WebSphere Message Broker
Version 8 Release 0

Scenario: Reusing common application logic

Note

Before using this information and the product it supports, read the information in “Notices” on page 85.

Published date: 02 October 2013

Contents

Figures

Chapter 1. Introduction

You use subflows to group common functionality that you want to use several times within multiple integration solutions. This scenario provides information so that you can choose the type of subflow that meets your requirements, and how WebSphere® Message Broker handles each type of subflow at the different phases of your integration project.

About this task

When you create an integration solution, you can create an application, a service, or a Message Broker project which are the containers for the message flows and subflows that define the functionality. Your solution can reuse existing functionality or can define new functionality that can be reused by other solutions in your organization.

In WebSphere Message Broker Version 8.0, you create a subflow to group common functionality so it can be used by one or more solutions. You create new subflows as `.subflow` files. Subflows created as `.msgflow` files are available for compatibility with earlier versions.

Your can also use legacy subflows which are subflows created as `.msgflow` files in earlier versions of WebSphere Message Broker. These legacy subflows can be reused in new integration solution developments in WebSphere Message Broker Version 8.0, but you should convert them to a `.subflow` file.

To choose which type of subflow to create, you must consider the limitations that apply when you design a subflow and how you build the BAR file for deployment. In addition, you must understand how you update a subflow, and whast are the performance parameters that you can tune on a subflow.

What to do next

Review the following topics to understand how you can use subflows to reuse functionality across one or more integration solutions:

- "Context"
- "Technical solution" on page 3
  - "Technical solution: Creating a subflow as a `.subflow` file (Recommended)" on page 4
  - "Technical solution: Creating a subflow as a `.msgflow` file (Maintained for compatibility with previous versions)" on page 5
  - "Technical solution: Converting legacy subflows" on page 6

Context

This scenario explains how you can use subflows to reuse common logic in an integration solution; how a subflow behaves at runtime based on its design and implementation; and how you convert a legacy subflow into a WebSphere Message Broker Version 8 subflow.

When wanting to reuse common functionality in new integration solutions, there are a number of different points you might be starting from. This scenario covers the following situations:

- You might be a new user of WebSphere Message Broker who needs to develop an integration solution for the first time. You want to learn how to design and implement common logic that you know can be reused by future solutions in your organization.
- You might be an existing user of WebSphere Message Broker who has recently migrated to version 8. You are in the process of developing a new integration solution that reuses legacy subflows.
- You might be an existing user of WebSphere Message Broker who needs to develop common functionality that will be used by future integration solutions.

Regardless of your starting point, this scenario has been divided into sub-scenarios to help you understand how to use subflows in WebSphere Message Broker:

- Sub-scenario 1: Creating a new subflow as a `.subflow` file
- Sub-scenario 2: Creating a new subflow as a `.msgflow` file
- Sub-scenario 3: Converting a legacy subflow

Creating a new subflow as a `.subflow` file

This is the recommended approach for creating subflows.

- You are a new user of WebSphere Message Broker, or you are an existing user that needs to create new subflows.
- Your company may have multiple applications, such as SAP or Siebel, as well as numerous web services that use WebSphere Message Broker as the integration backbone through which all data transformation takes place.
- Your company might also have a corporate requirement that stipulates that all systems that handle sensitive client information must comply fully with audit requirements.

To comply with the corporate requirement, you can design and implement an audit subflow that can be used by any message flow. You can create a subflow as a `.subflow` file or as a `.msgflow` file. However, you are recommended to create the subflow as a `.subflow` file. The subflow defines new functionality that will be reused by other solutions.

Creating a new subflow as a `.msgflow` file

This option is maintained for subflow compatibility with previous releases.

- You are an existing user of WebSphere Message Broker.
- You have migrated your legacy subflows to version 8.
- Due to resource constraints, you cannot convert your subflows into `.subflow` files.
- The existing subflows are widely used across multiple solutions.

You continue to create subflows as `.msgflow` files whenever these subflows include existing subflows. In any other cases, you create your new subflows as `.subflow` files.

Converting a legacy subflow

- You are an existing user of WebSphere Message Broker.

- You have migrated your legacy subflows to version 8.
- You want to convert your legacy subflows into `.subflow` files.

You convert your subflows created as `.msgflow` files into `.subflow` files by using the conversion function provided by WebSphere Message Broker.

Related information:

"Technical solution"
To reuse common application logic in multiple integration solutions, you create a subflow. You must choose whether you create the subflow as a `.subflow` file or as a `.msgflow` file, or whether you must convert your legacy subflow to a `.subflow` file. You must understand how WebSphere Message Broker handles subflows at the deployment and run time, so that you can configure the subflow properties based on your solution requirements.

Technical solution

To reuse common application logic in multiple integration solutions, you create a subflow. You must choose whether you create the subflow as a `.subflow` file or as a `.msgflow` file, or whether you must convert your legacy subflow to a `.subflow` file. You must understand how WebSphere Message Broker handles subflows at the deployment and run time, so that you can configure the subflow properties based on your solution requirements.

The following table summarizes the recommendations regarding subflow type in WebSphere Message Broker:

Table 1. Summary table with subflow type adoption recommendations

	New subflow created as a `.subflow` file	**New subflow created as a `.msgflow` file**	**Legacy subflow (`.msgflow` file)**
WebSphere Message Broker Version 6	NOT supported	Supported	Supported
WebSphere Message Broker Version 7	NOT supported	Supported	Supported
WebSphere Message Broker Version 8	Supported and recommended for all new developments	Supported, but not recommended for new developments (This option is available for compatibility with earlier versions.)	Supported, however, you are recommended to convert your legacy subflows to a `.subflow` file

The scenario has been divided into three sub-scenarios that explain the different subflow adoption options:

- "Technical solution: Creating a subflow as a `.subflow` file (Recommended)" on page 4
- "Technical solution: Creating a subflow as a `.msgflow` file (Maintained for compatibility with previous versions)" on page 5
- "Technical solution: Converting legacy subflows" on page 6

Related tasks:

Chapter 2, "Implementing the solution," on page 7
Whether you create a subflow as a `.subflow` file or as a `.msgflow` file, or whether you convert your legacy subflow to a `.subflow` file, you must understand how WebSphere Message Broker handles subflows at deployment and run time, so that you can design and configure your subflow based on your solution requirements.

Technical solution: Creating a subflow as a `.subflow` file (Recommended)

To reuse common application logic in multiple integration solutions, you create a subflow as a `.subflow` file.

Subflows are resources that you can include in message flows. Message flows define the functionality provided by an application, a service, or a Message Broker project.

When you use subflows as part of your integration solution, you must complete the following steps before you create a subflow:

1. You must verify that your subflow follows the design guidelines for a subflow created as a `.subflow` file. For more information, see "Designing a subflow created as a `.subflow` file" on page 10.
2. You must choose whether to create the subflow within a library, an application, a service, or a Message Broker project.

   For new WebSphere Message Broker Version 8 developments, you create applications or services. You use libraries for sharing resources between applications and services. Message Broker projects are maintained for compatibility with earlier versions. For more information, see ../../com.ibm.etools.mft.doc/bc88710_.dita.
3. You must decide which method to use to set the version of the subflow based on where do you plan to check its value at run time. For more information, see "Versioning a subflow" on page 13.

Note: WebSphere Message Broker releases earlier than version 8 only supported subflows created as a `.msgflow` file. However, if you plan to develop new integration solutions, you should create subflows created as `.subflow` files.

After you create your subflow and include it in a message flow, you must complete the following steps before deploying the message flow into a broker run time:

1. You must choose how to package your integration solution resources into a BAR file. The BAR file contains your message flows and subflows. For more information, see "Creating a BAR file that includes a subflow created as a `.subflow` file" on page 23.

   Note: You are recommended to work with WebSphere Message Broker Version 8.0.0.1 or later.

   Note: Building a BAR file for an application or service with the build option **Compile and in-line resources** not selected is the recommended approach to packaging integration solutions in WebSphere Message Broker Version 8.0.0.1 and later versions.
2. You must understand how WebSphere Message Broker handles threads at run time based on your subflow design, and how you configure the overall

message flow BAR file deployment properties and the Input node deployment properties. For more information, see "Configuring a message flow that includes subflows" on page 32.

After your solution is running in a broker run time, you may need to update a version of a subflow. For more information, see "Updating a subflow" on page 31.

Related tasks:

"Implementing the solution: Creating a subflow as a `.subflow` file" on page 10
To reuse common application logic in multiple integration solutions, you can create a subflow as a `.subflow` file. You must understand how WebSphere Message Broker handles subflows at deployment and run time, so that you can configure the subflow properties based on your solution requirements.

Technical solution: Creating a subflow as a `.msgflow` file (Maintained for compatibility with previous versions)

To reuse common application logic in multiple integration solutions, you can create a new subflow as a `.msgflow` file. This is not a recommended approach for the development of new subflows. This type of subflow is supported in WebSphere Message Broker Version 8 for compatibility with earlier versions.

A subflow created as a `.msgflow` file cannot be deployed as a source object. The subflow contents are always copied into the parent message flow. The subflow is deployed as part of the main message flow compiled CMF object. Subflows created as a `.msgflow` file cannot be deployed as an independent resource. It always becomes a part of the parent message flow when it is deployed.

When you use subflows as part of your integration solution, you must complete the following steps before you create a subflow:

1. You must verify that your subflow follows the guidelines for a subflow created as a `.msgflow` file. For more information, see "Designing a subflow created as a `.msgflow` file" on page 40.
2. You must choose whether to create a subflow within a library, an application, a service, or a Message Broker project.

   For new WebSphere Message Broker Version 8 developments, you create applications or services. You use libraries for sharing resources between applications and services. Message Broker projects are maintained for compatibility with earlier versions. For more information, see ../../com.ibm.etools.mft.doc/bc88710_.dita.
3. You must decide which method to use to set the version of the subflow. For more information, see "Versioning a subflow" on page 13.

Note: WebSphere Message Broker releases earlier than version 8 only supported subflows created as a `.msgflow` file. However, if you plan to develop new integration solutions, you should create subflows created as `.subflow` files.

After you create your subflow and include it in a message flow, you must complete the following steps before deploying a message flow that includes a subflow into a broker run time:

1. You must choose how to package your integration solution resources into a BAR file. The BAR file contains your message flows and subflows. For more information, see "Creating a BAR file that includes a subflow created as a `.msgflow` file" on page 52.

Note: You are recommended to work with WebSphere Message Broker Version 8.0.0.1 or a later version.

Note: Building a BAR file for an application or service with the build option **Compile and in-line resources** selected is the recommended approach to packaging integration solutions in WebSphere Message Broker Version 8.0.0.1 and later versions.

2. You must understand how WebSphere Message Broker handles threads at run time based on your subflow design, and how you configure the overall message flow BAR file deployment properties and the Input node deployment properties. For more information, see "Configuring a message flow that includes subflows" on page 32.

After your solution is running in a broker run time, you may need to update a version of a subflow. For more information, see "Updating a subflow" on page 31.

Related tasks:

"Implementing the solution: Creating a subflow as a `.msgflow` file" on page 39
To reuse common application logic in multiple integration solutions, you can create a subflow as a `.msgflow` file. You must understand how WebSphere Message Broker handles subflows at deployment and run time, so that you can configure the subflow properties based on your solution requirements.

Technical solution: Converting legacy subflows

To reuse common application logic in multiple integration solutions, you can reuse your legacy subflows. However, you should convert them to subflows created as a `.subflow` file before you reuse them in new integration solutions.

Legacy subflows are subflows that you created in earlier versions of WebSphere Message Broker. These subflows are deployed at run time inlined into the message flow that includes them. Legacy subflows allow you to reuse the common flow logic at the design and development time.

You must complete the following steps to convert a legacy subflow:

1. Learn how WebSphere Message Broker Version 8 handles legacy subflows at deployment and run time. For more information, see "Deploying a subflow created as a `.msgflow` file" on page 57.
2. Learn how WebSphere Message Broker Version 8 handles subflows created as `.subflow` files at deployment and run time. For more information, see "Deploying a subflow created as a `.subflow` file" on page 28.
3. Identify other resources included in your subflow that require conversion, for example, WebSphere Message Broker Version 7.0 Mapping nodes.
4. Convert your legacy subflows created as `.msgflow` files to `.subflow` files.
5. Update existing solutions to use the converted subflow instead of the legacy subflow.

Related tasks:

"Implementing the solution: Converting legacy subflows" on page 68
To convert legacy subflows into subflows created as `.subflow` files, you use the **Convert to subflow** function in WebSphere Message Broker.

Chapter 2. Implementing the solution

Whether you create a subflow as a `.subflow` file or as a `.msgflow` file, or whether you convert your legacy subflow to a `.subflow` file, you must understand how WebSphere Message Broker handles subflows at deployment and run time, so that you can design and configure your subflow based on your solution requirements.

Before you begin

Read the “Technical solution” on page 3 section, and decide which type of subflow you want to create.

Procedure

Follow the relevant sub-scenario:

- If you are developing a new subflow in WebSphere Message Broker, follow the sub-scenario: Creating a subflow as a `.subflow` file.
- If you have legacy subflows from earlier versions, and you still want to develop a new subflow in WebSphere Message Broker as a `.msgflow` file, follow the sub-scenario: Creating a subflow as a `.msgflow` file.
- If you want to convert your legacy subflows, follow the sub-scenario: Converting legacy subflows.

What to do next

When you know which type of subflow you want to create or whether you want to convert a legacy subflow, follow the relevant sub-scenario:

- If you want to create a subflow as a `.subflow` file, see “Implementing the solution: Creating a subflow as a `.subflow` file” on page 10.
- If you want to create a subflow as a `.msgflow` file, see “Implementing the solution: Creating a subflow as a `.msgflow` file” on page 39.
- If you want to convert a legacy subflow, see “Implementing the solution: Converting legacy subflows” on page 68.

Creating the initial configuration for the Error Handler sample

This scenario was developed by using a sample initial configuration. You can set up the sample initial IT configuration to try out the scenario in the same way as it was originally developed.

Before you begin

- Download a copy of the file Error_Handler_sample.zip.
- Make sure you have access to a WebSphere Message Broker runtime environment and a WebSphere Message Broker Toolkit installation with the default configuration deployed. For more information on installing WebSphere Message Broker components, see Installing in the IBM® WebSphere Message Broker information center.

About this task

Complete the following steps to set up the sample initial configuration that was used to develop the scenario.

Procedure

1. From the WebSphere Message Broker Toolkit menu, click **File** > **Import** > **Other** > **Project Interchange** and then click **Next**. The Import Projects dialog box opens.
2. Next to the drop-down menu for the "From zip file" field, click **Browse**, navigate to the `Error_Handler_sample.zip` file and click **Open**. The Import Projects dialog box is populated with the list of resources from the `Error_Handler_sample.zip` file.
3. Ensure that all the resources are selected and click **Finish**. In the Broker Development pane you should see the following entries under the **Independent resources** folder:
   - One Message Broker project named **Error Handler Message Flows**.
   - This project contains the following resources:
     - A message flow named **Main_Flow.msgflow** inside the **Flows** folder.
     - A subflow named **Error_handler.msgflow** inside the **Flows** folder.
     - Two esql files.
4. To deploy the `errorHandlersampleBAR.bar` BAR file, right-click on the BAR file name, click **Deploy**, select an execution group, and click **Finish**. The resources are deployed to the broker.

Creating the initial configuration for the Coordinated Request Reply sample

This scenario was developed by using a sample initial configuration. You can set up the sample initial IT configuration to try out the scenario in the same way as it was originally developed.

Before you begin

- Download a copy of the file coordinatedRequestReplyMQAppsample.zip.
- Make sure you have access to a WebSphere Message Broker runtime environment and a WebSphere Message Broker Toolkit installation with the default configuration deployed. For more information on installing WebSphere Message Broker components, see Installing in the IBM WebSphere Message Broker information center.

About this task

Complete the following steps to set up the sample initial configuration that was used to develop the scenario.

Procedure

1. From the WebSphere Message Broker Toolkit menu, click **File** > **Import** > **Other** > **Project Interchange** and then click **Next**. The Import Projects dialog box opens.
2. Next to the drop-down menu for the "From zip file" field, click **Browse**, navigate to the `coordinatedRequestReplyMQAppsample.zip` file and click **Open**.

The Import Projects dialog box is populated with the list of resources from the `coordinatedRequestReplyMQAppsample.zip` file.

3. Ensure that all the resources are selected and click **Finish**. In the Broker Development pane you should see the following entries under the **Broker Development** tab:
   - One application project named **Coordinated request Reply MQ Application**.
   - One library project named **Coordinated Request Reply Library**.
4. To deploy the `coordinated_request_reply.bar` BAR file, right-click on the BAR file name, click **Deploy**, select an execution group, and click **Finish**. The resources are deployed to the broker.

Creating the initial configuration for the MembersMFP sample

This scenario was developed by using a sample initial configuration. You can set up the sample initial IT configuration to try out the scenario in the same way as it was originally developed.

Before you begin

- Download a copy of the file MembersMFP.zip.
- Make sure you have access to a WebSphere Message Broker runtime environment and a WebSphere Message Broker Toolkit installation with the default configuration deployed. For more information on installing WebSphere Message Broker components, see Installing in the IBM WebSphere Message Broker information center.

About this task

Complete the following steps to set up the sample initial configuration that was used to develop the scenario.

Procedure

1. From the WebSphere Message Broker Toolkit menu, click **File** > **Import** > **Other** > **Project Interchange** and then click **Next**. The Import Projects dialog box opens.
2. Next to the drop-down menu for the "From zip file" field, click **Browse**, navigate to the `MembersMFP.zip` file and click **Open**. The Import Projects dialog box is populated with the list of resources from the `MembersMFP.zip` file.
3. Ensure that all the resources are selected and click **Finish**. In the Broker Development pane you should see the following entries under the **Independent resources** folder:
   - Two Message Broker projects named **MembersMFP** and **MembersMSP**.
   - This **MembersMFP** project contains the following resources:
     - A message flow named **Main_Flow.msgflow** inside the **Flows** folder.
     - A subflow named **MembersMFPFlow.msgflow** inside the **Flows** folder.
     - One map named **MembersMFPFlow_Mapping.msgmap**.
     - One esql file.
4. To deploy the sample, create a BAR file, right-click on the BAR file name, click **Deploy**, select an execution group, and click **Finish**. The resources are deployed to the broker.

Implementing the solution: Creating a subflow as a `.subflow` file

To reuse common application logic in multiple integration solutions, you can create a subflow as a `.subflow` file. You must understand how WebSphere Message Broker handles subflows at deployment and run time, so that you can configure the subflow properties based on your solution requirements.

About this task

To create a subflow created as a `.subflow` file, see http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/topic/com.ibm.etools.mft.doc/bc31700_.html.

Note: WebSphere Message Broker releases earlier than version 8 only supported subflows created as a `.msgflow` file. However, if you plan to develop new integration solutions, you should create subflows of type `.subflow`.

