WebSphere Message Broker V8 Scenarios
Version 1 Release 0

Converting a message map to a graphical data map

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 37.

Contents

Figures

Chapter 1. Introduction to the "Converting a message map to a graphical data map" scenario

This scenario shows how you can convert a message map (`.msgmap`) file to a graphical data map (`.map`)) type, what the differences between the mapping transforms are, and how to configure converted transforms.

About this task

Message maps are used in WebSphere® Message Broker Version 6.1 and WebSphere Message Broker Version 7.0. Graphical data maps are used in WebSphere Message Broker Version 8.0 and later. Graphical data maps benefit from being based on XML schema and XPath standards, which provide a greater range of transforms, including XPath 2.0 expressions.

You cannot edit or develop WebSphere Message Broker Version 6.0 or Version 7.0 message maps in WebSphere Message Broker Version 8.0 or later versions. To edit message maps by using the Graphical Data Mapping editor as part of WebSphere Message Broker Version 8.0 and later, you must convert your message maps to graphical data maps.

Review the topics to complete the scenario:

- “Context”
- “Technical solution” on page 3
  - “Before: initial message map behavior” on page 4
  - “After: the converted graphical data map” on page 7
- Chapter 2, “Implementing the solution,” on page 9
  - “Creating the scenario message map configuration” on page 10
  - “Converting the message map” on page 11
  - “Configuring the converted transforms” on page 17
- Chapter 3, “Verifying the solution,” on page 29

Context

This scenario explains how to convert a message map to a graphical data map, how to check that all transforms work as intended in the mapping solution, and explains some of the logic of the mapping so that you can make decisions about your mapping solutions.

The scenario message map contains a simple transformation that takes an input generic message with details of a person and transforms it to a message required by a member database system that adds a member record for the person. In the converted map, you can see automatically converted transforms, and transforms that must be configured manually.

The message map that is used in this scenario demonstrates:

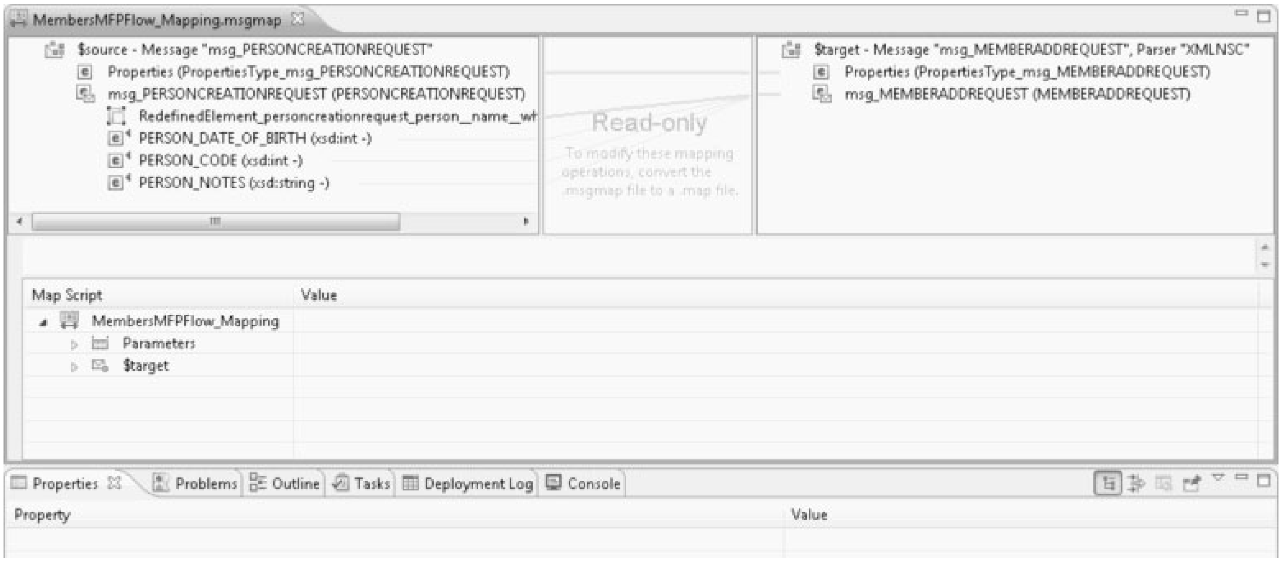
- The differences between message maps and graphical data maps
- Which transform types you can benefit from when you are using graphical data maps
- How to solve conversion problems

The scenario prepares you for converting your message maps so that you can edit them in later versions of WebSphere Message Broker or IBM® Integration Toolkit, and enables you to make updates to your mapping solutions.

If you import your message map solutions from WebSphere Message Broker Version 6.1 or Version 7.0, you can still compile and deploy message flows that use message map (`.msgmap`) files. However, in WebSphere Message Broker Version 8.0 and later, message map files are accessible only in read-only mode, and cannot be modified. For example, you cannot take advantage of deploying broker or integration node resources as source, so if you continue to use message maps, you cannot deploy reusable subflows independently from all flows that are using them.

To be able to take advantage of editing, and the transforms available by using the Graphical Data Mapping editor in WebSphere Message Broker Version 8.0 and later, you must use a graphical data map (`.map`) file type.

The Graphical Data Mapping editor is based on XML schema and XPath standards so that you can use the full set of XPath 2.0 expressions for controlling and implementing transformations. The Graphical Data Mapping editor provides a greater range of transforms compared to the Message Mapping editor. The Graphical Data Mapping editor also includes developer accelerators, such as the Auto map wizard, which automatically creates transforms based on correlation of the names of input and output elements, and content assist for paths and expressions. The WebSphere Message Broker Version 8.0/IBM Integration Bus runtime has a dedicated Java based runtime execution engine, enabling the map execution to benefit from Java JIT optimization.



MembersMFPFlow_Mapping.msgmap

$source - Message "msg_PERSONCREATIONREQUEST"
Properties (PropertiesType_msg_PERSONCREATIONREQUEST)
msg_PERSONCREATIONREQUEST (PERSONCREATIONREQUEST)
RedefinedElement_personcreationrequest_person__name__wh
PERSON_DATE_OF_BIRTH (xsd:int -)
PERSON_CODE (xsd:int -)
PERSON_NOTES (xsd:string -)

Read-only

To modify these mapping operations, convert the .msgmap file to a .map file.

$target - Message "msg_MEMBERADDREQUEST", Parser "XMLNSC"
Properties (PropertiesType_msg_MEMBERADDREQUEST)
msg_MEMBERADDREQUEST (MEMBERADDREQUEST)

Map Script	Value
MembersMFPFlow_Mapping	
Parameters	
$target	

Property	Value

Figure 1. Message map. The scenario message map. Message maps are read-only in WebSphere Message Broker Version 8.0 or later.

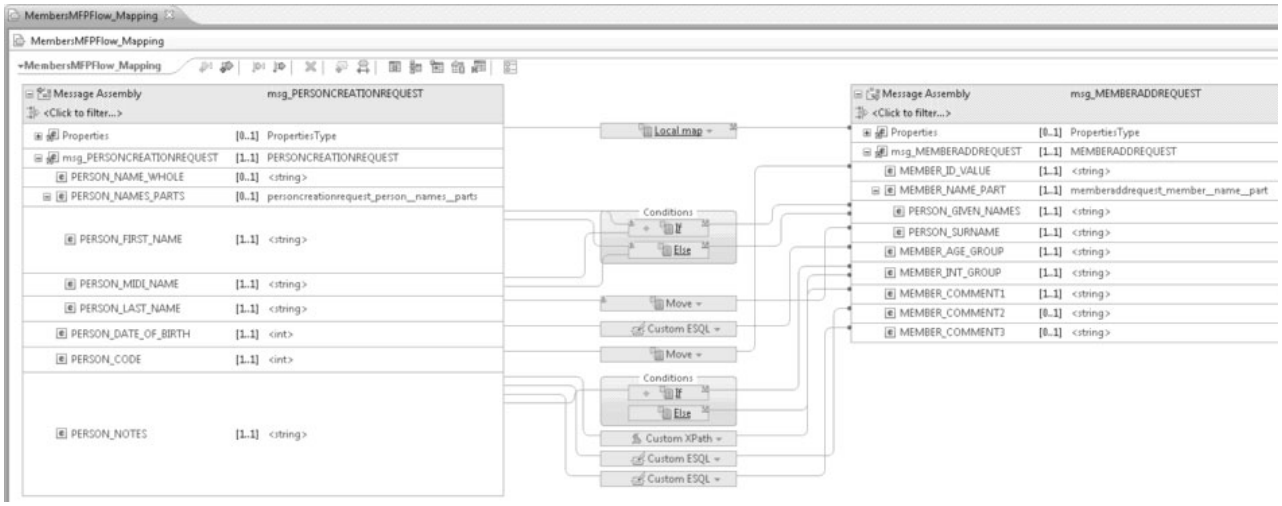

MembersMFPFlow_Mapping
MembersMFPFlow_Mapping
MembersMFPFlow_Mapping
Message Assembly
msg_PERSONCREATIONREQUEST
<Click to filter...>
Properties [0..1] PropertiesType
msg_PERSONCREATIONREQUEST [1..1] PERSONCREATIONREQUEST
PERSON_NAME_WHOLE [0..1] <string>
PERSON_NAMES_PARTS [0..1] personcreationrequest_person__names__parts
PERSON_FIRST_NAME [1..1] <string>
PERSON_MIDI_NAME [1..1] <string>
PERSON_LAST_NAME [1..1] <string>
PERSON_DATE_OF_BIRTH [1..1] <int>
PERSON_CODE [1..1] <int>
PERSON_NOTES [1..1] <string>
Local map
Conditions
If
Else
Move
Custom ESQL
Move
Conditions
If
Else
Custom XPath
Custom ESQL
Custom ESQL
Message Assembly
msg_MEMBERADDREQUEST
<Click to filter...>
Properties [0..1] PropertiesType
msg_MEMBERADDREQUEST [1..1] MEMBERADDREQUEST
MEMBER_ID_VALUE [1..1] <string>
MEMBER_NAME_PART [1..1] memberaddrequest_member__name__part
PERSON_GIVEN_NAMES [1..1] <string>
PERSON_SURNAME [1..1] <string>
MEMBER_AGE_GROUP [1..1] <string>
MEMBER_INT_GROUP [1..1] <string>
MEMBER_COMMENT1 [1..1] <string>
MEMBER_COMMENT2 [0..1] <string>
MEMBER_COMMENT3 [0..1] <string>

Figure 2. Graphical data map. The scenario message map after it is converted to a graphical data map.

To observe what changes take place during conversion, review “Technical solution.”

Technical solution

To complete the scenario, you must convert a message map, and verify the transform behavior of the converted map.

Use the scenario steps to go through the conversion process, review the new map to check the created transform logic, and if necessary, complete any required manual steps to provide equivalent behavior for the converted transform.

When you convert a message map, the transform logic is automatically converted so that it becomes valid behavior for a graphical data map.

During conversion, the transforms can change in the following ways:

- Automatically convert to a valid mapping transform. After conversion, you can then test and verify that the resultant behavior is not changed.
- Automatically convert to a valid mapping transform, but with possible minor changes in the behavior you might want to consider. For example, a date or time function might give a slightly different output format.
- Cannot be converted automatically, and requires manual reconfiguration by using the facilities in the Graphical Data Mapping editor to provide equivalent behavior. This requirement is indicated by a Task transform in your converted map.

Most transformations that are used in `.msgmap` files automatically convert to working transforms in a `.map` file. However, in complex maps you might need to make changes to the graphical data map, or check that the behavior matches what you expect after the conversion.

If you experience problems with the behavior of a converted transform in the runtime, it is always best to try to resolve those problems in the mapping, by using the Graphical Data Mapping editor, rather than create a workaround in the runtime. This ensures that your mapping solutions remain robust for future versions of WebSphere Message Broker/IBM Integration Bus. If the problem is not resolvable in the mapping, then contact your IBM service provider.

You cannot convert a `.map` file back into a `.msgmap` file, but a backup of the file is created when you start conversion. The backup file name is in the format *`filename`*`.msgmap_backup`. To restore your message map, rename the file extension.

The following topics explain the initial configuration that is used to start the scenario, and the target configuration that is the result of following the steps that are documented in the scenario:

- "Before: initial message map behavior"
- "After: the converted graphical data map" on page 7

Before: initial message map behavior

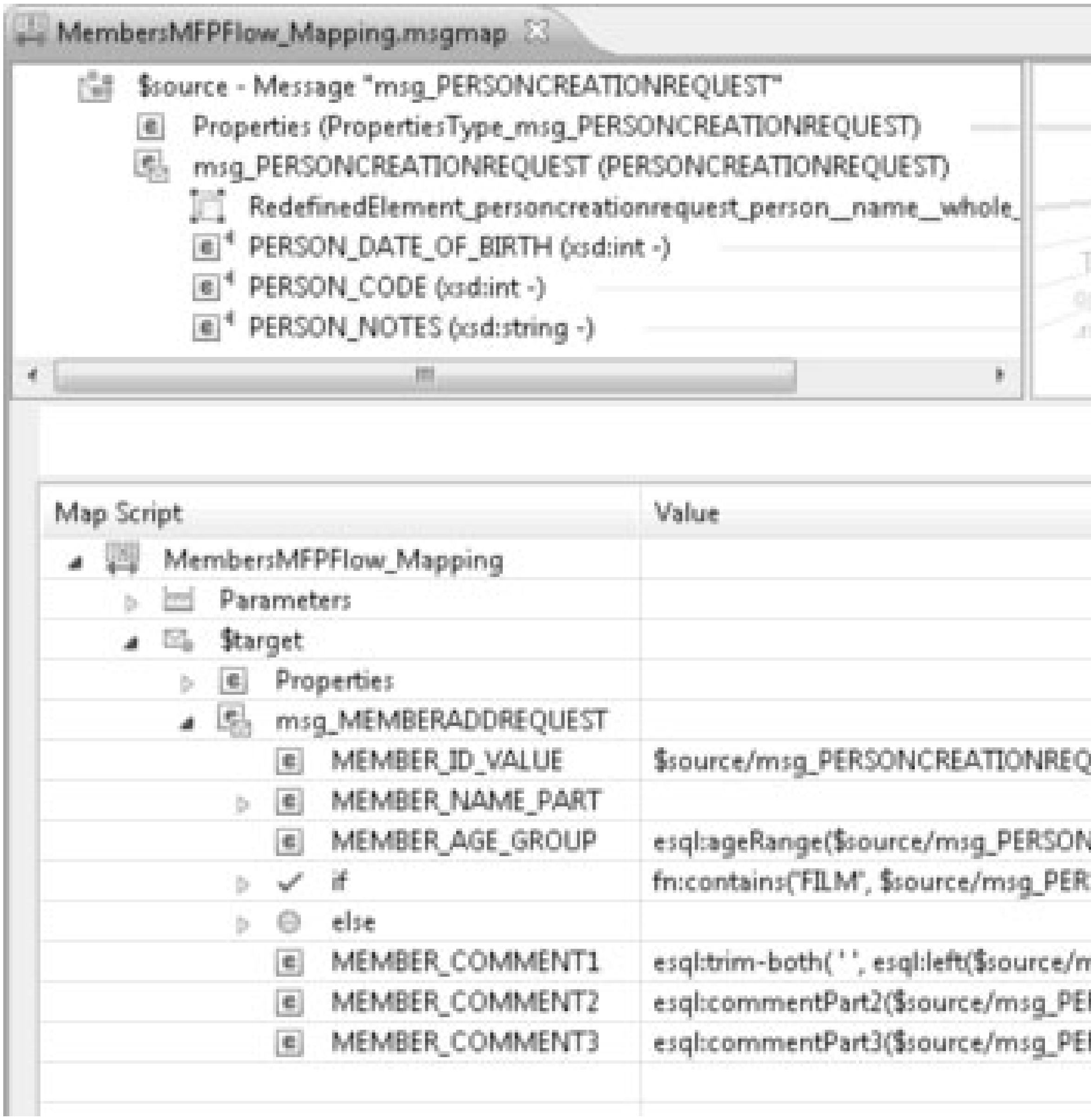
Review this topic for a description of the initial message map file for the scenario, its transformations and what changes you make when you implement the scenario solution.

