6

Transforming Messages

Objectives

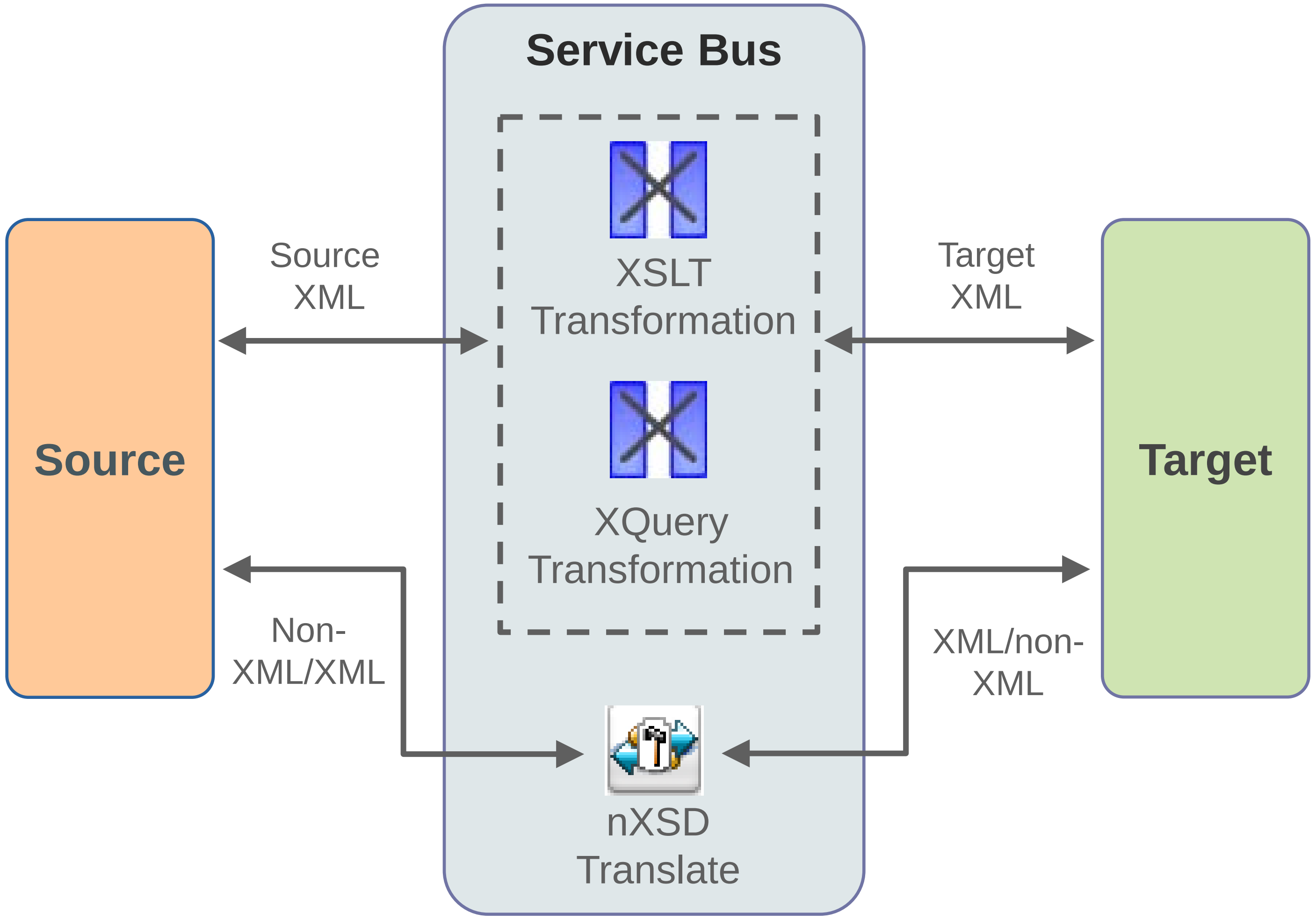
After completing this lesson, you should be able to:

- Describe the role of XPath, XQuery, and XSLT in the way Service Bus handles data
- Describe standard and custom XPath functions
- Describe how the XSLT Mapper can be used to create XSL transformations
- Use the XQuery Mapper to create XQuery transformations
- Transform non-XML data to XML data with nXSD

Agenda

- XPath
- XSLT transformation
- XQuery transformation
- nXSD

Message Transformation in Service Bus



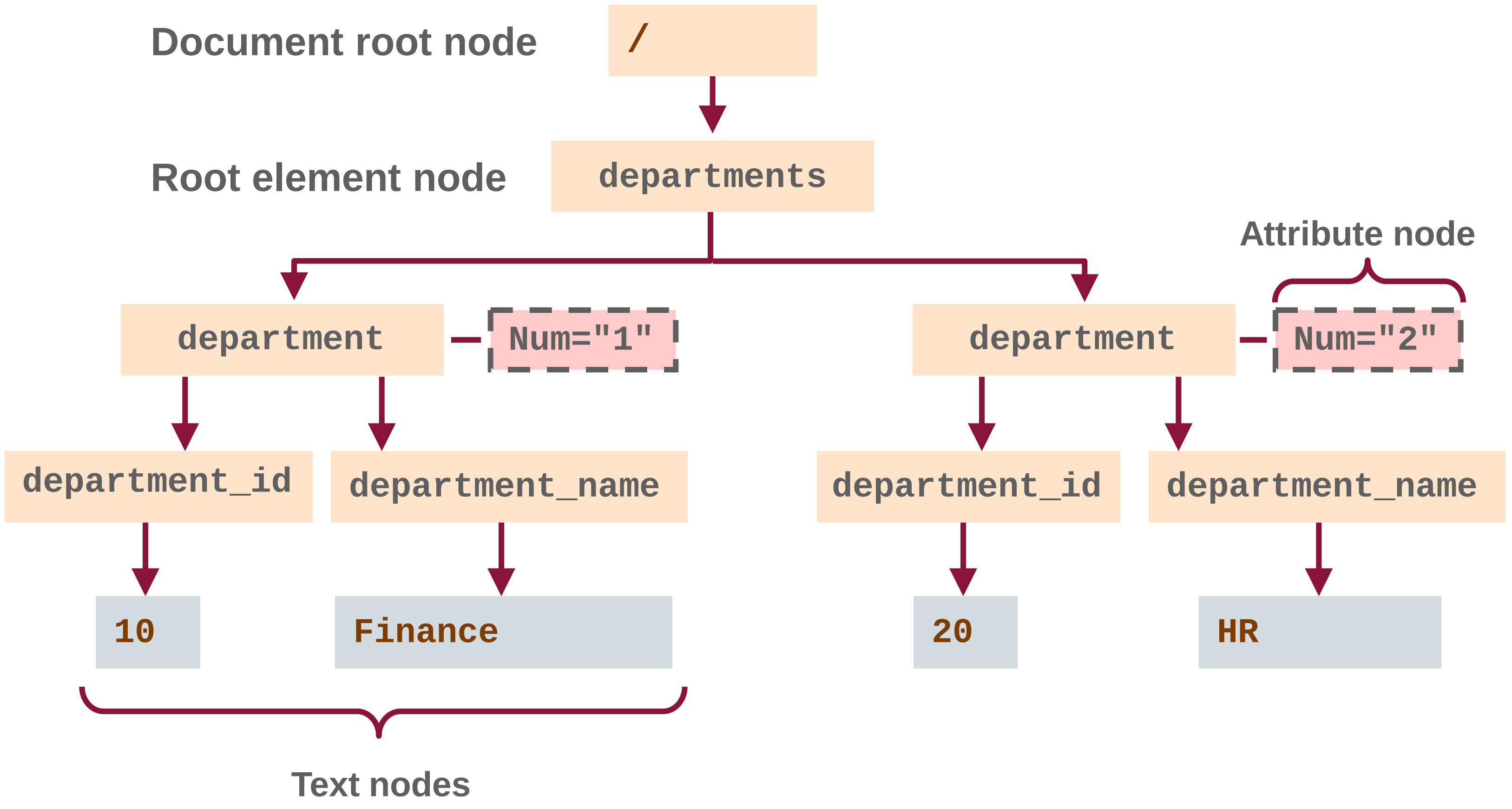
Terminology and Concepts

- **XPath:** A language used for addressing (finding, retrieving, or manipulating) values in XML sources.
- **XML Stylesheet Language for Transformations (XSLT):** A language used for transforming data from one XML structure to another.
- **XQuery:** A language designed for querying (finding and extracting elements and attributes), transforming, and accessing XML and relational data.

XPath

- Treats XML documents as trees of nodes
- Uses path expressions and conditions to select nodes or node-sets in an XML document
- Is a fundamental component of XSLT and XQuery

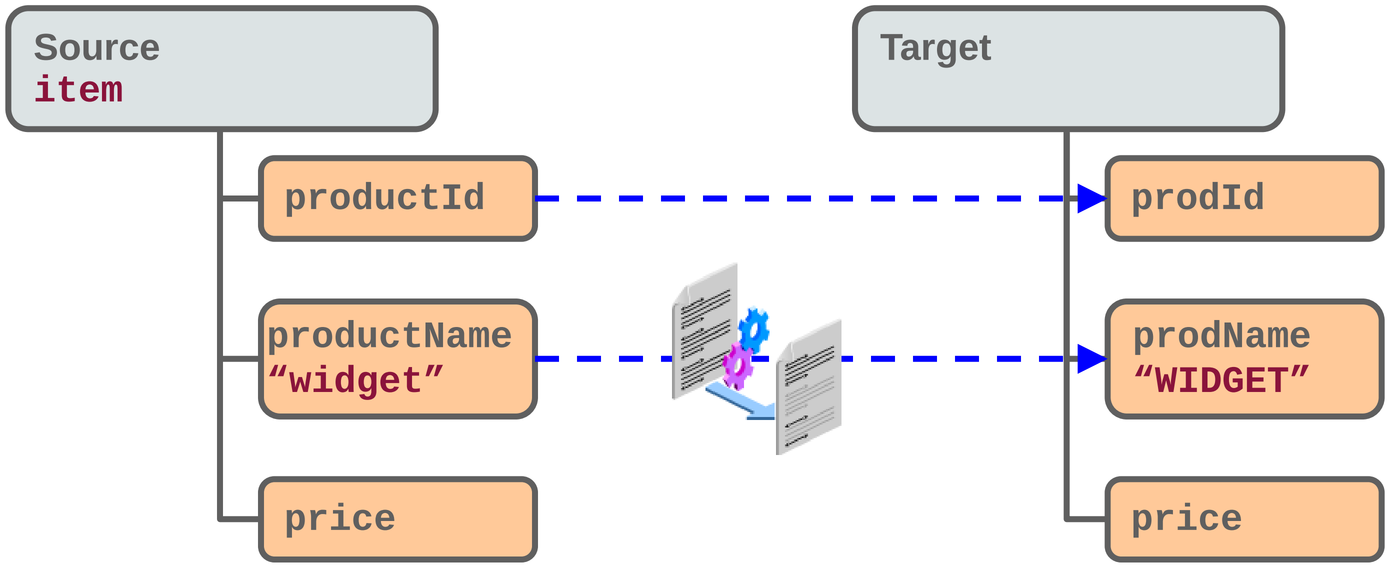
XPath: Examples



Example:

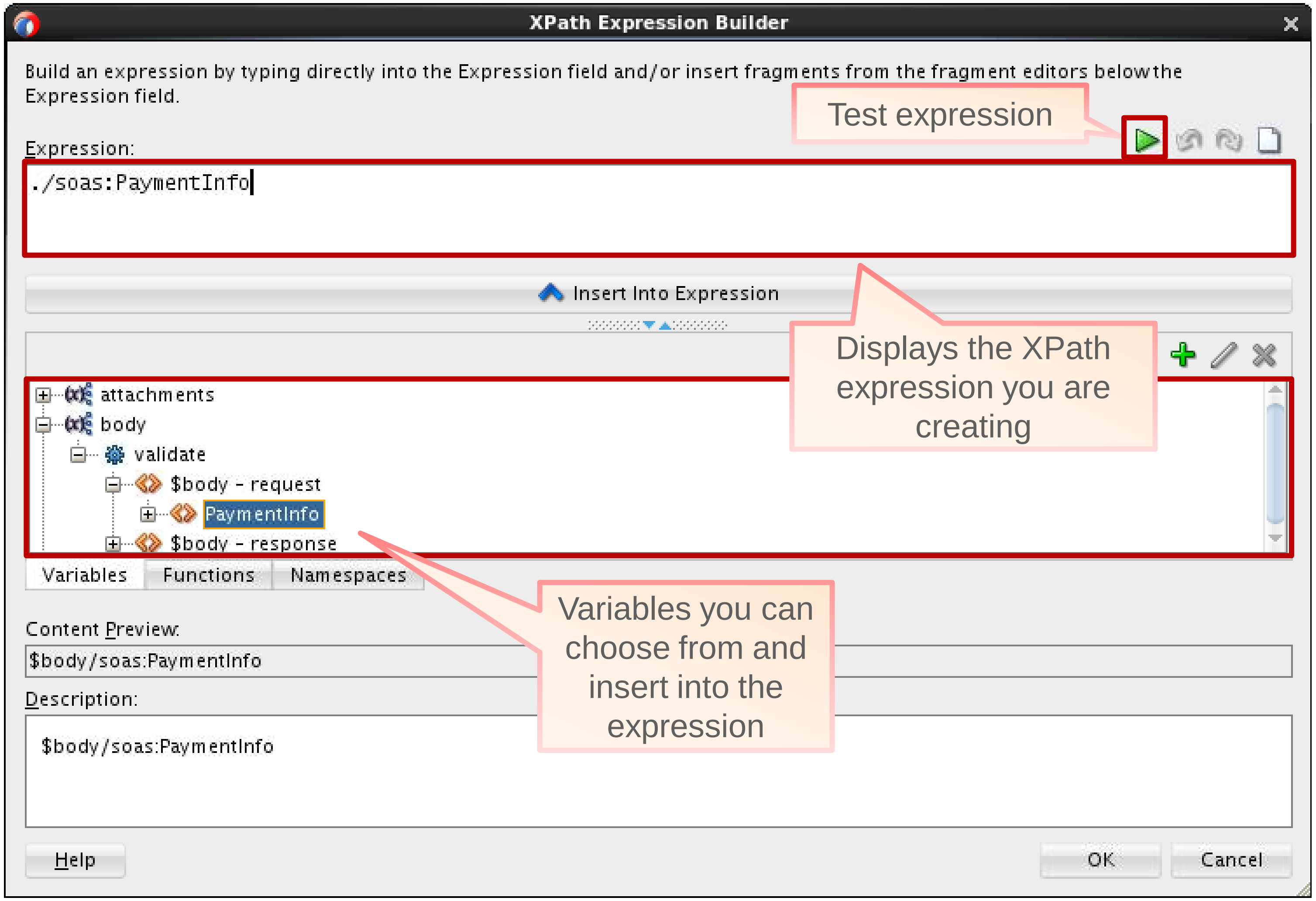
- /departments/department/department_id
- /departments/department[@Num=“1”]

XPath Functions



```
<ns1:item>
    <ns1:prodId>
        <xsl:value-of select="imp1:productID"/>
    </ns1:prodId>
    <ns1:prodName>
        <xsl:value-of select ="xp20:upper-case(imp1:productName)"/>
    </ns1:prodName>
</ns1:item>
```

XPath Expression Builder



XPath Expression Builder

Build an expression by typing directly into the Expression field and/or insert fragments from the fragment editors below the Expression field.

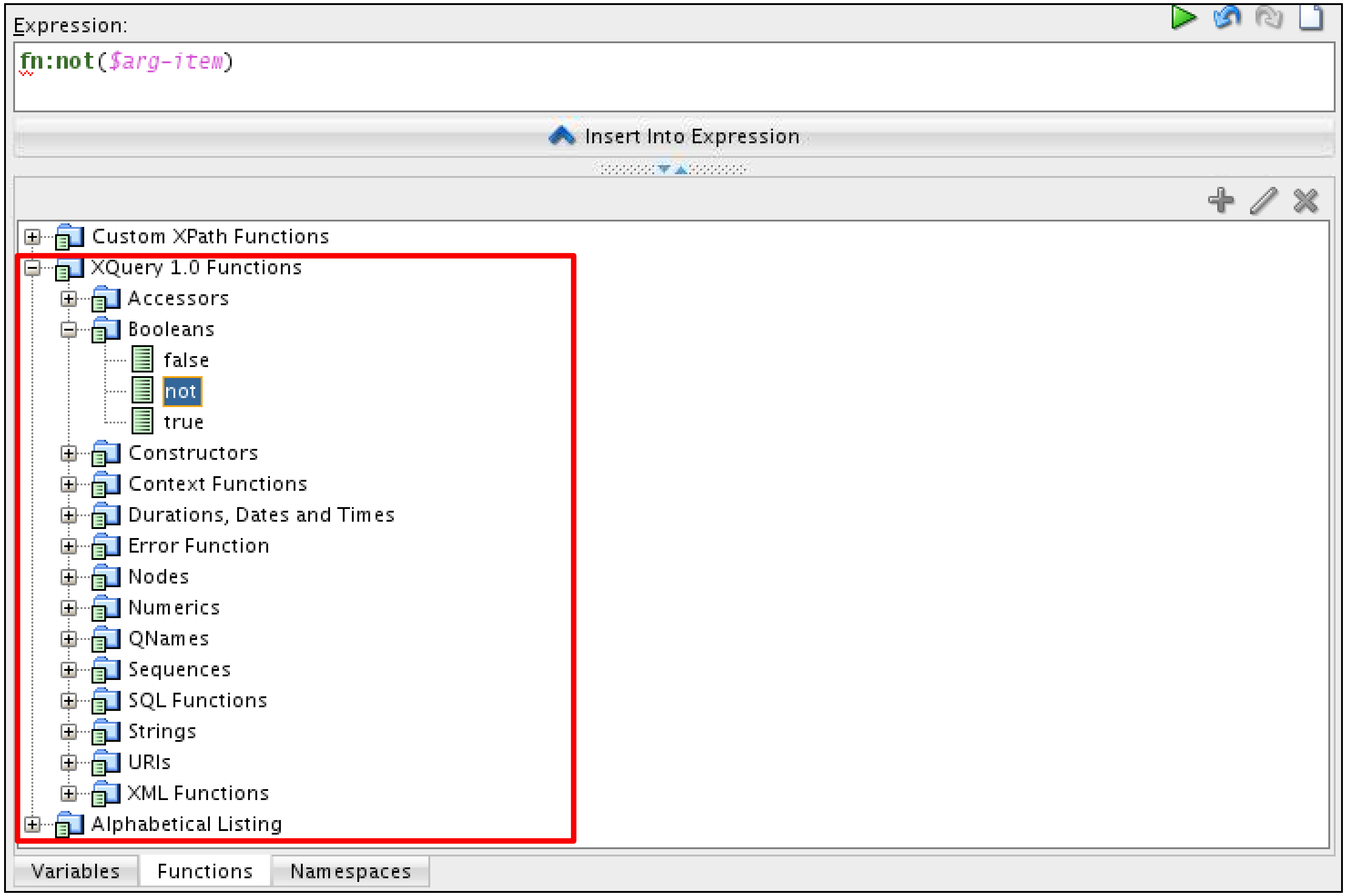
Expression:

```
./soas:PaymentInfo
```

Content Preview: $body/soas:PaymentInfo

Description: $body/soas:PaymentInfo

Standard XPath Functions



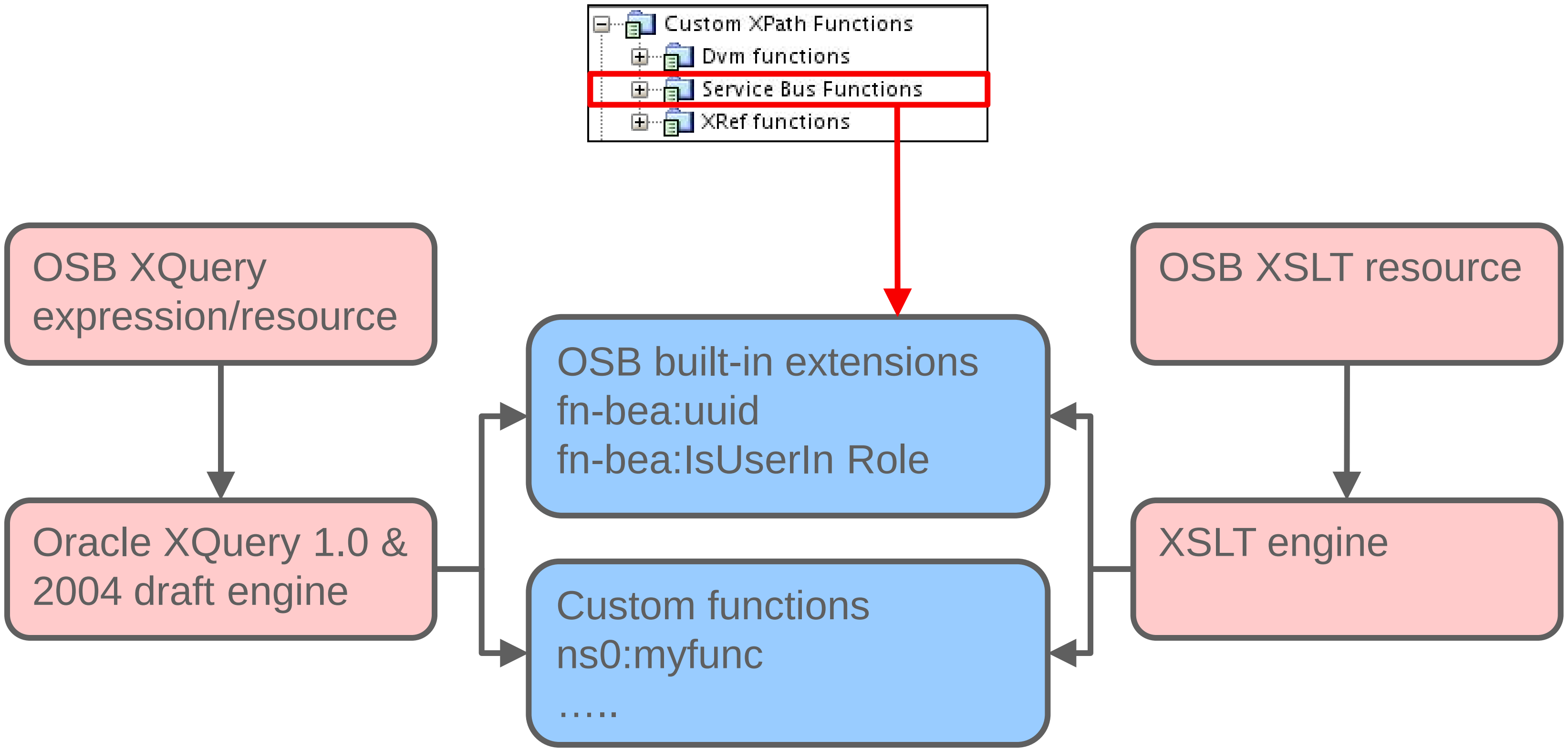
Expression:

```
fn:not($arg-item)
```

Note: XPath 2.0, XQuery 1.0, and XSLT 2.0 share the same functions library.

