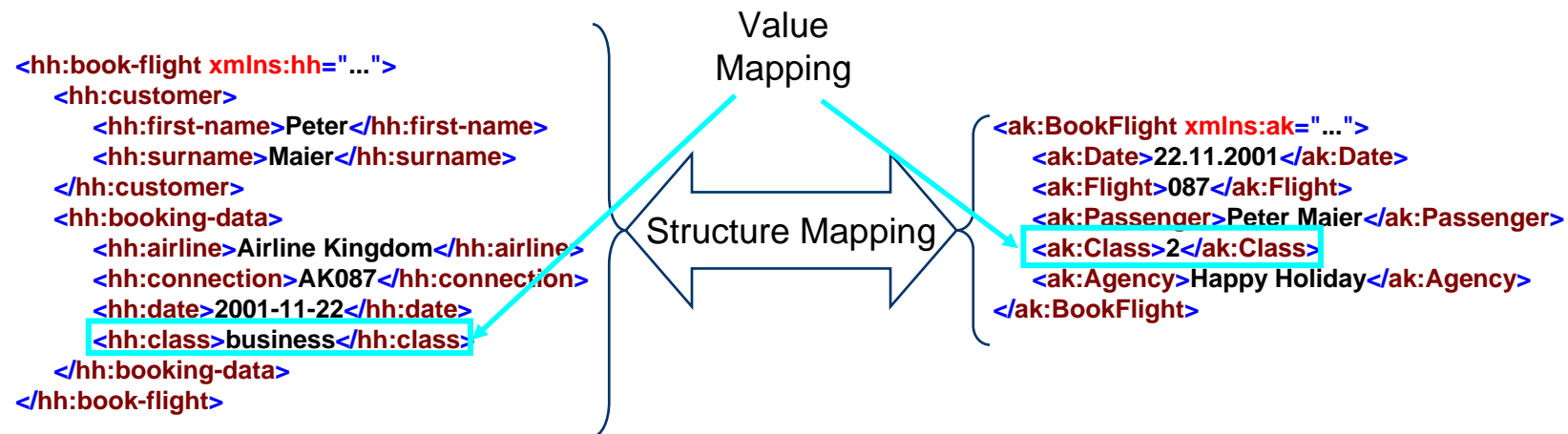
Das grafische Message Mapping

- Konzepte
- Standard Funktionen
- Werte-Mapping
- Benutzerdefinierte Funktionen

Zusammenfassung

Mapping

- Transformation von Nachrichtenstrukturen
- Transformation von Nachrichteninhalten
- Transformationen werden in SAP XI als *Mapping-Programme* angelegt