The initial map file for the scenario is called `MembersMFPFlow_Mapping.msgmap`. The message map takes a generic input message with details of a person and transforms it to a message required by a member database system that adds a member record for the person.

Tip: This scenario was developed by using a sample of an initial message map configuration. If you want to try out the scenario, you can either use your own integration solutions, or set up a copy of the sample configuration as described in "Creating the scenario message map configuration" on page 10.

There are aspects to consider before you convert your maps to ensure that your broker (or integration node in IBM Integration Bus) operates as intended. For more information, see Considerations for converting a message map to a graphical data map [Page in the information center on ibm.com]. Although this scenario includes the necessary information from that topic, if you are using your own maps you might want to review that topic to understand what changes you might encounter.

The following image shows how the `MembersMFPFlow_Mapping.msgmap` map file looks before conversion:



MembersMFPFlow_Mapping.msgmap

$source - Message "msg_PERSONCREATIONREQUEST"
Properties (PropertiesType_msg_PERSONCREATIONREQUEST)
msg_PERSONCREATIONREQUEST (PERSONCREATIONREQUEST)
RedefinedElement_personcreationrequest_person__name__whole_
PERSON_DATE_OF_BIRTH (xsd:int -)
PERSON_CODE (xsd:int -)
PERSON_NOTES (xsd:string -)

Map Script	Value
MembersMFPFlow_Mapping	
Parameters	
$target	
Properties	
msg_MEMBERADDREQUEST	
MEMBER_ID_VALUE	$source/msg_PERSONCREATIONREQ
MEMBER_NAME_PART	
MEMBER_AGE_GROUP	esql:ageRange($source/msg_PERSON
if	fn:contains("FILM", $source/msg_PER
else	
MEMBER_COMMENT1	esql:trim-both(' ', esql:left($source/m
MEMBER_COMMENT2	esql:commentPart2($source/msg_PE
MEMBER_COMMENT3	esql:commentPart3($source/msg_PE

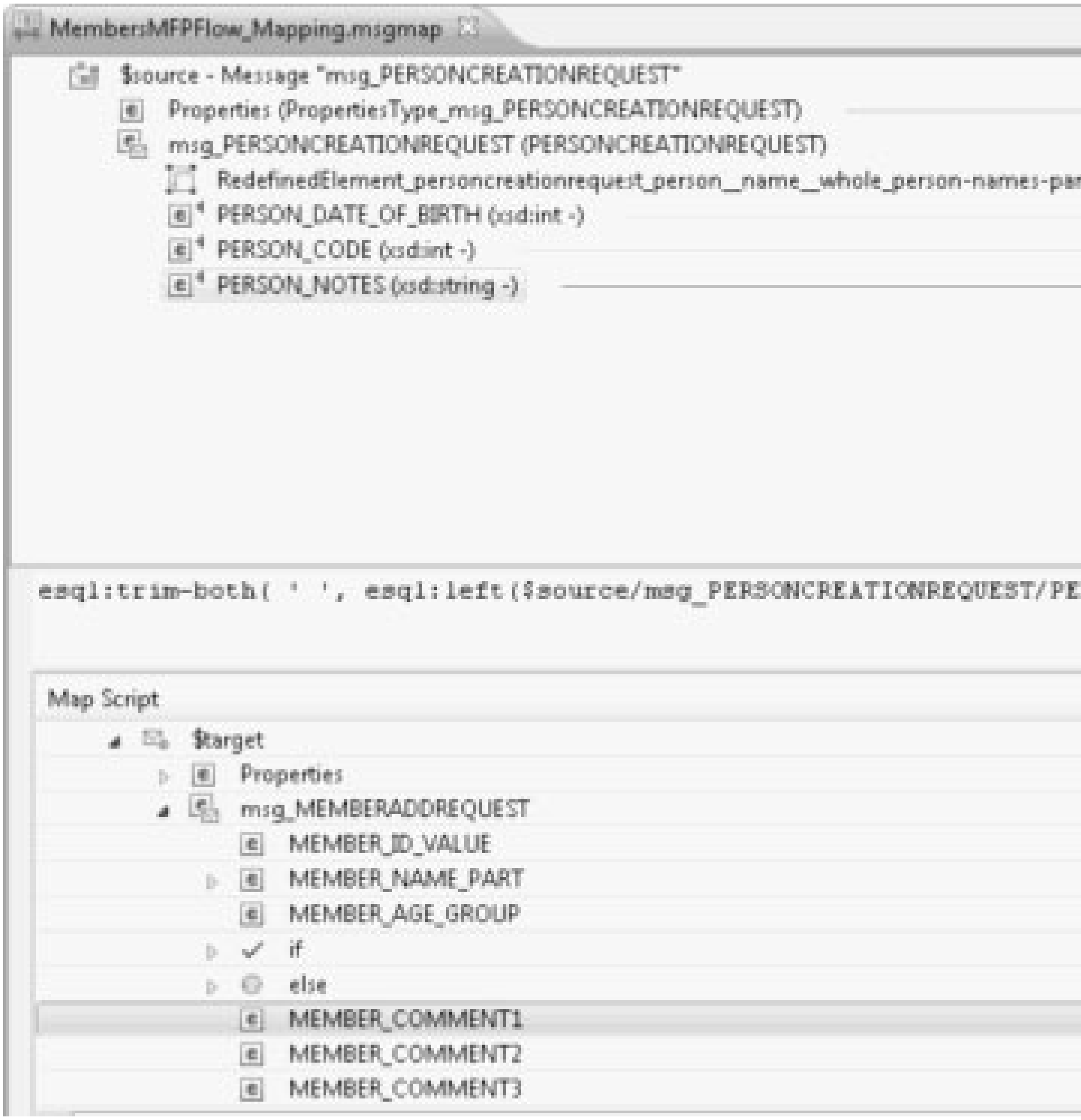
The image shows some, but not all of the transformations that are used in the message map. The message map includes the following significant transformation logic:

- *MEMBER_NAME_PART.PERSON_GIVEN_NAMES* is set by using an `If Else` statement, which is using proprietary built-in ESQL functions that were provided in the Message Mapping editor to create a conditional expression that determines whether the input message has non-white-space characters in the value of *PERSON_MIDI_NAME*.
- *MEMBER_AGE_GROUP* is set by using a call to a user-written ESQL function named `ageRange()`.
- *MEMBER_INT_GROUP* is set by using an `If Else` construct, which is using a standard XPath expression to control the construct.
- *MEMBER_COMMENT1* is set by using proprietary built-in ESQL functions that were provided in the previous message map tool.
- *MEMBER_COMMENT2* and *MEMBER_COMMENT3* are set by calling user-written ESQL functions `commentPart2()` and `commentPart3()`. These

functions are coded so that if there is not enough input data in the input element *PERSON_NOTES*, they return an ESQL NULL. Setting an output element in ESQL to NULL causes the element to be deleted so that the *MEMBER_COMMENT2* and *MEMBER_COMMENT3* output elements do not show in the produced message.

One of the transformations for *MEMBER_COMMENT1* uses an ESQL function, esql:trim-both, that is not supported for graphical data maps. The list of non-supported ESQL functions is listed in Considerations for converting a message map to a graphical data map [Page in the information center on ibm.com].

See the following image for the construction of the expression using esql:trim-both:


MembersMFPFlow_Mapping.msgmap
$source - Message "msg_PERSONCREATIONREQUEST"
Properties (PropertiesType_msg_PERSONCREATIONREQUEST)
msg_PERSONCREATIONREQUEST (PERSONCREATIONREQUEST)
RedefinedElement_personcreationrequest_person__name__whole_person-names-par
PERSON_DATE_OF_BIRTH (xsd:int -)
PERSON_CODE (xsd:int -)
PERSON_NOTES (xsd:string -)
Map Script
$target
Properties
msg_MEMBERADDREQUEST
MEMBER_ID_VALUE
MEMBER_NAME_PART
MEMBER_AGE_GROUP
if
else
MEMBER_COMMENT1
MEMBER_COMMENT2
MEMBER_COMMENT3

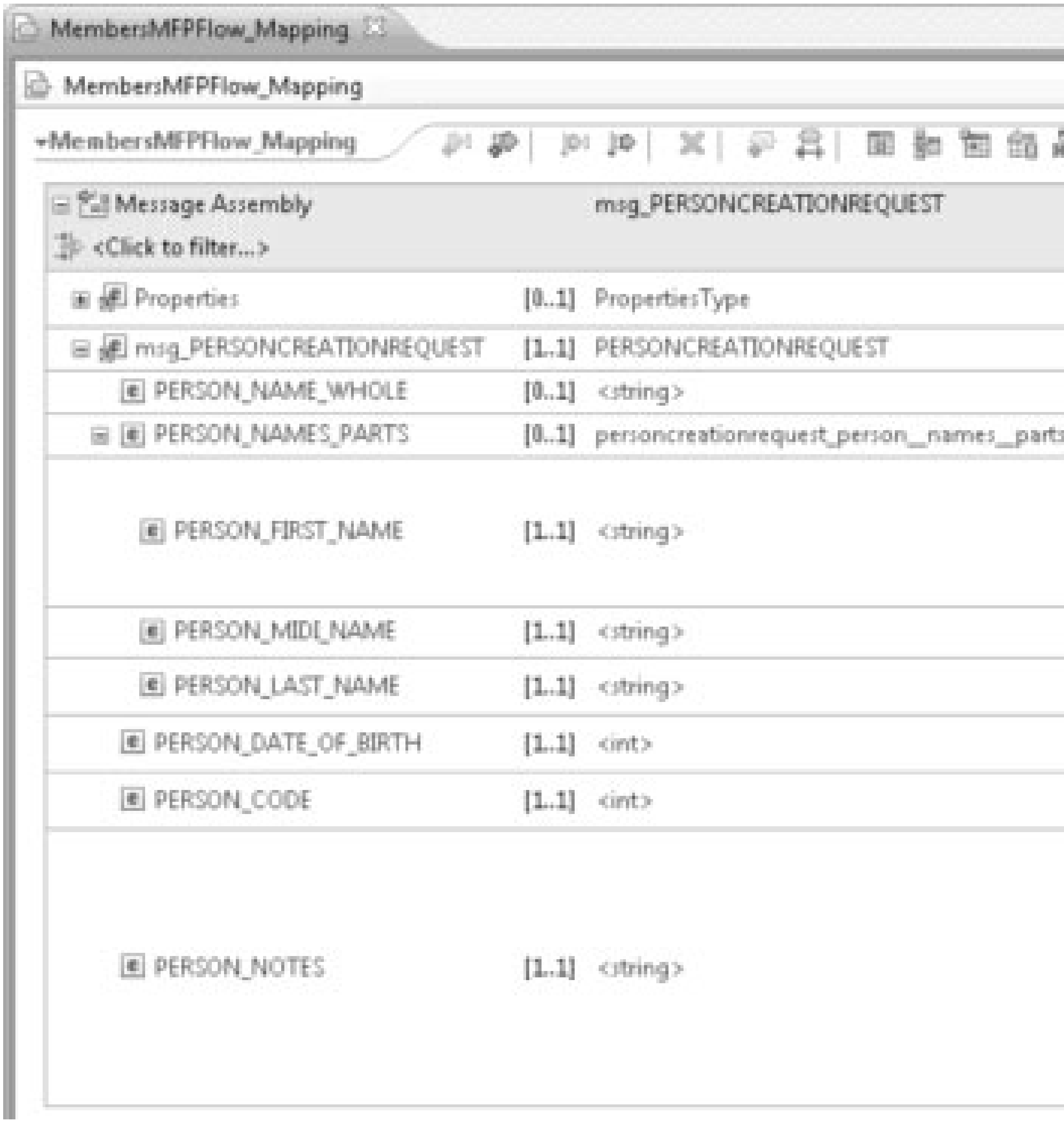
```
esql:trim-both( ' ', esql:left($source/msg_PERSONCREATIONREQUEST/PE
```

Review the next topic to see what changes are made to transformations when they are converted in a graphical data map. "After: the converted graphical data map."

After: the converted graphical data map

After you update the message map, you should review and amend the transforms in the graphical data map so that it behaves in the same way as the original message map.

The following image shows the converted graphical data map configuration for the map named `MembersMFPFlow_Mapping.map`:



MembersMFPFlow_Mapping

MembersMFPFlow_Mapping

▾MembersMFPFlow_Mapping

Message Assembly		msg_PERSONCREATIONREQUEST
<Click to filter...>		
Properties	[0..1]	PropertiesType
msg_PERSONCREATIONREQUEST	[1..1]	PERSONCREATIONREQUEST
PERSON_NAME_WHOLE	[0..1]	<string>
PERSON_NAMES_PARTS	[0..1]	personcreationrequest_person__names__parts
PERSON_FIRST_NAME	[1..1]	<string>
PERSON_MIDL_NAME	[1..1]	<string>
PERSON_LAST_NAME	[1..1]	<string>
PERSON_DATE_OF_BIRTH	[1..1]	<int>
PERSON_CODE	[1..1]	<int>
PERSON_NOTES	[1..1]	<string>

The transforms are updated as part of the solution, so that even though the transforms are now of a different type, they behave in the same way as in the message map. You can test this by following the steps in Chapter 3, “Verifying the solution,” on page 29 after you follow the steps for Chapter 2, “Implementing the solution,” on page 9.

Graphical data maps use all standard XPath 2.0 functions and allow user-written functions in Java and ESQL. Message maps used basic XPath 1.0, some proprietary built-in ESQL mapping functions, and user-written Java and ESQL functions.

Not all ESQL functions can be directly translated to XPath equivalents, but these functions are resolved in the scenario solution. In the screen capture, you can observe these changes to the following elements:

- *MEMBER_NAME_PART.PERSON_GIVEN_NAMES* is now set by using standard XPath expressions to check whether the input message has non-white-space characters in *PERSON_MIDI_NAME*. The Graphical Data Mapping editor provides a warning here because the map's logic assumes that the input message has *MEMBER_NAME_PARTS* present, which is defined as a choice in the message schema model with *MEMBER_NAME_WHOLE*. For this scenario, the original message map author has assumed that the input data will always include *MEMBER_NAME_PARTS*, so the warning can be ignored on the same assumption.