Custom XPath Functions

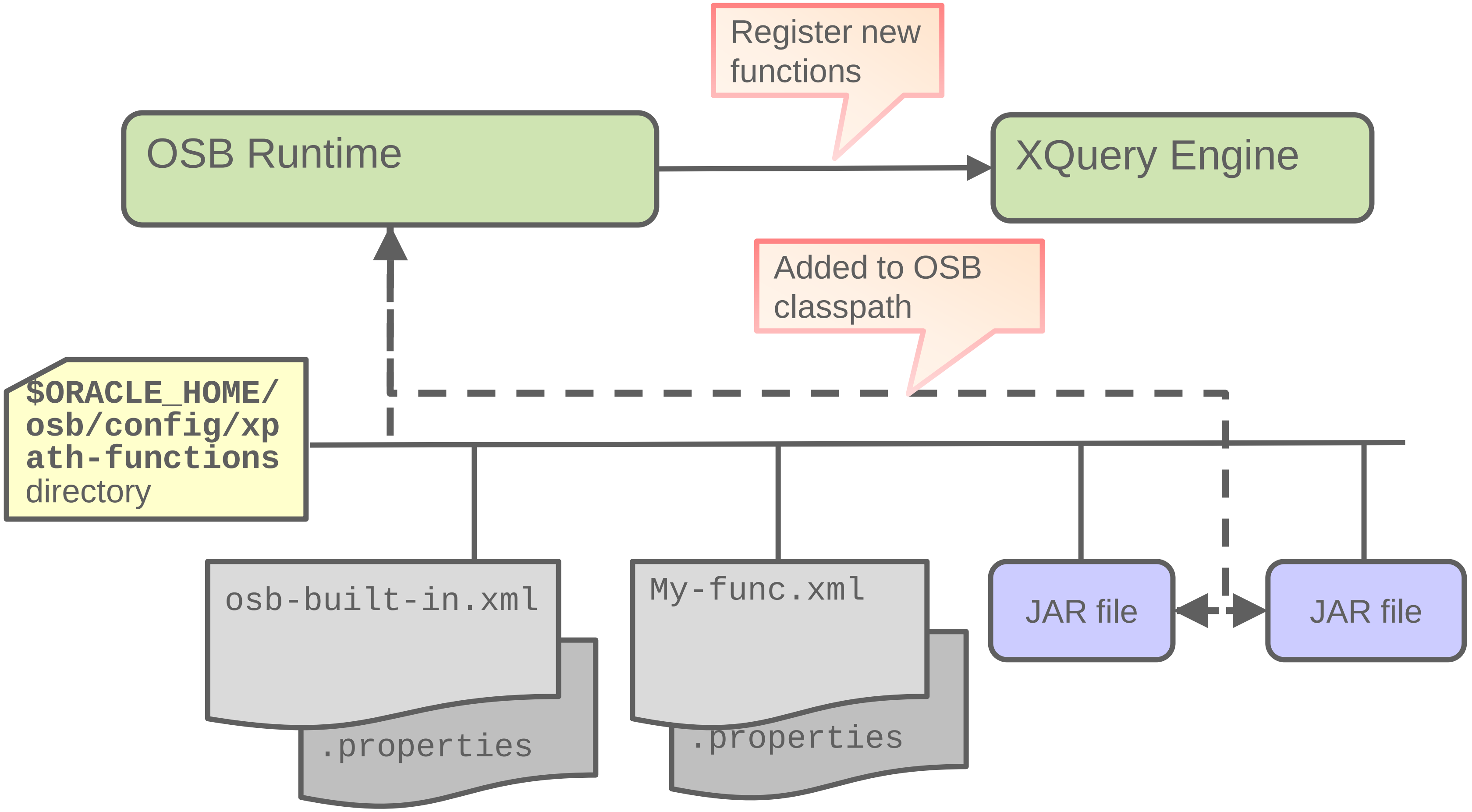
Service Bus provides an extensible framework for creating custom XPath functions that you can use in the XQuery Expression Editors.



Creating and Packaging Custom XPath Functions

1. Create Java class and method for a custom function.
2. Package the custom function class in a JAR file.
3. Place the JAR file in the OSB XPath-functions directory:
   `$ORACLE_HOME/osb/config/xpath-functions/`
4. Register the custom function.

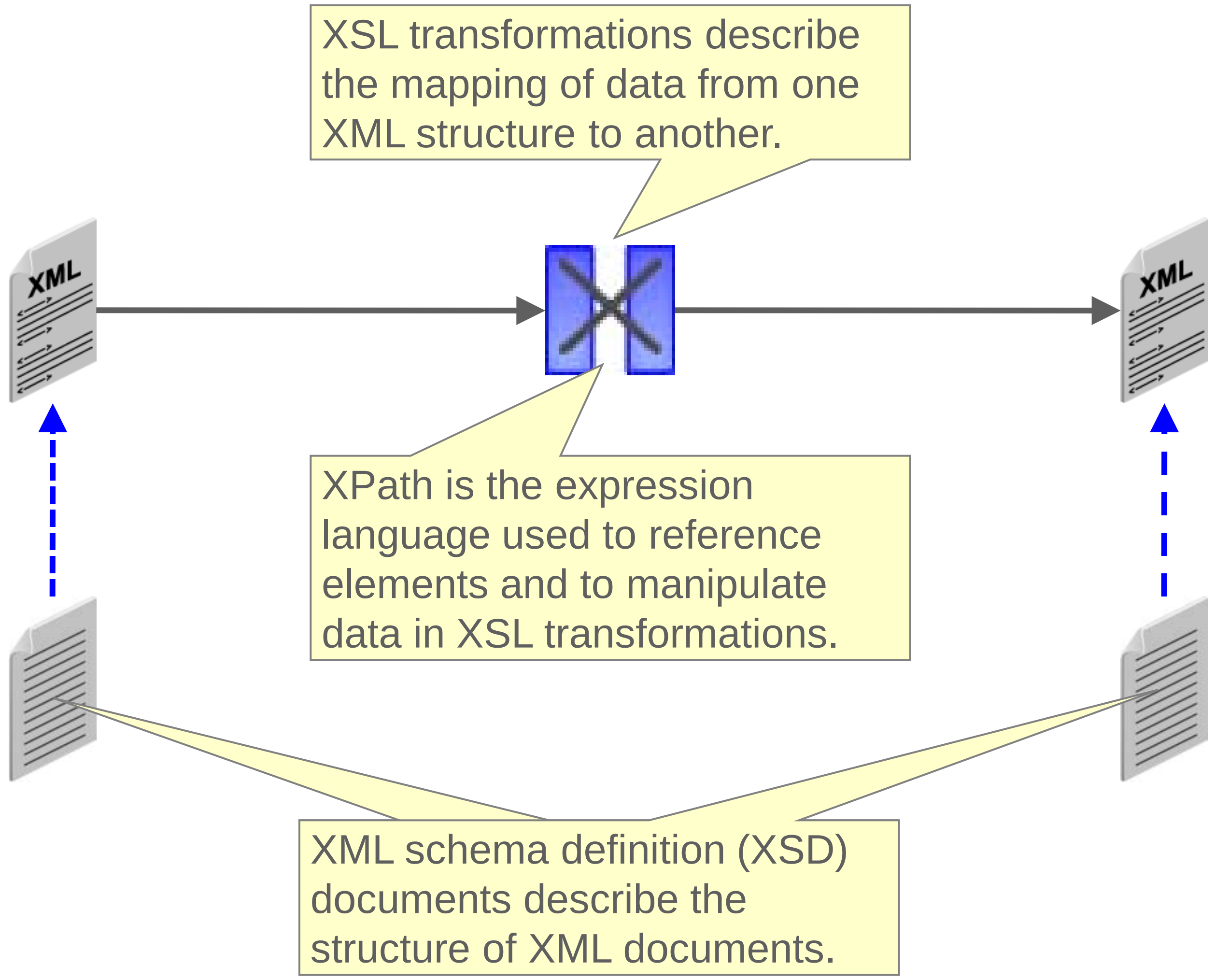
Registering a Custom Function



Agenda

- XPath
- XSLT transformation
- XQuery transformation
- nXSD

Data Standards



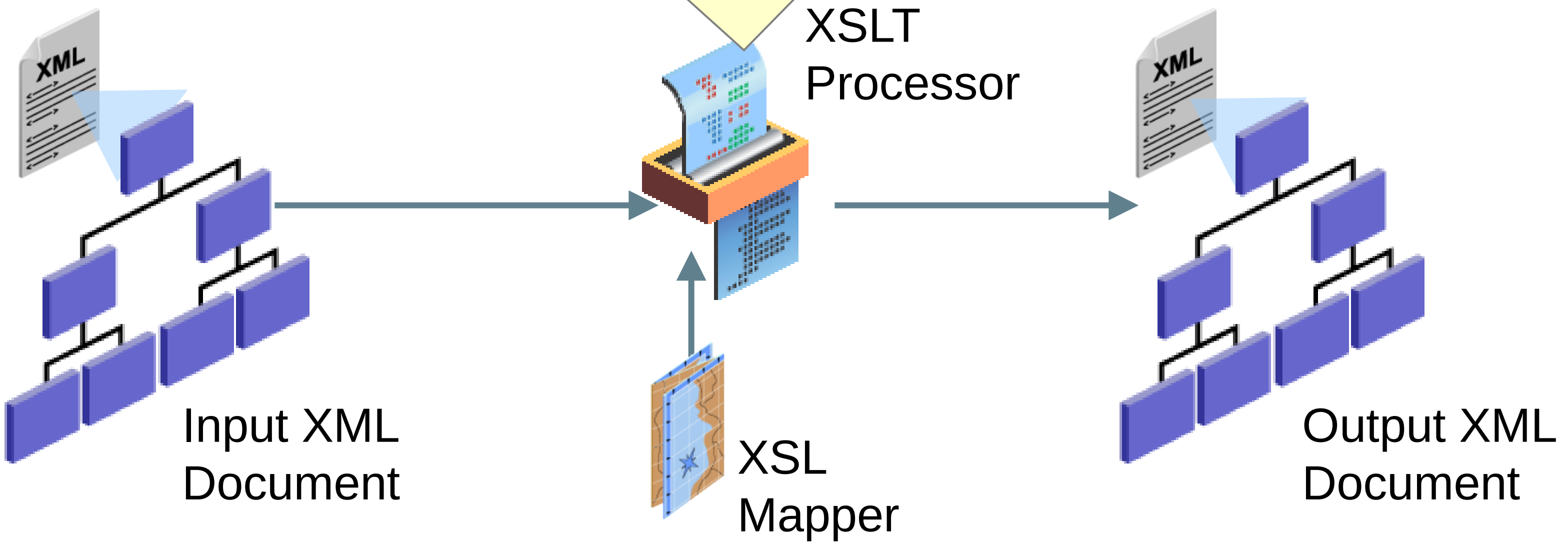
XSL transformations describe the mapping of data from one XML structure to another.

XPath is the expression language used to reference elements and to manipulate data in XSL transformations.

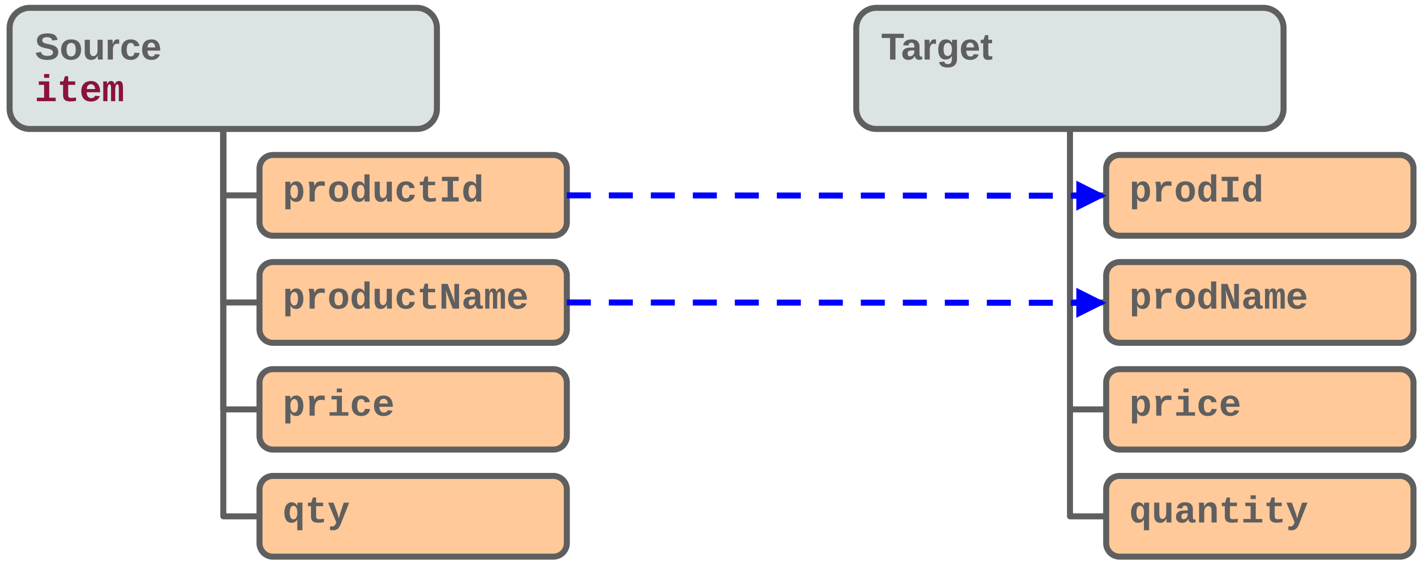
XML schema definition (XSD) documents describe the structure of XML documents.

Transforming Data

eXtensible Stylesheet Language Transformation (XSLT) describes the mapping of data from one XML structure to another.