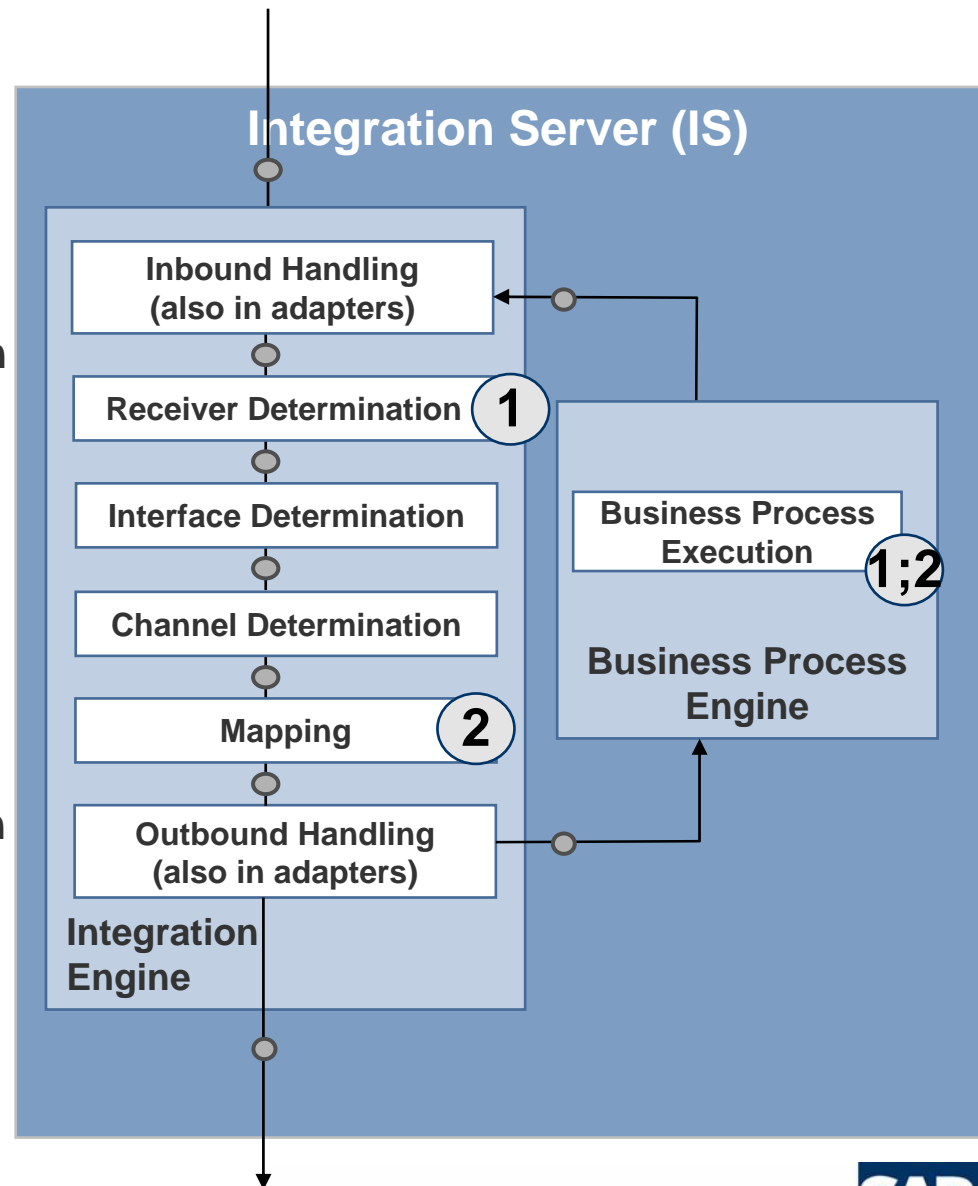


Mapping zur Laufzeit

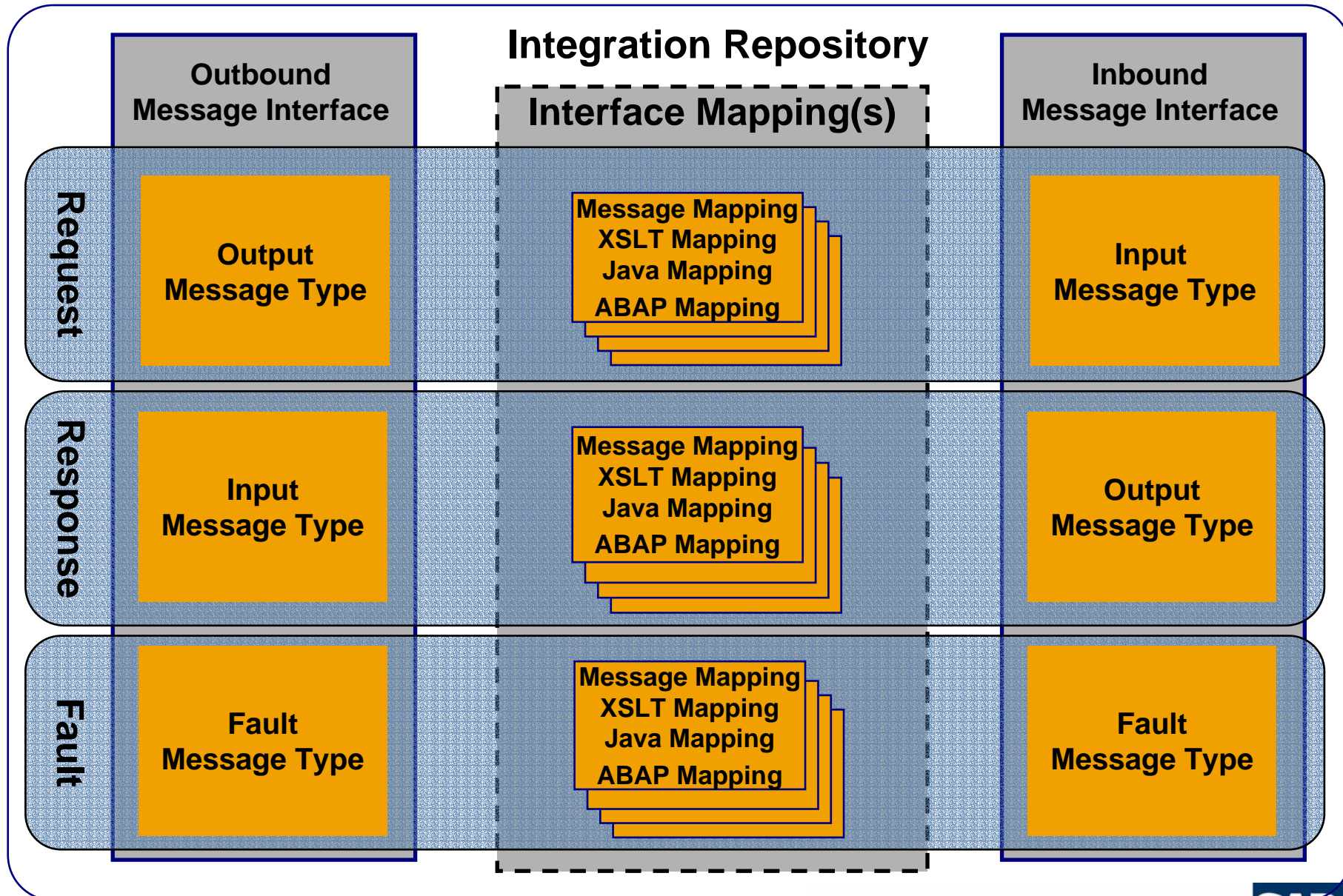
XI-Nachrichten enthalten ein Sender-Interface

IS wertet die Konfigurationsdaten des Integration Directory aus

- Im logischen Routing wird das Empfänger Interface (1) ermittelt.
- Das Interface-Paar referenziert ein Interface Mapping
- Das Interface Mapping referenziert Mapping Programme (Für Request, Response und/oder Fehler Nachrichten)
- IS führt die Mapping Programme aus (2).
- Empfänger-Ermittlung und Mapping können auch im Rahmen des ccBPM (Cross Component BPM) (1;2) durchgeführt werden.



Übersicht – Mapping Programme (I)



Message Mapping

- Werden mit Hilfe des graphischen Mapping Editors des Integration Repository angelegt
- Erzeugt Java Code

Importierte Archive

- Extern definierte Mapping Programme können ins Repository importiert werden
- Java Mapping Klassen
- XSLT Mapping:
 - Runtime Unterstützung im Java und ABAP Stack des SAP Web AS
 - Aufruf von Java Methoden aus dem XSLT

ABAP Mapping

Mapping Programme können sequentiell gerufen werden

Übersicht Mapping Programme

- Einführung
- **Java Mapping**
- XSLT Mapping
- ABAP Mapping

Das grafische Message Mapping

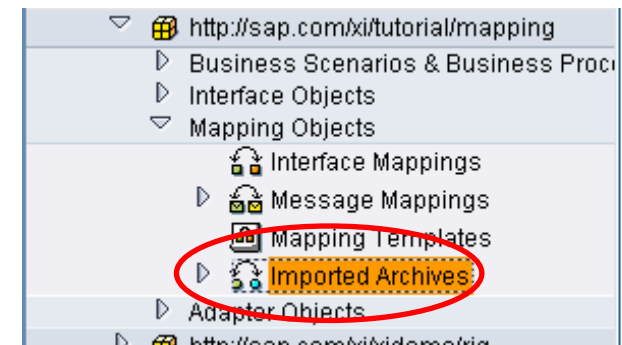
- Konzepte
- Standard Funktionen
- Werte-Mapping
- Benutzerdefinierte Funktionen

Zusammenfassung

Java Mapping (I)

Mapping Programme können in Java angelegt werden

1. Entwickeln Sie Ihr Java Coding lokal
z.B. mit Hilfe einer IDE
2. Erzeugen Sie ein .jar oder ein .zip
 - 2.1 Entweder über den jar Befehl des jdk
 - 2.2 Oder über eine IDE Funktion
3. Legen Sie ein Imported Archive an und importieren Sie das .jar in das Integration Repository



Java Mapping (II) - Beispiel

```
import com.sap.aii.mapping.api.StreamTransformation;
import com.sap.aii.mapping.api.StreamTransformationConstants;

public class JavaMapping implements StreamTransformation {
    private Map myParam;

    public void setParameter(Map param) {
        myParam = param;
    }

    public void execute(InputStream in, OutputStream out) {
        // ...
        String senderName = (String)myParam.get(
            StreamTransformationConstants.SENDER_NAME);
        // ...
    }
}
```