  To avoid this warning, and the one on the PERSON_SURNAME mapping, you can add an `If / Else` check for *MEMBER_NAME_PARTS* or *MEMBER_NAME_WHOLE*.
- *MEMBER_AGE_GROUP* is set by using a call to a user-written ESQL function `ageRange()`. This is unchanged.
- *MEMBER_INT_GROUP* is set by using an `If / Else` construct, which continues to use a standard XPath expression for the transform condition.
- *MEMBER_COMMENT1* is now set by using standard XPath expressions that replace the previous proprietary ESQL functions in the Message Mapping editor.
- *MEMBER_COMMENT2* and *MEMBER_COMMENT3* are set by calling user-written ESQL functions `commentPart2()` and `commentPart3()`. It was necessary to add an XPath conditional expression to prevent the `commentPart3()` ESQL being invoked when the input data is fewer than 48 characters long. This corrects a difference in behavior for user ESQL functions that can return an ESQL `NULL` value. For more information on ESQL changes in mapping, see Changes in behavior in graphical data maps converted from message maps.

You can now re-create these changes by using either your own files or the scenario files that are provided by following the steps as part of Chapter 2, “Implementing the solution,” on page 9.

Chapter 2. Implementing the solution

Implementing the solution in this scenario involves converting the message mapping solution to a graphical data map.

Before you begin

The starting point for this scenario is an existing and working message map for the initial messaging infrastructure. If you want to try out the scenario, you can either follow the instructions to convert your message map to a graphical data map, or set up a copy of the sample message map scenario as described in “Creating the scenario message map configuration” on page 10.

About this task

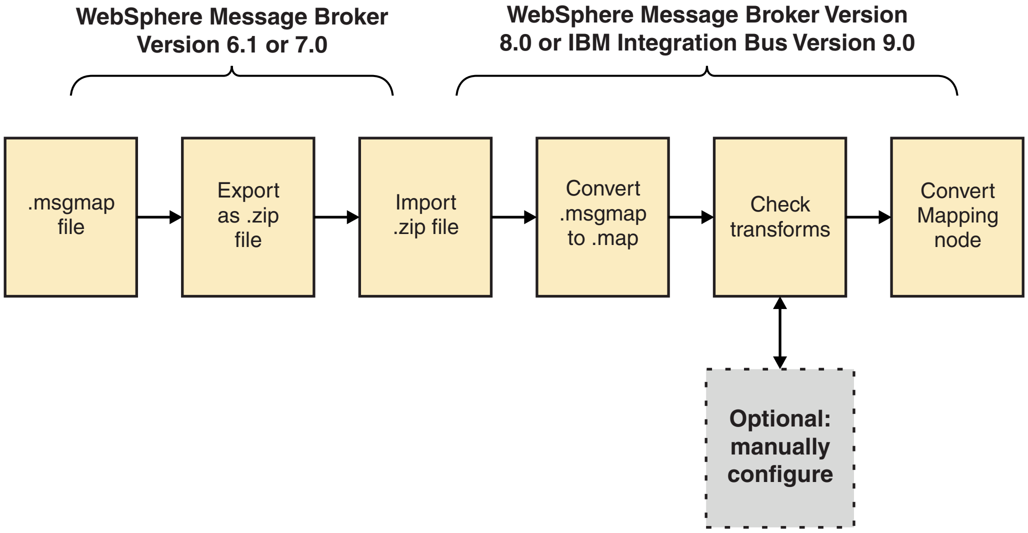

WebSphere Message Broker Version 6.1 or 7.0
WebSphere Message Broker Version 8.0 or IBM Integration Bus Version 9.0
.msgmap file
Export as .zip file
Import .zip file
Convert .msgmap to .map
Check transforms
Convert Mapping node
Optional: manually configure

Figure 3. Order of tasks to complete for this scenario.

To implement the scenario, you must complete the following steps:

1. If you are using the provided sample files for the scenario, download and check the configuration. See “Creating the scenario message map configuration” on page 10.
2. Convert the message map to a graphical data map. See “Converting the message map” on page 11.
3. When the map is converted, check the transform behavior and configure. See “Configuring the converted transforms” on page 17.
4. Verify that the message flow works in the runtime. See Chapter 3, “Verifying the solution,” on page 29.

If you experience any technical problems during the process of implementing the scenario, check the error log for your system.

- On UNIX and Linux systems, the local error log is the `syslog`. Where `syslog` messages are sent depends on how you configure your UNIX or Linux system.

- On z/OS®, the local error log is the operator console.
- On Windows, The Windows Event Viewer is where IBM Integration Bus writes records to the local system. Use Windows system facilities to view this log:
  1. Open a command prompt.
  2. At the prompt, type `eventvwr`. The Windows Event Viewer opens.
  3. Click **Windows Logs** > **Application**.
  4. Review the error or errors listed.

For more information on error logs, see Local error logs in the product Information Center.

To see what the scenario files look like before and after conversion, review:
- "Before: initial message map behavior" on page 4
- "After: the converted graphical data map" on page 7

Creating the scenario message map configuration

This scenario was developed by using a sample initial configuration. You can either follow the instructions to convert your own message map and configure the output graphical data map, or set up the sample message map to try out the scenario in the same way as it was originally developed.

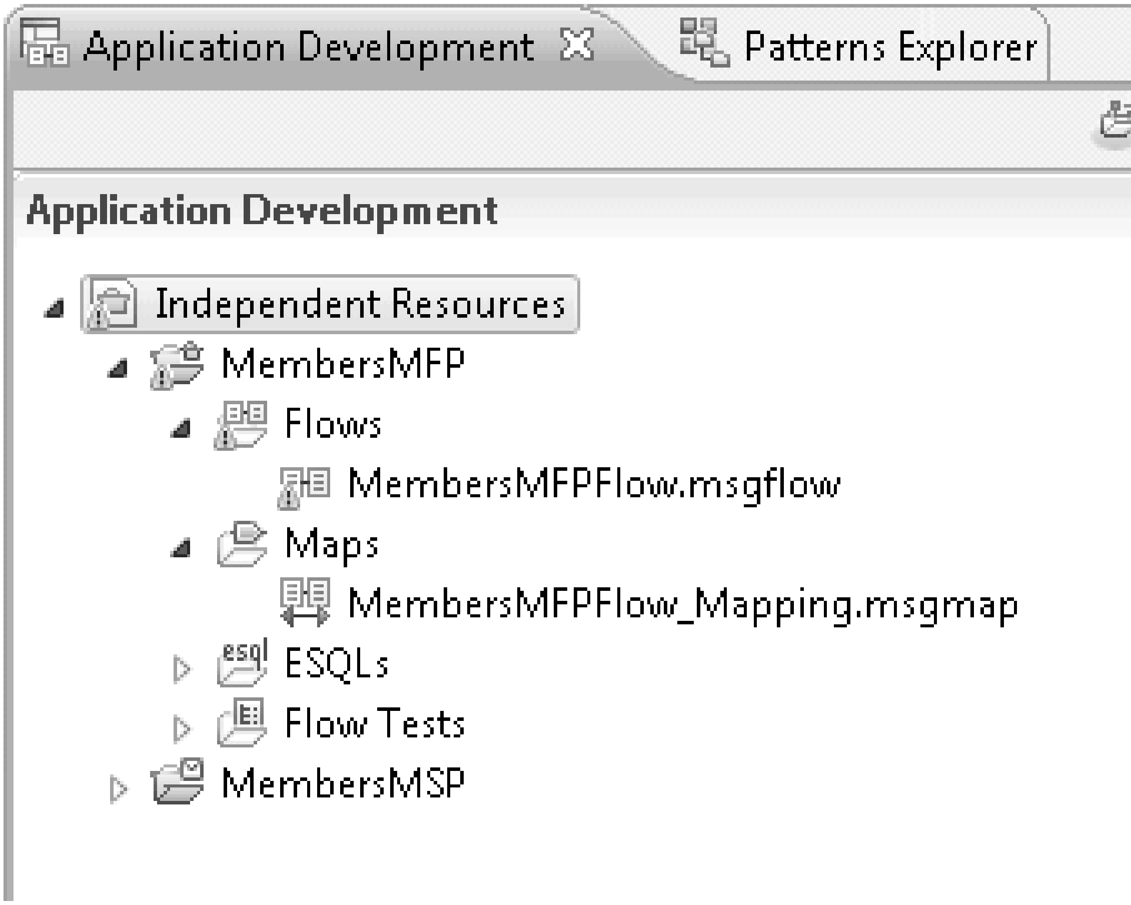
About this task

Complete the following steps to set up the sample initial configuration that was used to develop the scenario. If you are importing message flow projects into IBM Integration Toolkit Version 9.0, they are converted automatically to Integration projects.

Procedure

1. Install the WebSphere Message Broker Version 8.0 Toolkit or IBM Integration Toolkit.
2. Download a copy of the scenario files as a `.zip` file. Download the file `scnmap_01_scenario.zip` from the IBM Integration Community.
3. Import the `scnmap_01_scenario.zip` file:
   a. Click **File** > **Import**. The Import wizard opens.
   b. Expand **Other**, click **Project Interchange**, then click **Next**.
   c. Specify the location of `scnmap_01_scenario.zip`.
   d. Specify the location of the open workspace.
   e. Select the projects that you want to import into your workspace. For this scenario, select all projects. Then, click **Finish**.

   For the scenario, you can see the `MembersMFP` project in the Application Development view:



4. Open `MembersMFPFlow_Mapping.msgmap`. You can now check the transforms that are used in the message map and confirm that they match the transforms as shown in "Before: initial message map behavior" on page 4.

Results

You imported the scenario message map files and validated them.

What to do next

Follow the steps for "Converting the message map."

Converting the message map

After the map resources are imported, you can then convert the map to a graphical data map.

Before you begin

If you are using your own message map for this scenario, review "Before: initial message map behavior" on page 4, or the topic Changes in behavior in graphical data maps converted from message maps in the Information Center.

About this task

The steps for converting the message map for this scenario are taken from the steps that are described in Converting a message map from a `.msgmap` file to a `.map` file in the in the IBM Integration Bus Information Center. These steps have been tailored to the scenario, and so do not include all possible options and methods available. Review the Information Center topic for generic conversion steps.

If you import your messaging solutions from WebSphere Message Broker Version 6.1 or WebSphere Message Broker Version 7.0, you can still compile and deploy message flows that use message map (`.msgmap`) files, provided that the message flow is built in a BAR file that has the option **compile and in-line resources** set. However, in WebSphere Message Broker Version 8.0 and later, message map files are accessible in read-only mode and cannot be modified by using the Toolkit.

In WebSphere Message Broker Version 8.0 and later, if you want to modify mapping operations that are defined in a message map, you must first convert your message map to a graphical data map (`.map`) file.

When you convert a message map, message flows that reference the unconverted message map cannot be deployed until you modify each message flow to reference the new graphical data map instead.

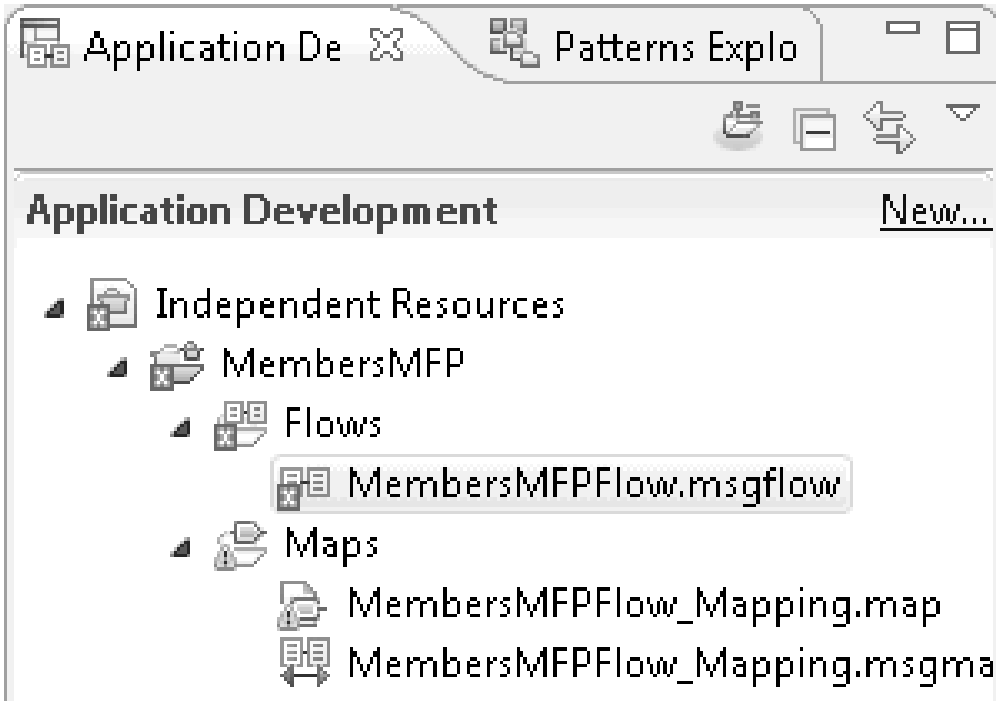
To convert the scenario message map to a deployable graphical data map by using the Toolkit, complete the following steps:

Procedure

1. In the **Application Development** view, right-click `MembersMFPFlow_Mapping.msgmap` (or a `.msgmap` that you want to convert), and click **Convert Message Map from .msgmap to .map**.

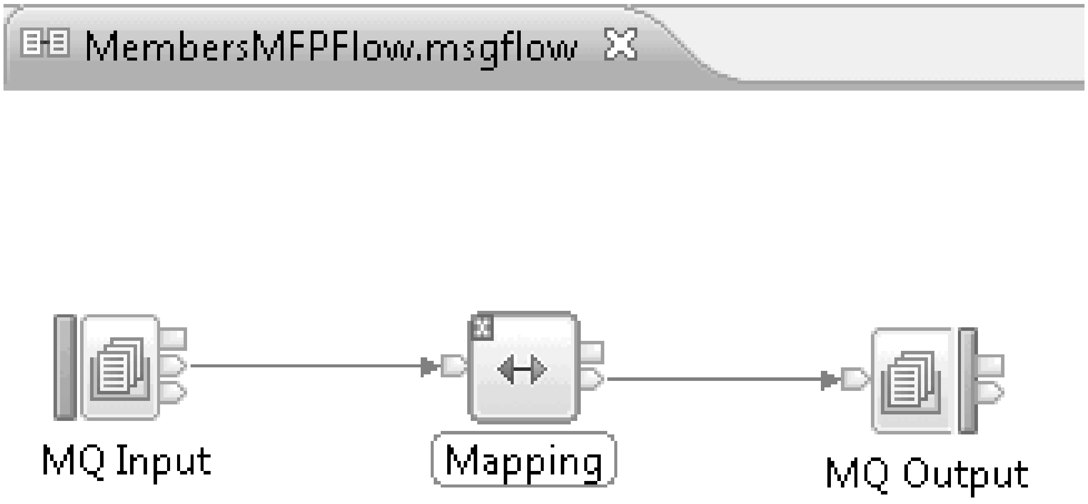
   Your new graphical data map is created, and is displayed in the **Application Development** view. Your message map is renamed *`MembersMFPFlow_Mapping`*`.msgmap_backup`, and is displayed after your new graphical data map in the **Application Development** view.