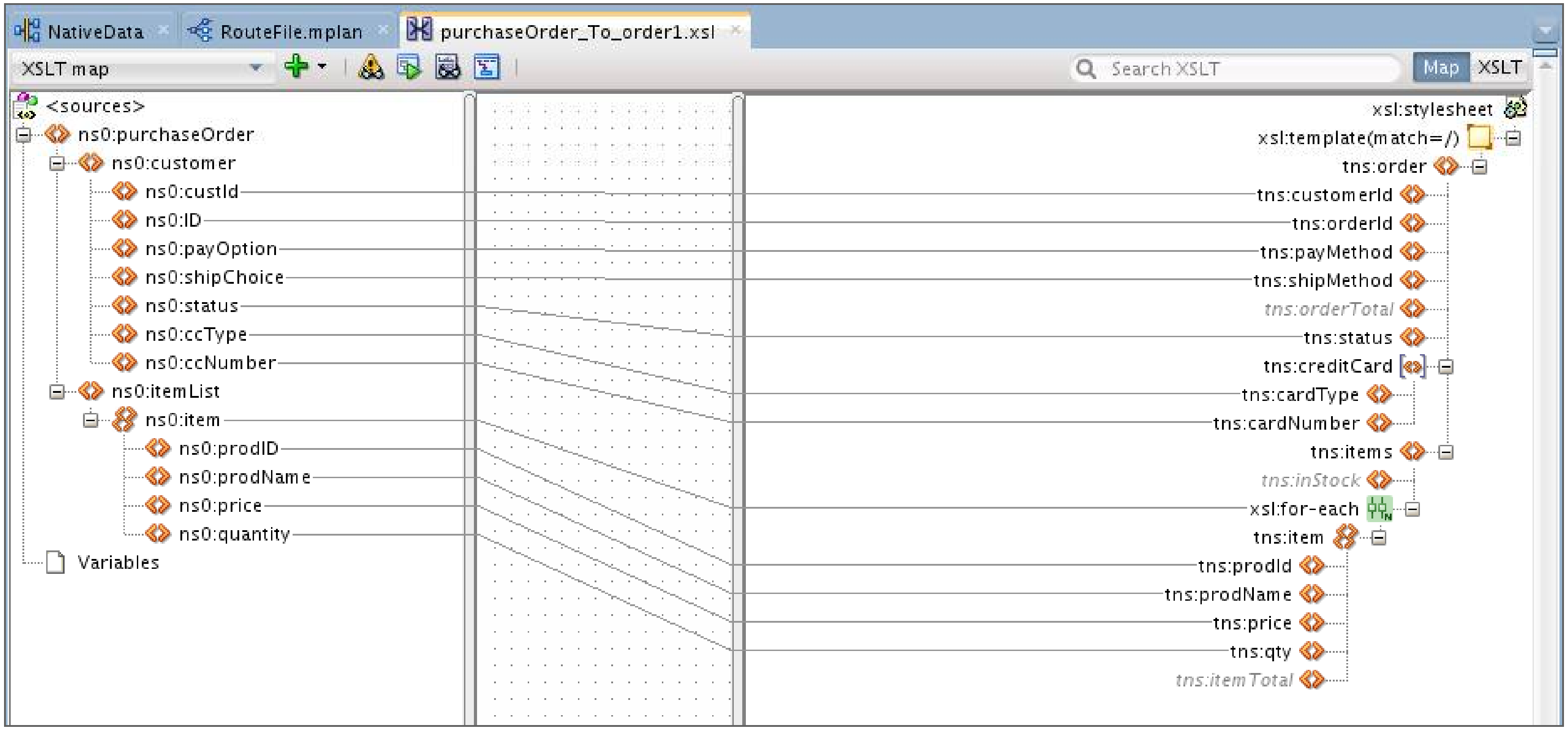


XSL Transformations

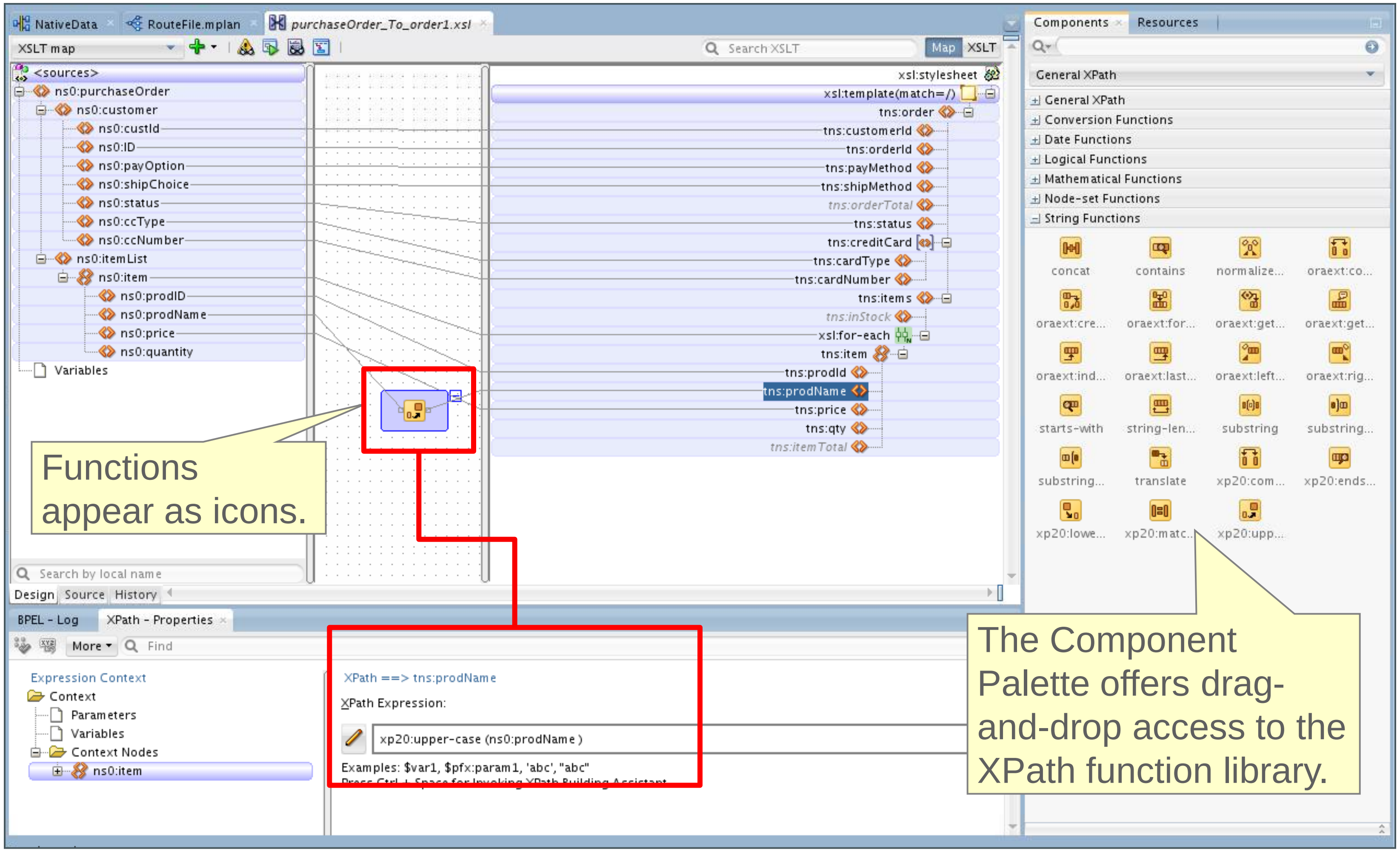


```
<ns1:item>
    <ns1:prodId>
        <xsl:value-of select="imp1:productID"/>
    </ns1:prodId>
    <ns1:prodName>
        <xsl:value-of select="imp1:productName"/>
    </ns1:prodName>
</ns1:item>
```

XSLT Mapper


NativeData
RouteFile.mplan
purchaseOrder_To_order1.xsl
XSLT map
Search XSLT
Map
XSLT
<sources>
ns0:purchaseOrder
ns0:customer
ns0:custId
ns0:ID
ns0:payOption
ns0:shipChoice
ns0:status
ns0:ccType
ns0:ccNumber
ns0:itemList
ns0:item
ns0:prodID
ns0:prodName
ns0:price
ns0:quantity
Variables
xsl:stylesheet
xsl:template(match=/)
tns:order
tns:customerId
tns:orderId
tns:payMethod
tns:shipMethod
tns:orderTotal
tns:status
tns:creditCard
tns:cardType
tns:cardNumber
tns:items
tns:inStock
xsl:for-each
tns:item
tns:prodId
tns:prodName
tns:price
tns:qty
tns:itemTotal

Using XPath Functions in XSLT Mapper



XPath ==> tns:prodName

XPath Expression:

```
xp20:upper-case (ns0:prodName )
```

Examples: $var1, $pfx:param1, 'abc', "abc"

Press Ctrl + Space for Invoking XPath Building Assistant

Functions appear as icons.

The Component Palette offers drag-and-drop access to the XPath function library.

Agenda

- XPath
- XSLT transformation
- XQuery transformation
- nXSD

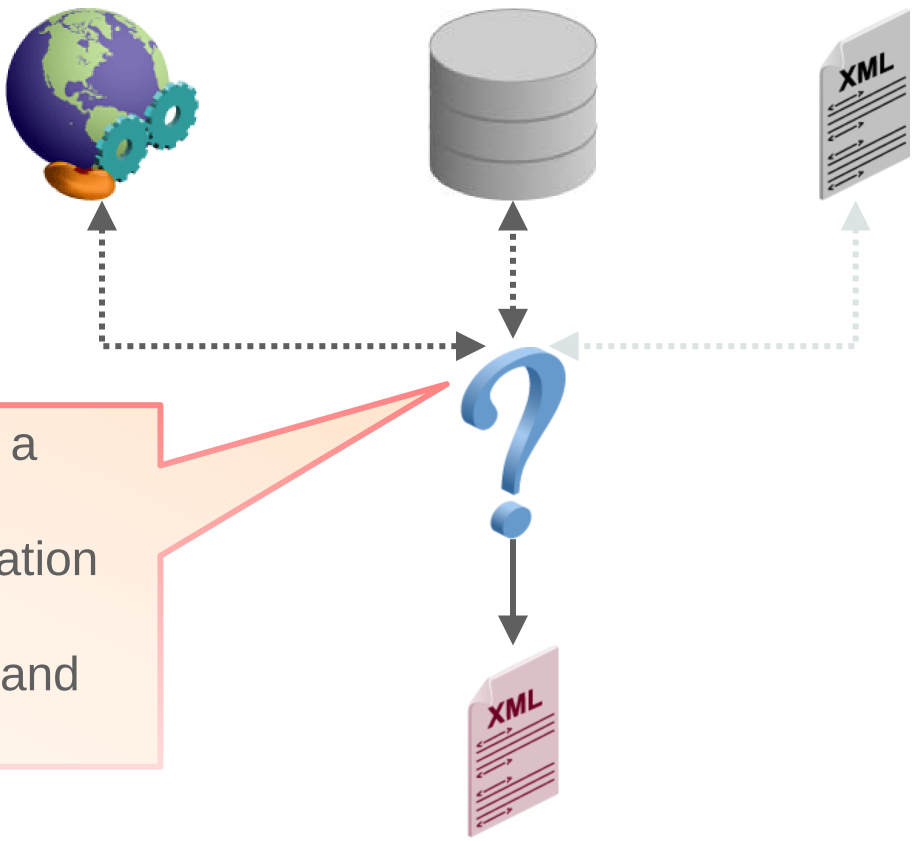
XQuery: Introduction

XQuery:

- Allows you to select the XML data elements of interest, reorganize and transform them, and return the results in a structure of your choosing
- Uses and extends XPath to help navigate and extract elements and attributes from an XML document

Uses of XQuery

- XQuery is built on XPath expressions.
- Applications of XQuery include:
  - XML querying
  - XML generation
  - XML transformation



XQuery provides a mechanism for extracting information from relational databases, files, and web services.