Übersicht Mapping Programme

- Einführung
- Java Mapping
- **XSLT Mapping**
- ABAP Mapping

Das grafische Message Mapping

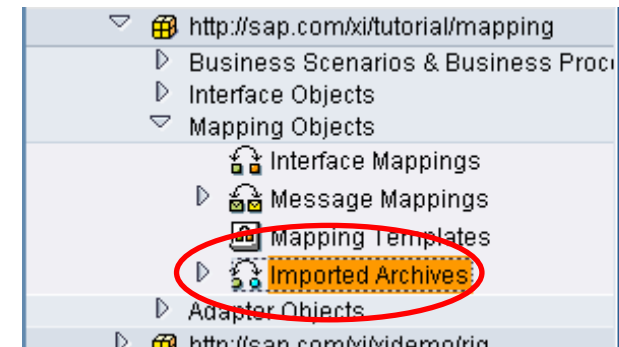
- Konzepte
- Standard Funktionen
- Werte-Mapping
- Benutzerdefinierte Funktionen

Zusammenfassung

XSLT Mapping (I)

Ähnliches Vorgehen wie bei Java Mappings

1. Entwickeln Sie das XSLT-File
2. Erzeugen Sie ein .jar oder ein .zip
 - 2.1 Entweder über den jar Befehl des jdk
 - 2.2 Oder über eine IDE Funktion
3. Legen Sie ein Imported Archive an und importieren Sie das .jar in das Integration Repository
4. Geladene XSLT-Programme können im XI Repository mittels eines einfachen Editors bearbeitet werden



XSLT Mapping (II) - Beispiel

XSLT Mapping with Java Erweiterung

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:javamap="java:com.company.group.MappingClass">
  <xsl:param name="first" select="//first-name"/>
  <xsl:param name="last" select="//last-name"/>
  <xsl:template match="/">
    ...
    <name>
      <xsl:if test="function-available('javamap:merge')">
        <xsl:value-of select="javamap:merge($first, $last)"/>
      </xsl:if>
    </name>
    ...
  </xsl:template>
</xsl:stylesheet>
```

Ruft Methode merge:

```
package com.company.group;

public class MappingClass {
  ...
  public static String merge(String first, String last) {
    return first + ' ' + last;
  }
  ...
}
```

Übersicht Mapping Programme

- Einführung
- Java Mapping
- XSLT Mapping
- **ABAP Mapping**

Das grafische Message Mapping

- Konzepte
- Standard Funktionen
- Werte-Mapping
- Benutzerdefinierte Funktionen

Zusammenfassung

Mapping Programme können in ABAP implementiert werden

1. Entwickeln Sie den ABAP Code im Object Navigator (se80)
2. Legen Sie eine ABAP Klasse an, die das Interface IF_MAPPING des Software-Pakets SAI_MAPPING implementiert
3. Programmieren Sie die Methode EXECUTE aus

ABAP Mapping (III) - Template

```
method if_mapping~execute.

* initialize iXML
type-pools: ixml.
class cl_ixml definition load.

* create stream factory
data: streamfactory type ref to if_ixml_stream_factory.
streamfactory = ixmlfactory->create_stream_factory( ).

* create input stream
data: istream type ref to if_ixml_istream.
istream = streamfactory->create_istream_xstring( source ).

* ...

* get XI parameter
data: l_sender_service type string.
l_sender_service = param->get( if_mapping_param=>sender_service ).
* ...

* create output stream
data: ostream type ref to if_ixml_ostream.
ostream = streamfactory->create_ostream_xstring( result ).
* ...