In the **Application Development** view, you can see that the scenario message flow has a problem that is indicated by a red box with a cross:

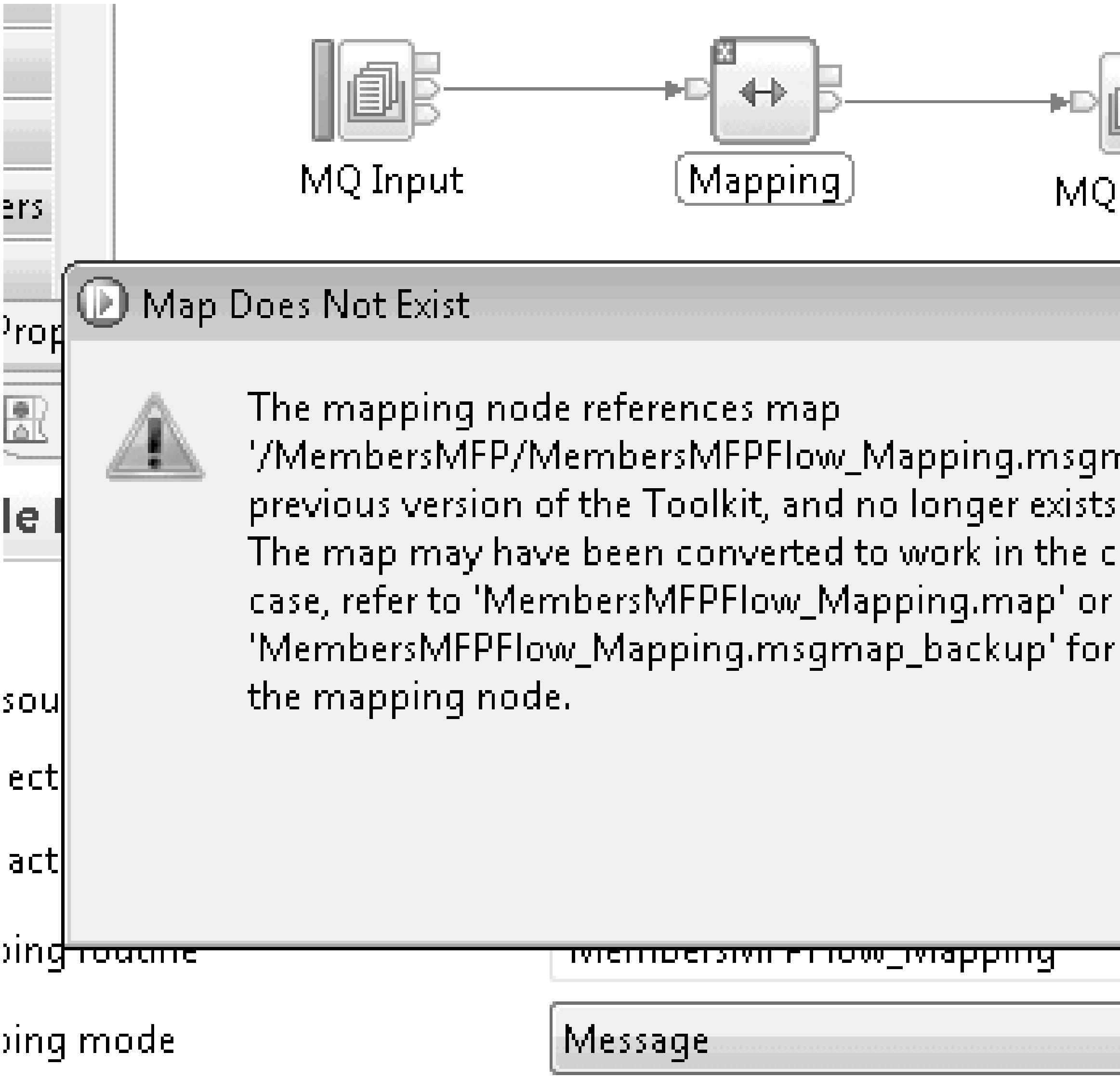


You must now modify the message flow that referenced the previous message map so that the message flow instead references your new graphical data map. In this scenario, `MembersMFPFlow.msgflow` uses `MembersMFPFlow_Mapping.map`.

2. In the **Application Development** view, double-click `MembersMFPFlow.msgflow`. The message flow opens in the Message Flow editor.



3. In the Message Flow editor, identify the Mapping node from a previous version of WebSphere Message Broker. If you click an old Mapping node, you get a warning message box, that informs you that the node cannot find your previous message map.



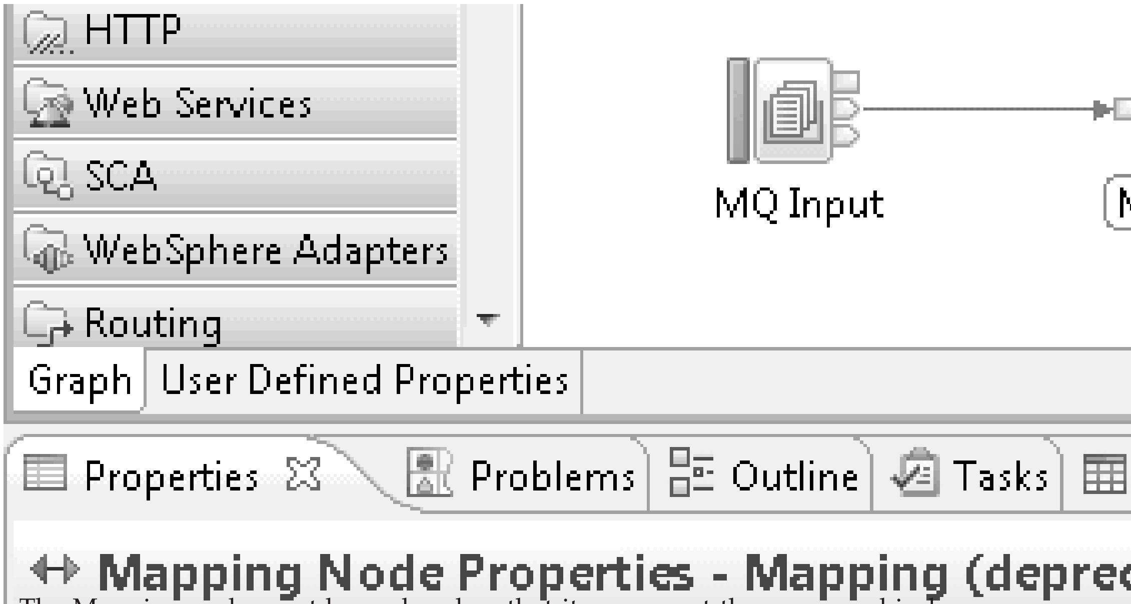
Map Does Not Exist

The mapping node references map
'/MembersMFP/MembersMFPFlow_Mapping.msgm
previous version of the Toolkit, and no longer exists
The map may have been converted to work in the c
case, refer to 'MembersMFPFlow_Mapping.map' or
'MembersMFPFlow_Mapping.msgmap_backup' for
the mapping node.

ping routine MembersMFPFlow_Mapping

ping mode Message

You can also identify a previous version of a Mapping node by clicking it. In the Properties view, the node title includes the type as `Mapping (deprecated)`.



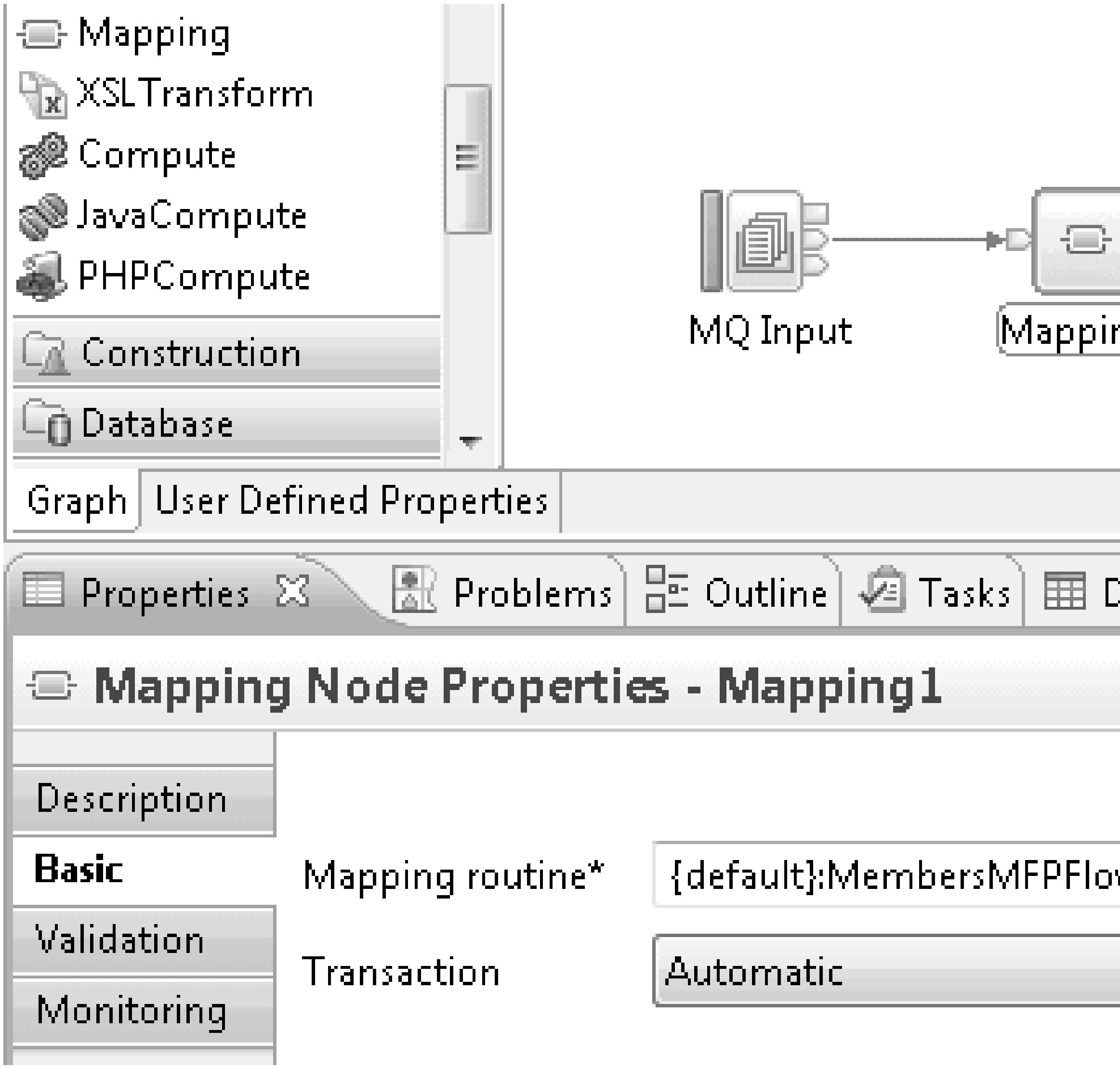
The Mapping node must be replaced so that it can accept the new graphical data map.

Check the **Basic** tab in the Properties view of the deprecated Mapping node. The **Mapping routine** field contains the value `MembersMFPFlow_Mapping`. This value is the name of the routine for the map, that is required for the new Mapping node.

4. In the Palette, expand the **Transformation** section, then drag a new Mapping node from the Palette to the canvas of the Message Flow Editor. A new Mapping node is added to your message flow, and is assigned a default name. If you rename the node, the name that you choose must be unique in the message flow.

   Note: If you do not change the default name now, you can change it later by using the name field in the **Description** tab.
5. Select your new Mapping node. The node properties are displayed in the Properties view.
6. In the Properties view, the *Mapping routine* property contains a default value, which must be replaced. Click **Browse...** to locate it, or specify your `.map` file in the format `{`*`BrokerSchemaName`*`}:`*`MapName`*.

   `{default}` indicates that no broker or integration node schema is used by the graphical data map. There is no schema for the scenario, and so `{default}` is shown. For the scenario, replace the *MapName* value with `MembersMFPFlow_Mapping`.



Mapping Node Properties - Mapping1

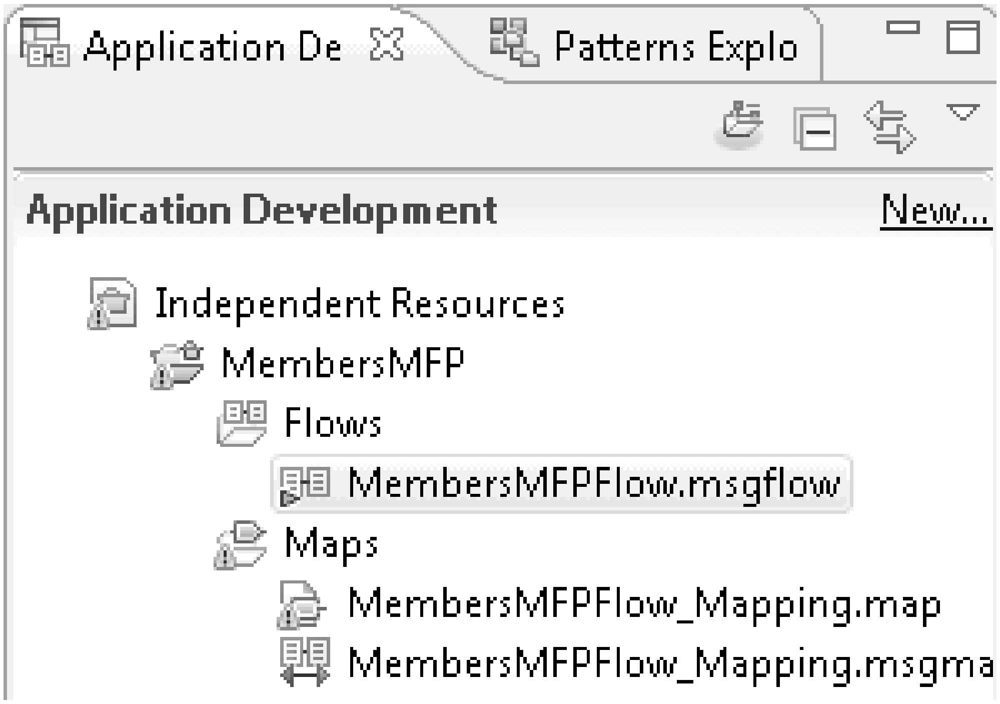
Mapping routine* {default}:MembersMFPFlov

Transaction Automatic

7. Move the existing connections from your previous Mapping node to your new Mapping node. Drag the connectors by using the mouse. You can also use the Terminal Selection dialog box to move the connections, but this usage is not covered in this scenario; for more information, see Creating node connections with the Terminal Selection dialog box in the Information Center.
8. Select your previous Mapping node, and press the delete key (del) to remove it from your message flow.
9. You have replaced the Mapping node and updated the message flow to reference the new graphical data map. Save your changes.

Results

The message map is converted to a graphical data map that can be modified by using the Graphical Data Mapping editor, and `MembersMFPFlow.msgflow` accepts the new map, as shown in the following image:



What to do next

After you convert a message map, open and check the map for any warnings or errors on transforms. In this scenario, you must configure the transforms to resolve errors and warnings. Next, follow the tasks for "Configuring the converted transforms"

Configuring the converted transforms

When the map is converted, automatically or manually resolve transforms with errors or warnings.