Procedure

To implement the technical solution, complete the following tasks:

1. "Designing a subflow created as a `.subflow` file"
2. "Versioning a subflow" on page 13
   - "Defining a keyword to be used as the container of a subflow version" on page 14
   - "Versioning a subflow by using a keyword in the Label property of a Passthrough node" on page 15
   - "Versioning a subflow by using a keyword as a Passthrough node name" on page 17
   - "Versioning a subflow by setting the Version property of the subflow" on page 19
3. "Creating a BAR file that includes a subflow created as a `.subflow` file" on page 23
4. "Deploying a subflow created as a `.subflow` file" on page 28
5. "Updating a subflow" on page 31
6. "Configuring a message flow that includes subflows" on page 32

Designing a subflow created as a `.subflow` file

When designing a subflow, you need to choose whether to use a subflow created as a `.subflow` file or as a `.msgflow` file. In WebSphere Message Broker Version 8, you create your subflows as `.subflow` files for new integration solutions.

You can add subflows into your message flows if either of the following statements is true:

- The subflow that you want to add to a message flow is defined in a library.

  Applications and services can reference libraries.

  Note: A library is a logical grouping of related code, data, or both that typically contains reusable subflows, and other type of resources.
- The subflow that you want to add to a message flow is defined in the same Message Broker project, application, or service as the message flow.

During the development phase, in the WebSphere Message Broker Toolkit, you can define a subflow once, and then reuse it in more than one message flow.

- If you use a subflow across multiple applications or services, you should define the subflow within a library. This library can then be referenced by any application or service that needs access to the resources it contains.
- If you use a subflow multiple times within one application or within one service, then you can define the subflow as part of the application, the service, or the library.

Note: When the WebSphere Message Broker run time encounters a subflow within a message flow, it creates a copy of all the message flow nodes that define that subflow. If the subflow is used more than once, it creates a copy of all the message flow nodes that define the subflow every time it occurs. This increases resource usage, which can affect your overall message flow performance.

Types of subflows supported

In WebSphere Message Broker Version 8, you can create a subflow as a `.subflow` file or as a `.msgflow` file. However, if you develop new integration solutions, you must create subflows of type `.subflow`.

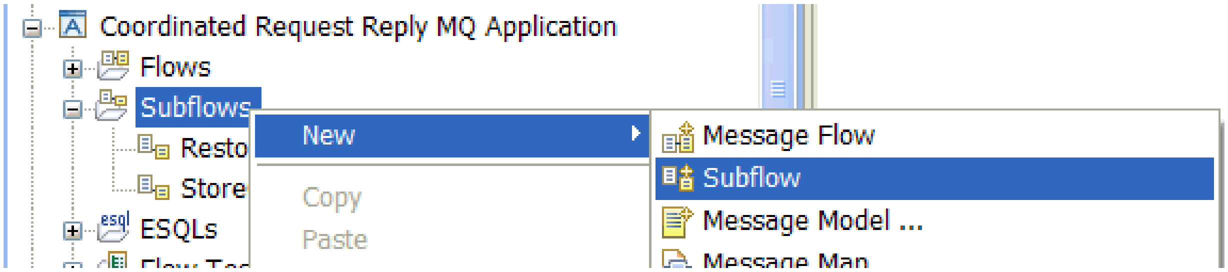
Subflow content

You define subflow content in the same way as you define message flow content, by adding, configuring, and connecting message flow nodes.

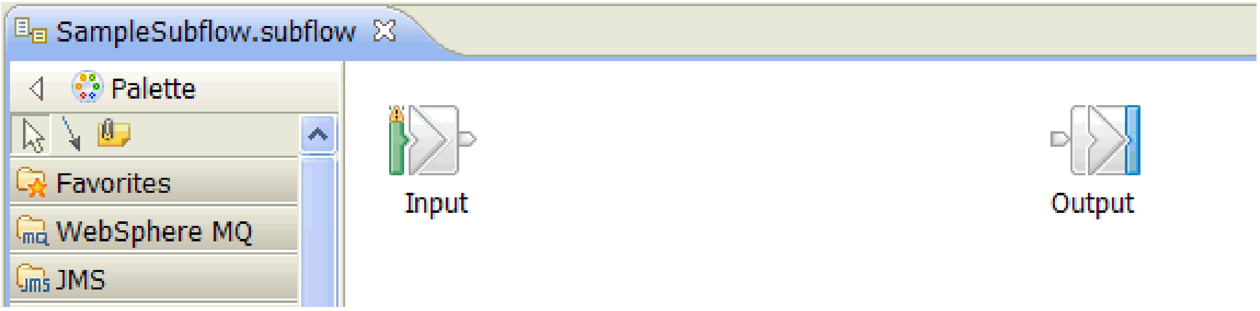
A subflow can have one or more input nodes, and zero, one, or more output nodes.

- If you want to use a subflow as the first node in your message flow, you need to add at least one Input node, such as the MQInput node, and one generic Output node.
- If you want to use a subflow in the middle of a message flow, you need to add at least one generic Input node. You only need to add a generic Output node if you want to connect additional nodes in the message flow after you add the subflow.
- If you want to use a subflow as the last node in a message flow, you need to add at least one generic Input node, and at least one Output node, such as the MQOutput node.

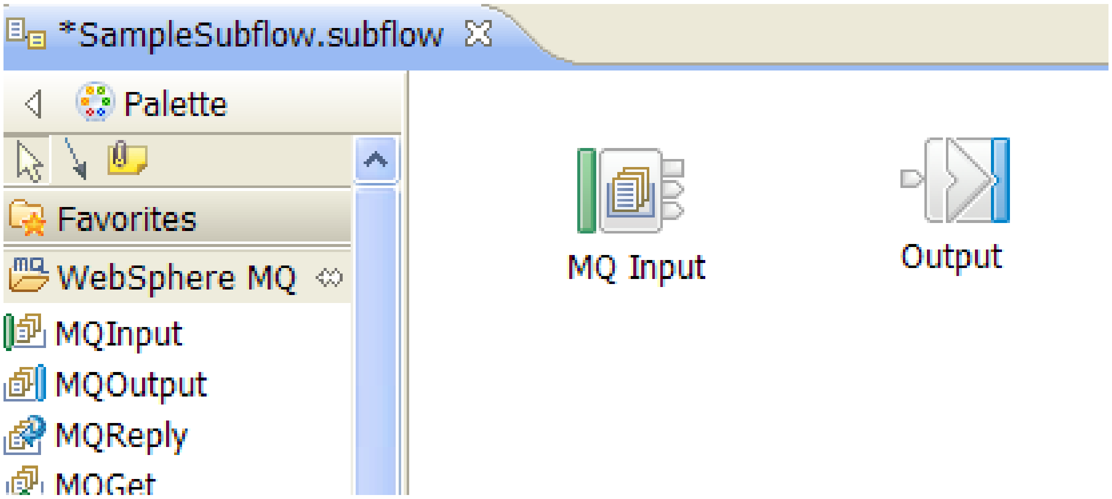
Subflows created as `.subflow` files are grouped in the folder named **Subflows** within your application, service, or Message Broker project. The following figure illustrates the location in a project where you can create subflows or see all the subflows defined under that project:



When you create a subflow created as a .subflow file, the Message Flow editor automatically creates a generic input node and a generic output node.



If you want to use a subflow created as a .subflow file as the first node in a message flow, you must add an Input node such as an MQInput node instead of the generic input node.



You cannot use the following nodes in subflows created as .subflow files:

- Nodes representing subflows that are created as .msgflow files.
- WebSphere Message Broker Version 7.0 Mapping nodes.
- User-defined nodes created from subflows that are created as .msgflow files.
- MQOptimizedFlow nodes.

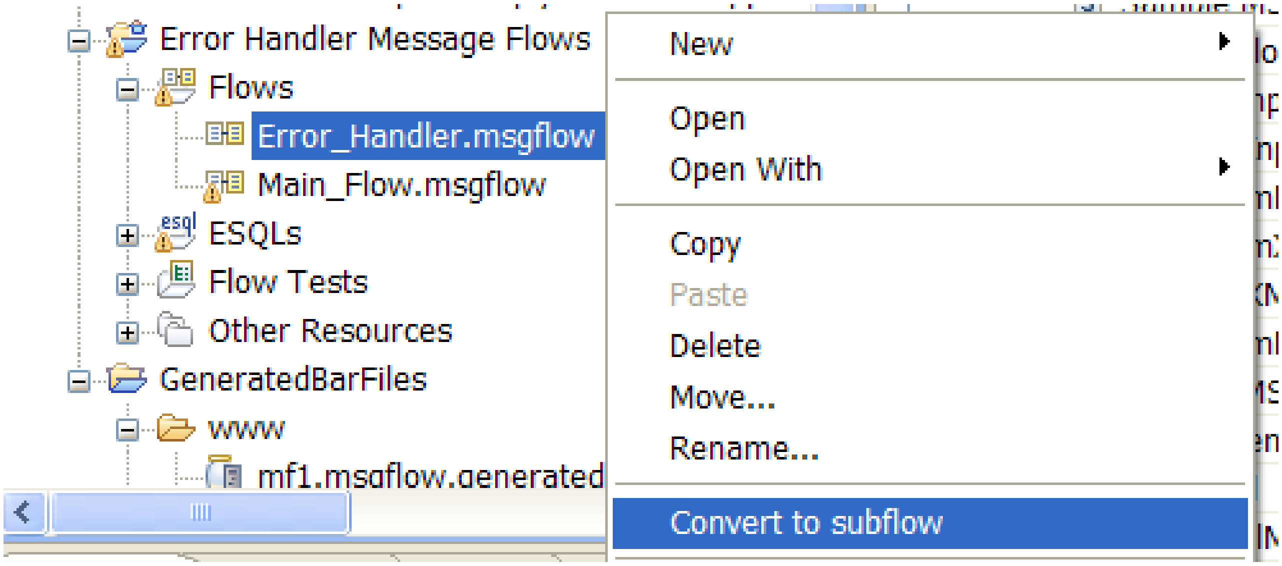
Design considerations

When you design a subflow, you should consider the following recommendations:

- Reuse logic by creating a subflow created as a .subflow file.
- Define subflows of type .subflow in libraries. Create references to a library from your applications or services to be able to reuse the common logic.
- Limit the number of nested subflows.
- Group logic that is defined in consecutive compute nodes into one single compute node. This will improve performance, since the number of nodes loaded at runtime will be reduced.
- Add a version to the subflow. For more information, see “Versioning a subflow” on page 13.

- Do not nest subflows of different types, that is, do not add a subflow created as a `.msgflow` file into a subflow created as a `.subflow` file or vice versa.
- To reuse a subflow created as a `.msgflow` file into a subflow created as a `.subflow` file, convert the subflow created as a `.msgflow` file into a `.subflow` file.

  To convert a subflow, right-click the subflow name, and select **Convert to subflow**. For more information, see "Converting subflows" on page 69.



Versioning a subflow

To enable version control of a subflow at run time, you can set the version by using a keyword or by setting the subflow Version property.

About this task

You set the version of a subflow based on where you want to check its value at run time. The following table summarizes the different components that you can use to check the version of a subflow.

Table 2. Summary of the components that you can use to check the version of a subflow

Methods	WebSphere Message Broker Toolkit	WebSphere Message Broker Explorer	WebSphere Message Broker console
Configure a keyword in the Label property of a Passthrough node	Valid to check the value at development time	Valid to check the value at run time	Valid to check the value at run time
Name a Passthrough node with a keyword	Valid to check the value at development time	Valid to check the value at run time	Valid to check the value at run time
Configure the Version property of a subflow	Valid to check the value at development time	Valid to check the value at run time	Not a valid option

Procedure

To enable version control of a subflow at run time, you can set the version of a subflow in any of the following ways:

- Configure a keyword in the Label property of a Passthrough node. For more information, see "Defining a keyword to be used as the container of a subflow version," and then "Versioning a subflow by using a keyword in the Label property of a Passthrough node" on page 15.

- Name a Passthrough node with a keyword. For more information, see "Defining a keyword to be used as the container of a subflow version," and then "Versioning a subflow by using a keyword as a Passthrough node name" on page 17.
- Configure the Version property of a subflow. For more information, see "Versioning a subflow by setting the Version property of the subflow" on page 19.

What to do next

After you set the version of a subflow, create a BAR file. For more information, see "Creating a BAR file that includes a subflow created as a `.subflow` file" on page 23 or "Creating a BAR file that includes a subflow created as a `.msgflow` file" on page 52.

Defining a keyword to be used as the container of a subflow version

You can use a *keyword* to add more information to a message flow or subflow.

About this task

The information that is passed through keywords is only accessible at run time when the object for which it has been defined is deployed and is running in the broker runtime environment.

You can check the value of a keyword in any of the following ways:

- In the WebSphere Message Broker console, by using the `mqsireadbar` command to read the properties stored in the broker archive (BAR) file.
- In the WebSphere Message Broker Explorer, on the properties of a deployed message flow or subflow as last deployed to a particular broker.
- In the runtime environment, if you enable user trace for that message flow.

Procedure

To define a keyword that you can use as the container of a subflow version, complete the following step:

Define the syntax of the keyword as follows: `$MQSI` *`name of your subflow`* `Subflow Version = Version` *`xxxx`* `MQSI$` where *name of your subflow* is the name of your subflow; and *xxxx* is your subflow version identifier.
You can define your own keyword to define the version. For guidance on defining *keywords*, see Guidance for defining keywords. You can define your own naming convention to assign a version value to a subflow.

What to do next

Use the keyword to set the version of a subflow. For more information, see "Versioning a subflow by using a keyword in the Label property of a Passthrough node" on page 15 or "Versioning a subflow by using a keyword as a Passthrough node name" on page 17.

Related information:

mqsireadbar command

Versioning a subflow by using a keyword in the Label property of a Passthrough node

To define the version of a subflow, you can define a keyword in the Label property of a Passthrough node.

Before you begin

Set up the initial configuration. See “Creating the initial configuration for the Error Handler sample” on page 7.

About this task

When you set a subflow version by defining a keyword in the Label property of a Passthrough node, you can check the value at run time in the WebSphere Message Broker console and in WebSphere Message Broker Explorer.

Procedure

You must complete the following steps in the WebSphere Message Broker Toolkit:

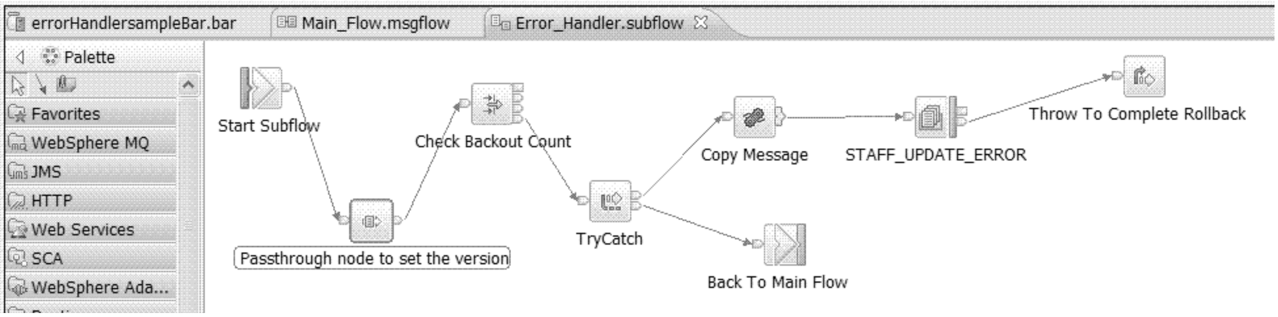
1. Open the subflow in the Message Flow editor.
2. Add a Passthrough node to a subflow.
3. Connect the Passthrough node to other nodes in the subflow or leave it unconnected.
4. Define a keyword. For more information, see “Defining a keyword to be used as the container of a subflow version” on page 14.
5. Set the keyword in the **Label** property of the Passthrough node.

Example

For example, in the Error Handler sample provided with WebSphere Message Broker, you can add and check the version of the subflow by completing the following steps:

1. In the WebSphere Message Broker Toolkit, open the **Error_Handler.subflow** subflow in the Message Flow editor:

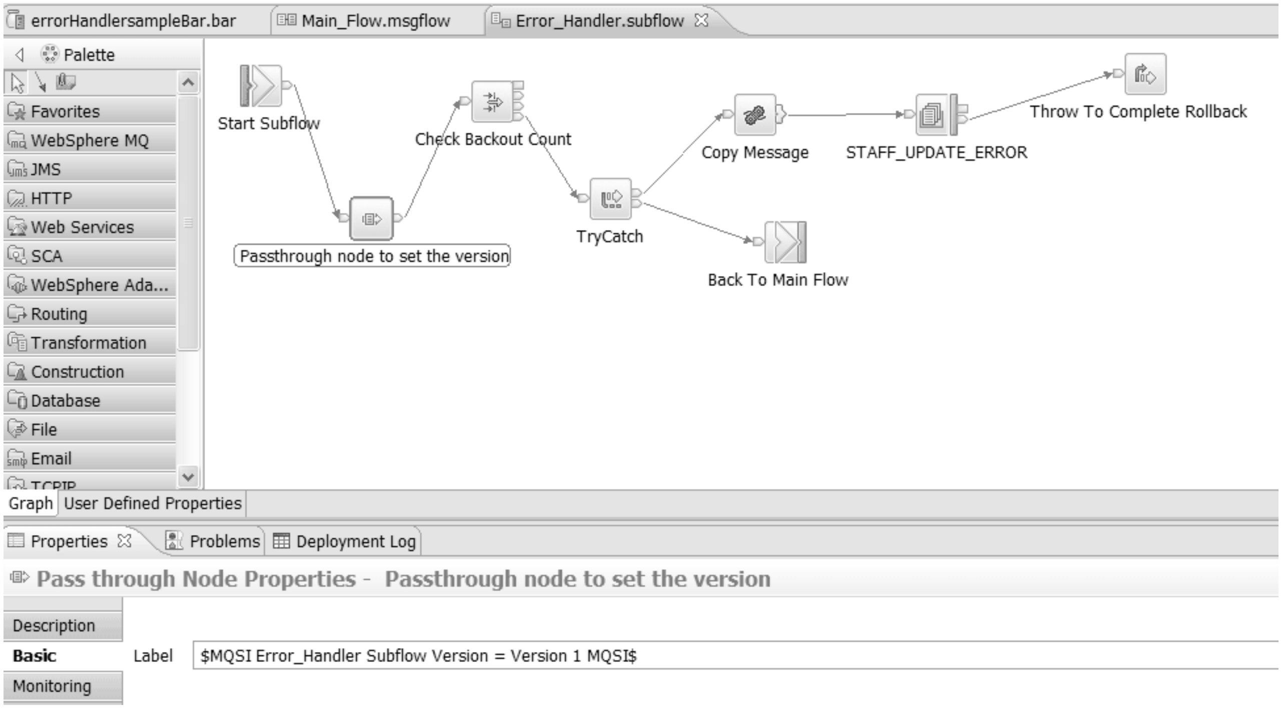
   In the **Broker Development** view, right-click the name of the subflow, and then select **Open with** > **Message Flow editor**.
2. In the **Error_Handler.subflow**, connect a Passthrough node.