endmethod.
```

Übersicht Mapping Programme

- Einführung
- Java Mapping
- XSLT Mapping
- ABAP Mapping

Das grafische Message Mapping

- **Konzepte**
- Standard Funktionen
- Werte-Mapping
- Benutzerdefinierte Funktionen

Zusammenfassung

Allgemeine Konzepte - Mapping Editor (I)

The screenshot displays the SAP Message Mapping Editor interface. At the top, the window title is "ContactToCustomer: Edit Message Mapping". Below the title bar, there are tabs for "Message Mapping", "Navigation", "Edit", and "View". The main area is divided into several sections:

- Edit Message Mapping:** A metadata section with fields for Name (ContactToCustomer), Namespace (http://sap.com/xi/tutorial/mapping), Software Component Version (XI RIG, 3.0 of Stefan), and Description.
- Strukturübersicht (Structure Overview):** Two tree views showing the source and target message structures. The source structure (left) includes Contact, RefNo, Name, and Address (with sub-elements Street, ZIPCode, and City). The target structure (right) includes Customer, CustomerID, CustomerName (with sub-elements Fullname and Surname), and CustomerAddress (with sub-elements Street, ZIPCode, and City). A red dashed box highlights the ZIPCode elements in both structures, with a red arrow pointing from the source to the target.
- Data-Flow Editor:** A diagram showing a mapping between two "ZIPCode" data flows. A red oval highlights this mapping, which is labeled "Zielfeld-Mapping".
- Functions:** A list of available functions at the bottom, including Text, substring, concat, equalsS, indexOf, lastIndexOf, compare, replaceString, length, endsWith, and startsWith.

Einbinden von Quell- und Zielstrukturen

- **Aus dem Integration Repository**
 - XSD, die im Integration Repository entwickelt wurden
 - Importierte externe Definitionen (WSDL, XSD, DTD)
- **Aus dem lokalen File-Verzeichnis (XSD oder XML)**
 - Nicht alle XML-Tags werden unterstützt (Es gilt die Regel, dass XSDs, die mit dem Data Type Editor entwickelt wurden, importiert werden können)
 - `<xsd:include>` und `<xsd:import>` werden nicht unterstützt
- **Der Mapping Editor zeigt eine Strukturübersicht der XSD und XML**

- **Zielfeld-Mappings** werden im Data-Flow Editor angelegt und editiert
- Ein **Message Mapping** setzt sich aus mehreren Zielfeld-Mappings
- Alle Pflicht-Zielfelder (rot markiert in der Strukturübersicht) müssen gemappt werden

Allgemeine Konzepte - Elemente

Message Type: ContactIn

	type	Occurences
ContactIn	ContactIn	1..1
@Type		optional
Name		1..1
Firstname	xsd:string	1..unbounded
Surname	xsd:string	1..1
Address		0..3
Street	xsd:string	1..1
CityCode	xsd:string	0..1
City	xsd:string	0..1
Phone	xsd:string	0..unbounded
Email	xsd:string	0..unbounded

Attribut

Pflichtfeld, kann sich wiederholen ($minOccurs > 0, maxOccurs > 1$)

Pflichtfeld ($minOccurs > 0$)

Knoten, kann sich wiederholen ($maxOccurs > 1$)

Feld, kann sich wiederholen ($maxOccurs > 1$)

Allgemeine Konzepte – Feldzuordnung (I)

Doppelklicken = Einfügen in den Data-Flow Editor

Doppelklicken = in existierendes Mapping springen

drag&drop

drag&drop

drag&drop

ZIPCode → ZIPCode

Field	Type	Cardinality
Contact	Contact	1..1
RefNo	xsd:string	1..1
Name	Name	1..1
Address	Address	1..1
Street	xsd:string	1..1
ZIPCode	xsd:string	1..1
City	xsd:string	1..1

Field	Type	Cardinality
Customer	customer	1..1
CustomerID	xsd:string	1..1
CustomerName	CustomerName	1..1
Fullname	xsd:string	1..1
Surname	xsd:string	1..1
CustomerAddress	Address	1..1
Street	xsd:string	1..1
ZIPCode	xsd:string	1..1
City	xsd:string	1..1

Functions: Text, substrings, concat, equalsS, indexOf, indexOf, lastIndexOf, lastIndexOf, compare, replaceString, length, endsWith, startsWith

Allgemeine Konzepte – Feldzuordnung (II)

1. Knoten markieren

2. Zuordnungsfunktion auswählen

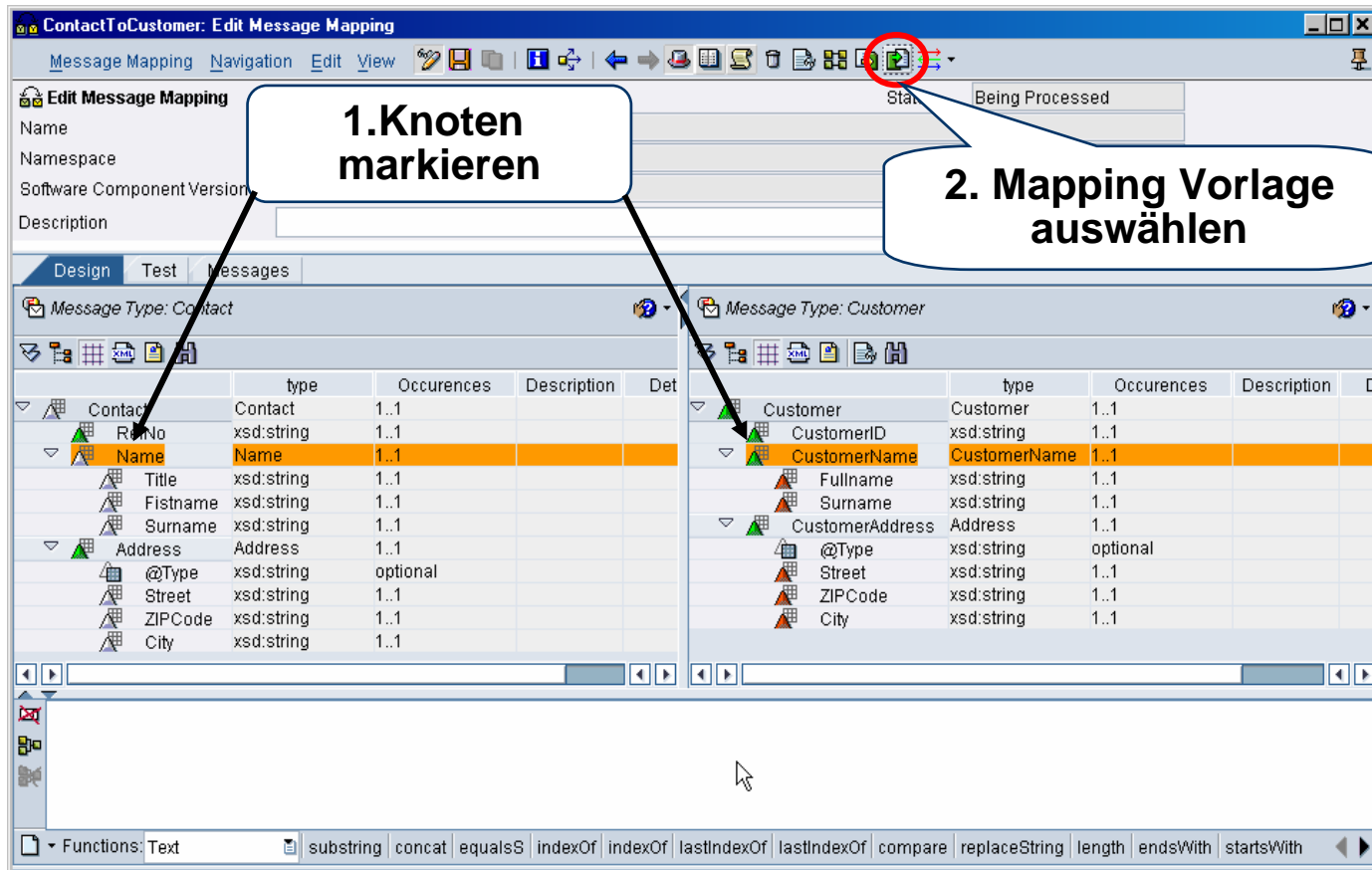
	type	Occurrences	Description	De
Contact	Contact	1..1		
RefNo	xsd:string	1..1		
Name	Name	1..1		
Title	xsd:string	1..1		
Firstname	xsd:string	1..1		
Surname	xsd:string	1..1		
Address	Address	1..1		
@Type	xsd:string	optional		
Street	xsd:string	1..1		
ZIPCode	xsd:string	1..1		
City	xsd:string	1..1		

	type	Occurrences	Description	D
Customer	Customer	1..1		
CustomerID	xsd:string	1..1		
CustomerName	CustomerName	1..1		
Fullname	xsd:string	1..1		
Surname	xsd:string	1..1		
CustomerAddress	Address	1..1		
@Type	xsd:string	optional		
Street	xsd:string	1..1		
ZIPCode	xsd:string	1..1		
City	xsd:string	1..1		

Functions: Text substring concat equalsS indexOf indexOf lastIndexOf lastIndexOf compare replaceString length endsWith startsWith

Mappings für gleichnamige Felder werden automatisch angelegt

Allgemeine Konzepte – Mapping Vorlagen



Voraussetzungen:

- Mapping Vorlagen basieren auf Daten Typen
- Unterknoten müssen diese Daten Typen referenzieren

Allgemeine Konzepte – Textanzeige

The screenshot shows the 'Edit Message Mapping' window for 'ContactToCustomer'. It features a toolbar with a 'Text' icon circled in red, labeled 'Textanzeige'. Below the toolbar, the 'Design' tab is active, showing two message type trees: 'Contact' and 'Customer'. The 'Customer' tree has 'CustomerAddress' highlighted in orange, with a callout 'Textanzeige pro Zielfeld' pointing to its 'XML' icon, also circled in red. At the bottom, the 'Zielfeld-Mappings' section contains the following XSLT snippets:

```

/ins:Customer/CustomerID=/ins:Contact/RefNo
/ins:Customer/CustomerName/Fullname=concat([delimiter]=concat([delimiter]=/ins:Contact/Name/Title , /ins:Contact/Name/Fistname) , /ins:Contact/Name/Surname)
/ins:Customer/CustomerName/Surname=/ins:Contact/Name/Surname
/ins:Customer/CustomerAddress=/ins:Contact/Address
/ins:Customer/CustomerAddress/@Type=/ins:Contact/Address/@Type
/ins:Customer/CustomerAddress/Street=/ins:Contact/Address/Street
/ins:Customer/CustomerAddress/ZIPCode=/ins:Contact/Address/ZIPCode
/ins:Customer/CustomerAddress/City=/ins:Contact/Address/City
    