About this task

After your maps are converted, your transforms are changed to the appropriate type or configuration for a graphical data map. However, some transforms might show the following indicators:

- Annotations are represented by a green arrow (➡) on the transform. The transform was successfully converted, but if you review the transform and find the behavior did not convert as expected, you can opt to reject conversion.
- Warnings are represented by a triangular exclamation mark box (⚠). If a warning is present on one or more of the transforms in your converted graphical data map, the behavior or output of the converted transform might be different from the behavior or output of the mapping structure in your original message map.
- Required tasks are represented by a red cross on a Task transform (✖). You must complete required tasks so that the transform works in your map.

To ensure that your transforms operate as intended, you must review each transform that has an annotation, warning, or task before you use your converted graphical data map in a messaging solution. The converted map will work without reviewing the transforms, but might not work as expected. Task transforms are ignored at runtime, so you must resolve them before you use your maps.

Warnings are often specific to the data that is used in the map, so the scenario covers how to resolve annotations and Task transforms:

Procedure

1. Process the conversion annotations.
2. Configure the Task transforms.

What to do next

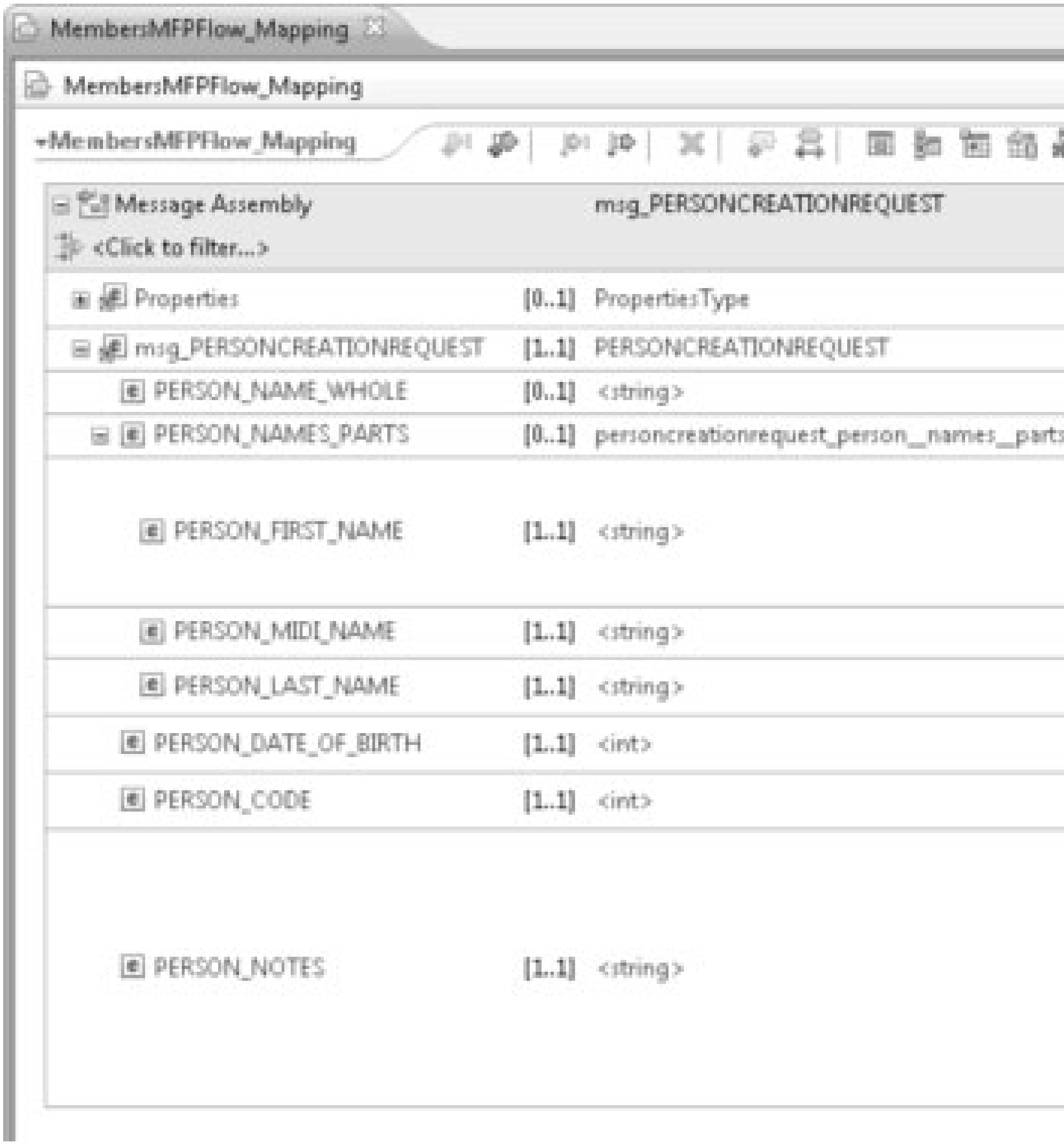
After you have configured the transforms in the graphical data map, you can test the changes by Chapter 3, “Verifying the solution,” on page 29.

Processing conversion annotations

An annotation on a converted transform shows that although the transform is successful, you might want to review that it behaves as you expect. You can either confirm or reject the annotated transform in your graphical data map.

About this task

If you are using the scenario files, after you follow the steps in “Converting the message map” on page 11 your `MembersMFPFlow_Mapping.map` file looks like the following image:



MembersMFPFlow_Mapping

MembersMFPFlow_Mapping

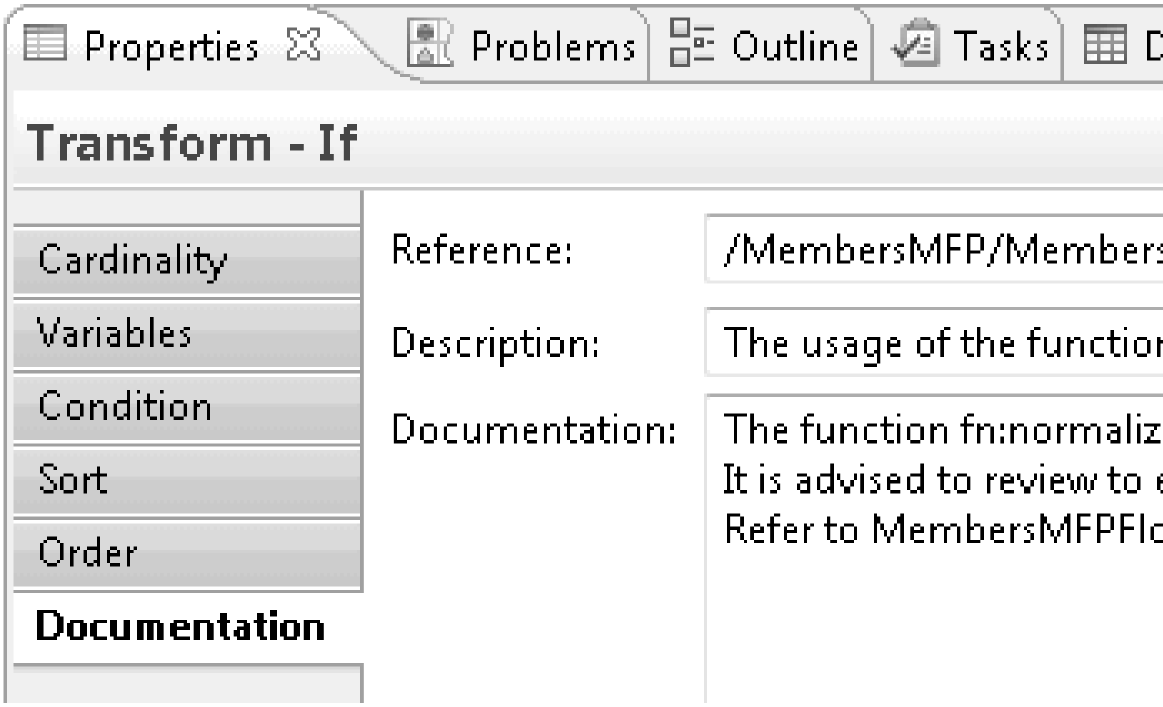
▾MembersMFPFlow_Mapping

Message Assembly		msg_PERSONCREATIONREQUEST
<Click to filter...>		
Properties	[0..1]	PropertiesType
msg_PERSONCREATIONREQUEST	[1..1]	PERSONCREATIONREQUEST
PERSON_NAME_WHOLE	[0..1]	<string>
PERSON_NAMES_PARTS	[0..1]	personcreationrequest_person__names__parts
PERSON_FIRST_NAME	[1..1]	<string>
PERSON_MIDL_NAME	[1..1]	<string>
PERSON_LAST_NAME	[1..1]	<string>
PERSON_DATE_OF_BIRTH	[1..1]	<int>
PERSON_CODE	[1..1]	<int>
PERSON_NOTES	[1..1]	<string>

You can see that *MEMBER_NAME_PART.PERSON_GIVEN_NAMES* now takes the output from both an If and Else transform. Both transforms have warnings, and there is an annotation on the If transform. Follow the steps to review an annotated transform.

Procedure

1. In the Graphical Data Mapping editor, select the If transform that shows a conversion annotation icon. The transform properties are displayed in the Properties view.
2. In the Properties view, click the **Documentation** tab to review details about the transform.



Transform - If

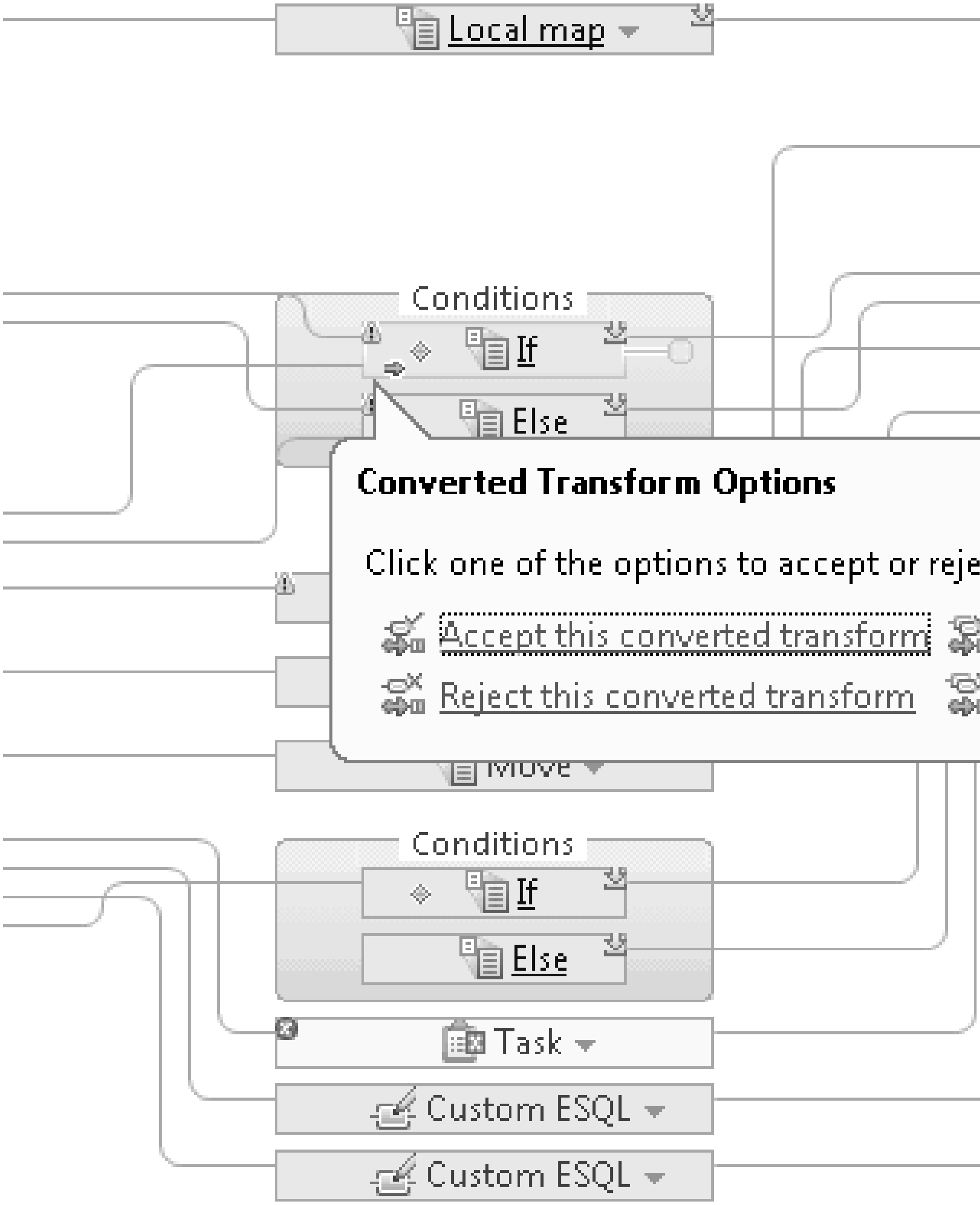
Reference: /MembersMFP/Members

Description: The usage of the functio

Documentation: The function fn:normaliz
It is advised to review to e
Refer to MembersMFPFlo

The If transform **Documentation** field advises that the `esql:trim` function used in the previous message map, which is not a valid type in a graphical data map, is converted to an XPath `fn:normalize-space` function. However, the behavior might be changed, and so the text in the field suggests that you should review the transformation.

3. Review the transformation logic that was created by the conversion to ensure that it produces the correct output for your application. You can review the online documentation for XPath expressions to check how `fn:normalize-space` works. In the scenario, the If transform behavior is correct, so you can accept it.
4. In the Graphical Data Mapping editor, right-click the transform. Click **Accept** to remove the warning from this transform.


Local map
Conditions
If
Else
Converted Transform Options
Click one of the options to accept or reje
Accept this converted transform
Reject this converted transform
Move
Conditions
If
Else
Task
Custom ESQL
Custom ESQL

Alternatively, after you review all the transforms in a converted map, you can use a menu action to accept or reject all annotated transforms. Select an area of the map so that no transform is selected and right-click the area to open the menu.

5. Save your updated map.

Results

You resolved all annotations in your graphical data map.

There are still warnings on the child transforms of *MEMBER_NAME_PART* in the scenario map. The Graphical Data Mapping editor provides a warning here because the map's logic assumes that the input message has *PERSON_NAME_PARTS* present, which is defined as a choice in the message schema model with *PERSON_NAME_WHOLE*. For the scenario, this assumption comes from the original message map, and therefore it is valid.

To avoid this warning, and the one on the *PERSON_SURNAME* mapping, you can add an If or Else check for *PERSON_NAME_PARTS* or *PERSON_NAME_WHOLE*. This is an optional task, because warnings do not affect the behavior of the map. You do not have to remove the warnings to complete the scenario.

Configuring Task transforms

If your message map contained mapping structures for which no direct replacement exists in the conversion, your new graphical data map includes Task transforms to assist you in manually re-creating those structures. You must review and replace each Task transform before you can use your graphical data map in a messaging solution.