3. In the Properties tab of the Passthrough node, use the **Label** field to set the subflow version. This must be in the format:

   `$MQSI` *name of your subflow* `Subflow Version = Version` *xxxx* `MQSI$`.

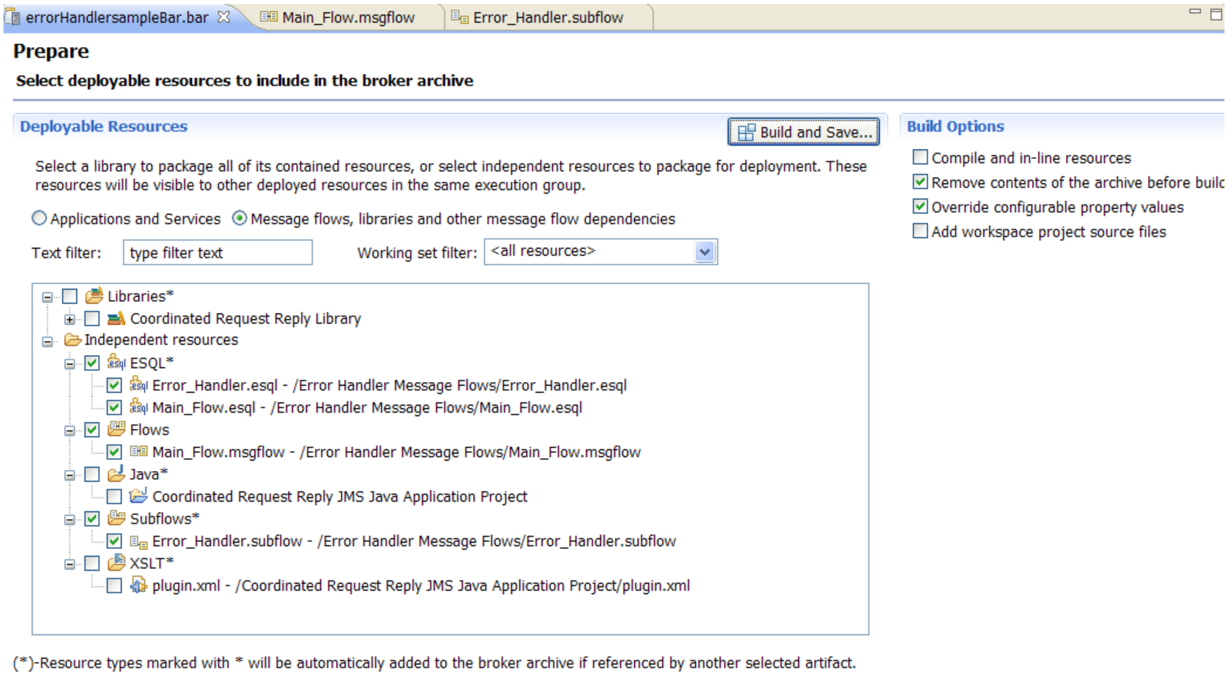
   where *name of your subflow* is the name of your subflow; and *xxxx* is your subflow version identifier.



Pass through Node Properties - Passthrough node to set the version

Label: $MQSI Error_Handler Subflow Version = Version 1 MQSI$

4. Create a BAR file that includes the **Main_flow.msgflow**.



Prepare

Select deployable resources to include in the broker archive

Deployable Resources Build and Save...

Select a library to package all of its contained resources, or select independent resources to package for deployment. These resources will be visible to other deployed resources in the same execution group.

○ Applications and Services ● Message flows, libraries and other message flow dependencies

Text filter: type filter text Working set filter: <all resources>

- □ Libraries*
  - □ Coordinated Request Reply Library
- Independent resources
  - ☑ ESQL*
    - ☑ Error_Handler.esql - /Error Handler Message Flows/Error_Handler.esql
    - ☑ Main_Flow.esql - /Error Handler Message Flows/Main_Flow.esql
  - ☑ Flows
    - ☑ Main_Flow.msgflow - /Error Handler Message Flows/Main_Flow.msgflow
  - □ Java*
    - □ Coordinated Request Reply JMS Java Application Project
  - ☑ Subflows*
    - ☑ Error_Handler.subflow - /Error Handler Message Flows/Error_Handler.subflow
  - □ XSLT*
    - □ plugin.xml - /Coordinated Request Reply JMS Java Application Project/plugin.xml

Build Options

- □ Compile and in-line resources
- ☑ Remove contents of the archive before builc
- ☑ Override configurable property values
- □ Add workspace project source files

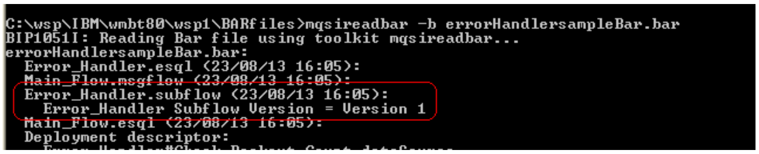
(*)-Resource types marked with * will be automatically added to the broker archive if referenced by another selected artifact.

5. Deploy the message flow into a broker runtime environment. For more information, see Deploying a broker archive file.
6. In the WebSphere Message Broker console, run the `mqsireadbar` command to check the version of the subflow.

   You run the command using the following syntax:

   `mqsireadbar -b` *`your BAR file name`*

   where *your BAR file name* is the name of the BAR file you have created to deploy the sample into the run time.



```
C:\wsp\IBM\wmbt80\wsp1\BARfiles>mqsireadbar -b errorHandlersampleBar.bar
BIP1051I: Reading Bar file using toolkit mqsireadbar...
errorHandlersampleBar.bar:
  Error_Handler.esql (23/08/13 16:05):
  Main_Flow.msgflow (23/08/13 16:05):
  Error_Handler.subflow (23/08/13 16:05):
    Error_Handler Subflow Version = Version 1
  Main_Flow.esql (23/08/13 16:05):
  Deployment descriptor:
```

What to do next

After you set the version of a subflow, create a BAR file. For more information, see "Creating a BAR file that includes a subflow created as a `.subflow` file" on page 23 or "Creating a BAR file that includes a subflow created as a `.msgflow` file" on page 52.

Versioning a subflow by using a keyword as a Passthrough node name

To define the version of a subflow, you can set the version of a subflow by adding a Passthrough node to the subflow with a keyword as the node name.

Before you begin

Set up the initial configuration. See "Creating the initial configuration for the Error Handler sample" on page 7.

About this task

When you set a subflow version by adding a Passthrough node to the subflow with a keyword as the node name, you can check the value at run time in the WebSphere Message Broker console and in WebSphere Message Broker Explorer.

Procedure

To define the version of a subflow, you must complete the following steps in the WebSphere Message Broker Toolkit:

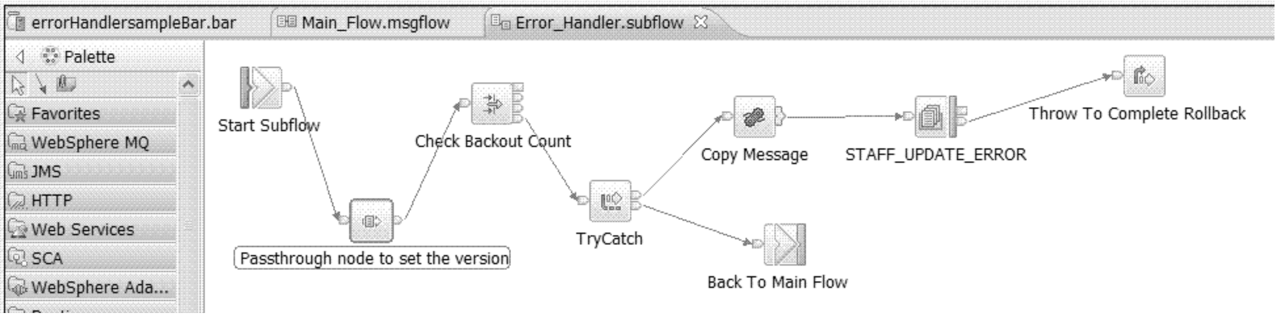
1. Open the subflow in the Message Flow editor.
2. Add a Passthrough node to a subflow.
3. Connect the Passthrough node to other nodes in the subflow or leave it unconnected.
4. Define a keyword. For more information, see "Defining a keyword to be used as the container of a subflow version" on page 14.
5. Set the keyword in the **Node name** property of the Passthrough node.

Example

For example, in the Error Handler sample provided with WebSphere Message Broker, you can add and check the version of the subflow as follows:

1. In the WebSphere Message Broker Toolkit, open the **Error_Handler.subflow** subflow in the Message Flow editor:

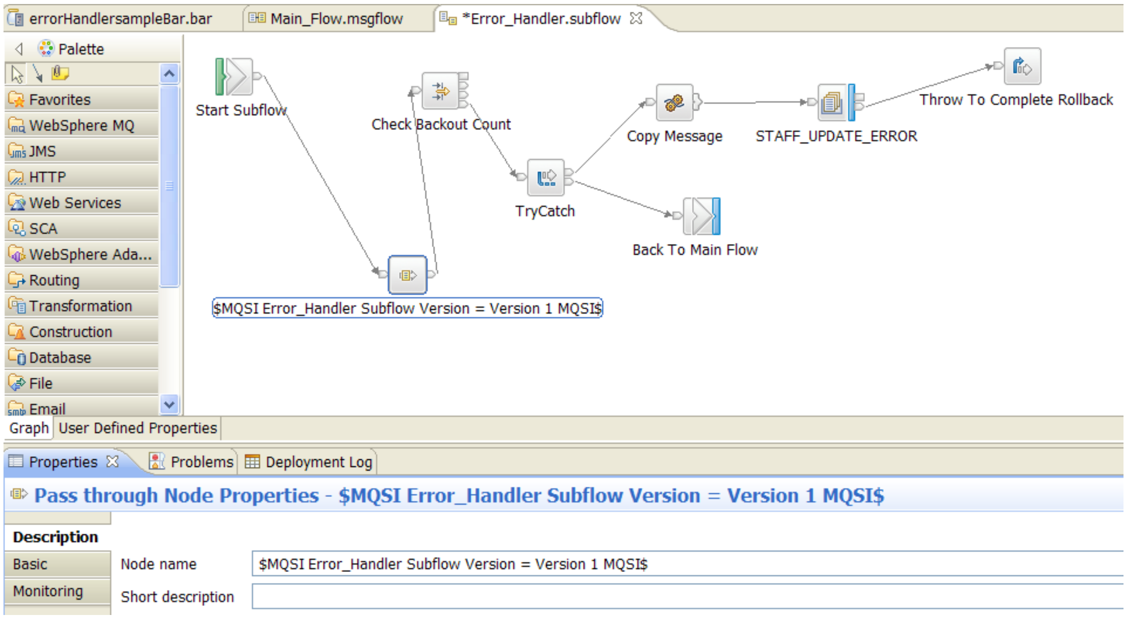
   In the **Broker Development** view, right-click the name of the subflow, and then select **Open with** > **Message Flow editor**.
2. In the **Error_Handler.subflow**, connect a Passthrough node.



3. In the Properties tab of the Passthrough node, use the **Node name** field to set the subflow version. This must be in the format:

   `$MQSI` *`name of your subflow`* `Subflow Version = Version` *`xxxx`* `MQSI$.`

   where *name of your subflow* is the name of your subflow; and *xxxx* is your subflow version identifier.

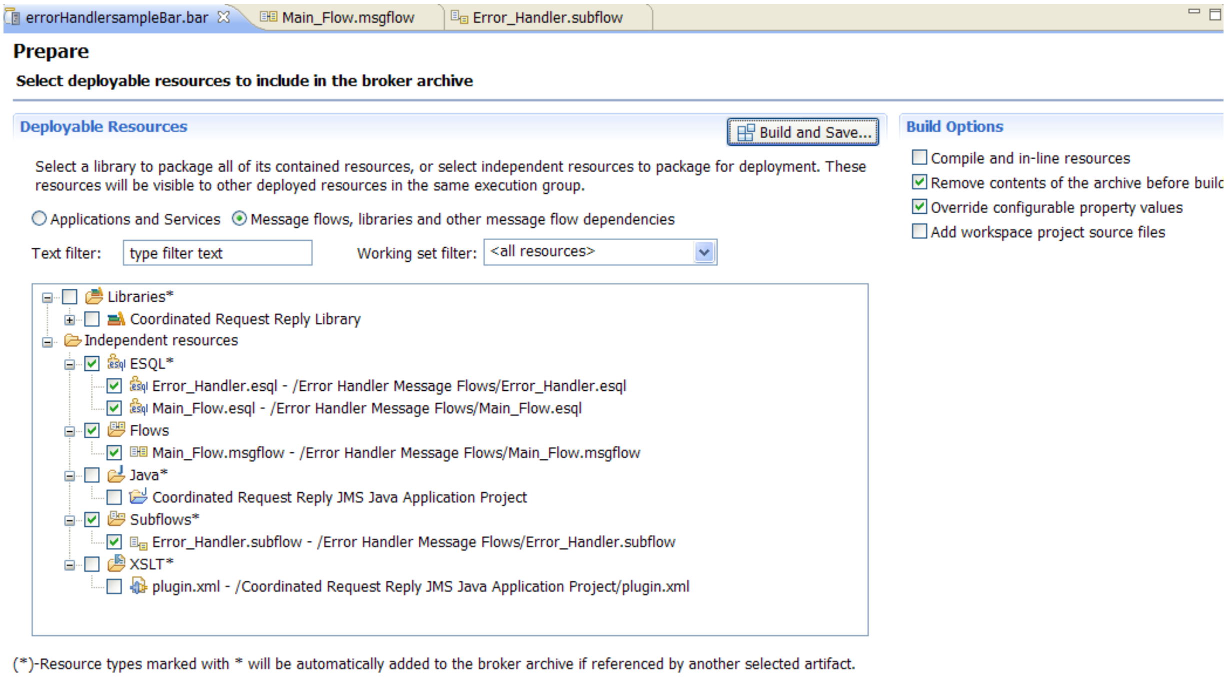


Pass through Node Properties - $MQSI Error_Handler Subflow Version = Version 1 MQSI$

Node name: $MQSI Error_Handler Subflow Version = Version 1 MQSI$

Short description:

4. Create a BAR file that includes the **Main_flow.msgflow**.



errorHandlersampleBar.bar ☒ | Main_Flow.msgflow | Error_Handler.subflow

Prepare

Select deployable resources to include in the broker archive

Deployable Resources [Build and Save...]

Select a library to package all of its contained resources, or select independent resources to package for deployment. These resources will be visible to other deployed resources in the same execution group.

○ Applications and Services ● Message flows, libraries and other message flow dependencies

Text filter: type filter text   Working set filter: <all resources>

- □ Libraries*
  - □ Coordinated Request Reply Library
- Independent resources
  - ☑ ESQL*
    - ☑ Error_Handler.esql - /Error Handler Message Flows/Error_Handler.esql
    - ☑ Main_Flow.esql - /Error Handler Message Flows/Main_Flow.esql
  - ☑ Flows
    - ☑ Main_Flow.msgflow - /Error Handler Message Flows/Main_Flow.msgflow
  - □ Java*
    - □ Coordinated Request Reply JMS Java Application Project
  - ☑ Subflows*
    - ☑ Error_Handler.subflow - /Error Handler Message Flows/Error_Handler.subflow
  - □ XSLT*
    - □ plugin.xml - /Coordinated Request Reply JMS Java Application Project/plugin.xml

(*)-Resource types marked with * will be automatically added to the broker archive if referenced by another selected artifact.

Build Options

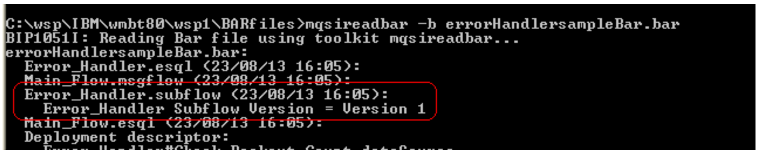
□ Compile and in-line resources
☑ Remove contents of the archive before builc
☑ Override configurable property values
□ Add workspace project source files

5. Deploy the message flow into a broker runtime environment. For more information, see Deploying a broker archive file.
6. In the broker console, run the `mqsireadbar` command to check the version of the subflow.

   You run the command using the following syntax:

   `mqsireadbar -b` *`your BAR file name`*

   where *your BAR file name* is the name of the BAR file you have created to deploy the sample into the run time.



```
C:\wsp\IBM\wmbt80\wsp1\BARfiles>mqsireadbar -b errorHandlersampleBar.bar
BIP1051I: Reading Bar file using toolkit mqsireadbar...
errorHandlersampleBar.bar:
  Error_Handler.esql (23/08/13 16:05):
  Main_Flow.msgflow (23/08/13 16:05):
  Error_Handler.subflow (23/08/13 16:05):
    Error_Handler Subflow Version = Version 1
  Main_Flow.esql (23/08/13 16:05):
  Deployment descriptor:
```

What to do next

After you set the version of a subflow, create a BAR file. For more information, see “Creating a BAR file that includes a subflow created as a `.subflow` file” on page 23 or “Creating a BAR file that includes a subflow created as a `.msgflow` file” on page 52.

Versioning a subflow by setting the Version property of the subflow

To define the version of a subflow, you can set the value of the subflow **Version** property.

Before you begin

Set up the initial configuration. See “Creating the initial configuration for the Error Handler sample” on page 7.

About this task

When you set the Version property of the subflow, you can check the value at run time in the WebSphere Message Broker Toolkit and in WebSphere Message Broker Explorer.