```

Allgemeine Konzepte – Abhängigkeiten

The screenshot displays the SAP Message Mapping interface. At the top, the 'Where-Used List' button is circled in red, with a callout box containing the text 'Abhängigkeiten'. Below this, the 'Status' is shown as 'Being Processed'. The main area is split into two panels: 'Message Type: Contact' on the left and 'Message Type: Customer' on the right. Both panels show a tree view of message components with columns for 'type', 'Occurrences', and 'Description'. Red lines connect the 'Name' component in the Contact message type to the 'CustomerName' component in the Customer message type. A callout box points to these connections with the text 'Alle Abhängigkeiten der ausgewählten Unterknoten werden dargestellt'. At the bottom, a 'Functions' bar is visible with various mapping functions like 'substring', 'concat', etc.

Component	type	Occurrences	Description
Contact	Contact	1..1	
RefNo	xsd:string	1..1	
Name	Name	1..1	
Title	xsd:string	1..1	
Fistname	xsd:string	1..1	
Surname	xsd:string	1..1	
Address	Address	1..1	
@Type	xsd:string	optional	
Street	xsd:string	1..1	
ZIPCode	xsd:string	1..1	
City	xsd:string	1..1	

Component	type	Occurrences	Description
Customer	Customer	1..1	
CustomerID	xsd:string	1..1	
CustomerName	CustomerName	1..1	
Fullname	xsd:string	1..1	
Surname	xsd:string	1..1	
CustomerAddress	Address	1..1	
@Type	xsd:string	optional	
Street	xsd:string	1..1	
ZIPCode	xsd:string	1..1	

Allgemeine Konzepte – Mappings testen (I)

The screenshot shows the 'Edit Message Mapping' window in SAP NetWeaver. The window title is 'ContactToCustomer: Edit Message Mapping'. The status bar at the top right indicates 'Status: Being Processed'. The main area is divided into 'Design', 'Test', and 'Messages' tabs. The 'Test' tab is active, showing a 'Test Case' field and a 'Trace Level' dropdown set to 'Warning'. Below this, there is a tree view of the message structure. The 'Contact' node is expanded, showing fields like 'RefNo', 'Name', 'Title', 'Firstname', 'Surname', 'Address', '@Type', 'Street', 'ZIPCode', and 'City'. The 'Surname' field is highlighted with a red box and contains the value 'Miller'. A callout bubble points to this field with the text '1. Strukturvorlage zum Erzeugen von Testfällen'. Another callout bubble points to the 'Test' tab icon with the text '2. Ausführen des Tests'. The 'Messages' tab at the bottom is empty.

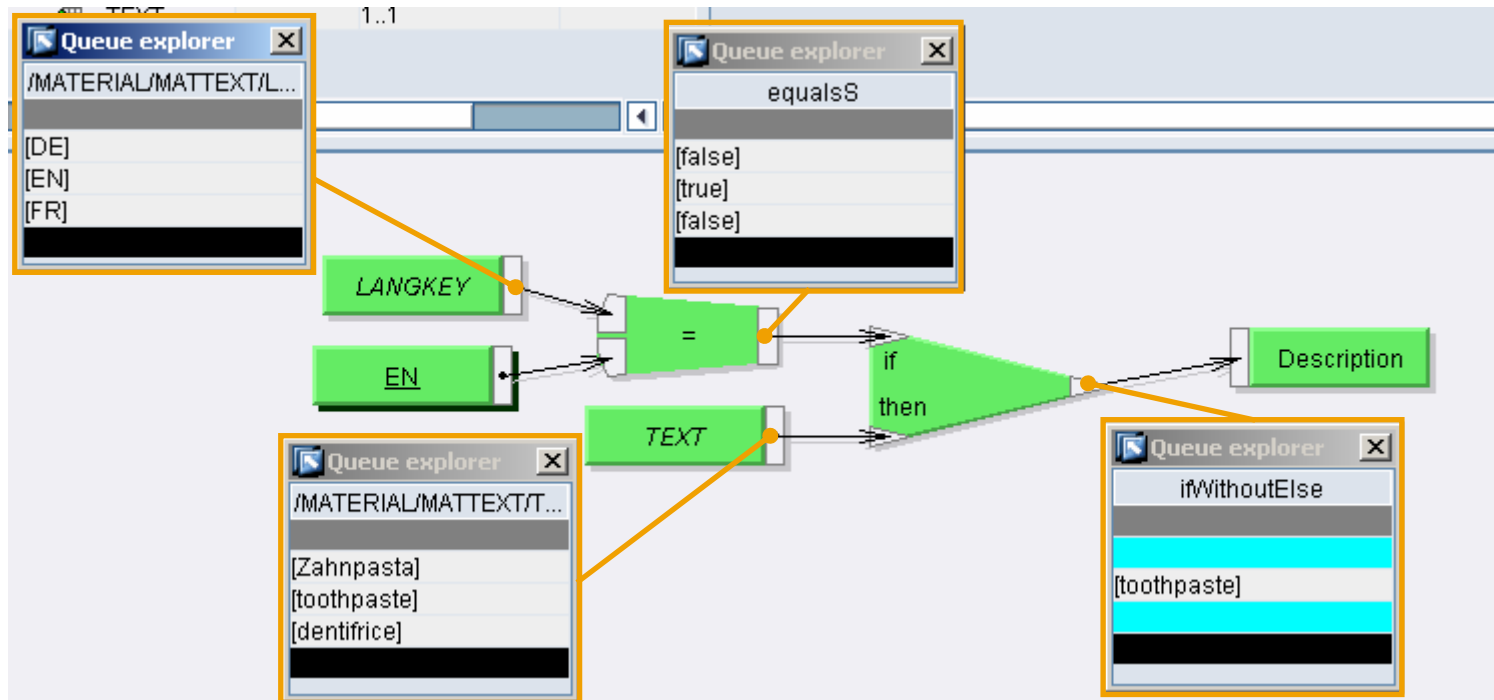
Field	Value
RefNo	
Name	
Title	Mr.
Firstname	John
Surname	Miller
Address	
@Type	
Street	
ZIPCode	
City	

Allgemeine Konzepte – Mappings testen (II)

The screenshot shows the 'ContactToCustomer: Edit Message Mapping' window. The top toolbar contains icons for Design, Test, and Messages. The 'Test' tab is active, showing a 'Test Case' field and a 'Trace Level' dropdown set to 'Warning'. Below this, the 'Instance: Message Type: Contact' section is visible, with a 'Result:(Not a File)' indicator. The main area displays an XML instance for a 'Contact' message type. A callout bubble points to the 'Test' icon in the toolbar, labeled '2. Ausführen des Tests'. Another callout bubble points to the 'Instance: Message Type: Contact' section, labeled '1. Laden einer xml Instanz vom Filesystem...'. A third callout bubble points to the XML text area, labeled '... oder Eintippen des xml im Editor.' The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ns:Contact xmlns:ns="http://sap.com/xi/tutorial" >
  <RefNo>777235</RefNo>
  <Name>
    <Title>Mrs.</Title>
    <Firstname>Paula</Firstname>
    <Surname>Jones</Surname>
  </Name>
  <Address Type="">
    <Street>123 Main St</Street>
    <ZIPCode>12345</ZIPCode>
    <City />
  </Address>
  <custBirthday />
</ns:Contact>
```