About this task

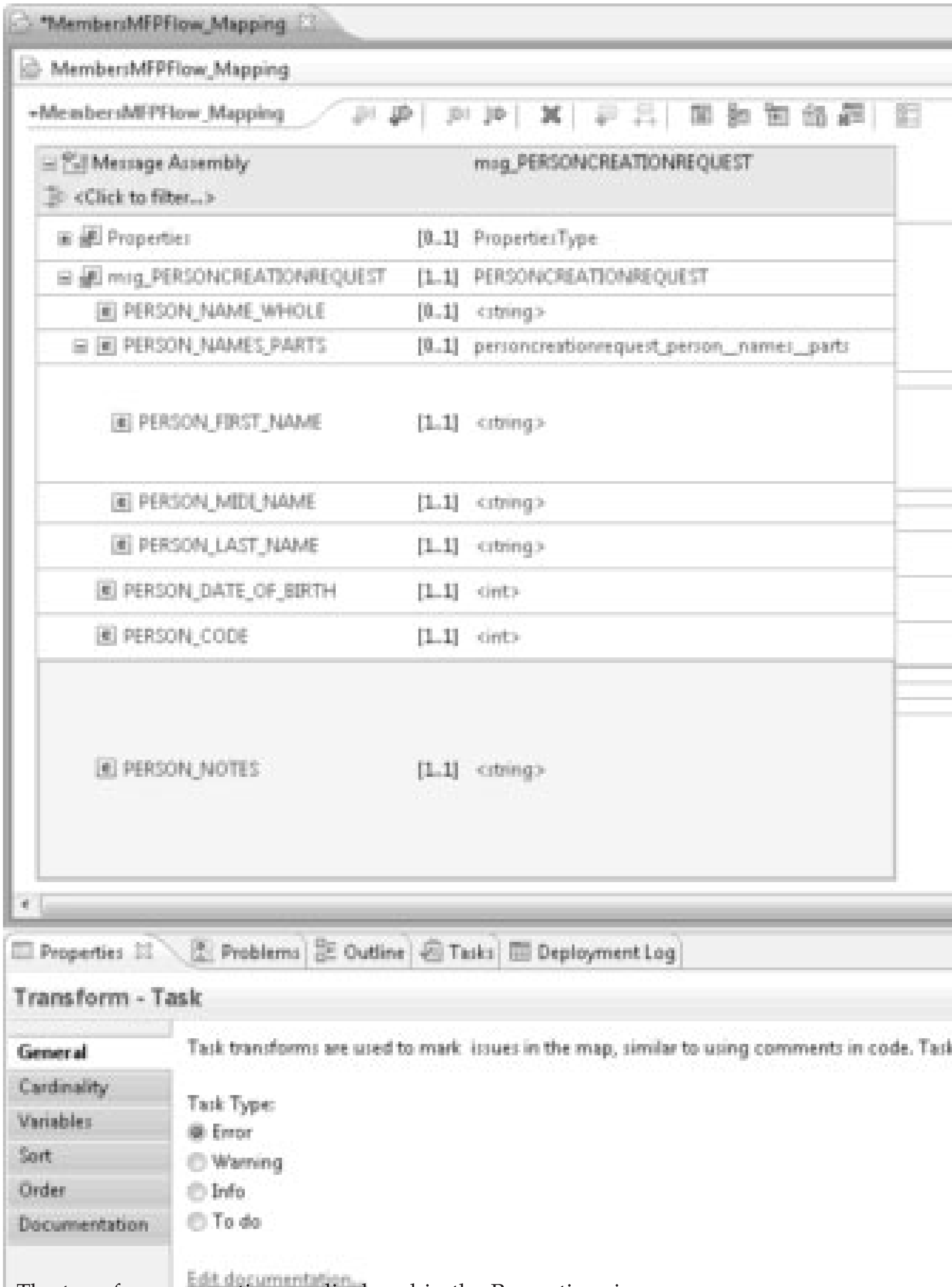
In the scenario map, `MembersMFPFlow_Mapping.map`, you must resolve a Task transform. If you review Changes in behavior in graphical data maps converted from message maps in the information center, there is information about what can and cannot be automatically converted. In the scenario map, `esql:trim-both` is a function that cannot be automatically converted, and so is now a Task transform.

If you have a Task transform in a submap, in the main map this is represented by a red cross on the transform, and more information is provided if you hover the cursor over the transform. Task transforms are listed in the Problems view.

You must configure the Task transform so that it becomes a valid transform type. Follow the steps to resolve a Task transform:

Procedure

1. In the Graphical Data Mapping editor, select the Task transform.



*MembersMFPFlow_Mapping

MembersMFPFlow_Mapping

MembersMFPFlow_Mapping

Message Assembly		msg_PERSONCREATIONREQUEST
<Click to filter...>		
Properties	[0..1]	PropertiesType
msg_PERSONCREATIONREQUEST	[1..1]	PERSONCREATIONREQUEST
PERSON_NAME_WHOLE	[0..1]	<string>
PERSON_NAMES_PARTS	[0..1]	personcreationrequest_person__names__parts
PERSON_FIRST_NAME	[1..1]	<string>
PERSON_MIDL_NAME	[1..1]	<string>
PERSON_LAST_NAME	[1..1]	<string>
PERSON_DATE_OF_BIRTH	[1..1]	<int>
PERSON_CODE	[1..1]	<int>
PERSON_NOTES	[1..1]	<string>

Transform - Task

Task transforms are used to mark issues in the map, similar to using comments in code. Task

Task Type:

● Error

○ Warning

○ Info

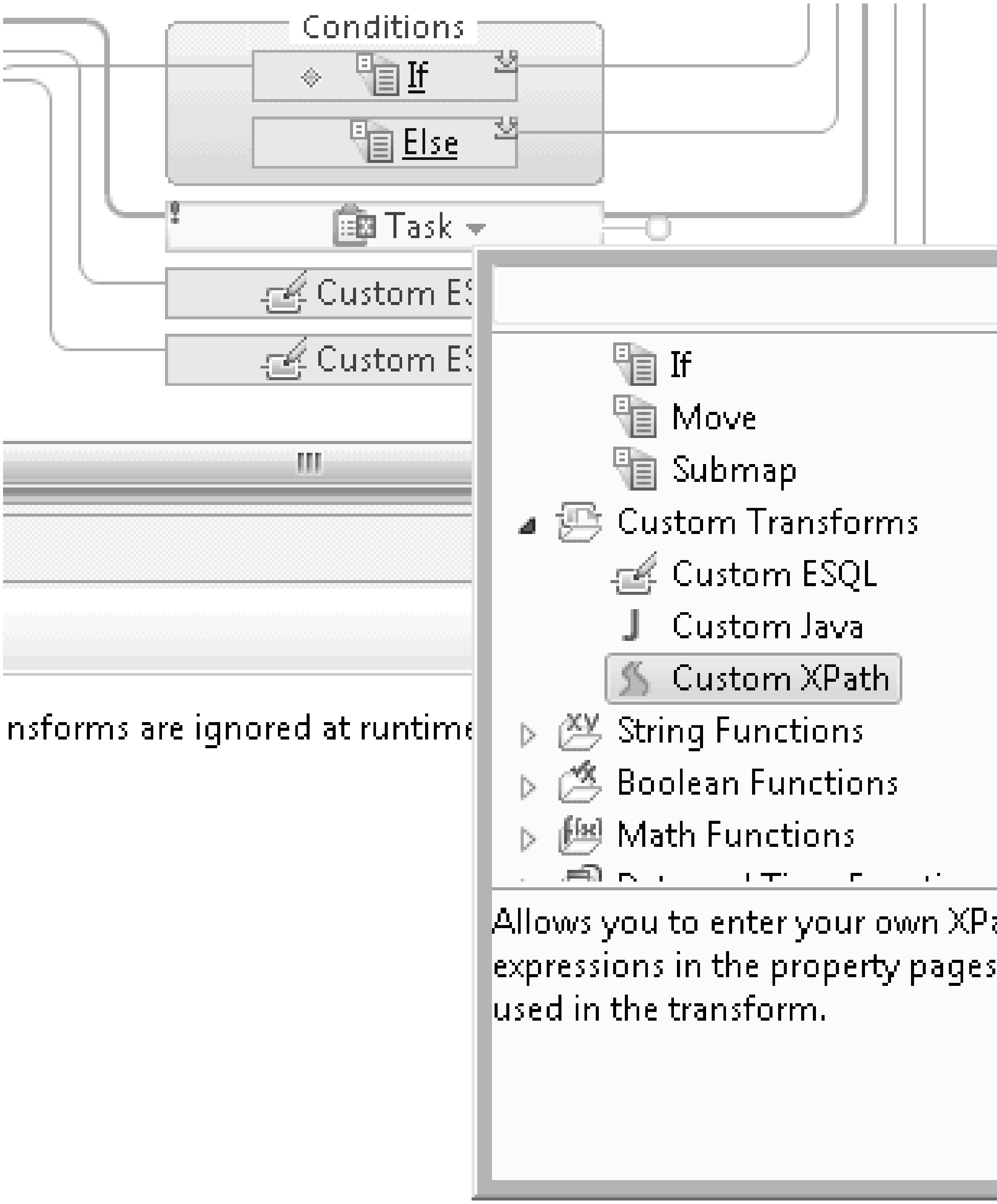
○ To do

Edit documentation...

The transform properties are displayed in the Properties view.

2. In the Properties view, click the **Documentation** tab to review details about the mapping structure that was not re-created by the conversion process. You can see that in the scenario, the information in the **Documentation** tab explains how to resolve the Task: by creating an equivalent expression for the function.

3. In the Graphical Data Mapping editor, change the Task transform into a functional transform type by clicking the arrow in the transform box, then selecting from the list of available transforms. For the scenario, select **Custom XPath**.

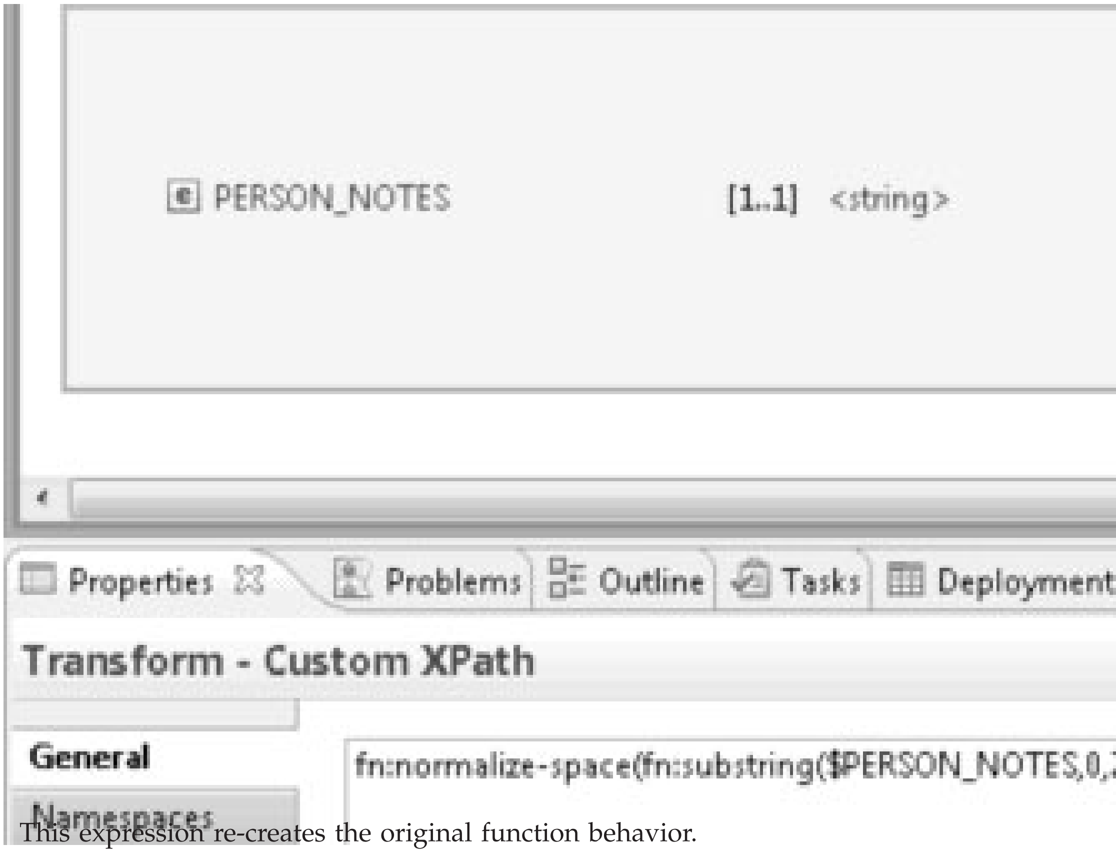


Allows you to enter your own XPa
expressions in the property pages
used in the transform.

4. In the Properties view, click the General tab.
5. In the General tab field, enter the following XPath expression:

```
fn:normalize-space( fn:substring( $PERSON_NOTES, 0, 24))
```

Use the content assist (Ctrl + Space) when you construct your XPath expression to ensure that you always use a valid variable name because the input element could be different across different transforms. There is an example of the importance of using content assist in step 9 on page 27.



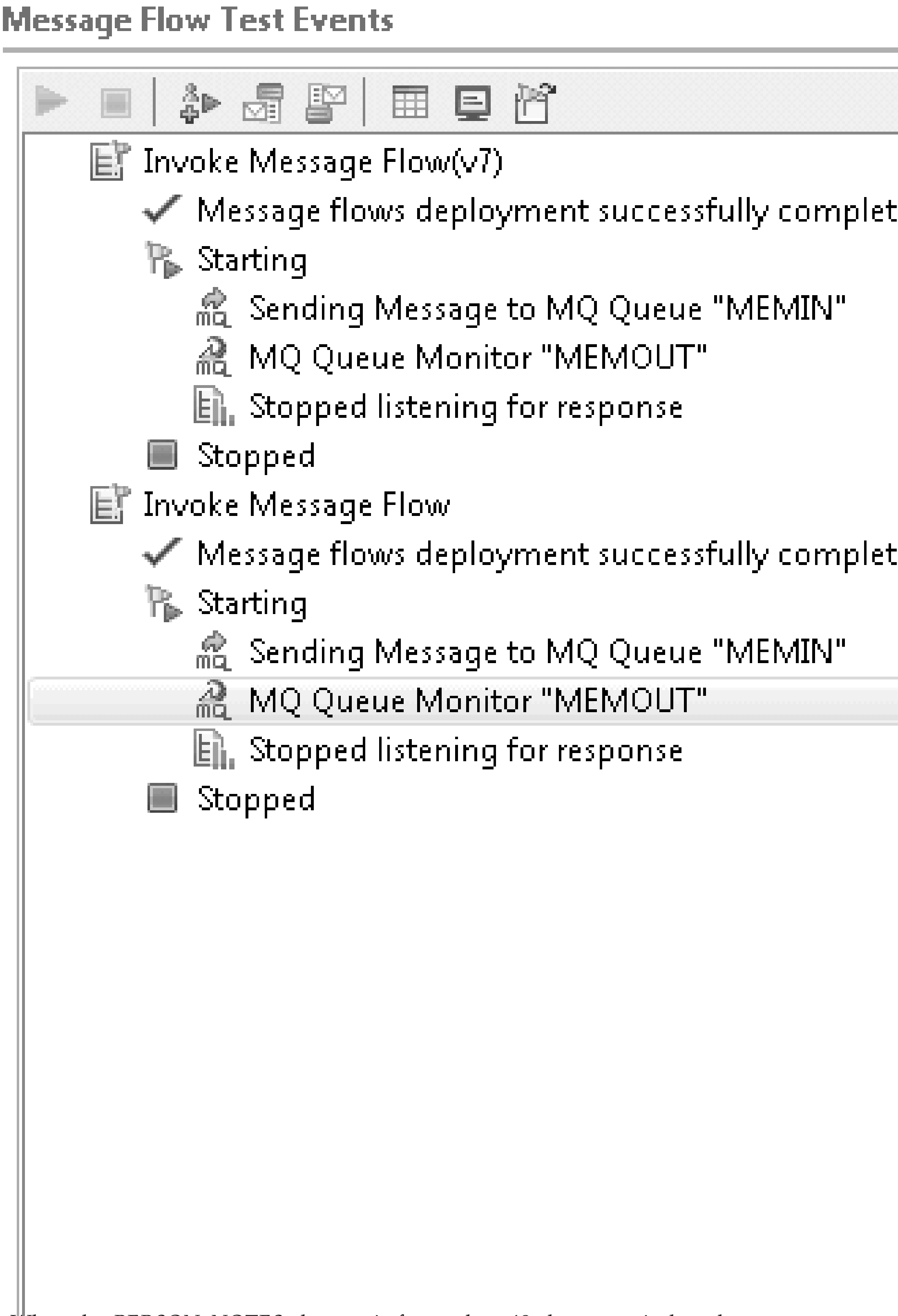
```
fn:normalize-space(fn:substring($PERSON_NOTES,0,
```

This expression re-creates the original function behavior.