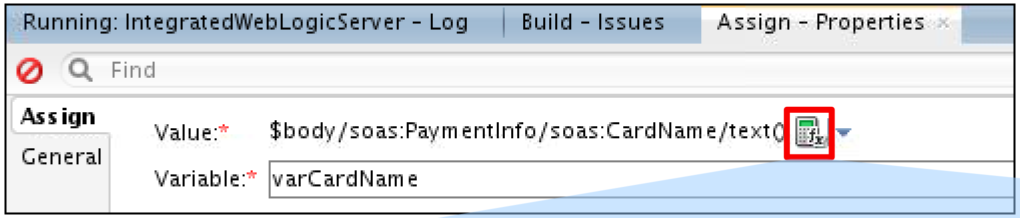
XQuery Support in Service Bus

Service Bus:

- Makes use of XQuery resources for various activities, like transformations, data selection, condition evaluation, and data manipulation
- Fully supports:
  - XQuery 1.0
  - XQuery 2004

Tool Type	Recommended Use	JDeveloper	Service Bus Console
XQuery Expression Builder	Script transformations using XQuery	X	X
XQuery Mapper	Create complex mappings	X	

XQuery Expression Builder

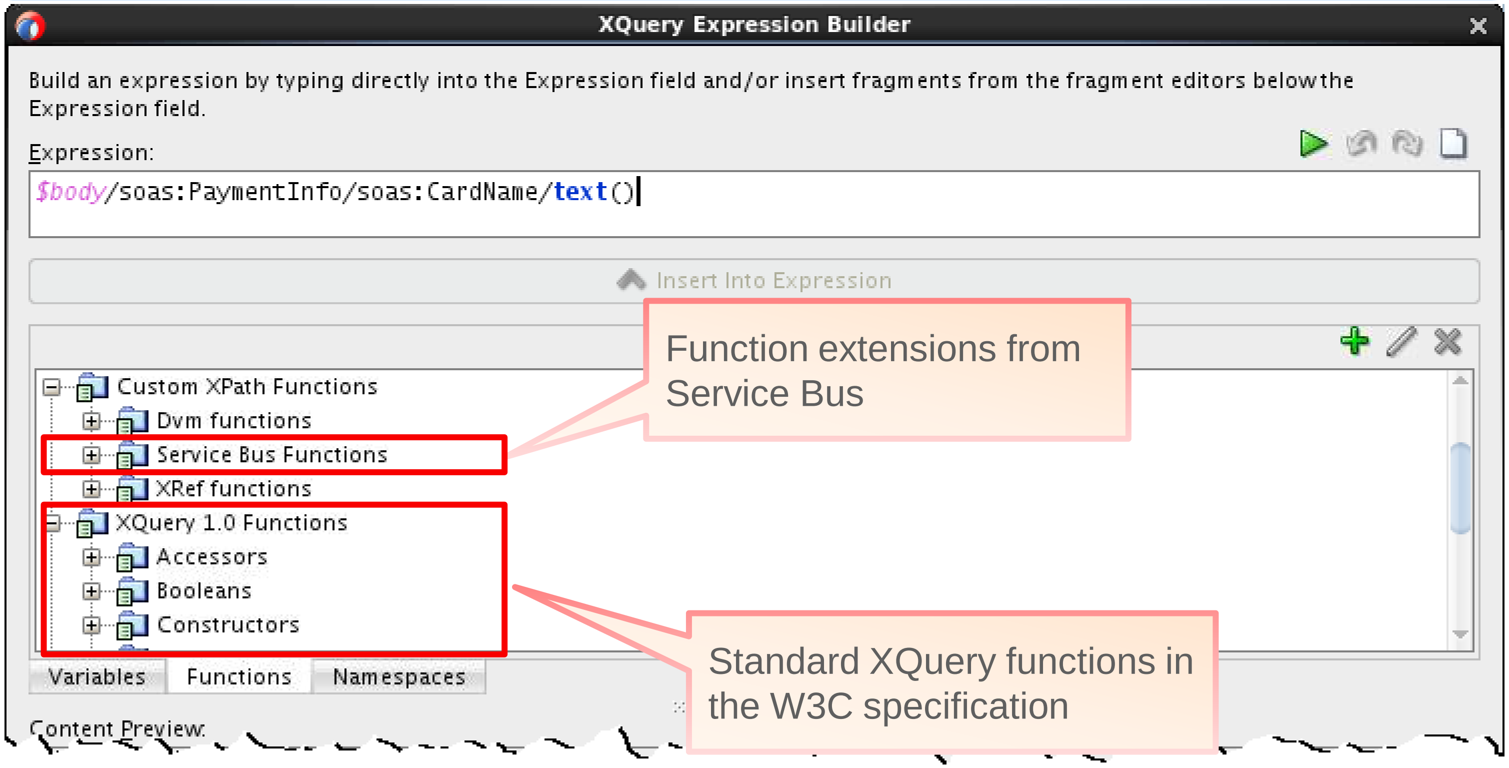


Running: IntegratedWebLogicServer - Log | Build - Issues | Assign - Properties

Find

Assign
General

Value:* $body/soas:PaymentInfo/soas:CardName/text()
Variable:* varCardName



XQuery Expression Builder

Build an expression by typing directly into the Expression field and/or insert fragments from the fragment editors below the Expression field.

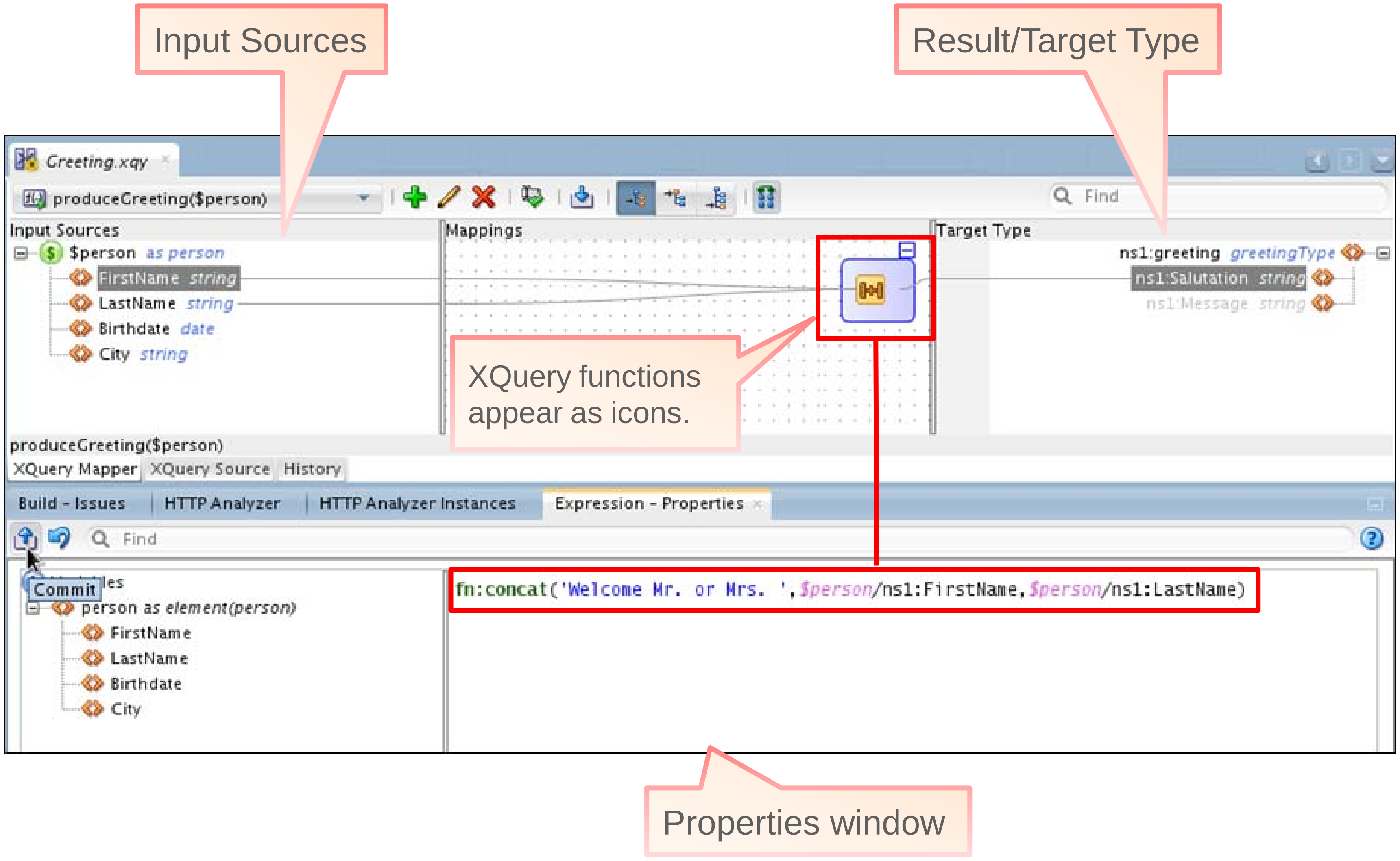
Expression:

```
$body/soas:PaymentInfo/soas:CardName/text()
```

Function extensions from Service Bus

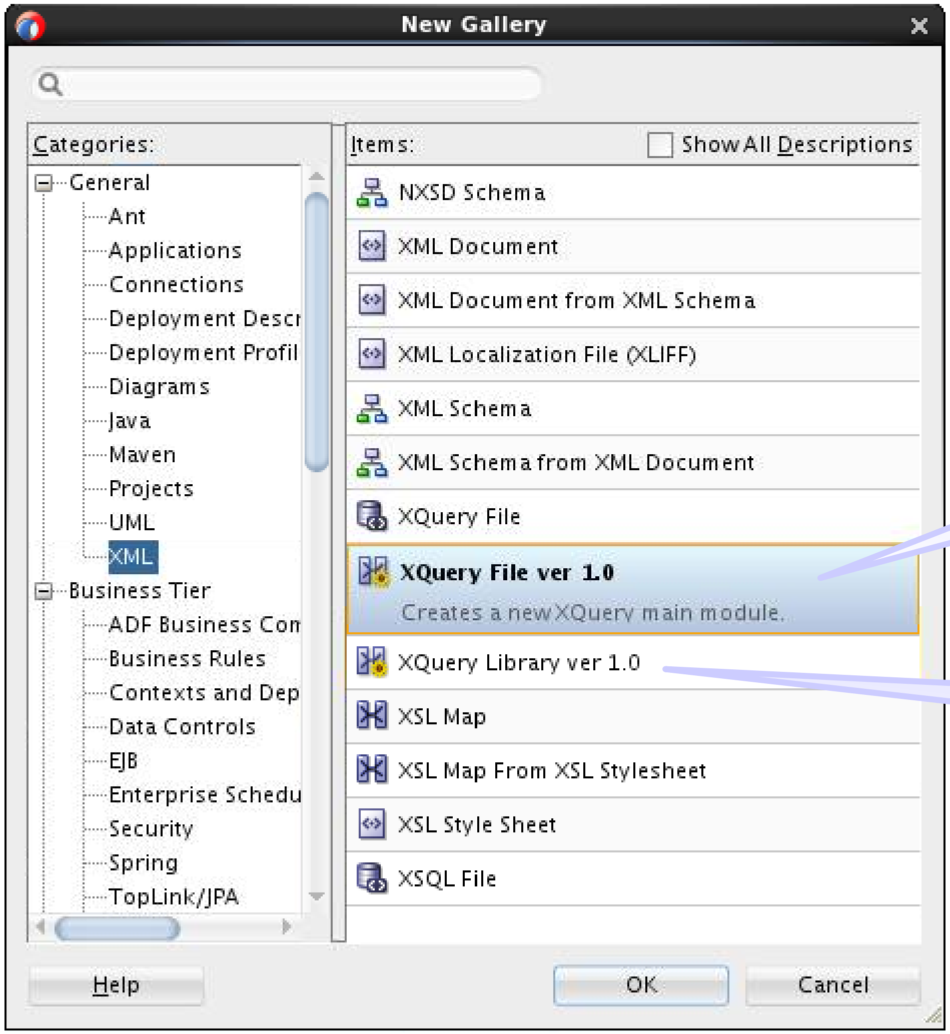
Standard XQuery functions in the W3C specification

XQuery Mapper



```
fn:concat('Welcome Mr. or Mrs. ',$person/ns1:FirstName,$person/ns1:LastName)
```