Procedure

You must complete the following steps in the WebSphere Message Broker Toolkit:

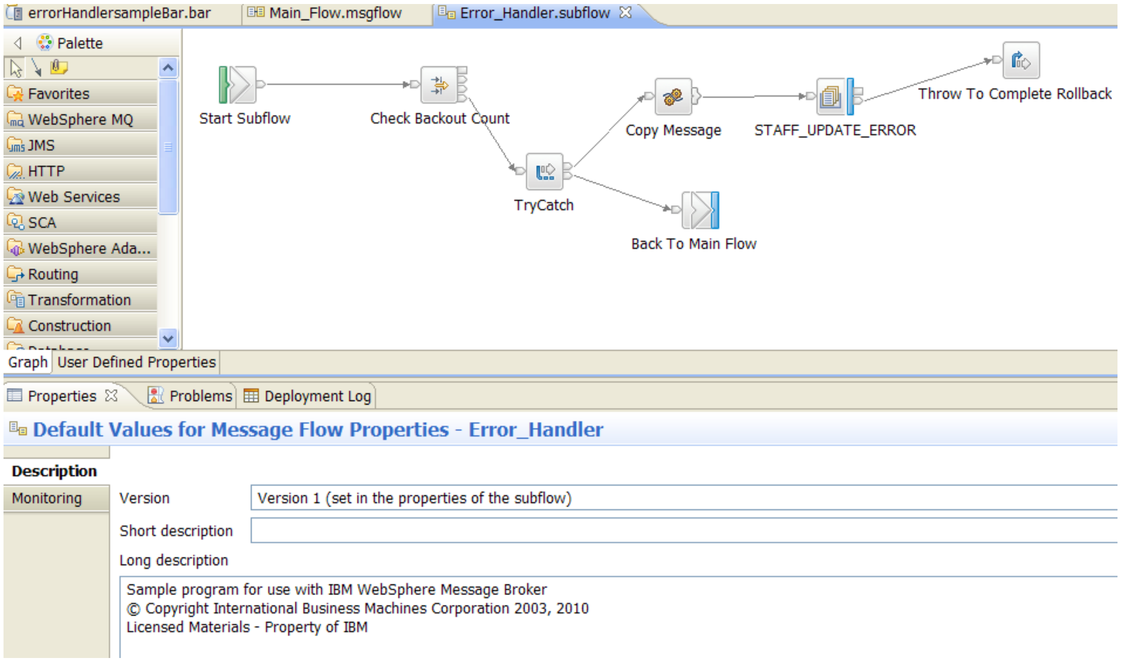
1. Open the subflow in the Message Flow editor.
2. Configure the **Version** property of the subflow.

Example

For example, in the Error Handler sample provided with WebSphere Message Broker, you can add and check the version of the subflow as follows:

1. In the WebSphere Message Broker Toolkit, open the **Error_Handler.subflow** subflow in the Message Flow editor:

   In the **Broker Development** view, right-click the name of the subflow, and then select **Open with** > **Message Flow editor**.
2. In the **Error_Handler.subflow**, configure the **Version** property of the subflow.



Default Values for Message Flow Properties - Error_Handler

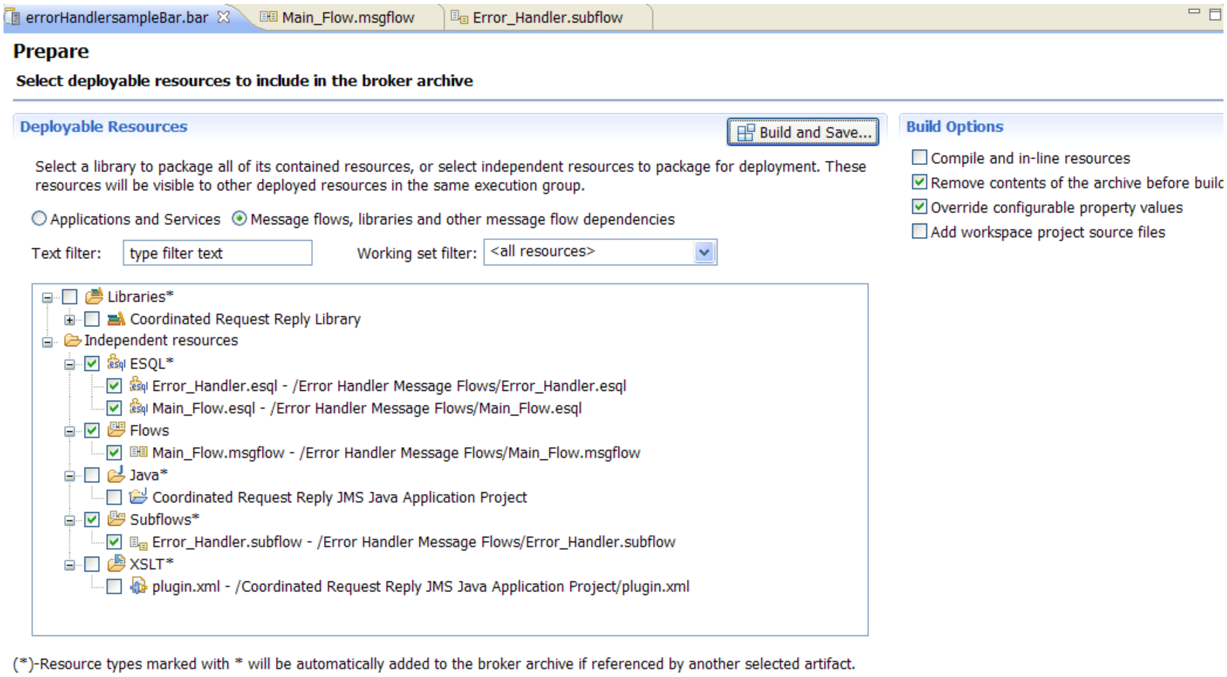
Version: Version 1 (set in the properties of the subflow)

Short description:

Long description:

Sample program for use with IBM WebSphere Message Broker
© Copyright International Business Machines Corporation 2003, 2010
Licensed Materials - Property of IBM

3. Create a BAR file that includes the **Main_flow.msgflow**.



Prepare

Select deployable resources to include in the broker archive

Deployable Resources — Build and Save...

Select a library to package all of its contained resources, or select independent resources to package for deployment. These resources will be visible to other deployed resources in the same execution group.

○ Applications and Services ● Message flows, libraries and other message flow dependencies

Text filter: type filter text Working set filter: <all resources>

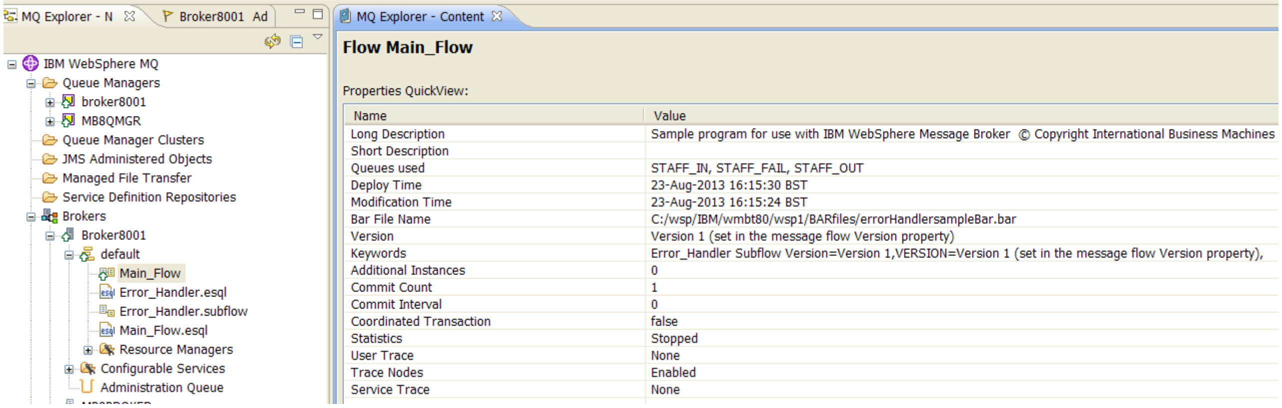
- □ Libraries*
  - □ Coordinated Request Reply Library
- Independent resources
  - ☑ ESQL*
    - ☑ Error_Handler.esql - /Error Handler Message Flows/Error_Handler.esql
    - ☑ Main_Flow.esql - /Error Handler Message Flows/Main_Flow.esql
  - ☑ Flows
    - ☑ Main_Flow.msgflow - /Error Handler Message Flows/Main_Flow.msgflow
  - □ Java*
    - □ Coordinated Request Reply JMS Java Application Project
  - ☑ Subflows*
    - ☑ Error_Handler.subflow - /Error Handler Message Flows/Error_Handler.subflow
  - □ XSLT*
    - □ plugin.xml - /Coordinated Request Reply JMS Java Application Project/plugin.xml

Build Options

□ Compile and in-line resources
☑ Remove contents of the archive before builc
☑ Override configurable property values
□ Add workspace project source files

(*)-Resource types marked with * will be automatically added to the broker archive if referenced by another selected artifact.

4. Deploy the message flow into a broker runtime environment. For more information, see Deploying a broker archive file.
5. In the WebSphere Message Broker Explorer, double-click the message flow that includes the subflow. In our example, double-click **Main_Flow**. The Properties QuickView panel opens. In this panel, you can check the version of the subflow under **Keywords**.



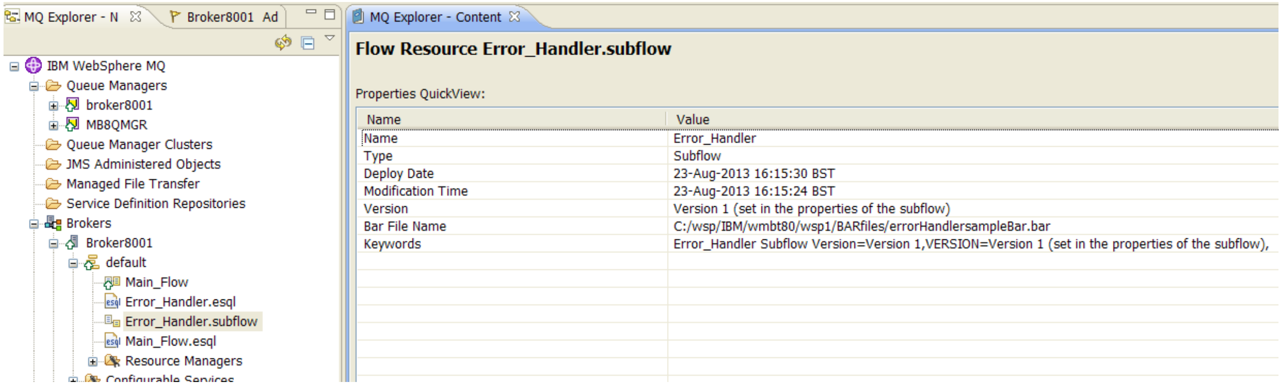
- IBM WebSphere MQ
  - Queue Managers
    - broker8001
    - MB8QMGR
  - Queue Manager Clusters
  - JMS Administered Objects
  - Managed File Transfer
  - Service Definition Repositories
  - Brokers
    - Broker8001
      - default
        - Main_Flow
        - Error_Handler.esql
        - Error_Handler.subflow
        - Main_Flow.esql
        - Resource Managers
      - Configurable Services
      - Administration Queue

Flow Main_Flow

Properties QuickView:

Name	Value
Long Description	Sample program for use with IBM WebSphere Message Broker © Copyright International Business Machines
Short Description	
Queues used	STAFF_IN, STAFF_FAIL, STAFF_OUT
Deploy Time	23-Aug-2013 16:15:30 BST
Modification Time	23-Aug-2013 16:15:24 BST
Bar File Name	C:/wsp/IBM/wmbt80/wsp1/BARfiles/errorHandlersampleBar.bar
Version	Version 1 (set in the message flow Version property)
Keywords	Error_Handler Subflow Version=Version 1,VERSION=Version 1 (set in the message flow Version property),
Additional Instances	0
Commit Count	1
Commit Interval	0
Coordinated Transaction	false
Statistics	Stopped
User Trace	None
Trace Nodes	Enabled
Service Trace	None

6. In the WebSphere Message Broker Explorer, double-click the subflow. In our example, double-click **Error_Handler.subflow**. The Properties QuickView panel opens. In this panel, you can check the version of the subflow under **Keywords**.



Flow Resource Error_Handler.subflow

Properties QuickView:

Name	Value
Name	Error_Handler
Type	Subflow
Deploy Date	23-Aug-2013 16:15:30 BST
Modification Time	23-Aug-2013 16:15:24 BST
Version	Version 1 (set in the properties of the subflow)
Bar File Name	C:/wsp/IBM/wmbt80/wsp1/BARfiles/errorHandlersampleBar.bar
Keywords	Error_Handler Subflow Version=Version 1,VERSION=Version 1 (set in the properties of the subflow),

What to do next

After you set the version of a subflow, create a BAR file. For more information, see "Creating a BAR file that includes a subflow created as a `.subflow` file" on page 23 or "Creating a BAR file that includes a subflow created as a `.msgflow` file" on page 52.

Checking the Version of a subflow at run time

You can check the version of a subflow at run time by using the WebSphere Message Broker console or the WebSphere Message Broker Explorer.

Before you begin

Set up the initial configuration. See "Creating the initial configuration for the Error Handler sample" on page 7.

About this task

Depending on how you have defined the version of a subflow during the development phase, you have different options to check the version of a subflow at run time:

- If you defined a keyword in the Label property of a Passthrough node, you can use the WebSphere Message Broker console or the WebSphere Message Broker Explorer.
- If you added a Passthrough node to the subflow with a keyword as the node name, you can use the WebSphere Message Broker console or the WebSphere Message Broker Explorer.
- If you set the Version property of the subflow, you can use WebSphere Message Broker Explorer to see the subflow version.

Table 3. Summary of WebSphere Message Broker components where you can check the version of a subflow at run time

How you defined the version during development	WebSphere Message Broker console	WebSphere Message Broker Explorer	WebSphere Message Broker Toolkit
Using a keyword	x	x	It is not possible to check the version at run time.
Setting the Version property of a subflow created as a `.subflow` file	It is not possible to check the version at run time.	x	It is not possible to check the version at run time.

Table 3. Summary of WebSphere Message Broker components where you can check the version of a subflow at run time (continued)

How you defined the version during development	WebSphere Message Broker console	WebSphere Message Broker Explorer	WebSphere Message Broker Toolkit
Setting the Version property of the subflow created as a `.msgflow` file	It is not possible to check the version at run time.	x	x

Procedure

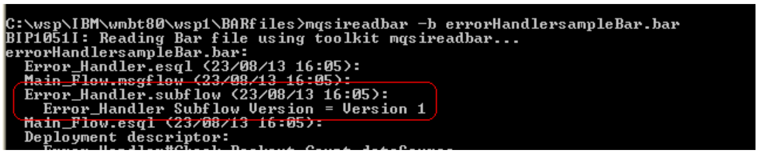
To check the version of a subflow at run time, choose one of the following options:

1. In the WebSphere Message Broker console, run the `mqsireadbar` command. This command reads the properties stored in the broker archive (BAR) file.

   You run the command using the following syntax:

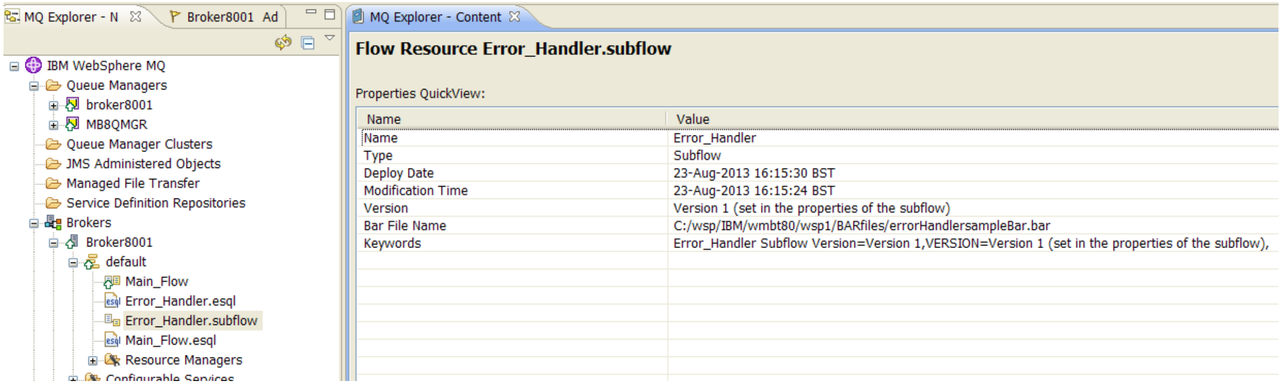
   `mqsireadbar -b` *`your BAR file name`*

   where *your BAR file name* is the name of the BAR file you have created to deploy into the run time.



```
C:\wsp\IBM\wmbt80\wsp1\BARfiles>mqsireadbar -b errorHandlersampleBar.bar
BIP1051I: Reading Bar file using toolkit mqsireadbar...
errorHandlersampleBar.bar:
  Error_Handler.esql (23/08/13 16:05):
  Main_Flow.msgflow (23/08/13 16:05):
  Error_Handler.subflow (23/08/13 16:05):
    Error_Handler Subflow Version = Version 1
  Main_Flow.esql (23/08/13 16:05):
  Deployment descriptor:
```

2. In the WebSphere Message Broker Explorer, double-click a subflow. The MQ Explorer - Content tab opens where you can see the Properties QuickView.



Flow Resource Error_Handler.subflow

Properties QuickView:

Name	Value
Name	Error_Handler
Type	Subflow
Deploy Date	23-Aug-2013 16:15:30 BST
Modification Time	23-Aug-2013 16:15:24 BST
Version	Version 1 (set in the properties of the subflow)
Bar File Name	C:/wsp/IBM/wmbt80/wsp1/BARfiles/errorHandlersampleBar.bar
Keywords	Error_Handler Subflow Version=Version 1,VERSION=Version 1 (set in the properties of the subflow),

What to do next

Return to “Updating a subflow” on page 31.

Creating a BAR file that includes a subflow created as a `.subflow` file

When deploying a subflow, you need to understand how WebSphere Message Broker packages subflows into BAR files. Building a BAR file for an application or service with the build option **Compile and in-line resources** not selected is the recommended approach to packaging integration solutions in WebSphere Message Broker Version 8 and later versions.

Before you begin

Set up the initial configuration. See “Creating the initial configuration for the Coordinated Request Reply sample” on page 8.

Procedure

To create a BAR file in WebSphere Message Broker Version 8.0.0.1, complete the following steps:

1. Create a broker archive (BAR file) for each configuration that you want to deploy to the run time. For more information, see Creating a broker archive.
2. Add files to the BAR file. For more information, see Adding files to a broker archive.
3. Configure the BAR file and its resources. For more information, see Configuring the start mode of flows and applications at development time.
4. Choose whether you want to deploy the resources in the BAR file in their compile mode or as objects. The recommended approach is to not select the **Compile and in-line resources** option. For more information, see “BAR builder behavior for subflows created as `.subflow` files.”
5. Save and build the BAR file in the BAR builder.

What to do next

Deploy a subflow created as a `.subflow` file. For more information, see “Deploying a subflow created as a `.subflow` file” on page 28.

BAR builder behavior for subflows created as `.subflow` files

Learn how a BAR file is created based on your message flow design, the subflows that you created as `.msgflow` files, and the build options that you choose. Notice the differences between WebSphere Message Broker Version 8.0.0.0 and WebSphere Message Broker Version 8.0.0.1.