6. Save your updated map.

There is an optional set of steps here to test the graphical data map now, because the test shows how your mapping behavior has changed due to the updated logic, and why it is important to check your conversions. If you want to create a working map only, go to step 8 on page 27.

7. Optional: Test the change to the ESQL `NULL` behavior in the scenario map.
   a. Try following the steps in Chapter 3, “Verifying the solution,” on page 29.



Message Flow Test Events

- Invoke Message Flow(v7)
  - Message flows deployment successfully complet
  - Starting
    - Sending Message to MQ Queue "MEMIN"
    - MQ Queue Monitor "MEMOUT"
    - Stopped listening for response
  - Stopped
- Invoke Message Flow
  - Message flows deployment successfully complet
  - Starting
    - Sending Message to MQ Queue "MEMIN"
    - MQ Queue Monitor "MEMOUT"
    - Stopped listening for response
  - Stopped

When the *PERSON_NOTES* element is fewer than 48 characters in length, the output element *MEMBER_COMMENT3* is created as an empty XML element.

b. Review the original Version 7.0 test results. You can see that *MEMBER_COMMENT3* was excluded in this condition. This is because the

ESQL `NULL` processing is different. In an XML/XPath standards-based graphical data map, you can model `NULL` only as an empty element, with `xsi:nil` set if the model defines the element as nillable. If such a change was unexpected and unwanted, it could lead to processing or validation errors. You must add an XPath condition expression to prevent the `commentPart3()` ESQL being invoked when the input data is fewer than 48 characters long.

   c. Continue the scenario from step 8.

8. In the Graphical Data Mapping editor, select the Custom ESQL transform that connects *PERSON_NOTES*. In the Properties view, click the Condition tab.
9. In the Condition tab, enter the following XPath expression in the field:

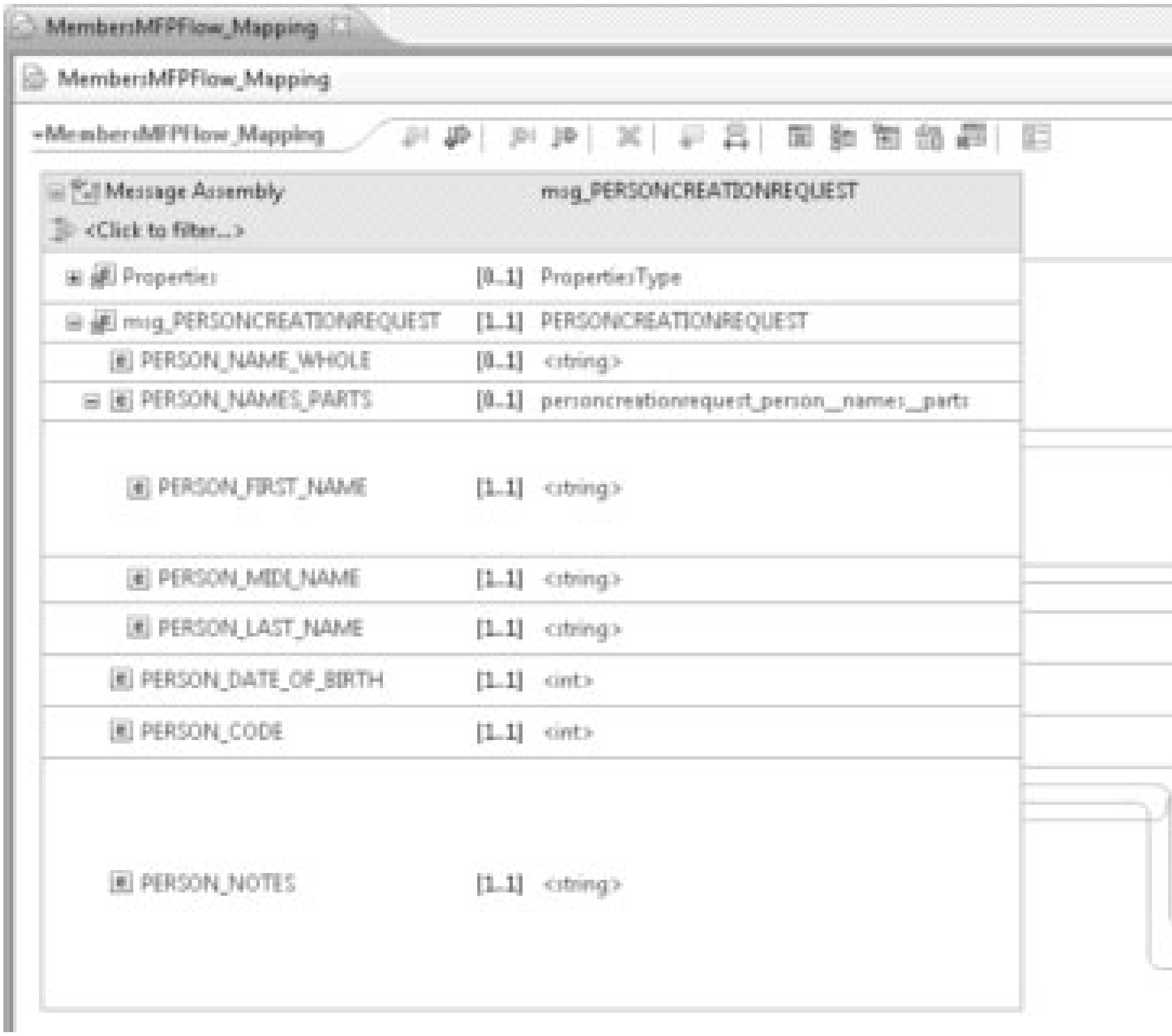
   ```
   fn:string-length( $PERSONNOTES ) > 48
   ```

   This condition prevents *MEMBER_COMMENT3* from appearing if the value of *PERSON_NOTES* is fewer than 48 characters.

   Use the content assist (Ctrl + Space) when you construct your XPath expression. The content assist might suggest a different value for `$PERSON_NOTES` if you modify the transforms outside of the steps provided. For example, content assist might suggest `$PERSON_NOTES1`. Accept this value, because it is correct for the scenario.
10. Your Task transform is resolved to a working Custom XPath transform. Save your updated map.
11. Save your updated map.

Results

You successfully removed the annotations and errors from the converted graphical data map. Your mapping solution is now ready to use.



MembersMFPFlow_Mapping

MembersMFPFlow_Mapping

▾MembersMFPFlow_Mapping

Message Assembly		msg_PERSONCREATIONREQUEST
<Click to filter...>		
Properties	[0..1]	PropertiesType
msg_PERSONCREATIONREQUEST	[1..1]	PERSONCREATIONREQUEST
PERSON_NAME_WHOLE	[0..1]	<string>
PERSON_NAMES_PARTS	[0..1]	personcreationrequest_person__names__parts
PERSON_FIRST_NAME	[1..1]	<string>
PERSON_MIDL_NAME	[1..1]	<string>
PERSON_LAST_NAME	[1..1]	<string>
PERSON_DATE_OF_BIRTH	[1..1]	<int>
PERSON_CODE	[1..1]	<int>
PERSON_NOTES	[1..1]	<string>

To check that your map works as intended, follow the steps in Chapter 3, “Verifying the solution,” on page 29.

Chapter 3. Verifying the solution

You can check that the solution works for your own files, or for the provided scenario.

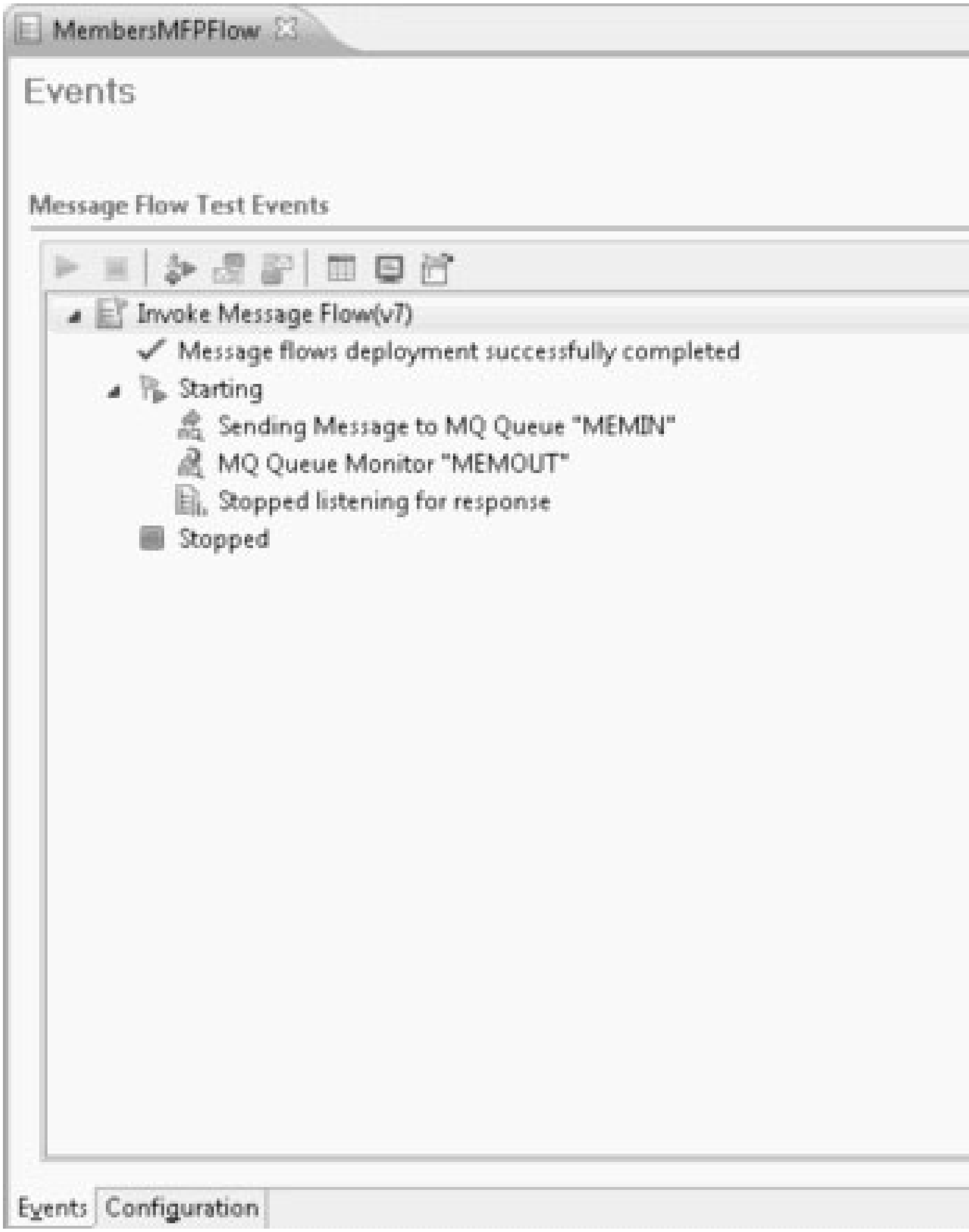
About this task

After you have completed the mapping conversion, and adjusted any flagged transforms, you can check that the map works as intended by running the Test Client. If you are using your own project files, create your own message flow tests by following the instructions for Testing a message flow that has WebSphere MQ nodes in the Information Center.

In the scenario, a message flow test file is already created for you to use. To verify the scenario:

Procedure

1. In the Application view, open `MembersMFPFlow.mbtest`. The Test Client opens. A previously run test named **Invoke Message Flow(v7)** is listed in the **Message Flow Test Events** pane.