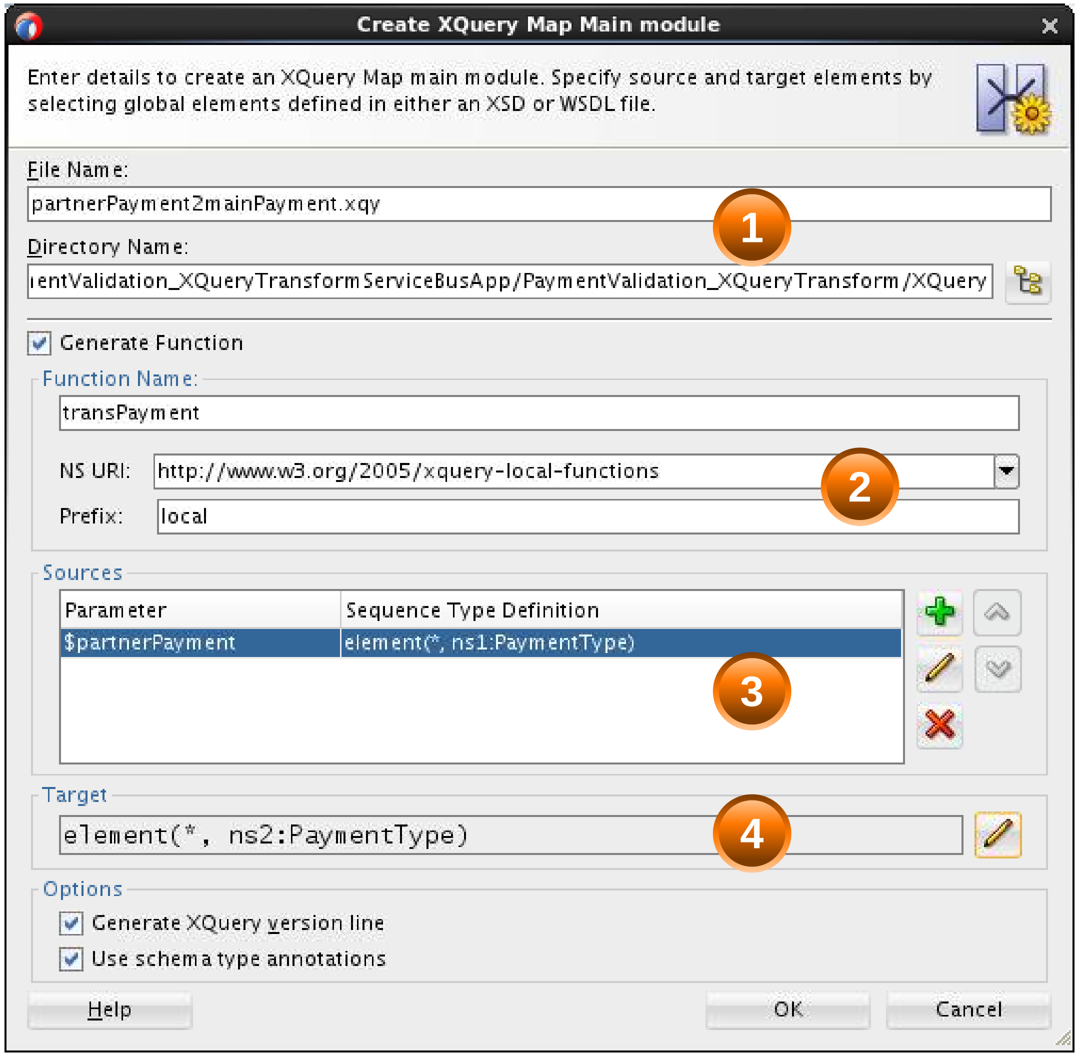
Types of XQuery Maps



XQuery main module

XQuery library module

Steps for Creating XQuery Map Files



Create XQuery Map Main module

Enter details to create an XQuery Map main module. Specify source and target elements by selecting global elements defined in either an XSD or WSDL file.

File Name: partnerPayment2mainPayment.xqy

Directory Name: ientValidation_XQueryTransformServiceBusApp/PaymentValidation_XQueryTransform/XQuery

[✓] Generate Function

Function Name: transPayment

NS URI: http://www.w3.org/2005/xquery-local-functions

Prefix: local

Sources

Parameter	Sequence Type Definition
$partnerPayment	element(*, ns1:PaymentType)

Target: element(*, ns2:PaymentType)

Options

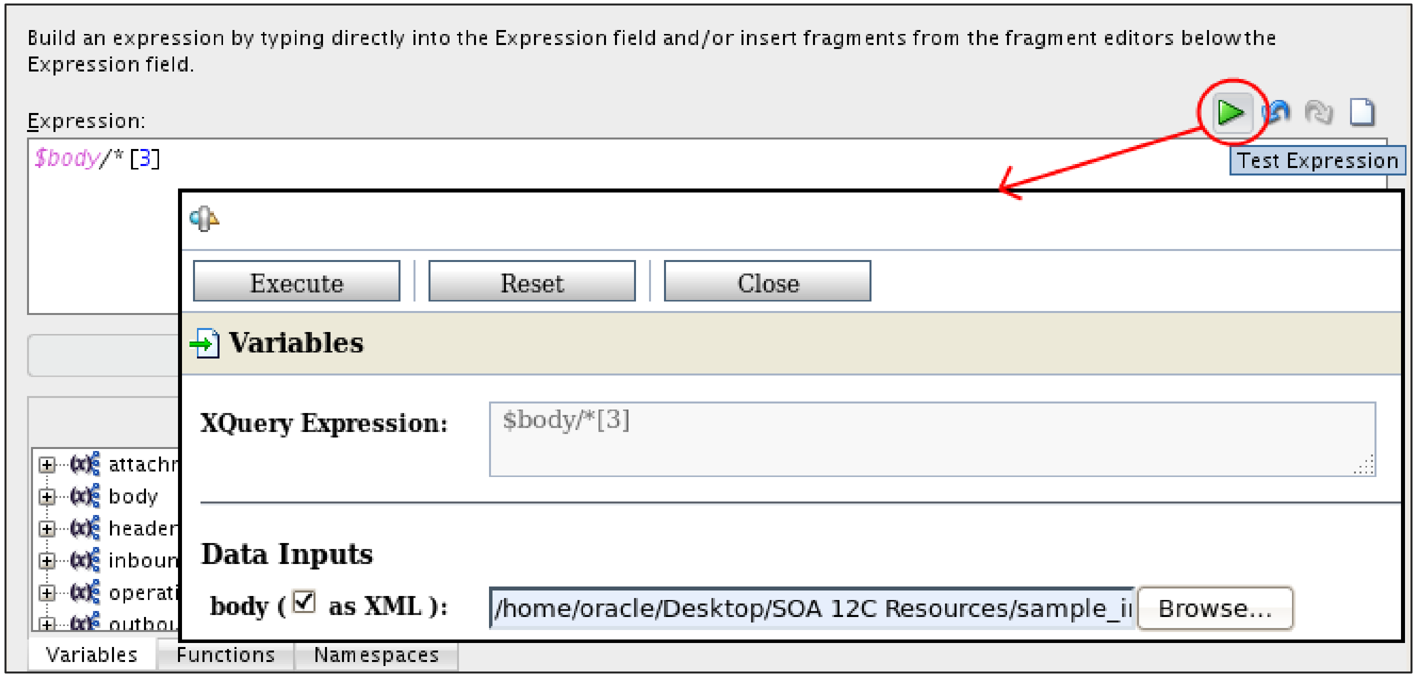
[✓] Generate XQuery version line

[✓] Use schema type annotations

XSLT Versus XQuery

- Similarities:
  - XPath
  - Data model
  - Functions and operators
- Differences:
  - Syntax
    - XQuery is similar to SQL.
    - XSLT stylesheets use XML syntax.
  - Performance
    - XSLT loads the entire input document in the memory.
    - XQuery loads only the objects that need to be used by the current statement.

Test Expressions at Design-Time



Build an expression by typing directly into the Expression field and/or insert fragments from the fragment editors below the Expression field.

Expression:

```
$body/* [3]
```

XQuery Expression: $body/*[3]

Data Inputs

body (✓ as XML): /home/oracle/Desktop/SOA 12C Resources/sample_i

Agenda

- XPath
- XSLT transformation
- XQuery transformation
- nXSD

XML to non-XML with nXSD

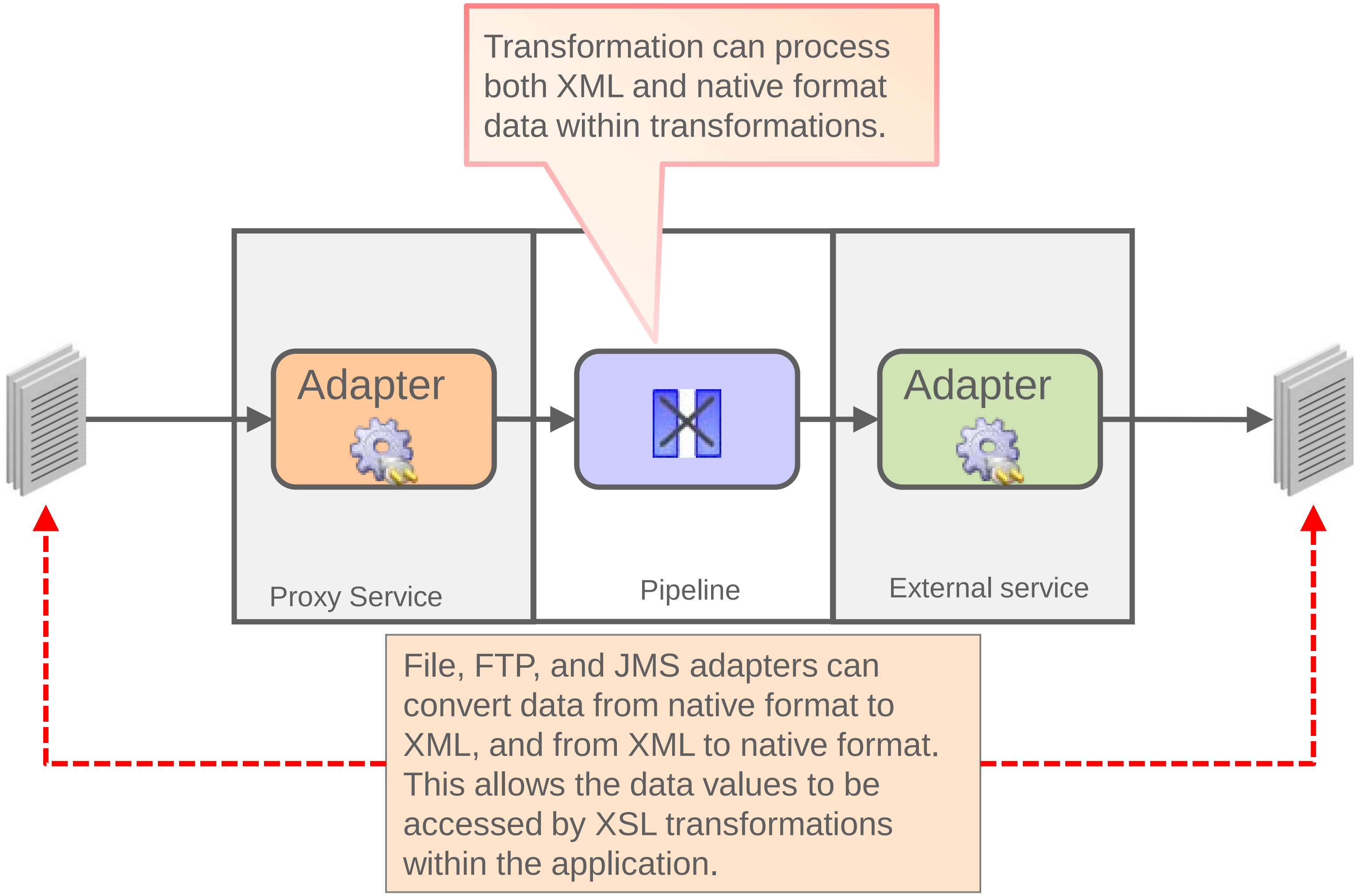
Requirement:

- Non-XML-based data should be processed by the Service Bus, and should be treated as structured.