Packaging considerations in WebSphere Message Broker Version 8.0.0.0

When you build a BAR file in WebSphere Message Broker Toolkit, you must consider the following BAR builder behavior:

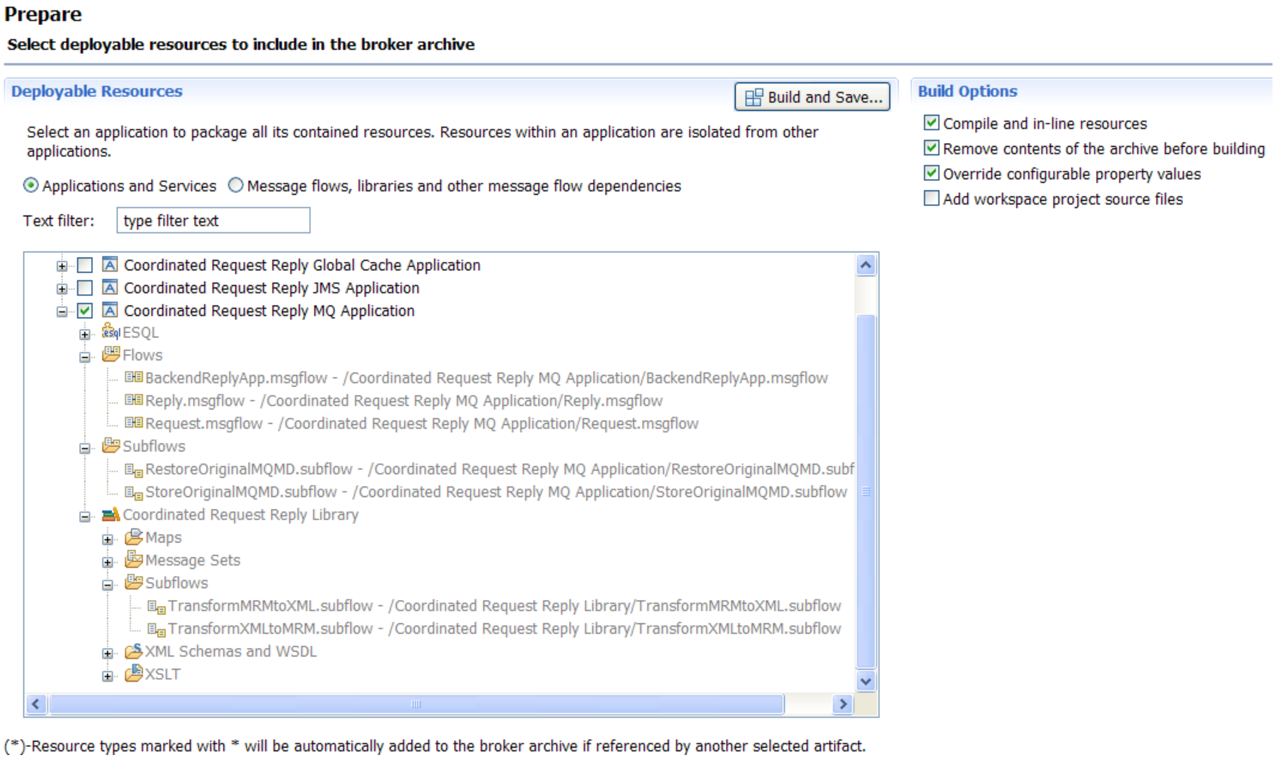
- When you add a message flow to a BAR file, the BAR builder will automatically include any required subflows.
- Message flows are included in the BAR file as compiled objects, that is, as `.cmf` files.

  Note: The BAR file creation option **Compile and in-line resources** is not available in WebSphere Message Broker Version 8.0.0.0, which means that a message flow is always added to a BAR file in its compiled form.
- If a subflow created as a `.subflow` file is included in a message flow, then the BAR builder creates a compiled message flow for the message flow and includes the subflow source file for the subflow created as a `.subflow` file.

Packaging considerations in WebSphere Message Broker Version 8.0.0.1 and later versions

When you build a BAR file in WebSphere Message Broker Toolkit, you must consider the following BAR builder behavior:

- When you add an application or a service to a BAR file, the BAR builder will automatically include any resources defined within the application project, or the service project, including all the required subflows.



Prepare

Select deployable resources to include in the broker archive

Deployable Resources [Build and Save...]

Select an application to package all its contained resources. Resources within an application are isolated from other applications.

● Applications and Services ○ Message flows, libraries and other message flow dependencies

Text filter: type filter text

- □ Coordinated Request Reply Global Cache Application
- □ Coordinated Request Reply JMS Application
- ☑ Coordinated Request Reply MQ Application
  - ESQL
  - Flows
    - BackendReplyApp.msgflow - /Coordinated Request Reply MQ Application/BackendReplyApp.msgflow
    - Reply.msgflow - /Coordinated Request Reply MQ Application/Reply.msgflow
    - Request.msgflow - /Coordinated Request Reply MQ Application/Request.msgflow
  - Subflows
    - RestoreOriginalMQMD.subflow - /Coordinated Request Reply MQ Application/RestoreOriginalMQMD.subf
    - StoreOriginalMQMD.subflow - /Coordinated Request Reply MQ Application/StoreOriginalMQMD.subflow
  - Coordinated Request Reply Library
    - Maps
    - Message Sets
    - Subflows
      - TransformMRMtoXML.subflow - /Coordinated Request Reply Library/TransformMRMtoXML.subflow
      - TransformXMLtoMRM.subflow - /Coordinated Request Reply Library/TransformXMLtoMRM.subflow
    - XML Schemas and WSDL
    - XSLT

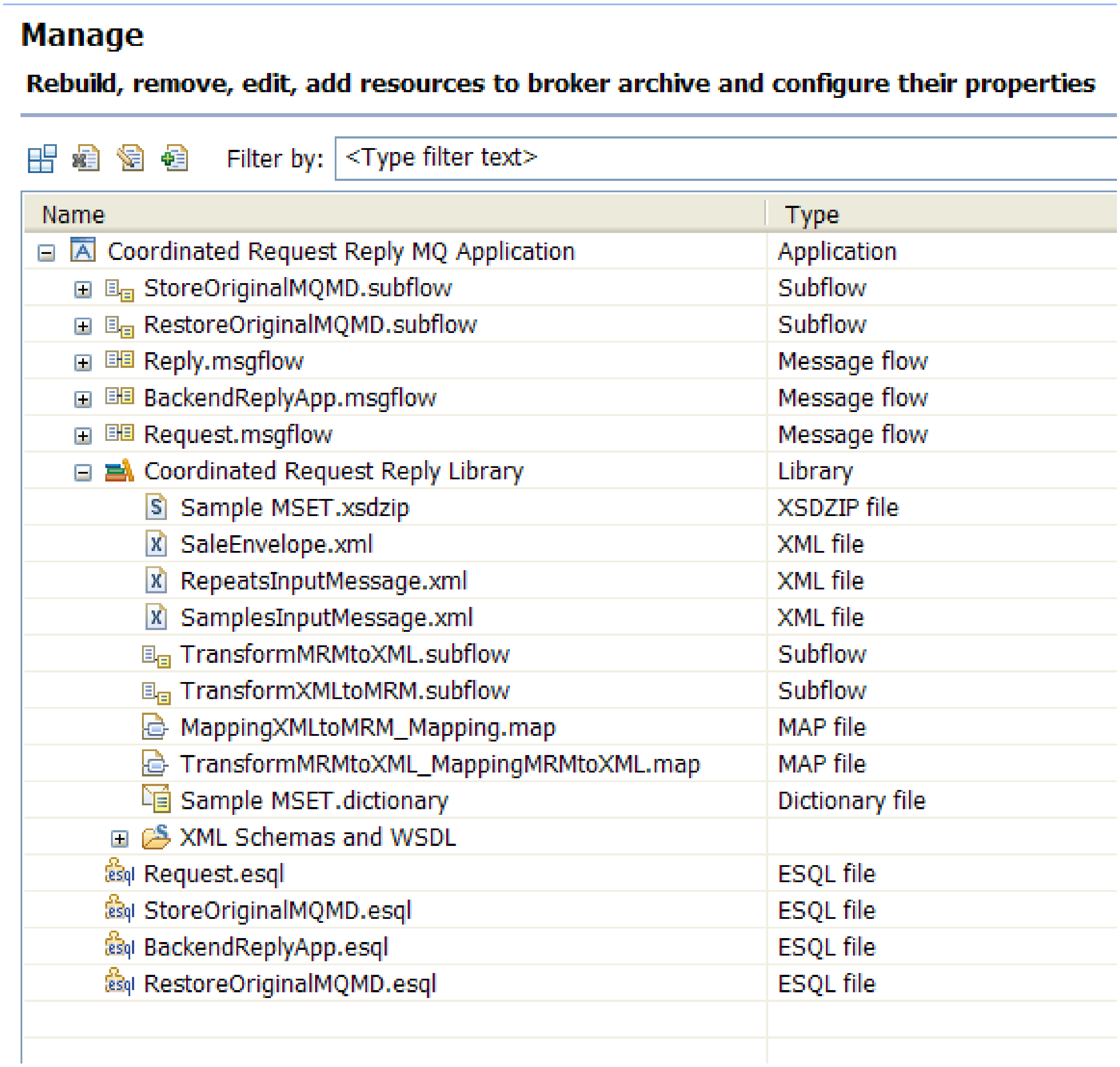
Build Options

☑ Compile and in-line resources
☑ Remove contents of the archive before building
☑ Override configurable property values
□ Add workspace project source files

(*)-Resource types marked with * will be automatically added to the broker archive if referenced by another selected artifact.

- You can leave the BAR file creation option **Compile and in-line resources** option not selected, so that a message flow is added to a BAR file as a `.msgflow` source file. Subflows created as `.msgflow` files cannot be embedded in the message flow. You can only add subflows created as `.subflow` files to the BAR file.

  Note: This is the recommended approach to creating a BAR file that includes message flows and subflows created as `.subflow` files.



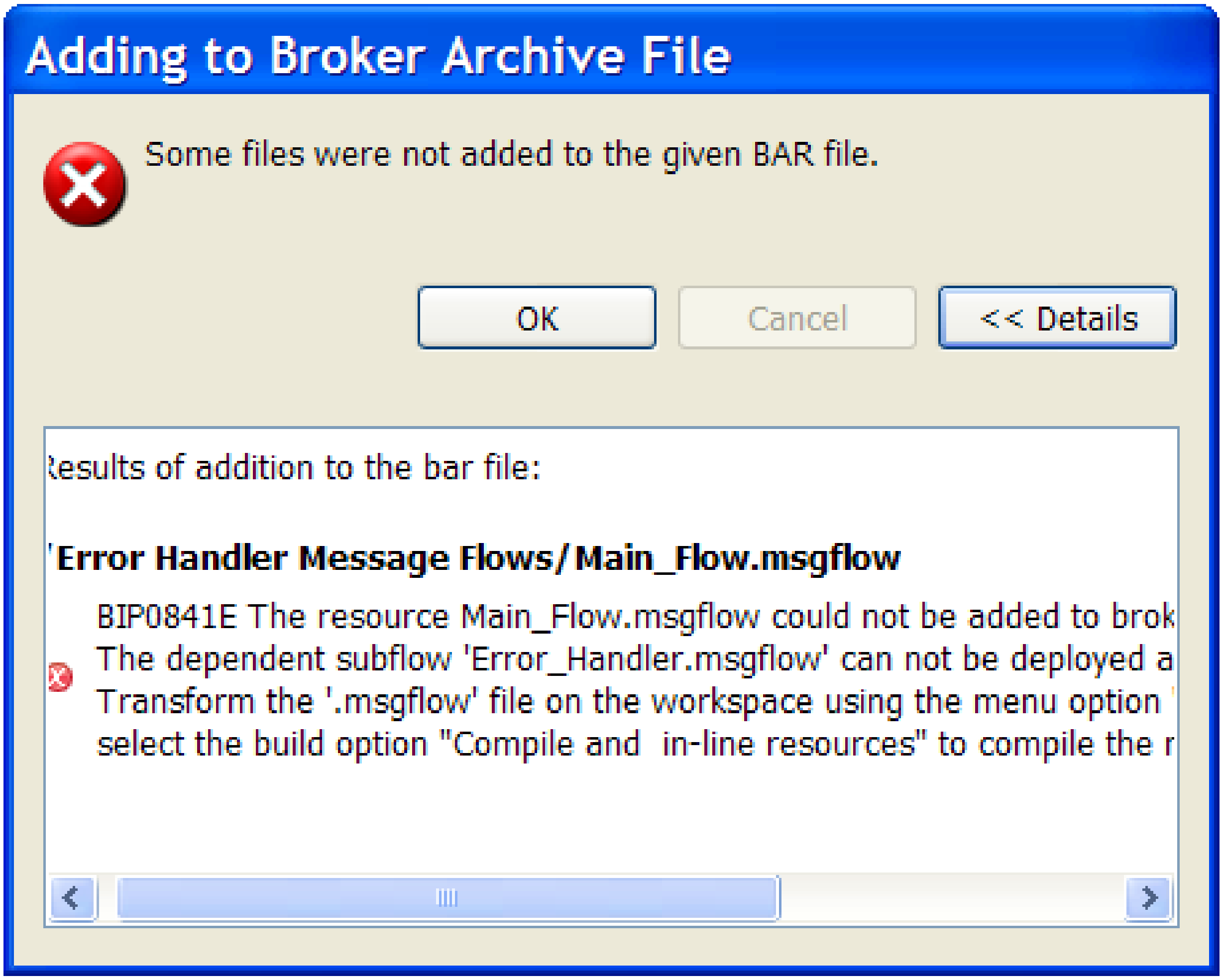
Manage

Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type
⊟ Coordinated Request Reply MQ Application	Application
⊞ StoreOriginalMQMD.subflow	Subflow
⊞ RestoreOriginalMQMD.subflow	Subflow
⊞ Reply.msgflow	Message flow
⊞ BackendReplyApp.msgflow	Message flow
⊞ Request.msgflow	Message flow
⊟ Coordinated Request Reply Library	Library
Sample MSET.xsdzip	XSDZIP file
SaleEnvelope.xml	XML file
RepeatsInputMessage.xml	XML file
SamplesInputMessage.xml	XML file
TransformMRMtoXML.subflow	Subflow
TransformXMLtoMRM.subflow	Subflow
MappingXMLtoMRM_Mapping.map	MAP file
TransformMRMtoXML_MappingMRMtoXML.map	MAP file
Sample MSET.dictionary	Dictionary file
⊞ XML Schemas and WSDL	
Request.esql	ESQL file
StoreOriginalMQMD.esql	ESQL file
BackendReplyApp.esql	ESQL file
RestoreOriginalMQMD.esql	ESQL file

Note: If you try to create a BAR file with the option **Compile and in-line resources** not selected, where the message flow has embedded subflows created as `.msgflow` files, you get the following error:



Adding to Broker Archive File

Some files were not added to the given BAR file.

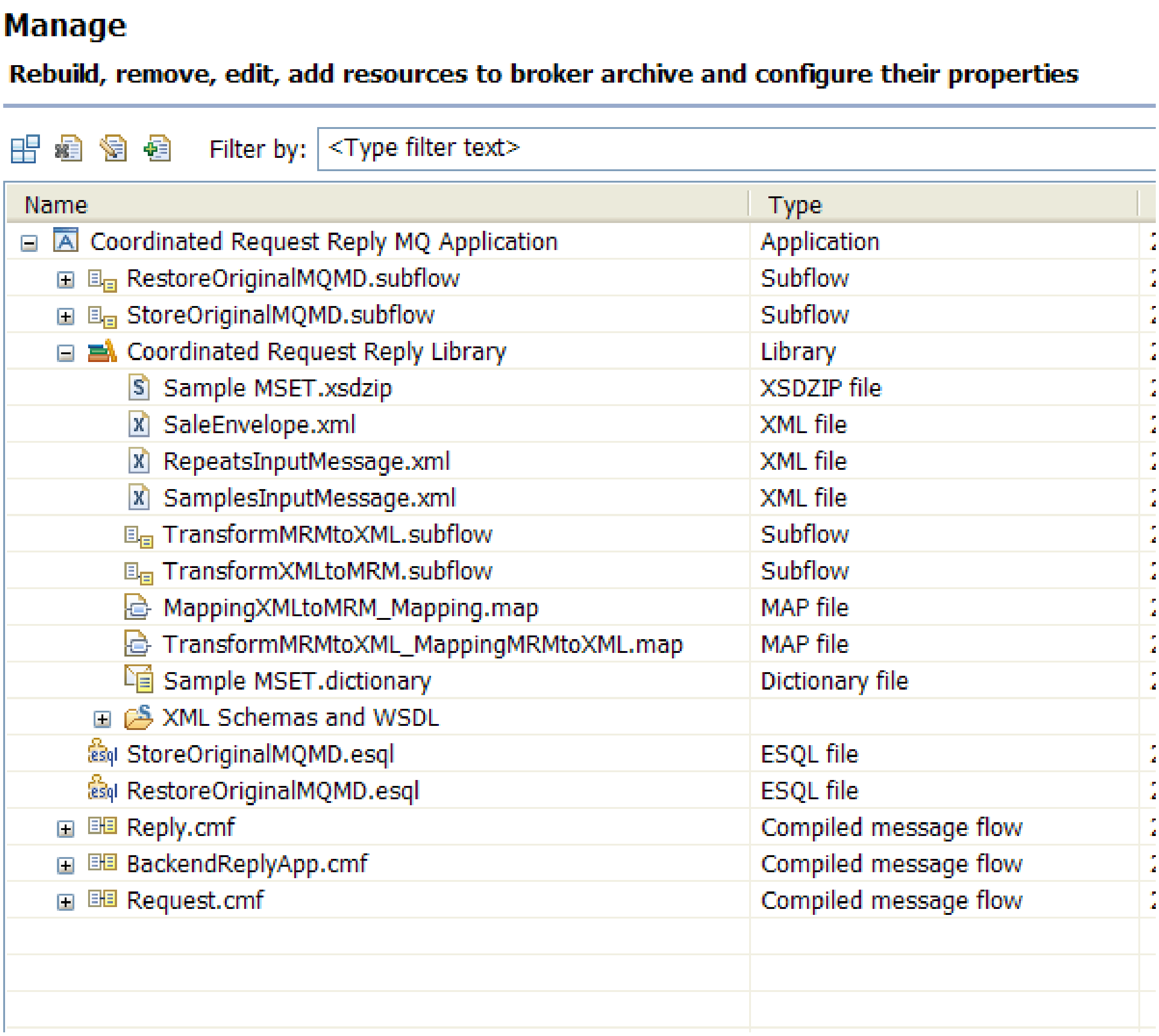
Results of addition to the bar file:

'Error Handler Message Flows/Main_Flow.msgflow

BIP0841E The resource Main_Flow.msgflow could not be added to brok
The dependent subflow 'Error_Handler.msgflow' can not be deployed a
Transform the '.msgflow' file on the workspace using the menu option '
select the build option "Compile and in-line resources" to compile the r

- You can choose to select the BAR file creation option **Compile and in-line resources** so that a message flow is added to a BAR file in its compiled form. Message flows are included in the BAR file as compiled objects, that is, `.cmf` files. Subflows are included as `.subflow` source files.

  Note: This is not a recommended approach. It is available for compatibility with previous releases of WebSphere Message Broker.