Allgemeine Konzepte – Mappings Debuggen



- “Queue Anzeigen” Funktion zum Darstellen der Feldwerte, die die Funktion verlassen.
- “Debugging” Funktion
- Auf jeden Mapping-Schritt anwendbar
- Mappings können Schritt für Schritt untersucht werden

Übersicht Mapping Programme

- Einführung
- Java Mapping
- XSLT Mapping
- ABAP Mapping

Das grafische Message Mapping

- Konzepte
- **Standard Funktionen**
- Werte-Mapping
- Benutzerdefinierte Funktionen


Zusammenfassung

Standardfunktionen: Einführung

X_1	$R = f(X_1, X_2)$	R
X_2		

I	$O = f(I)$	O
---	------------	---


X_1	$R = f(X_1, X_2, X_3)$	R
X_2		
X_3		

	$O =$ <i>Generated Value</i>	O
---	---------------------------------	---

- Alle Funktionen erfordern String-Werte als Eingabe und geben String-Werte zurück
- *Erzeugende Funktionen* erwarten keine Eingabe (z.B. Constant)
- Einige Standardfunktionen können über *Eigenschaften* spezifiziert werden (Sternmarkierung)
 - ◆ Z.B. können in der `concat` Funktion Trennzeichen angegeben werden
 - ◆ Eigenschaften werden durch Doppelklicken auf Funktion festgelegt

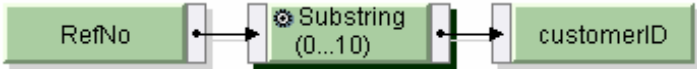
Standardfunktionen: *concat*

Kategorie: Text

Quellnachricht	Mapping	Zielnachricht
<pre><?xml version="1.0" encoding="UTF-8" ?> <Customer> <Firstname> Harry </Firstname> <Surname> Potter </Surname> </Customer></pre>	 <p><u>concat</u> Eigenschaft Delimiter=" "</p>	<pre><?xml version="1.0" encoding="UTF-8" ?> <Contact> <Fullname> Harry Potter </Fullname> </Contact></pre>

Standardfunktionen: *Substring*


Kategorie: Text

Quellnachricht	Mapping	Zielnachricht
<pre><?xml version="1.0" encoding="UTF-8" ?> <Contact > <RefNo > 0123456789-181170 </RefNo > </Contact ></pre>	 <p><u>Substring Eigenschaften</u> Beginn: 0 Anzahl: 10</p>	<pre><?xml version="1.0" encoding="UTF-8" ?> <Customer > <CustomerID > 0123456789 </customerID > </Customer ></pre>

- Zähler startet bei 0 !

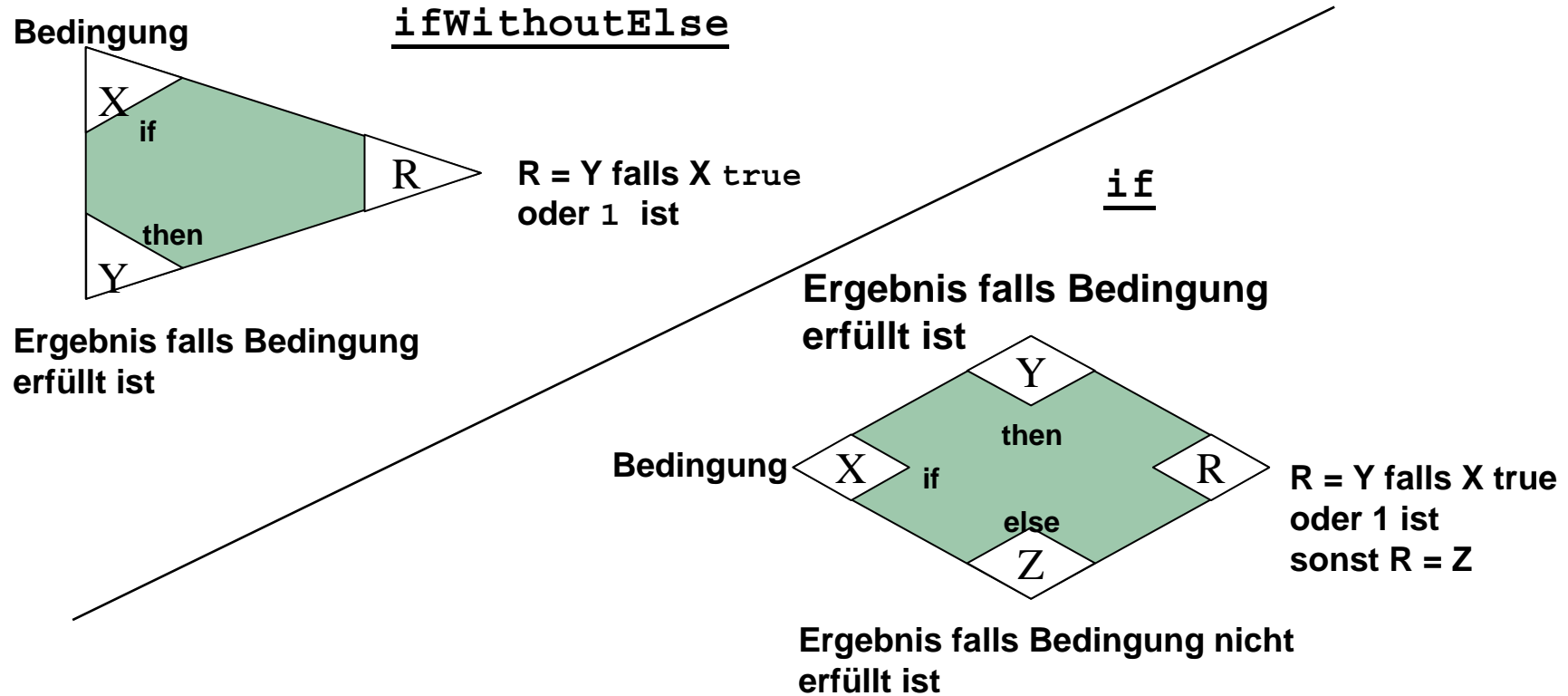
Standardfunktionen: *DateTrans*

Category: Date

Quellnachricht	Mapping	Zielnachricht
<pre><?xml version="1.0" encoding="UTF-8" ?> <Customer> <custBirthday> 181170 </custBirthday> </Customer></pre>	 <p><u>DateTrans Eigenschaften</u> src Format: ddMMyy dst Format: MM/dd/yyyy</p>	<pre><?xml version="1.0" encoding="UTF-8" ?> <Contact> <birthday> 11/18/1970 </birthday> </Contact></pre>