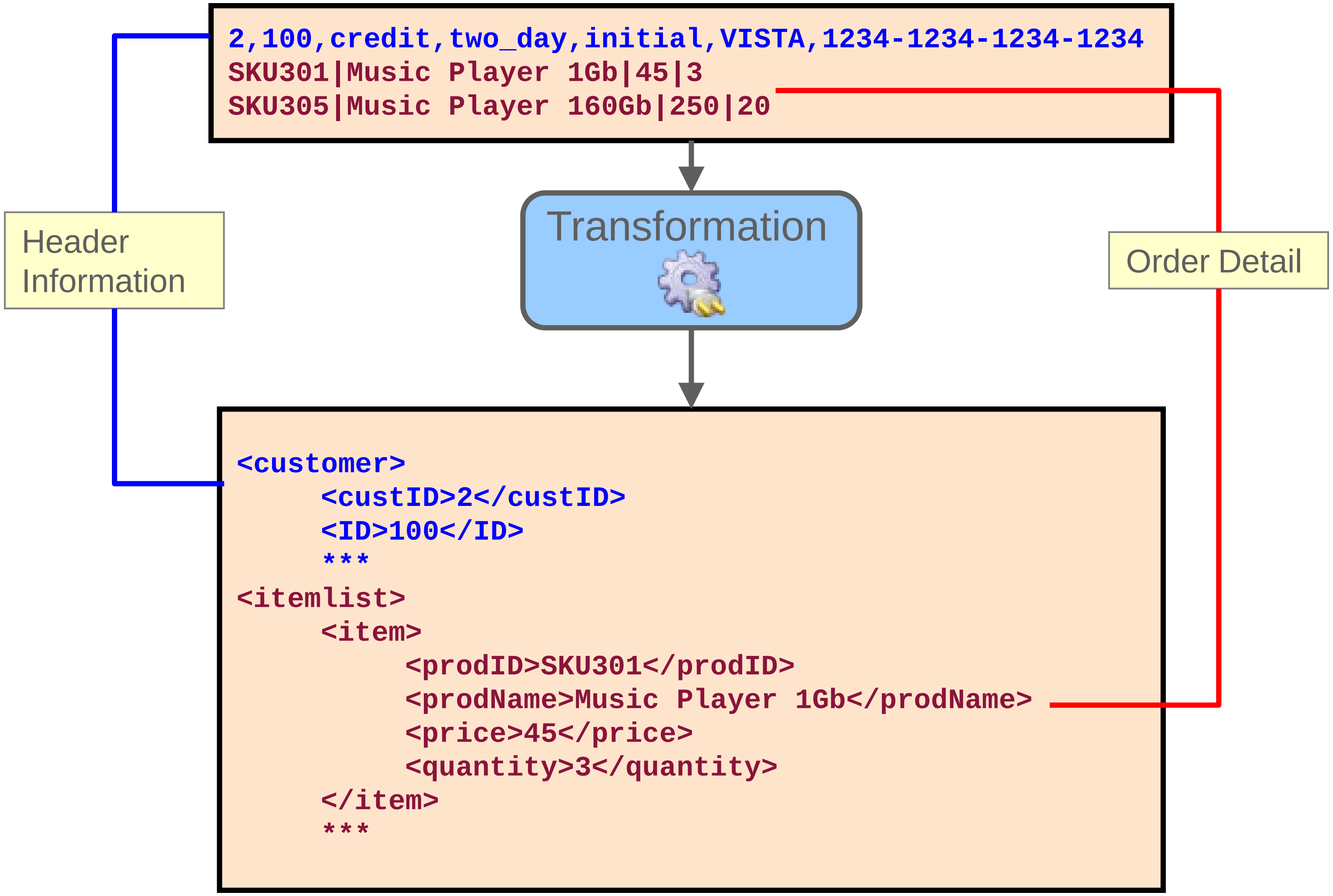
Solution options:

- nXSD
- MFL

Working with Native Format Data


Transformation can process both XML and native format data within transformations.
Adapter
Adapter
Proxy Service
Pipeline
External service
File, FTP, and JMS adapters can convert data from native format to XML, and from XML to native format. This allows the data values to be accessed by XSL transformations within the application.

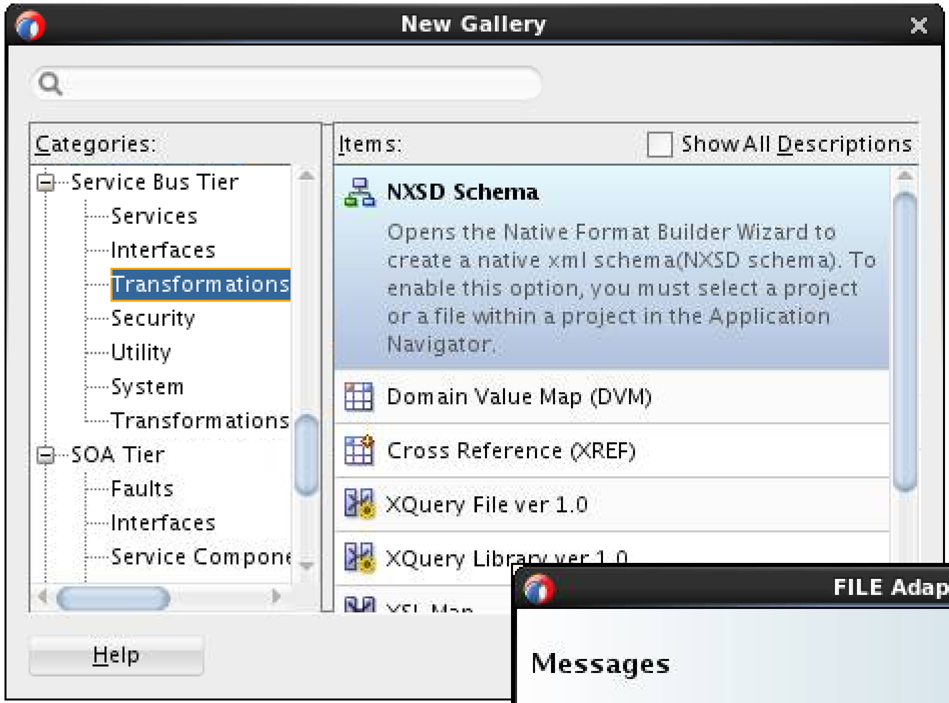
Native Data Transformation



```
2,100,credit,two_day,initial,VISTA,1234-1234-1234-1234
SKU301|Music Player 1Gb|45|3
SKU305|Music Player 160Gb|250|20
```

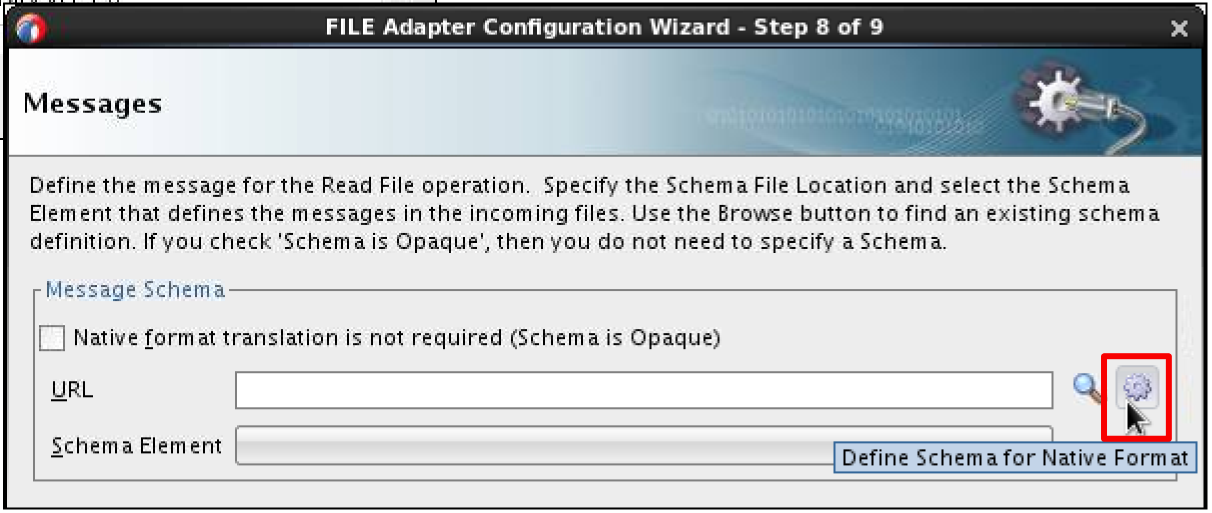
```
<customer>
     <custID>2</custID>
     <ID>100</ID>
     ***
<itemlist>
     <item>
          <prodID>SKU301</prodID>
          <prodName>Music Player 1Gb</prodName>
          <price>45</price>
          <quantity>3</quantity>
     </item>
     ***
```

Starting Native Format Builder Wizard



NXSD Schema

Opens the Native Format Builder Wizard to create a native xml schema(NXSD schema). To enable this option, you must select a project or a file within a project in the Application Navigator.

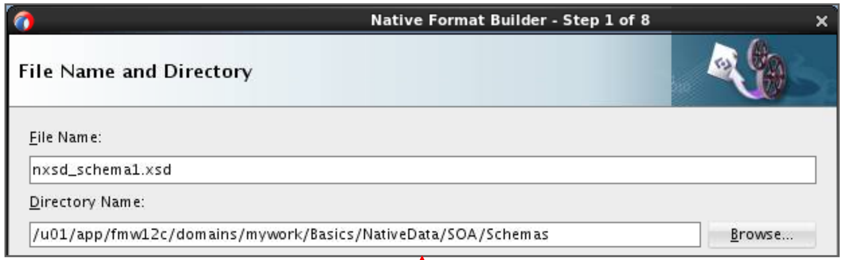


FILE Adapter Configuration Wizard - Step 8 of 9
Messages

Define the message for the Read File operation. Specify the Schema File Location and select the Schema Element that defines the messages in the incoming files. Use the Browse button to find an existing schema definition. If you check 'Schema is Opaque', then you do not need to specify a Schema.

Message Schema
Native format translation is not required (Schema is Opaque)
URL
Schema Element
Define Schema for Native Format

Specifying File Name and Native Data Format



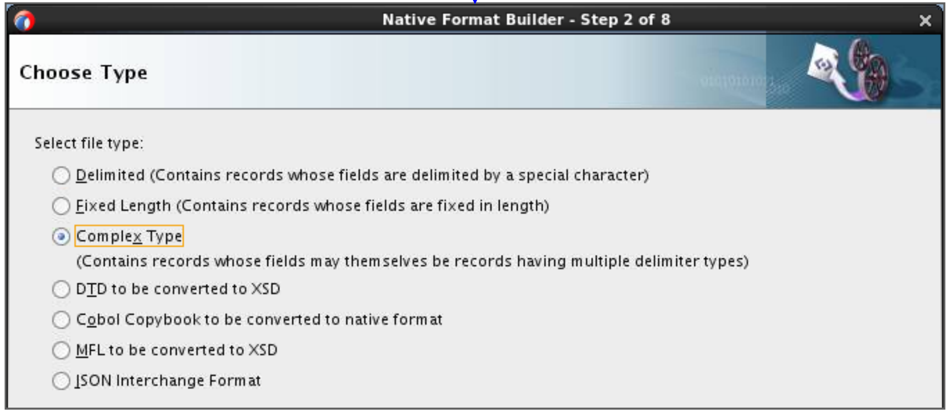
Native Format Builder - Step 1 of 8

File Name and Directory

File Name: nxsd_schema1.xsd

Directory Name: /u01/app/fmw12c/domains/mywork/Basics/NativeData/SOA/Schemas

Specify the name of the file to create, and the type of native data to process.