Manage

Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type
Coordinated Request Reply MQ Application	Application
RestoreOriginalMQMD.subflow	Subflow
StoreOriginalMQMD.subflow	Subflow
Coordinated Request Reply Library	Library
Sample MSET.xsdzip	XSDZIP file
SaleEnvelope.xml	XML file
RepeatsInputMessage.xml	XML file
SamplesInputMessage.xml	XML file
TransformMRMtoXML.subflow	Subflow
TransformXMLtoMRM.subflow	Subflow
MappingXMLtoMRM_Mapping.map	MAP file
TransformMRMtoXML_MappingMRMtoXML.map	MAP file
Sample MSET.dictionary	Dictionary file
XML Schemas and WSDL	
StoreOriginalMQMD.esql	ESQL file
RestoreOriginalMQMD.esql	ESQL file
Reply.cmf	Compiled message flow
BackendReplyApp.cmf	Compiled message flow
Request.cmf	Compiled message flow

Related tasks:

"Creating a BAR file that includes a subflow created as a `.subflow` file" on page 23
When deploying a subflow, you need to understand how WebSphere Message Broker packages subflows into BAR files. Building a BAR file for an application or service with the build option **Compile and in-line resources** not selected is the recommended approach to packaging integration solutions in WebSphere Message Broker Version 8 and later versions.

Deploying a subflow created as a `.subflow` file

There are multiple configurations that you can create to deploy a subflow created as a `.subflow` file into the broker run time. Configurations where the BAR file for an application or service is created with the build option **Compile and in-line resources** not selected are the recommended ones.

Procedure

To deploy a subflow created as a `.subflow` file, complete the following steps:

1. Check the conditions that apply when you deploy a subflow. For more information, see “Conditions that apply when you deploy a subflow created as a `.subflow` file.”
2. Verify that you have a valid deployment configuration. For more information, see “Deployment configurations for subflows created as `.subflow` files.”
3. Deploy the BAR file into the WebSphere Message Broker run time. For more information, see Deploying a broker archive file.

What to do next

After you have deployed a version of a subflow to the run time, you might want update the subflow. For more information, see “Updating a subflow” on page 31.

Conditions that apply when you deploy a subflow created as a `.subflow` file

When you deploy a subflow, you must deploy the application, the service, or the library where the subflow is contained.

You deploy a subflow to an execution group by sending a broker archive (BAR) file to an execution group in a broker, which unpacks and stores the contents ready for when your message flows are started.

You can deploy subflows created as `.subflow` files in any of the following ways:

- Deploy a subflow as part of an application.
- Deploy a subflow as part of a service.
- Deploy a subflow as part of a library.

The following conditions apply when you deploy a subflow created as a `.subflow` file as part of an integration solution:

- If you deploy a subflow that is contained in an application, you must deploy the application to an execution group, which results in a complete deployment of the application.
- If you deploy a subflow that is contained in a service, you must deploy the service to an execution group, which results in a complete deployment of the service.
- If you deploy a subflow that is contained in a library, you must deploy the library directly to an execution group or deploy the library as part of the application or service that references this library

Related tasks:

“Deploying a subflow created as a `.subflow` file” on page 28
There are multiple configurations that you can create to deploy a subflow created as a `.subflow` file into the broker run time. Configurations where the BAR file for an application or service is created with the build option **Compile and in-line resources** not selected are the recommended ones.

Deployment configurations for subflows created as `.subflow` files

You have different deployment configurations, that is combinations of resources that are supported in Websphere Message Broker run time, when you deploy applications and services into the broker runtime environment. Deployment configurations depends on the configuration choices you make when you build a BAR file and how you have designed your message flow and subflows.

In WebSphere Message Broker Version 8, you can deploy subflows created as a `.subflow` file as part of your application or your service.

Deployment considerations in WebSphere Message Broker Version 8.0.0.0

The following table summarizes the different combinations of resources that are supported in Websphere Message Broker run time when you deploy applications and services into the broker runtime environment:

Table 4. Summary of supported deployment configurations for applications and services in WebSphere Message Broker Version 8.0.0.0

Configurations	**1**	**2**	**3**	**4**	**5**
Message flow (.msgflow)	x	x	x		
Subflow (.subflow) with generic input node		x		x	
Subflow (.subflow) with an Input node such as MQInput node			x		x
Bar file resources	`.cmf` file	`.cmf` file and `.subflow` source file	`.cmf` file and `.subflow` source file	`.subflow` source file	`.subflow` source file
Configuration	valid	valid	valid	valid	valid

Deployment considerations in WebSphere Message Broker Version 8.0.0.1 and later versions

The following table summarizes the different combinations of resources that are supported in Websphere Message Broker run time when you deploy applications and services into the broker runtime environment and the **Compile and in-line resources** option to build a BAR file is not selected:

Table 5. Summary of supported deployment configurations for applications and services in WebSphere Message Broker Version 8.0.0.1 and later versions where the **Compile and in-line resources** *option to build a BAR file is not selected*

Configurations with the option Compile and in-line resources not selected (These are recommended configurations for WebSphere Message Broker Version 8.0.0.1 and later versions)	**1**	**2**	**3**	**4**	**5**
Message flow (.msgflow)	x	x	x		
Subflow (.subflow) with generic input node		x		x	
Subflow (.subflow) with an Input node such as MQInput node			x		x
Bar file resources	1 `.msgflow` file	1 `.msgflow` file and 1 `.subflow` source file	1 `.msgflow` file and 1 `.subflow` source file	1 `.subflow` source file	1 `.subflow` source file
Configuration	valid	valid	valid	valid	valid

The following table summarizes the different combinations of resources that are supported in Websphere Message Broker run time when you deploy applications and services into the broker runtime environment and the **Compile and in-line resources** option to build a BAR file is selected:

Table 6. Summary of supported deployment configurations for applications and services in WebSphere Message Broker Version 8.0.0.1 and later versions where the **Compile and in-line resources** *option to build a BAR file is selected*

Configurations with the option Compile and in-line resources selected (These configurations are available for compatibility with previous releases of WebSphere Message Broker.)	**1**	**2**	**3**	**4**	**5**
Message flow (.msgflow)	x	x	x		
Subflow (.subflow) with generic input node		x		x	
Subflow (.subflow) with an Input node such as MQInput node			x		x
Bar file resources	1 `.cmf` file	1 `.cmf` file and 1 `.subflow` source file	1 `.cmf` file and 1 `.subflow` source file	1 `.subflow` source file	1 `.subflow` source file
Configuration	valid	valid, but it is not recommended	valid, but it is not recommended	valid	valid

Note: Although you can define a message flow that includes subflows created as a `.subflow` file and subflows created as a `.msgflow` file, then create a BAR file that includes both types of subflows, and then deploy the BAR file objects into the runtime environment, this configuration is not recommended.

Related tasks:

"Deploying a subflow created as a `.subflow` file" on page 28
There are multiple configurations that you can create to deploy a subflow created as a `.subflow` file into the broker run time. Configurations where the BAR file for an application or service is created with the build option **Compile and in-line resources** not selected are the recommended ones.

Updating a subflow

When updating a subflow, you need to check the subflow version, and understand how you can deploy a new version of the subflow to the WebSphere Message Broker run time. For example, if a subflow is deployed as part of an application or a service, then you need to update your applications or services to include the new subflow version, and then redeploy the application or service.

Procedure

To update a subflow, complete the following steps:

1. Check the version of a subflow. For more information, see "Checking the Version of a subflow at run time" on page 22.
2. Update the value of the subflow version.

   You update the version of a subflow manually in the WebSphere Message Broker Toolkit. You can modify the **Label** property of the Passthrough node, the **Name** property of the Passthrough node, or the **Version** property of a subflow. For more information on how to define the version of a subflow, see "Versioning a subflow" on page 13.
3. Update the subflow implementation to include the new requirements, for example, add new nodes or modify the logic of a Compute node.

4. Update the BAR file and refresh its contents. For more information, see Refreshing the contents of a broker archive.
5. Deploy the BAR file into the WebSphere Message Broker run time. For more information, see Deploying a broker archive file.

   For subflows created as a `.subflow` file, you must consider the following behaviour when deploying a new version of a subflow:

   - If the subflow is deployed as part of an application, a service, or a library, then you need to update your applications, services, and libraries to include the new subflow version, and redeploy them.

     Note: If the application, service, or library has been deployed before, redeploying it removes the currently deployed application, service, or library and its contents, and replaces it with the deployed application, service, or library.
   - If the subflow is deployed separately from any of the message flows that use this subflow and you deploy a new version of the subflow, then all the message flows are updated automatically.

   For subflows created as a `.msgflow` file, you must consider the following behaviour when deploying a new version of a subflow:

   - You need to update your applications, services, and independent resources that use the subflow to include the new subflow version, and redeploy them.

     Note: Legacy subflows can only be updated at run time when the parent message flow is recompiled and deployed.

What to do next

Configure a message flow that includes subflows created as `.subflow` files. For more information, see "Configuring a message flow that includes subflows."

Configuring a message flow that includes subflows

Depending on your subflow design, that is, the number of Input nodes that you have modeled, and how you configure the overall message flow BAR file deployment properties and the Input node deployment properties, WebSphere Message Broker handles threads at run time differently. This can impact your solution performance and your message parallel processing behavior at run time.

About this task

When you design a message flow or a subflow, you connect message flow nodes and subflows to define the integration solution logic. In a message flow, you use one or more Input nodes, such as the MQinput node, to model how a message is received by the message flow. If the first node in a message flow is a subflow, the subflow must contain one or more Input nodes. You can also have subflows in the middle of a message flow that include Input nodes.

When you design a subflow that is included in a message flow and you configure the BAR file that contains your integration solution, consider the following questions:

1. How many Input nodes, such as MQInput node, your message flow has? How many Input nodes your subflows have?

   Note: The number of Input nodes determines the default number of threads allocated by WebSphere Message Broker to process messages.

2. Have you calculated how many threads are required by WebSphere Message Broker to run your message flow?
3. Are you planning to start all instances at message flow start time? If yes, consider how many threads are allocated at start up and what other solutions are competing for resources in your system.
4. Do you have any order requirements when processing messages? If yes, consider that an Input node can receive messages out of order, and it will wait for the correct message to arrive before it continues processing any messages. By adding additional instances you can cause resource bottlenecks and not improve your solution performance.
5. Does your message flow design and subflows have multiple Input nodes? Do you need to receive messages through these Input nodes in parallel before continuing the message flow logic? Does one of them take longer to receive a message?

The answers to these questions will help you decide how you want to configure your resources.

To decide which option to choose to create additional instances, consider the following rules:

- In order to create instances which are shared between multiple Input nodes, you must set the **Additional instances** property at the message flow level.
- In order to create instances which are allocated to a specific Input node, you must set the **Additional instances** property at the Input node level.

In addition, you must consider the following behavior based on how you configure additional instances:

- When you set additional instances on an Input node, the additional instances or threads only run against that specific Input node. For example, if you have Input node A with 15 additional instances and Input node B with 0 additional instances, then if there is work waiting on Queue B, it will only be processed one message at a time. The spare instances belonging to Input node A cannot be used to process messages received from Input node B.
- The disadvantage of setting thread pools at message flow level is that one busy Input node could starve other input nodes.

You must decide how you want to set additional instances based on the priority of work on different Input nodes and your subflows and message flow design.

Procedure

When you design a message flow or a subflow that includes one or more Input nodes, you must consider the impact on resource allocation and performance:

- To learn how you can add additional instances at message flow level so that WebSphere Message Broker allocates additional threads to run messages at run time, see "Adding additional instances at message flow level" on page 34.
- To learn how you can add additional instances at Input node level so that WebSphere Message Broker allocates additional threads to run messages at run time, see "Adding additional instances at message flow node level" on page 36.

What to do next

Allocate additional instances at message flow level or at Input node level. For more information, see "Adding additional instances at message flow level" on page 34 page 34 and “Adding additional instances at message flow node level” on page 36.

Adding additional instances at message flow level

To change the number of threads that a message flow can use at run time to process messages, you can configure the **Additional instances** property of a message flow in a BAR file. These instances can be started as you need them or you can choose to start all of them at once when the message flow starts by selecting the **Start additional instances when the flow starts** property.

About this task

The default value of the **Additional instances** property is 0 threads. The maximum value of the **Additional instances** property is 256 threads.

The following conditions apply when WebSphere Message Broker allocates threads to a message flow based on your configuration options:

- By default, one thread is allocated per Input node. The Input node can be defined in the main message flow or within a subflow.
- You can add more instances or threads by setting the **Additional instances** property of the message flow that is included in the BAR file to a value greater than 0 and up to 256 threads.

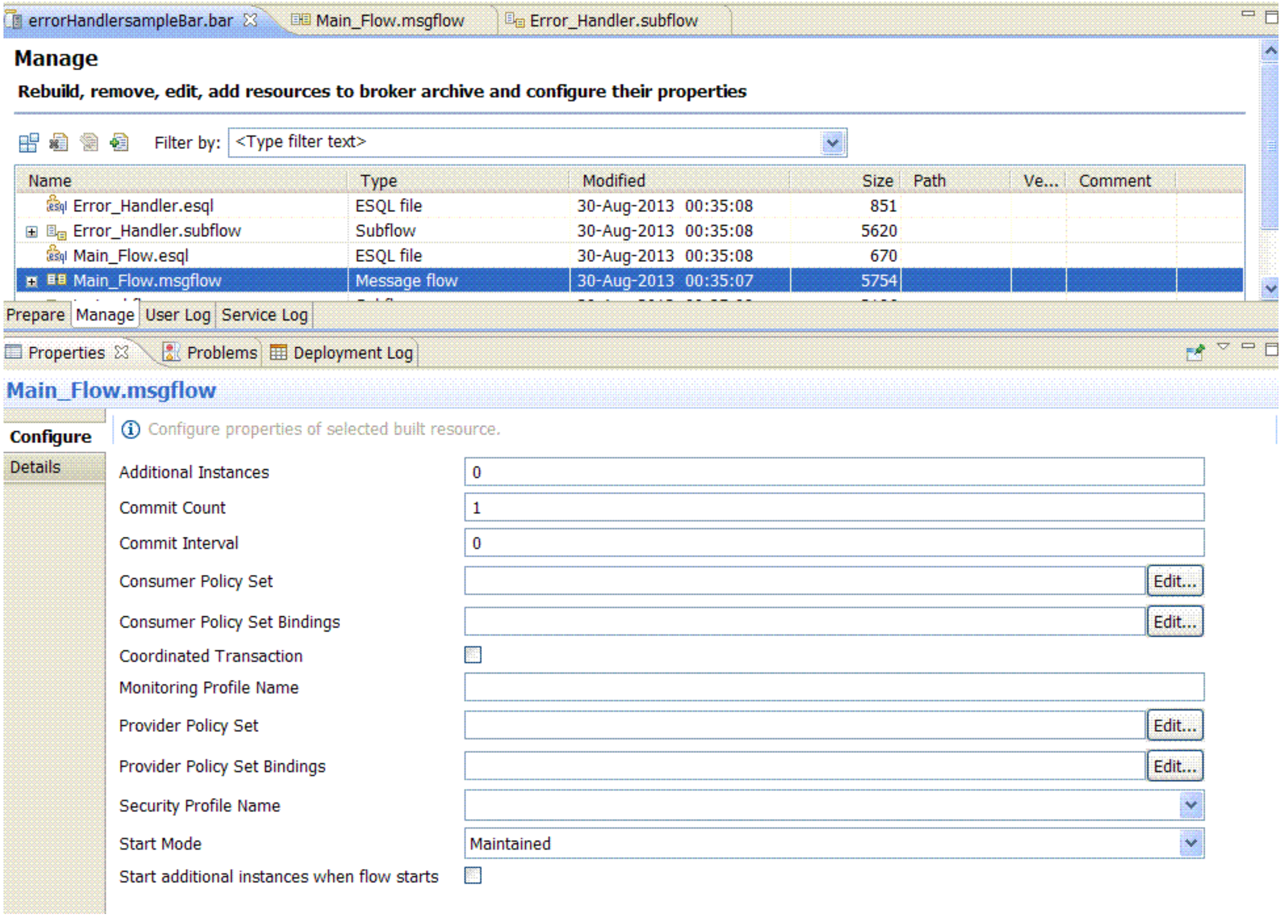
  Note: If you set the value to a number greater than 256, the BAR file will fail to deploy.
- If you have more than one Input node, the additional instances that you configure can be used by any Input node that receives messages. An additional instance is allocated to an Input node that has its default thread busy processing a previous message and needs to process an incoming message.
- The number of threads available to process messages when you start a message flow is by default equal to the number of Input nodes. WebSphere Message Broker will start additional instances as required.
- If you want to start all of the threads configured at once, including the ones configured through the **Additional instances** property, you must select the **Start additional instances when the flow starts** property associated with the message flow.
- Additional message flow instances are configured at flow level. If you have a subflow included in a message flow, you cannot configure additional instances for the subflow independently. This condition applies whether the subflow is used as the starting node in a message flow, as a middle node in a message flow, or as the last node in a message flow.

Procedure

To configure the **Additional instances** property of a message flow in a BAR file, complete the following steps:

1. Open the BAR file that contains the message flow and subflows for which you want to set the **Additional instances** property. In the **Broker Development** view, identify the project that contains the BAR file, and double-click the file to open it.
2. Set the **Additional instances** property of a message flow that is included in the BAR file to a value greater than 0 and up to 256.

   The following figure outlines the configuration properties of a message flow:



Manage

Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type	Modified	Size	Path	Ve...	Comment
Error_Handler.esql	ESQL file	30-Aug-2013 00:35:08	851			
Error_Handler.subflow	Subflow	30-Aug-2013 00:35:08	5620			
Main_Flow.esql	ESQL file	30-Aug-2013 00:35:08	670			
Main_Flow.msgflow	Message flow	30-Aug-2013 00:35:07	5754			

Main_Flow.msgflow

Configure properties of selected built resource.

Additional Instances	0
Commit Count	1
Commit Interval	0
Consumer Policy Set	Edit...
Consumer Policy Set Bindings	Edit...
Coordinated Transaction	□
Monitoring Profile Name	
Provider Policy Set	Edit...
Provider Policy Set Bindings	Edit...
Security Profile Name	
Start Mode	Maintained
Start additional instances when flow starts	□

Note: If you set the value to a number greater than 256, the BAR file will fail to deploy.