Muster orientieren sich an der Java Klasse java.text.DateFormat


Boole'sche Funktionen



Boole'sche Funktionen

- Verwenden true oder false String
- Interpretieren 1 and true als true-Wert

Kategorie: Arithmetic

Quellnachricht	Mapping	Zielnachricht
<pre><?xml version="1.0" encoding="UTF-8" ?> <Customer> <debt> 90 </debt> </Customer></pre>	 <p><u>FormatNum Eigenschaften</u> Number Format: ###,###,###.00</p> <p>(Muster siehe java.text.DecimalFormat)</p>	<pre><?xml version="1.0" encoding="UTF-8" ?> <Contact> <debt> 90.00 </debt> </Contact></pre>

- Berechnungen (add, multiply, etc.) werden mit Java Float Präzision durchgeführt
- Nachkomma Nullen werden abgeschnitten, z.B. $4.2 - 0.2 = 4$
- Um Nullen nach dem Komma zu halten muss *FormatNum* Funktion verwendet werden

Übersicht Mapping Programme

- Einführung
- Java Mapping
- XSLT Mapping
- ABAP Mapping

Das grafische Message Mapping

- Konzepte
- Standard Funktionen
- **Werte-Mapping**
- Benutzerdefinierte Funktionen

Zusammenfassung

Die **FixValues** Funktion stellt ein einfaches generisches Wertemapping im Integration Repository zur Verfügung

1. Zugriff über Standardfunktionen
Conversions -> FixValues
2. Doppelklicken um Eigenschaften zu definieren
 - ◆ Jedes Feldmapping hält seine eigene Wertetabelle
 - ◆ Gefüllte Tabellen können kopiert und in anderen Feldmappings verwendet werden.



The screenshot shows a dialog box titled 'FixValues' with a table for defining key-value mappings. The table has two columns: 'key' and 'value'. The 'Default' value is set to 'E'. The table contains three rows of data:

key	value
Economy	E
Business	B
First	F

At the bottom of the dialog box, there are two buttons: 'Save Table' and 'Cancel'.

Werte-Mappings (II) – Value mapping

Die Value mapping Funktion stellt eine spezifisches Werte-Mapping zur Verfügung

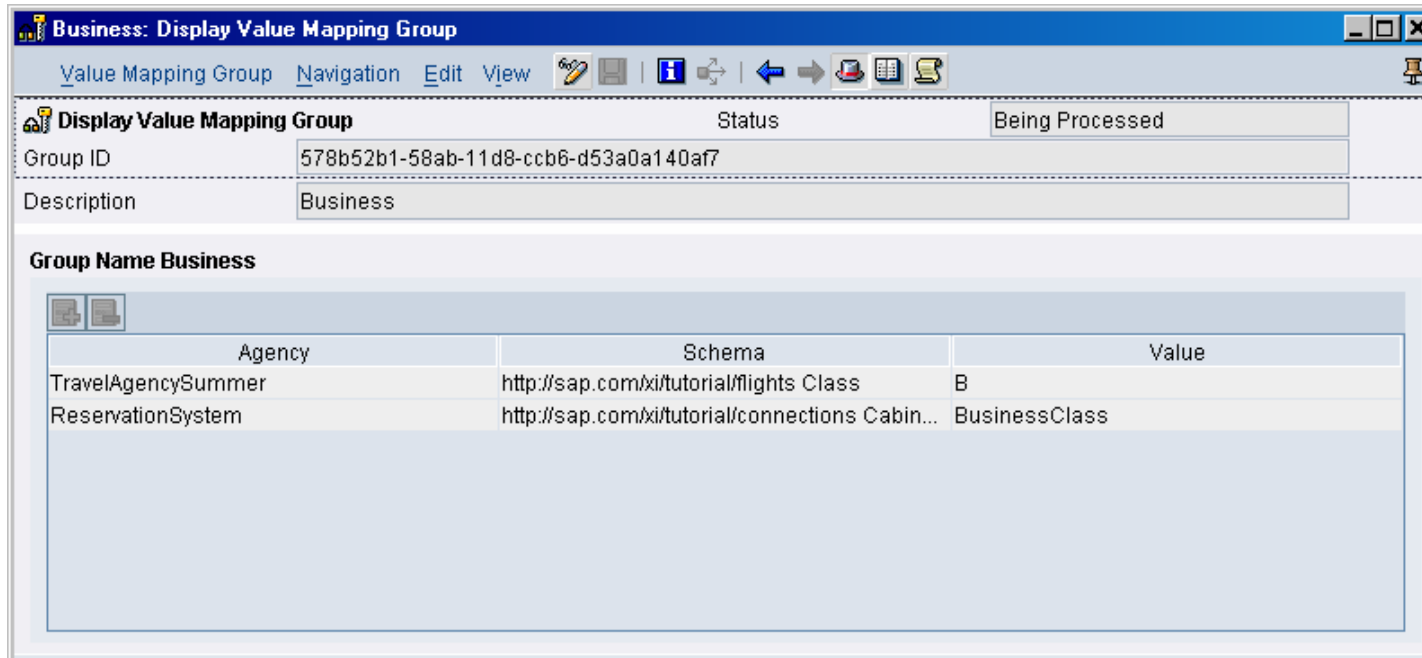
1. Zugriff über Standardfunktionen
Conversions -> Value mapping
2. Doppelklicken um Eigenschaften zu definieren
3. Legen Sie die Schlüssel für den Zugriff auf die Wertepaare fest:
 - ◆ Der Mapping Context definiert die Herkunft der Daten. Falls die Daten im Integration Repository angelegt wurden verwenden Sie:
`http://sap.com/xi/XI`
 - ◆ Agency und Schema für Sender- und Empfängernachrichten. Es können beliebige String Werte verwendet werden.



The screenshot shows the 'ValueMapping properties' dialog box. It has three sections for defining mapping contexts. Each section includes a 'Value Mapping Context' text field, a 'Notation' dropdown set to 'Any string value', an 'Agency' dropdown, and a 'Schema' text field. The first section has 'http://sap.com/xi/XI' in the context field, 'ReservationSystem' in the agency field, and 'http://sap.com/xi/tutorial/connections CabinClass' in the schema field. The second section has 'TravelAgencySummer' in the agency field and 'http://sap.com/xi/tutorial/flights Class' in the schema field. The third section is empty. At the bottom are 'OK' and 'Cancel' buttons.