Invoke Message Flow(v7)

✓ Message flows deployment successfully completed

Starting

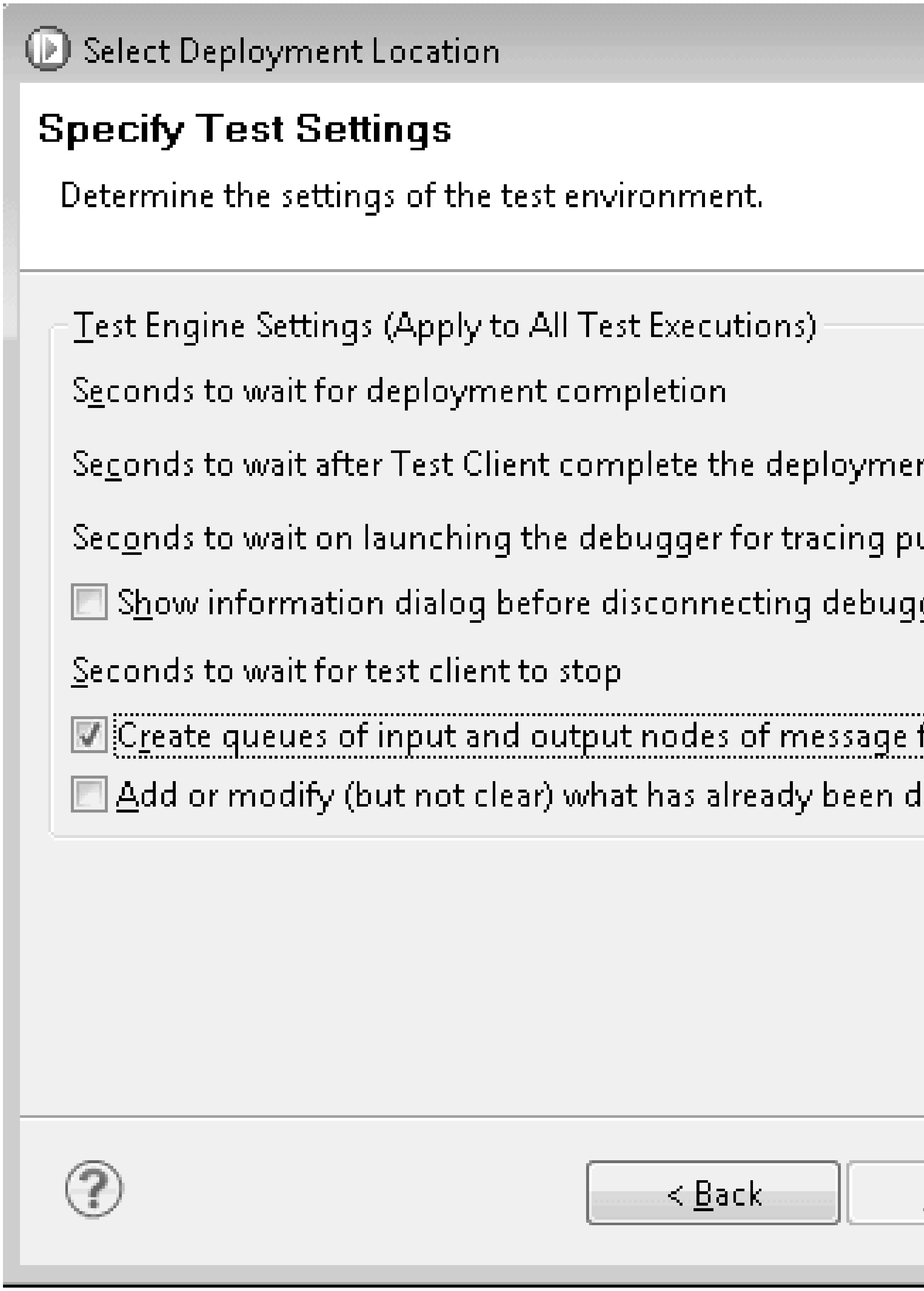
Sending Message to MQ Queue "MEMIN"

MQ Queue Monitor "MEMOUT"

Stopped listening for response

Stopped

2. Review the results of **Invoke Message Flow(v7)** in the **Detailed Properties** tab by clicking each stage of the test that is listed in **Message Flow Test Events**. These are the results from running the message flow before conversion.
3. Right-click **Invoke Message Flow(v7)** and click **Re-run**. The Select Deployment Location wizard opens.
   a. On the Specify Test Settings page, select **Create queues of input and output nodes of message flows when host name is localhost**. Click **Finish**.



Select Deployment Location

Specify Test Settings

Determine the settings of the test environment.

Test Engine Settings (Apply to All Test Executions)

Seconds to wait for deployment completion

Seconds to wait after Test Client complete the deploymer

Seconds to wait on launching the debugger for tracing pu

□ Show information dialog before disconnecting debugg

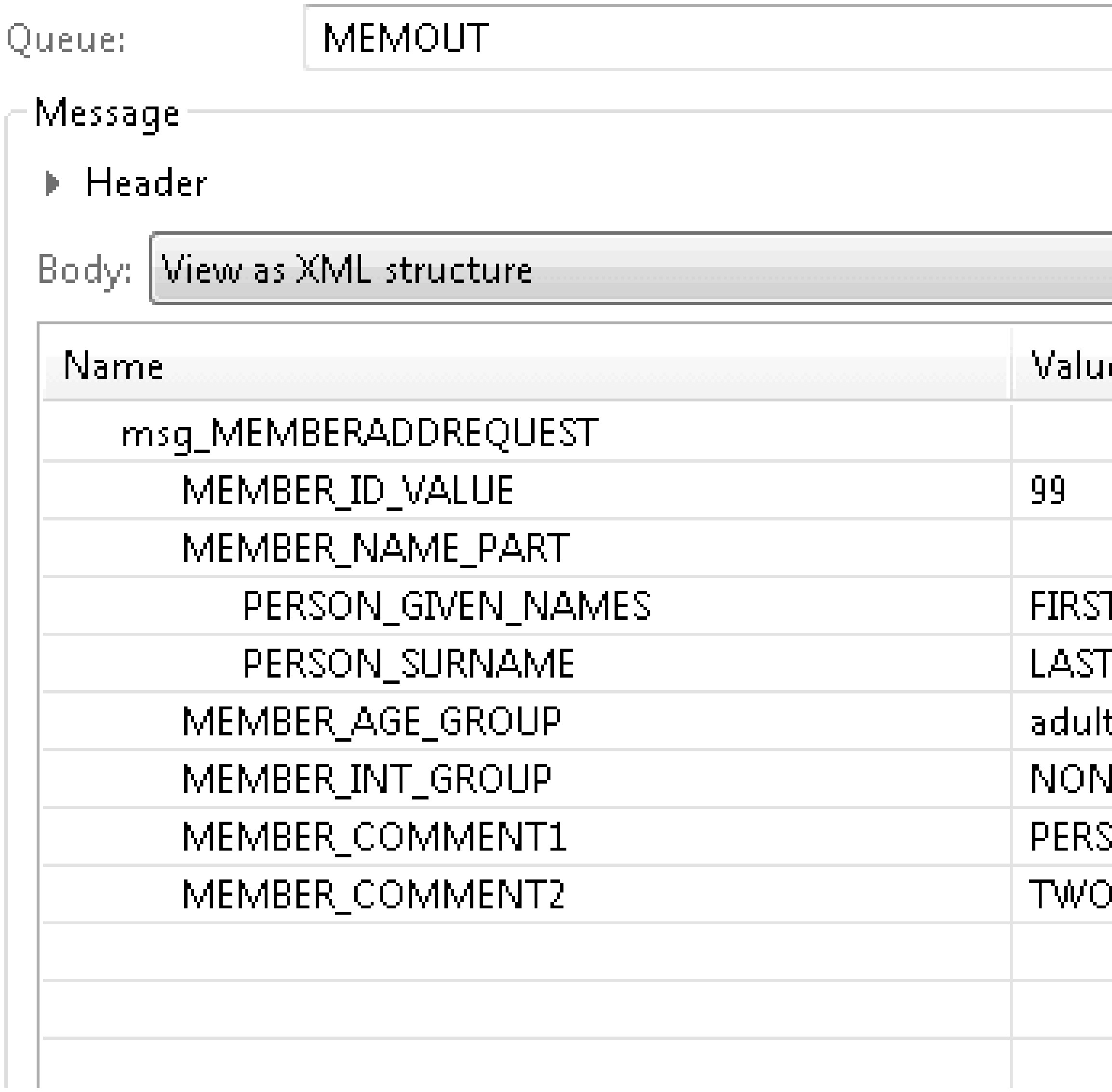
Seconds to wait for test client to stop

☑ Create queues of input and output nodes of message f

□ Add or modify (but not clear) what has already been d

The test is rerun; observe the results in the **Detailed Properties** tab.

4. In the **Message Flow Test Events** pane, select **MQ Queue Monitor "MEMOUT"**. Look at the output message that is displayed in the **Detailed Properties** pane. Compare the output message to that of the original test.



Queue: MEMOUT

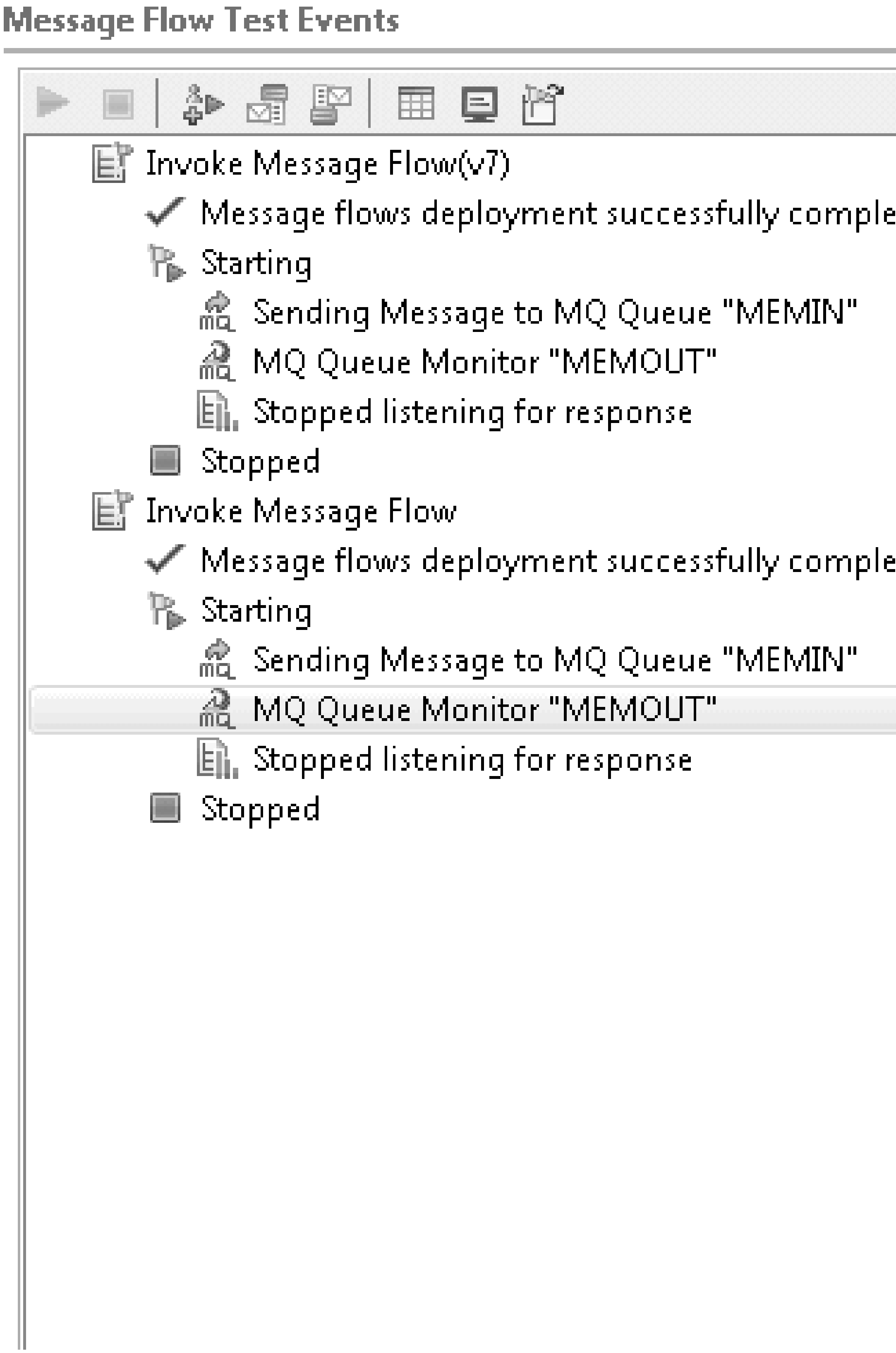
Message

Header

Body: View as XML structure

Name	Valu
msg_MEMBERADDREQUEST	
MEMBER_ID_VALUE	99
MEMBER_NAME_PART	
PERSON_GIVEN_NAMES	FIRST
PERSON_SURNAME	LAST
MEMBER_AGE_GROUP	adult
MEMBER_INT_GROUP	NON
MEMBER_COMMENT1	PERS
MEMBER_COMMENT2	TWO

a. Optional: If you are following the steps for the optional `NULL` behavior check in step 7 of Configuring Task transforms, you can see that *MEMBER_COMMENT3* is displayed, but the **Value** field is empty.



Message Flow Test Events

- Invoke Message Flow(v7)
  - ✓ Message flows deployment successfully comple
  - Starting
    - Sending Message to MQ Queue "MEMIN"
    - MQ Queue Monitor "MEMOUT"
    - Stopped listening for response
  - Stopped
- Invoke Message Flow
  - ✓ Message flows deployment successfully comple
  - Starting
    - Sending Message to MQ Queue "MEMIN"
    - MQ Queue Monitor "MEMOUT"
    - Stopped listening for response
  - Stopped

Results

You have successfully completed the scenario, by converting a message map, configuring the transforms, and verifying the output.

Chapter 4. Terminology

An explanation of the most common terms that are used in this scenario.

Message maps
The data map type that you can create and edit in WebSphere Message Broker Version 6.1 and Version 7.0. The file extension for a message map is `.msgmap`. Message maps are based on ESQL code. They can be deployed, but they cannot be edited in WebSphere Message Broker Version 8.0 or later.

Graphical data maps
The data map type that you can create and edit in WebSphere Message Broker Version 8.0 and IBM Integration Bus. The file extension for a graphical data map is `.map`. The Graphical Data Mapping editor has a dedicated runtime execution engine, instead of being just an ESQL code generation tool. Compared with message maps, graphical data maps have the following benefits:

- Faster at runtime
- Increased reliability and stability
- Greater range of options for transforms
- Easier to create and edit in the Graphical Data Mapping editor
- Are created in a common component across other IBM products, such as WebSphere Application Server.

Graphical Data Mapping editor
The editor for graphical data maps. The Graphical Data Mapping editor is a component that is part of WebSphere Message Broker Version 8.0 and IBM Integration Bus or later versions. It is also available as a stand-alone component, and as a component in other IBM products, such as WebSphere Application Server. The Graphical Data Mapping editor has a dedicated runtime execution engine that uses the broker or integration node Java virtual machine, whereas message mapping used ESQL statements. This change in underlying logic enables you to use a wider variety of transformations.

Mapping node
You can use the Mapping node to construct one or more new messages and populate them with various types of information. The Mapping node can transform a message from one format to another, by using a message map or graphical data map. The Mapping node that is used in Version 6.1 and Version 7.0 of WebSphere Message Broker is different from the Mapping node that is used in the later versions. It must be replaced by the latest type to accept graphical data maps.

Transforms
Transforms that are used in the Graphical Data Mapping editor use different underlying logic when compared to the types used in the message mapping editor. This change provides a wider range of transforms. You might find that you have to think differently about what transforms you might need in your messaging solutions, because ESQL map (esql:) functions are not used in graphical data maps.

For example, a `For` statement in a message map that produces one output for each input is converted to a ForEach transform in a graphical data map. It is flagged for review and is explained in more detail on the Documentation tab of the Properties view.

When a converted message map includes calls to predefined ESQL map (`esql:`) functions, each of these calls is converted to an XPath or Custom XPath transform. If there is no XPath equivalent of an ESQL map function, it is replaced with a Task transform in your graphical data map. You must replace each of these Task transforms with a Custom XPath, Custom Java, or Custom ESQL transform that re-creates the required behavior.

Task transforms

Task transforms in a converted graphical data map indicate that further configuration is required before the map can operate as intended. The Task transform is ignored at runtime.

Some information is provided on the Documentation tab to indicate the problem. When you click a Task transform, in the General tab you can define what problem requires resolving by selecting from one of four options:

- Error
- Warning
- Info
- To Do

If the map is complex, you can use the Task transform like code comments to indicate what further work needs to be done.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software for use with this program.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Important: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information (www.ibm.com/legal/copytrade.shtml).

Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To make comments about the functions of IBM products or systems, talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, to this address:

  User Technologies Department (MP095)
  IBM United Kingdom Laboratories
  Hursley Park
  WINCHESTER,
  Hampshire
  SO21 2JN
  United Kingdom
- By fax:
  - From outside the U.K., after your international access code use 44-1962-816151
  - From within the U.K., use 01962-816151
- Electronically, use the appropriate network ID:
  - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
  - IBMLink: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

Printed in USA