Results

Based on your message flow design and configuration, the following tables summarize the number of threads available when running the message flow in WebSphere Message Broker:

Table 7. Sample summary table of parallel threads available when running a message flow in a broker

Number of Input nodes in a message flow	Number of default instances or threads per message flow (1 thread per input node in a message flow)	Additional instances property (set at message flow level)	Maximum number of threads that can process messages in parallel
1	1	0	1
1	1	1	2
1	1	2	3
2	2	0	2
2	2	1	3
2	2	2	4
3	3	2	5

Table 8. Sample summary table of parallel threads available when running a message flow in a broker that contains a subflow with Input nodes

Number of Input nodes in a message flow (not included in a subflow)	Number of Input nodes modeled in a subflow that is included in a message flow	Number of default instances or threads per message flow (1 thread per input node in a message flow)	Additional instances property (set at message flow level)	Maximum number of threads that can process messages in parallel
0	1	1	0	1
1	1	2	0	2
0	1	1	1	2
1	1	2	1	3
0	1	1	2	3
1	1	2	2	4
0	2	2	0	2
2	2	4	2	6
0	2	2	1	3
0	2	2	2	4
0	3	3	2	5

What to do next

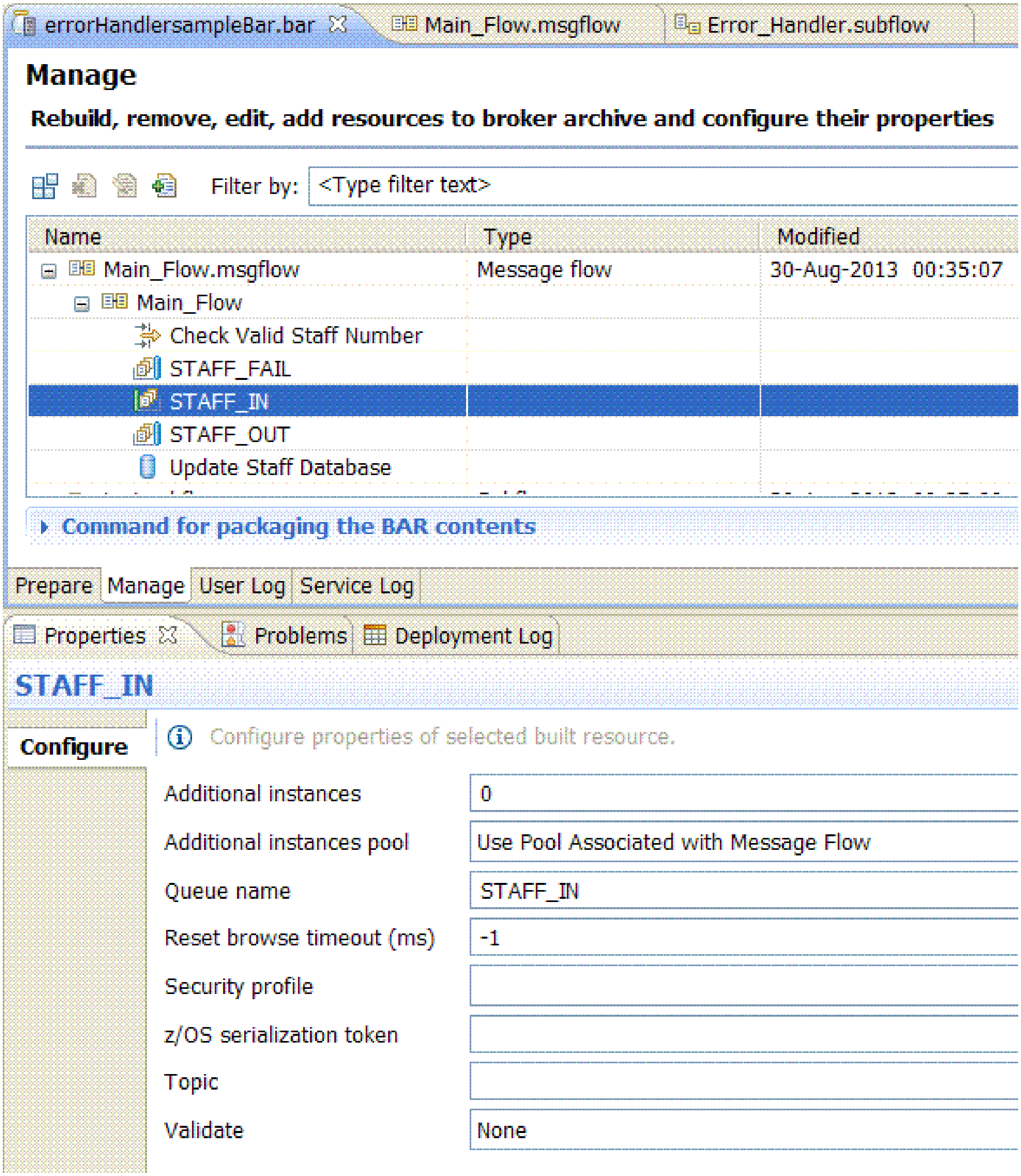
Return to “Configuring a message flow that includes subflows” on page 32.

Adding additional instances at message flow node level

To change the number of threads that a message flow node can use at run time to process messages, you can configure the **Additional instances** property of an Input node that is included in a message flow or subflow added to a BAR file.

About this task

The following figure outlines the configuration properties of a message flow node:



Manage

Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type	Modified
Main_Flow.msgflow	Message flow	30-Aug-2013 00:35:07
Main_Flow		
Check Valid Staff Number		
STAFF_FAIL		
STAFF_IN		
STAFF_OUT		
Update Staff Database		

STAFF_IN

Configure

Configure properties of selected built resource.

Additional instances	0
Additional instances pool	Use Pool Associated with Message Flow
Queue name	STAFF_IN
Reset browse timeout (ms)	-1
Security profile	
z/OS serialization token	
Topic	
Validate	None

WebSphere Message Broker allocates threads to a message flow as follows:

- By default, one thread is allocated per Input node. The Input node can be defined in the main message flow or within a subflow.
- The default value of the **Additional instances** property is 0 threads. The maximum value of the **Additional instances** property is 256 threads.
- All instances defined in the **Additional instances** property for an Input node are started by WebSphere Message Broker as required.

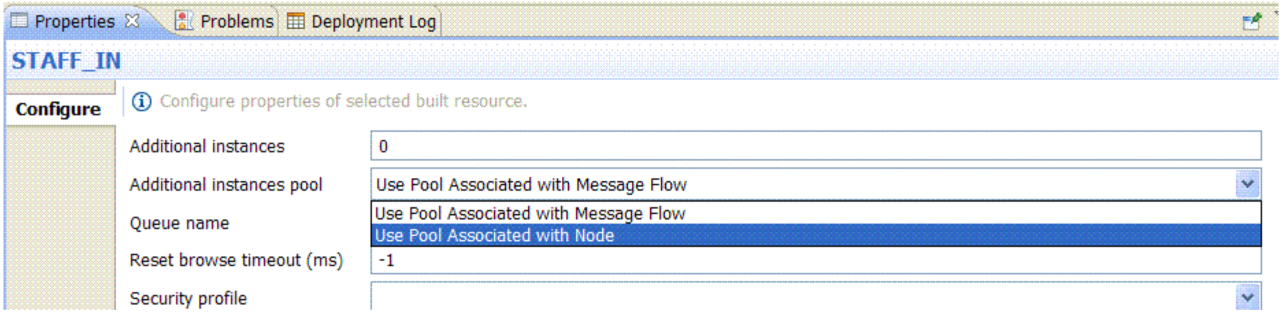
- The number of threads available to process messages when you start a message flow is by default equal to the number of Input nodes plus the number of additional instances configured for each Input node in the BAR file manage section.
- If you have a subflow included in a message flow, you can configure additional instances for each Input node modeled within the subflow. This condition applies whether the subflow is used as the starting node in a message flow, as a middle node in a message flow, or as the last node in a message flow.

Procedure

To add more threads to process messages by a specific Input node, you must complete the following steps:

1. Open the BAR file that contains the message flow and subflows for which you want to set the **Additional instances** property. In the **Broker Development** view, identify the project that contains the BAR file, and double-click the file to open it.
2. Set the **Additional instances** property of the Input node that is included in the BAR file to a value greater than 0 and up to 256.

   Note: If you set the value to a number greater than 256, the BAR file will fail to deploy.
3. Change the **Additional instances pool** to `Use Pool associated with Node`.



Additional instances: 0
Additional instances pool: Use Pool Associated with Message Flow
Use Pool Associated with Message Flow
Use Pool Associated with Node
Queue name
Reset browse timeout (ms): -1
Security profile

Results

Based on your message flow design and configuration, the following tables summarize the number of threads available when running the message flow in WebSphere Message Broker:

Table 9. Sample summary table of parallel threads available when running a message flow in a broker

Number of Input nodes in a message flow or subflow	Number of default instances or threads per message flow (1 thread per input node in a message flow)	Additional instances property (set at input node level)	Maximum number of threads that can process messages in parallel
1	1	0	(1+0) = 1
1	1	1	(1+1) = 2
1	1	2	(1+2) = 3
2	2	0	(1+0) + (1+0) = 2
2	2	Additional instances = 1 for all Input nodes	(1+1) + (1+1) = 4

Table 9. Sample summary table of parallel threads available when running a message flow in a broker (continued)

Number of Input nodes in a message flow or subflow	Number of default instances or threads per message flow (1 thread per input node in a message flow)	Additional instances property (set at input node level)	Maximum number of threads that can process messages in parallel
3	3	Additional instances = 2 for all Input nodes	(1 +2) + (1 +2) + (1 +2) = 9
3	3	Additional instances = 1 for one Input node; Additional instances = 2 for another Input node; Additional instances = 3 for another Input node	(1+1) + (1+2) + (1 + 3) = 9

What to do next

Return to "Configuring a message flow that includes subflows" on page 32.

Implementing the solution: Creating a subflow as a `.msgflow` file

To reuse common application logic in multiple integration solutions, you can create a subflow as a `.msgflow` file. You must understand how WebSphere Message Broker handles subflows at deployment and run time, so that you can configure the subflow properties based on your solution requirements.

About this task

To implement the technical solution, complete the following tasks:

1. "Designing a subflow created as a `.msgflow` file" on page 40
2. "Versioning a subflow" on page 13
   - "Defining a keyword to be used as the container of a subflow version" on page 14
   - "Versioning a subflow by using a keyword in the Label property of a Passthrough node" on page 15
   - "Versioning a subflow by using a keyword as a Passthrough node name" on page 17
   - "Versioning a subflow by setting the Version property of the subflow" on page 19
3. "Creating a BAR file that includes a subflow created as a `.msgflow` file" on page 52
4. "Deploying a subflow created as a `.msgflow` file" on page 57
5. "Updating a subflow" on page 31
6. "Configuring a message flow that includes subflows" on page 32

To create a subflow created as a `.msgflow` file, see Creating a message flow.

Note: This is not the preferred subflow type for new subflow development. Subflows created as a `.msgflow` file are available in WebSphere Message Broker Version 8 for compatibility with earlier versions.

Designing a subflow created as a `.msgflow` file

If you are migrating subflows from earlier versions of WebSphere Message Broker, you need to understand how to define a subflow created as a `.msgflow` file in WebSphere Message Broker Version 8.

You can add subflows created as a `.msgflow` file into your message flows if either of the following statements is true:

- The subflow that you want to add to a message flow is defined in a library. Applications and services can reference libraries.

  Note: A library is a logical grouping of related code, data, or both that typically contains reusable subflows, and other type of resources.
- The subflow that you want to add to a message flow is defined in the same Message Broker project, application, or service as the message flow.

During the development phase, in the WebSphere Message Broker Toolkit, you can define a subflow once, and then reuse it in more than one message flow.

- If you use a subflow in multiple message flows that are defined in one or more Message Broker projects, you can define the subflow within a library. This library can be referenced by any Message Broker project that needs access to the resources it contains.
- If you use a subflow in multiple message flows that are defined within the same Message Broker project, you can define the subflow within the Message Broker project.

Note: When the WebSphere Message Broker run time encounters a subflow within a message flow, it creates a copy of all the message flow nodes that define that subflow. If the subflow is used more than once, it creates a copy of all the message flow nodes that define the subflow every time it occurs. This increases resource usage, which can affect your overall message flow performance.

Types of subflows supported

In WebSphere Message Broker Version 8, you can create a subflow as a `.subflow` file or as a `.msgflow` file.

Note: If you have migrated subflows from earlier versions of WebSphere Message Broker Version 8, you have subflows created as `.msgflow` files.

Subflow content

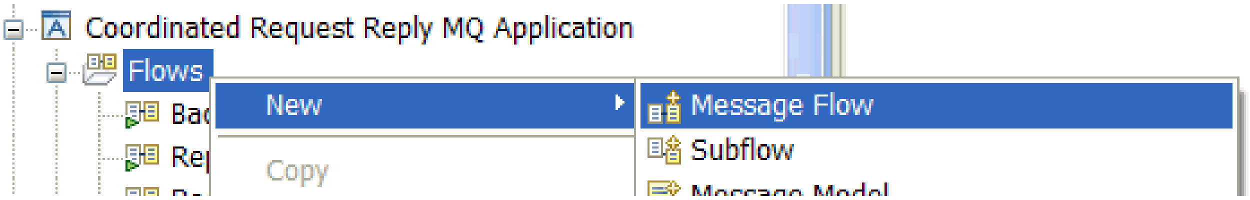
You define subflow content in the same way as you define message flow content, by adding, configuring, and connecting message flow nodes.

A subflow can have one or more input nodes, and zero, one or more output nodes.

- If you want to use a subflow as the first node in your message flow, you need to add at least one Input node, such as the MQInput node, and one generic Output node.
- If you want to use a subflow in the middle of a message flow, you need to add at least one generic Input node. You only need to add a generic Output node if you want to connect additional nodes in the message flow after you add the subflow.

- If you want to use a subflow as the last node in a message flow, you need to add at least one generic Input node, and at least one Output node, such as the MQOutput node.

Subflows created as `.msgflow` files are grouped in the folder named **Message Flow** within your application, service, or Message Broker project. The following figure illustrates where you can find subflows created as `.msgflow` files:



For subflows created as a `.msgflow` file, WebSphere Message Broker creates a blank flow to which you can add the message flow nodes that you need.

Design considerations

When you define the message flow nodes in a subflow created as a `.msgflow` file, you should consider the following recommendations:

- Define subflows in libraries. Create references to a library from your Message Broker projects to be able to reuse the common logic.
- Limit the number of nested subflows.
- Group logic that is defined in consecutive compute nodes into one single compute node. This will improve performance, since the number of nodes loaded at runtime will be reduced.
- Add a version to the subflow.
- Do not nest subflows of different types, that is, do not add a subflow created as a `.msgflow` file into a subflow created as a `.subflow` file or vice versa.

Development considerations

In WebSphere Message Broker releases earlier than version 8, you create message flows and other resources within Message flow projects.

Note: Message flow projects have been renamed to Message Broker projects in WebSphere Message Broker Version 8.
You create reusable logic as a message flow. You embed message flows into message flows.

Note: WebSphere Message Broker releases earlier than version 8 only supported subflows created as a `.msgflow` file. However, if you plan to develop new integration solutions, you should create subflows created as a `.subflow` file. You should convert your subflows created as `.msgflow` files into subflows created as `.subflow` files.

Versioning a subflow

To enable version control of a subflow at run time, you can set the version by using a keyword or by setting the subflow Version property.

About this task

You set the version of a subflow based on where you want to check its value at run time. The following table summarizes the different components that you can use to check the version of a subflow.

Table 10. Summary of the components that you can use to check the version of a subflow

Methods	WebSphere Message Broker Toolkit	WebSphere Message Broker Explorer	WebSphere Message Broker console
Configure a keyword in the Label property of a Passthrough node	Valid to check the value at development time	Valid to check the value at run time	Valid to check the value at run time
Name a Passthrough node with a keyword	Valid to check the value at development time	Valid to check the value at run time	Valid to check the value at run time
Configure the Version property of a subflow	Valid to check the value at development time	Valid to check the value at run time	Not a valid option

Procedure

To enable version control of a subflow at run time, you can set the version of a subflow in any of the following ways:

- Configure a keyword in the Label property of a Passthrough node. For more information, see “Defining a keyword to be used as the container of a subflow version” on page 14, and then “Versioning a subflow by using a keyword in the Label property of a Passthrough node” on page 15.
- Name a Passthrough node with a keyword. For more information, see “Defining a keyword to be used as the container of a subflow version” on page 14, and then “Versioning a subflow by using a keyword as a Passthrough node name” on page 17.
- Configure the Version property of a subflow. For more information, see “Versioning a subflow by setting the Version property of the subflow” on page 19.

What to do next

After you set the version of a subflow, create a BAR file. For more information, see “Creating a BAR file that includes a subflow created as a `.subflow` file” on page 23 or “Creating a BAR file that includes a subflow created as a `.msgflow` file” on page 52.

Defining a keyword to be used as the container of a subflow version

You can use a *keyword* to add more information to a message flow or subflow.

About this task

The information that is passed through keywords is only accessible at run time when the object for which it has been defined is deployed and is running in the broker runtime environment.

You can check the value of a keyword in any of the following ways:

- In the WebSphere Message Broker console, by using the `mqsireadbar` command to read the properties stored in the broker archive (BAR) file.

- In the WebSphere Message Broker Explorer, on the properties of a deployed message flow or subflow as last deployed to a particular broker.
- In the runtime environment, if you enable user trace for that message flow.

Procedure

To define a keyword that you can use as the container of a subflow version, complete the following step:

Define the syntax of the keyword as follows: `$MQSI` *`name of your subflow`* `Subflow Version = Version` *`xxxx`* `MQSI$` where *name of your subflow* is the name of your subflow; and *xxxx* is your subflow version identifier.
You can define your own keyword to define the version. For guidance on defining *keywords*, see Guidance for defining keywords. You can define your own naming convention to assign a version value to a subflow.

What to do next

Use the keyword to set the version of a subflow. For more information, see "Versioning a subflow by using a keyword in the Label property of a Passthrough node" on page 15 or "Versioning a subflow by using a keyword as a Passthrough node name" on page 17.

Related information:

mqsireadbar command

Versioning a subflow by using a keyword in the Label property of a Passthrough node

To define the version of a subflow, you can define a keyword in the Label property of a Passthrough node.

Before you begin

Set up the initial configuration. See "Creating the initial configuration for the Error Handler sample" on page 7.

About this task

When you set a subflow version by defining a keyword in the Label property of a Passthrough node, you can check the value at run time in the WebSphere Message Broker console and in WebSphere Message Broker Explorer.