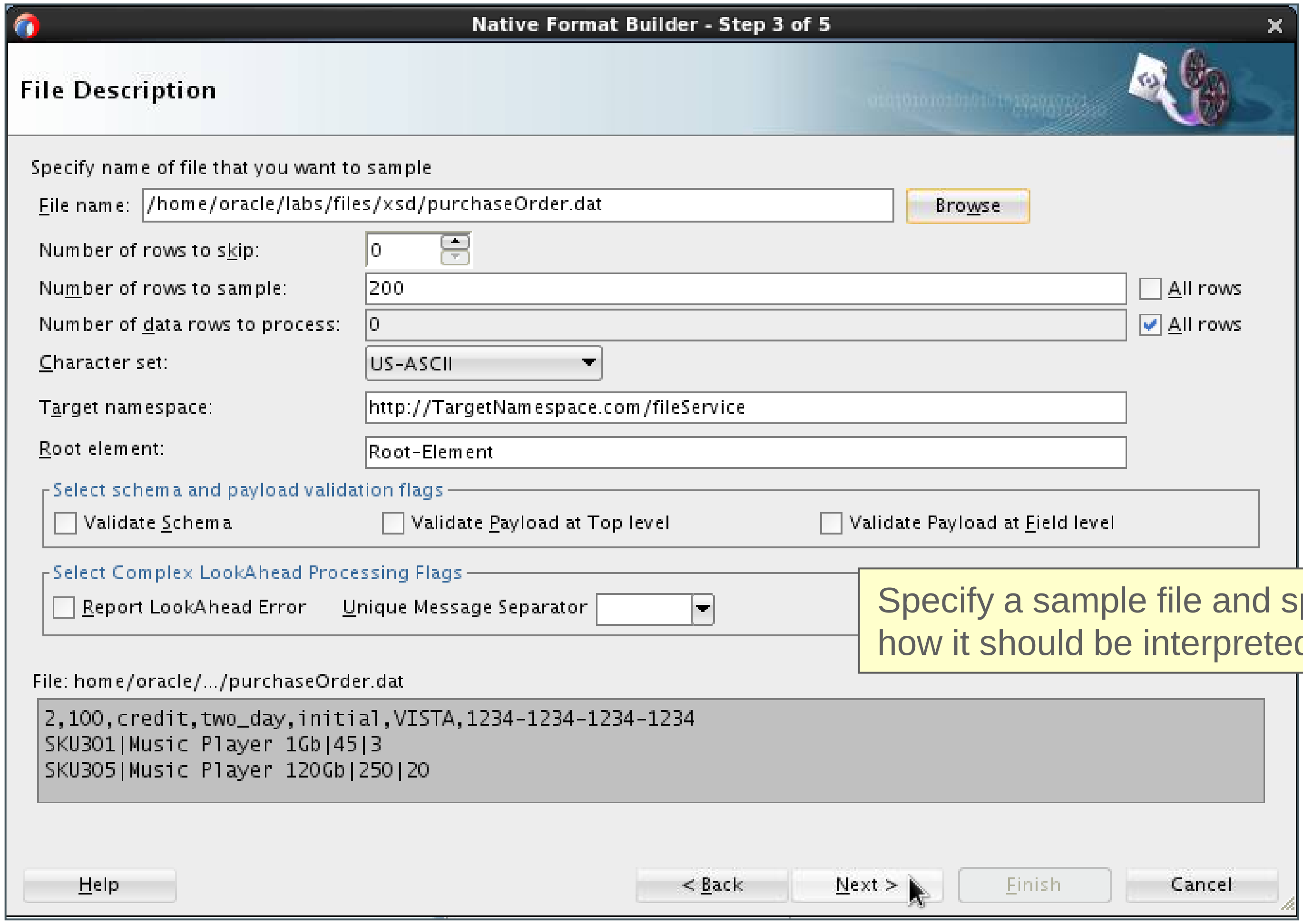
Native Format Builder - Step 2 of 8

Choose Type

Select file type:

- ( ) Delimited (Contains records whose fields are delimited by a special character)
- ( ) Fixed Length (Contains records whose fields are fixed in length)
- [✓ Complex Type] (Contains records whose fields may themselves be records having multiple delimiter types)
- ( ) DTD to be converted to XSD
- ( ) Cobol Copybook to be converted to native format
- ( ) MFL to be converted to XSD
- ( ) JSON Interchange Format

Specifying a Sample File



Native Format Builder - Step 3 of 5

File Description

Specify name of file that you want to sample

File name: /home/oracle/labs/files/xsd/purchaseOrder.dat

Number of rows to skip: 0

Number of rows to sample: 200 [ ] All rows

Number of data rows to process: 0 [✓] All rows

Character set: US-ASCII

Target namespace: http://TargetNamespace.com/fileService

Root element: Root-Element

Select schema and payload validation flags

[ ] Validate Schema [ ] Validate Payload at Top level [ ] Validate Payload at Field level

Select Complex LookAhead Processing Flags

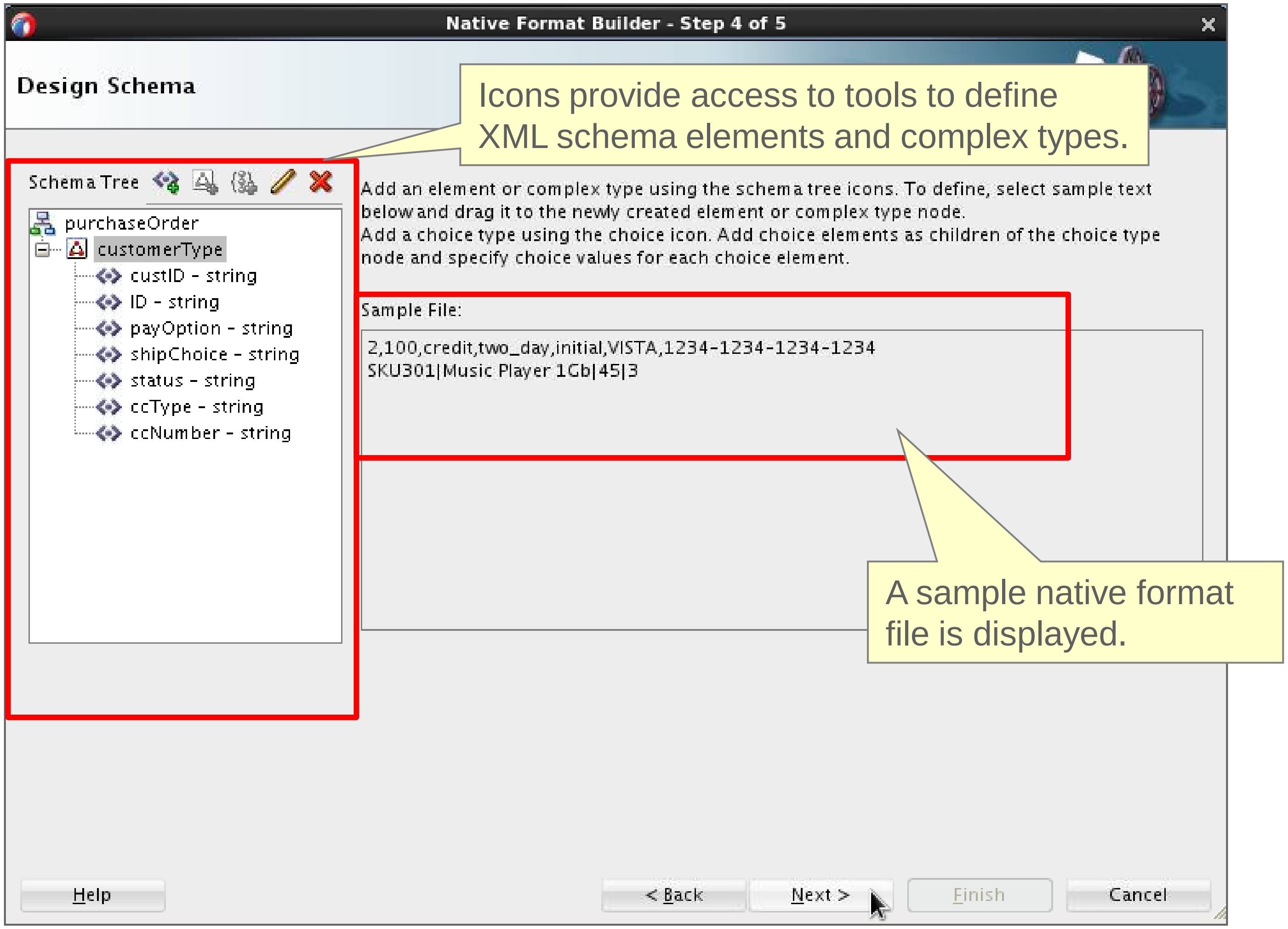
[ ] Report LookAhead Error Unique Message Separator

File: home/oracle/.../purchaseOrder.dat

```
2,100,credit,two_day,initial,VISTA,1234-1234-1234-1234
SKU301|Music Player 1Gb|45|3
SKU305|Music Player 120Gb|250|20
```

Specify a sample file and specify how it should be interpreted.

Defining a Schema for a Native Format



Native Format Builder - Step 4 of 5

Design Schema

Icons provide access to tools to define XML schema elements and complex types.

Add an element or complex type using the schema tree icons. To define, select sample text below and drag it to the newly created element or complex type node.
Add a choice type using the choice icon. Add choice elements as children of the choice type node and specify choice values for each choice element.

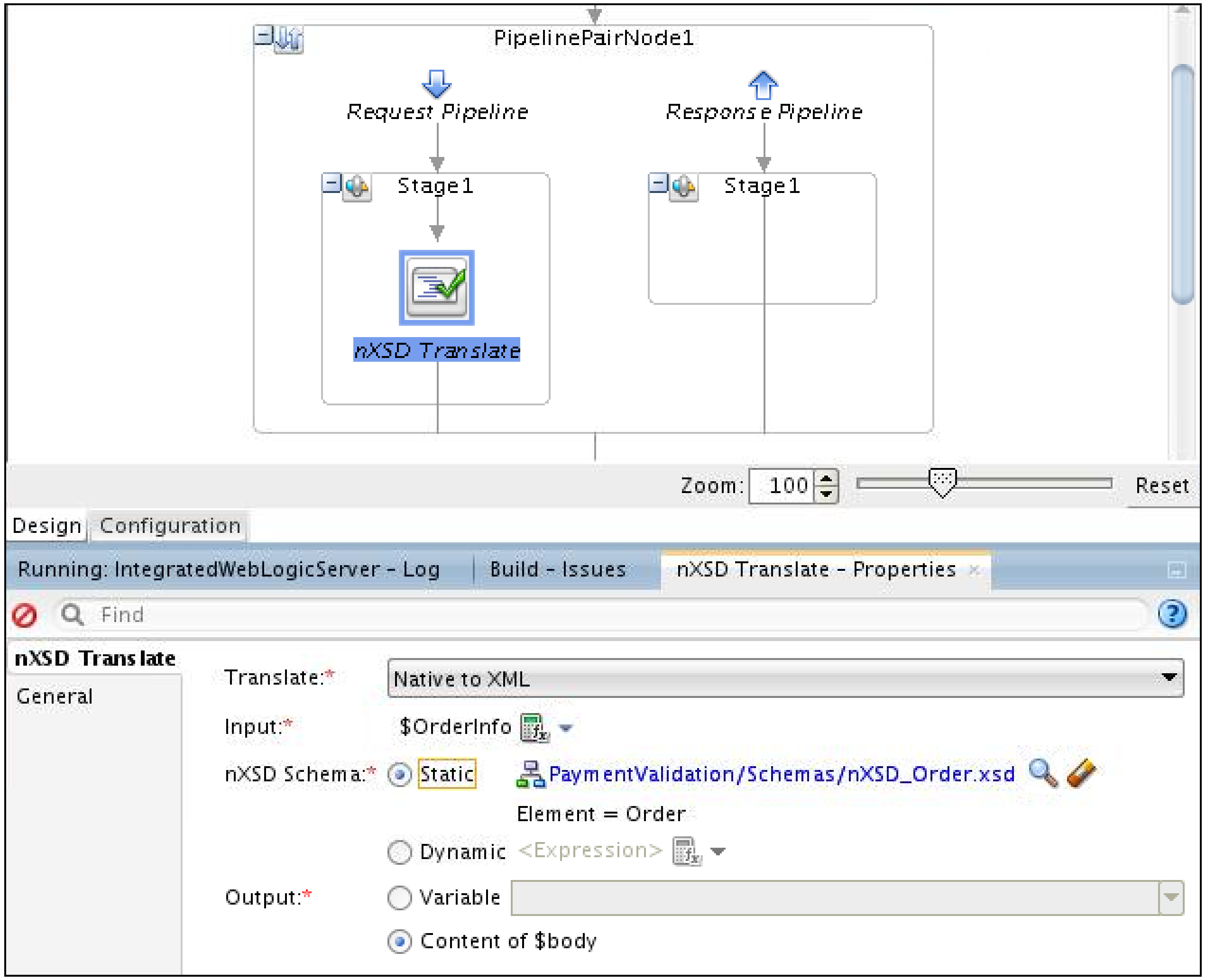
```
purchaseOrder
  customerType
    custID - string
    ID - string
    payOption - string
    shipChoice - string
    status - string
    ccType - string
    ccNumber - string
```

Sample File:

```
2,100,credit,two_day,initial,VISTA,1234-1234-1234-1234
SKU301|Music Player 1Gb|45|3
```

A sample native format file is displayed.

nXSD Translate Action



Translate:* Native to XML

Input:* $OrderInfo

nXSD Schema:* [✓ Static] PaymentValidation/Schemas/nXSD_Order.xsd

Element = Order

Dynamic <Expression>

Output:* Variable

[✓ Content of $body]

Summary

In this lesson, you should have learned how to:

- Describe the role of XPath, XQuery, and XSLT in the way Service Bus handles data
- Describe standard and custom XPath functions
- Describe how the XSLT Mapper can be used to create XSL transformations
- Use the XQuery Mapper to create XQuery transformations
- Transform non-XML data to XML data with nXSD

Practice 6: Overview

- 6-1: Transforming Data Using XQuery Transformation
- 6-2: Configuring a File Adapter Proxy with nXSD Transformation
- Appendix: Building the nXSD Transformation from the Beginning