Werte-Mapping (III) - Value Mapping Gruppen

- Semantisch gleiche Information wird in "Value Mapping Gruppen" zusammengefasst
- Pflege von Werte Mapping Gruppen:
 - im Integration Directory
 - senden von Nachrichten an das Message Interface "ValueMappingReplication" von SAP BASIS → SAP BASIS 6.40 → <http://sap.com/xi/XI/System>
- Notwendige Einträge
 - Agency
 - Schema (Namespace und Object Name)
 - Wert (der Gruppenrepräsentant der Agency–Schema Kombination)



The screenshot shows the SAP Business: Display Value Mapping Group interface. The window title is "Business: Display Value Mapping Group". The status is "Being Processed". The Group ID is "578b52b1-58ab-11d8-ccb6-d53a0a140af7" and the Description is "Business". Below this, there is a section titled "Group Name Business" which contains a table with three columns: Agency, Schema, and Value.

Agency	Schema	Value
TravelAgencySummer	http://sap.com/xi/tutorial/flights Class	B
ReservationSystem	http://sap.com/xi/tutorial/connections Cabin...	BusinessClass

Übersicht Mapping Programme

- Einführung
- Java Mapping
- XSLT Mapping
- ABAP Mapping

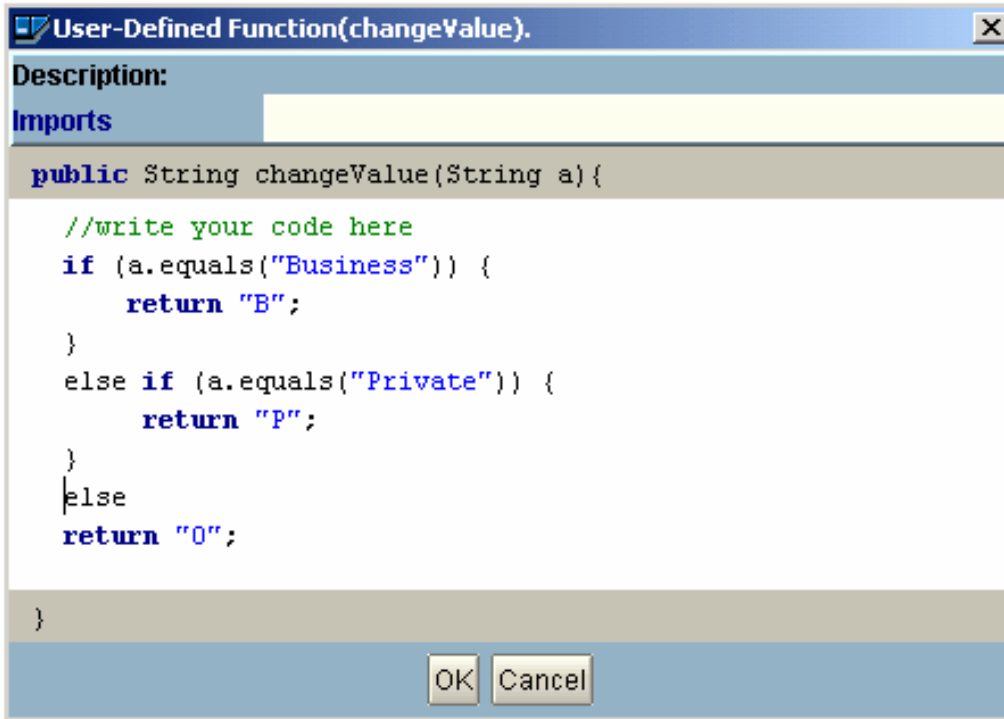
Das grafische Message Mapping

- Konzepte
- Standard Funktionen
- Werte-Mapping
- **Benutzerdefinierte Funktionen**

Zusammenfassung

Benutzerdefinierte Funktionen

- Erweiterung der Funktionalität durch benutzerdefinierte Java Methoden
- Typen
 - Einfache Funktionen (Verarbeitung von String-Werten)
 - Erweiterte Funktionen (Verarbeitung von String Arrays)
- Java Programme des *Imported Archive* der gleichen Software Komponenten Version können referenziert werden.



```
public String changeValue(String a){  
    //write your code here  
    if (a.equals("Business")) {  
        return "B";  
    }  
    else if (a.equals("Private")) {  
        return "P";  
    }  
    else  
        return "0";  
}
```

Übersicht Mapping Programme

- Einführung
- Java Mapping
- XSLT Mapping
- ABAP Mapping

Das grafische Message Mapping

- Konzepte
- Standard Funktionen
- Werte-Mapping
- Benutzerdefinierte Funktionen

Zusammenfassung

- In SAP Exchange Infrastructure können unterschiedliche Mapping Technologien eingesetzt werden
- Das grafische *Message Mapping* bietet die Möglichkeiten performante Mappings zu implementieren
- Sie haben nun einen Einblick in die Bedienung des Message Mappings erhalten

Weitere Informationen



Internet:

www.sap.de/netweaver



Kontaktperson

Joachim Orb, SAP AG



Schulungen

<http://www.sap.de/education>

RDXI30 - RKT Delta Workshop for SAP NetWeaver '04 - SAP XI 2.0 - 3.0.



Weitere Vorträge auf dem SAP NetWeaver Kongress

A8 „Verlängerung der Wertschöpfungskette im Einkauf bei Daimler Chrysler – Einsatz von SAP XI“, 24.Mai 14:15 Uhr - Hauptagenda



Ausstellung

Arbeitsplätze SAP NetWeaver -> SAP Exchange Infrastructure



Noch Fragen?



Bitte füllen Sie Ihren Feedbackbogen aus und geben ihn am Ende des Kongresses am Check-In ab.

Vielen Dank!

- Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die aus-drückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.
- Die von SAP AG oder deren Vertriebsfirmen angebotenen Softwareprodukte können Softwarekomponenten auch anderer Softwarehersteller enthalten.
- Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® und SQL Server® sind eingetragene Marken der Microsoft Corporation.
- IBM®, DB2®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix und Informix® Dynamic Server™ sind Marken der IBM Corporation in den USA und/oder anderen Ländern.
- ORACLE® ist eine eingetragene Marke der ORACLE Corporation.
- UNIX®, X/Open®, OSF/1® und Motif® sind eingetragene Marken der Open Group.
- Citrix®, das Citrix-Logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® und andere hier erwähnte Namen von Citrix-Produkten sind Marken von Citrix Systems, Inc.
- HTML, DHTML, XML, XHTML sind Marken oder eingetragene Marken des W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- JAVA® ist eine eingetragene Marke der Sun Microsystems, Inc.
- JAVASCRIPT® ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.
- MarketSet und Enterprise Buyer sind gemeinsame Marken von SAP AG und Commerce One.
- SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver und weitere im Text erwähnte SAP-Produkte und – Dienstleistungen sowie die entsprechenden Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und anderen Ländern weltweit. Alle anderen Namen von Produkten und Dienstleistungen sind Marken der jeweiligen Firmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.