Procedure

You must complete the following steps in the WebSphere Message Broker Toolkit:

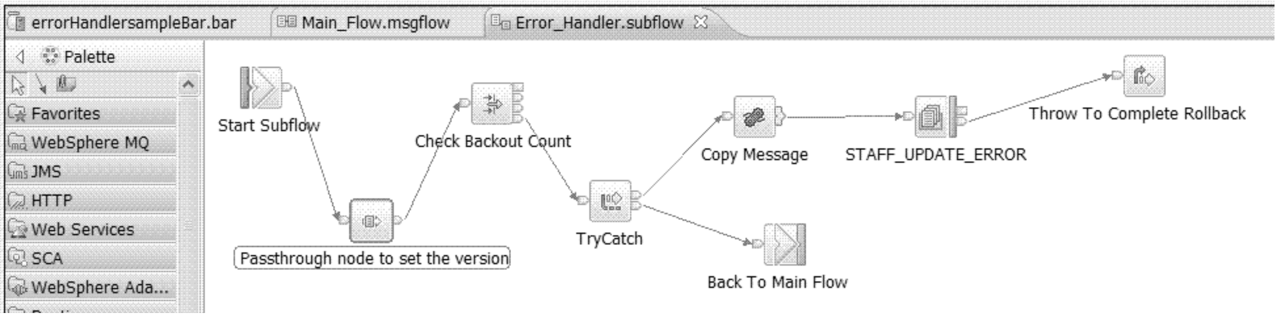
1. Open the subflow in the Message Flow editor.
2. Add a Passthrough node to a subflow.
3. Connect the Passthrough node to other nodes in the subflow or leave it unconnected.
4. Define a keyword. For more information, see "Defining a keyword to be used as the container of a subflow version" on page 14.
5. Set the keyword in the **Label** property of the Passthrough node.

Example

For example, in the Error Handler sample provided with WebSphere Message Broker, you can add and check the version of the subflow by completing the following steps:

1. In the WebSphere Message Broker Toolkit, open the **Error_Handler.subflow** subflow in the Message Flow editor:

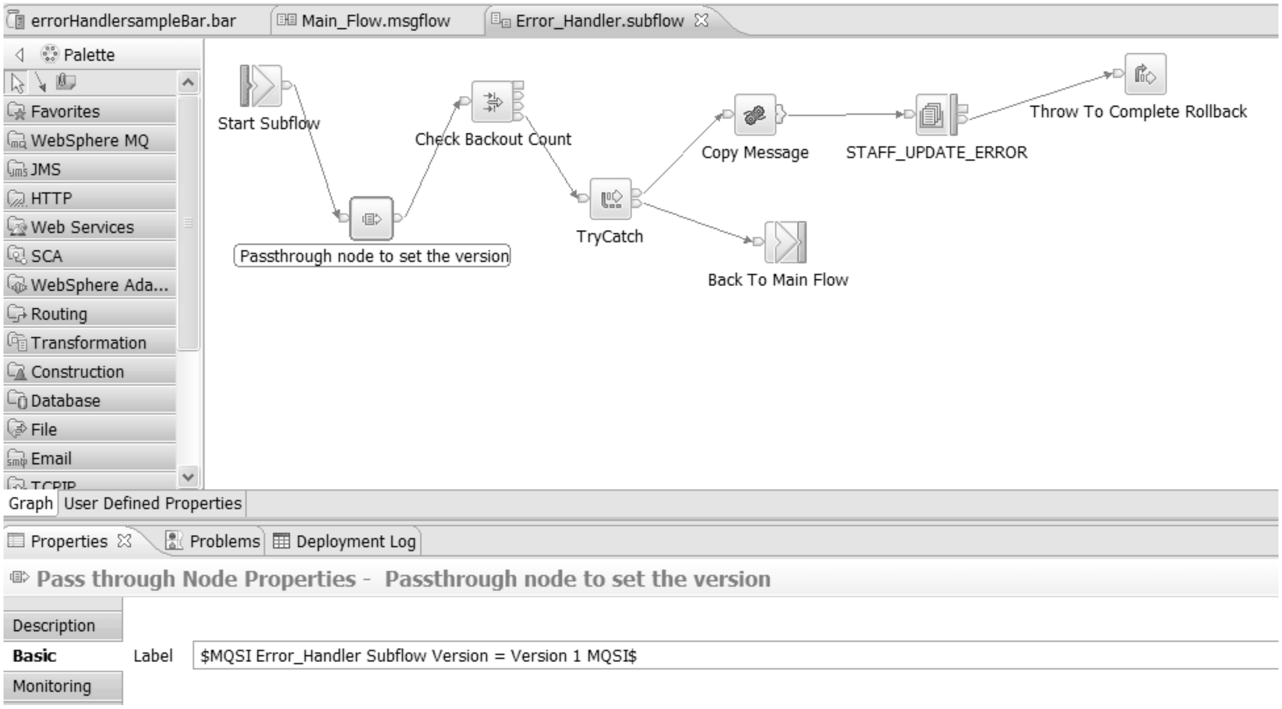
   In the **Broker Development** view, right-click the name of the subflow, and then select **Open with** > **Message Flow editor**.
2. In the **Error_Handler.subflow**, connect a Passthrough node.



3. In the Properties tab of the Passthrough node, use the **Label** field to set the subflow version. This must be in the format:

   `$MQSI` *`name of your subflow`* `Subflow Version = Version` *`xxxx`* `MQSI$.`

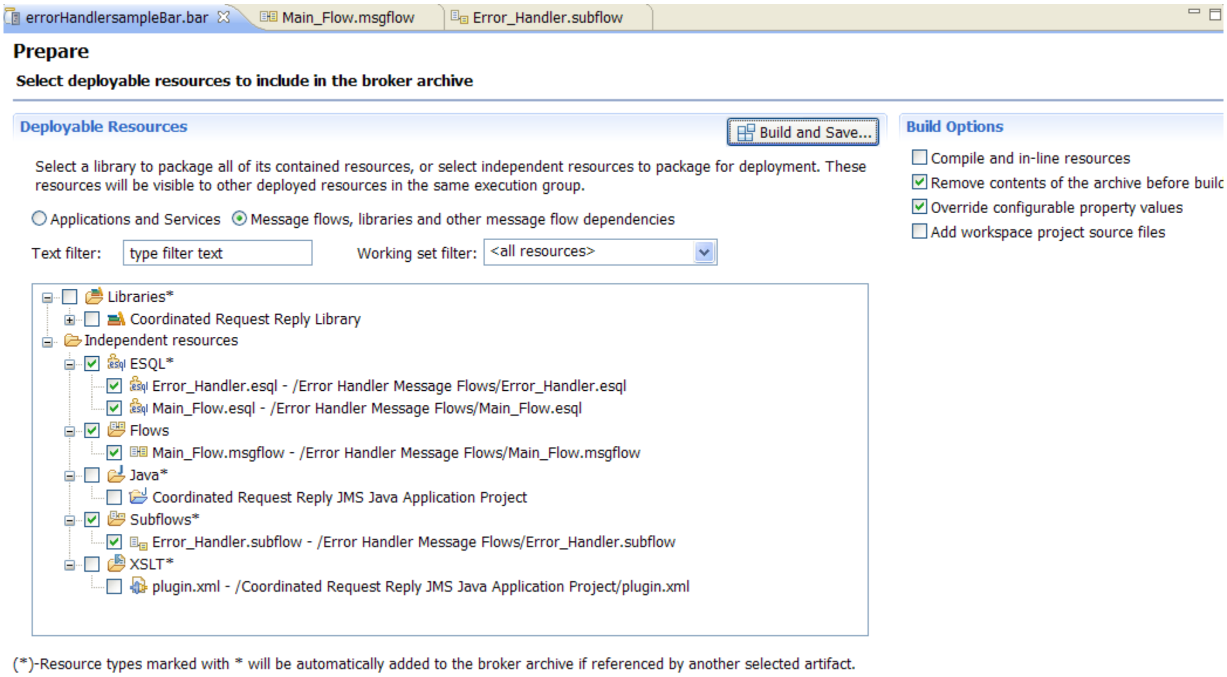
   where *name of your subflow* is the name of your subflow; and *xxxx* is your subflow version identifier.



Pass through Node Properties - Passthrough node to set the version

Label: $MQSI Error_Handler Subflow Version = Version 1 MQSI$

4. Create a BAR file that includes the **Main_flow.msgflow**.



Prepare

Select deployable resources to include in the broker archive

Deployable Resources

Select a library to package all of its contained resources, or select independent resources to package for deployment. These resources will be visible to other deployed resources in the same execution group.

○ Applications and Services ● Message flows, libraries and other message flow dependencies

Text filter: type filter text   Working set filter: <all resources>

```
⊟ □ Libraries*
   ⊞ □ Coordinated Request Reply Library
⊟ Independent resources
   ⊟ ☑ ESQL*
      ☑ Error_Handler.esql - /Error Handler Message Flows/Error_Handler.esql
      ☑ Main_Flow.esql - /Error Handler Message Flows/Main_Flow.esql
   ⊟ ☑ Flows
      ☑ Main_Flow.msgflow - /Error Handler Message Flows/Main_Flow.msgflow
   ⊟ □ Java*
      □ Coordinated Request Reply JMS Java Application Project
   ⊟ ☑ Subflows*
      ☑ Error_Handler.subflow - /Error Handler Message Flows/Error_Handler.subflow
   ⊟ □ XSLT*
      □ plugin.xml - /Coordinated Request Reply JMS Java Application Project/plugin.xml
```

(*)-Resource types marked with * will be automatically added to the broker archive if referenced by another selected artifact.

Build Options

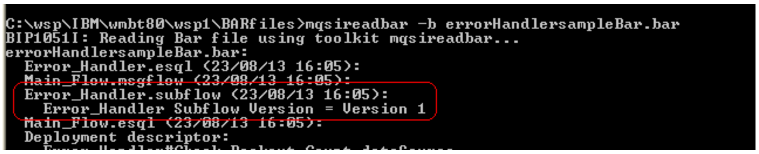
□ Compile and in-line resources
☑ Remove contents of the archive before builc
☑ Override configurable property values
□ Add workspace project source files

5. Deploy the message flow into a broker runtime environment. For more information, see Deploying a broker archive file.
6. In the WebSphere Message Broker console, run the `mqsireadbar` command to check the version of the subflow.

   You run the command using the following syntax:

   `mqsireadbar -b` *`your BAR file name`*

   where *your BAR file name* is the name of the BAR file you have created to deploy the sample into the run time.



```
C:\wsp\IBM\wmbt80\wsp1\BARfiles>mqsireadbar -b errorHandlersampleBar.bar
BIP1051I: Reading Bar file using toolkit mqsireadbar...
errorHandlersampleBar.bar:
  Error_Handler.esql (23/08/13 16:05):
  Main_Flow.msgflow (23/08/13 16:05):
  Error_Handler.subflow (23/08/13 16:05):
    Error_Handler Subflow Version = Version 1
  Main_Flow.esql (23/08/13 16:05):
  Deployment descriptor:
```

What to do next

After you set the version of a subflow, create a BAR file. For more information, see “Creating a BAR file that includes a subflow created as a `.subflow` file” on page 23 or “Creating a BAR file that includes a subflow created as a `.msgflow` file” on page 52.

Versioning a subflow by using a keyword as a Passthrough node name

To define the version of a subflow, you can set the version of a subflow by adding a Passthrough node to the subflow with a keyword as the node name.

Before you begin

Set up the initial configuration. See “Creating the initial configuration for the Error Handler sample” on page 7.

About this task

When you set a subflow version by adding a Passthrough node to the subflow with a keyword as the node name, you can check the value at run time in the WebSphere Message Broker console and in WebSphere Message Broker Explorer.

Procedure

To define the version of a subflow, you must complete the following steps in the WebSphere Message Broker Toolkit:

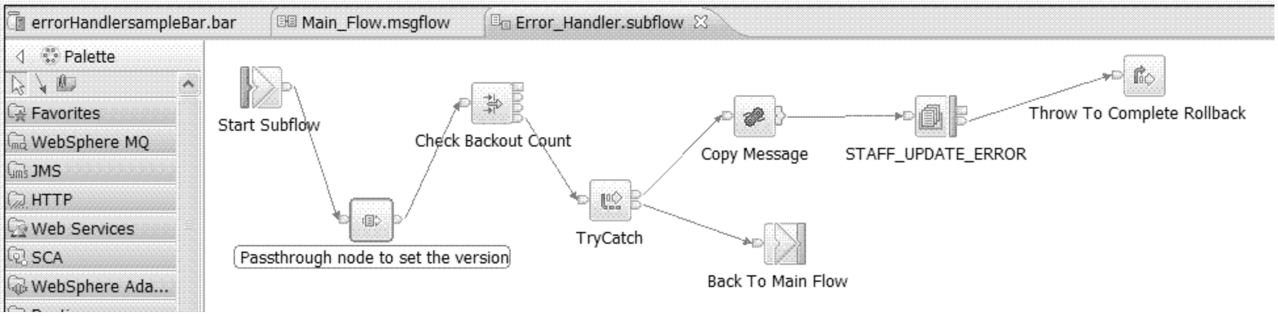
1. Open the subflow in the Message Flow editor.
2. Add a Passthrough node to a subflow.
3. Connect the Passthrough node to other nodes in the subflow or leave it unconnected.
4. Define a keyword. For more information, see "Defining a keyword to be used as the container of a subflow version" on page 14.
5. Set the keyword in the **Node name** property of the Passthrough node.

Example

For example, in the Error Handler sample provided with WebSphere Message Broker, you can add and check the version of the subflow as follows:

1. In the WebSphere Message Broker Toolkit, open the **Error_Handler.subflow** subflow in the Message Flow editor:

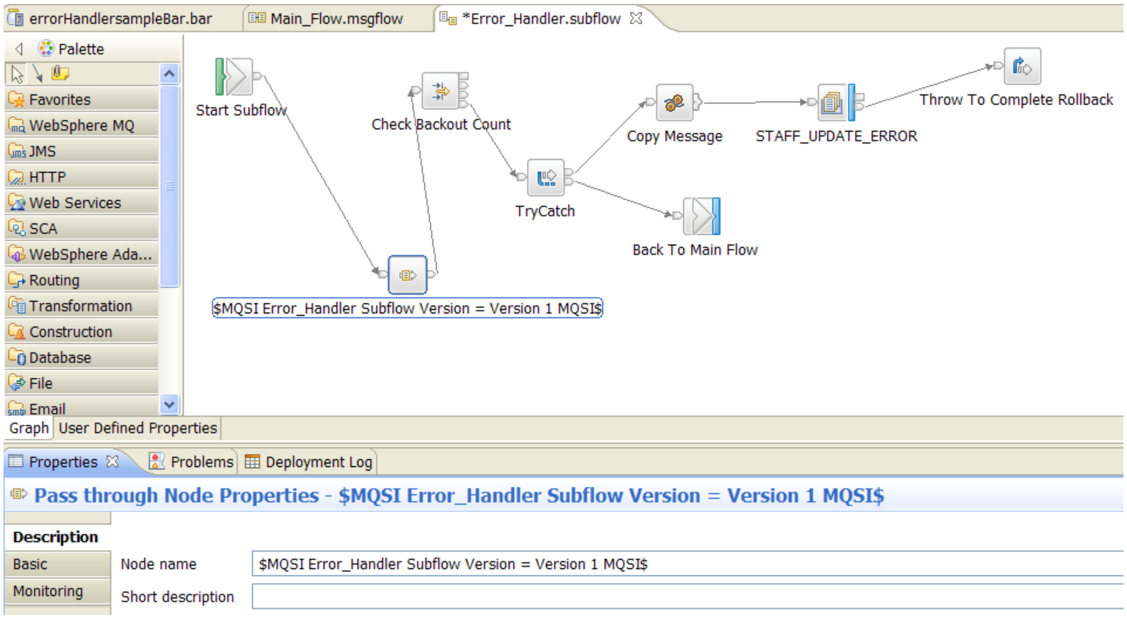
   In the **Broker Development** view, right-click the name of the subflow, and then select **Open with** > **Message Flow editor**.
2. In the **Error_Handler.subflow**, connect a Passthrough node.



3. In the Properties tab of the Passthrough node, use the **Node name** field to set the subflow version. This must be in the format:

   `$MQSI` *`name of your subflow`* `Subflow Version = Version` *`xxxx`* `MQSI$.`

   where *name of your subflow* is the name of your subflow; and *xxxx* is your subflow version identifier.

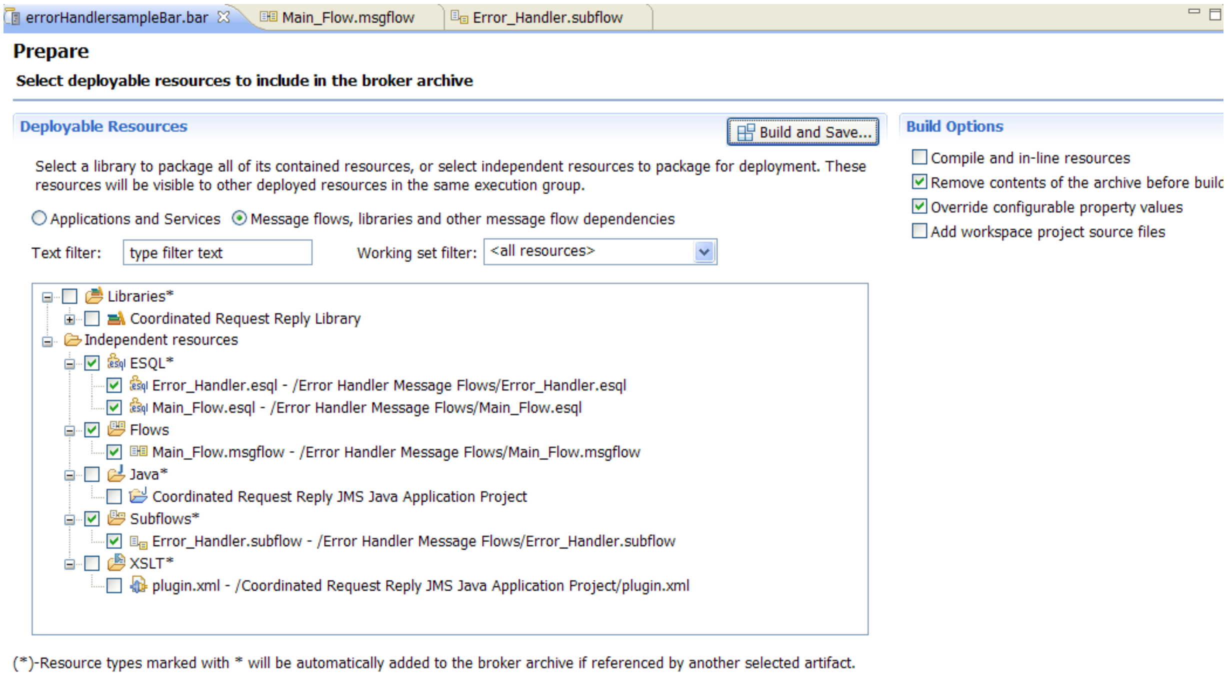


Pass through Node Properties - $MQSI Error_Handler Subflow Version = Version 1 MQSI$

Node name: $MQSI Error_Handler Subflow Version = Version 1 MQSI$

Short description:

4. Create a BAR file that includes the **Main_flow.msgflow**.



Prepare

Select deployable resources to include in the broker archive

Deployable Resources

Select a library to package all of its contained resources, or select independent resources to package for deployment. These resources will be visible to other deployed resources in the same execution group.

○ Applications and Services ● Message flows, libraries and other message flow dependencies

Text filter: type filter text Working set filter: <all resources>

- □ Libraries*
  - □ Coordinated Request Reply Library
- Independent resources
  - ☑ ESQL*
    - ☑ Error_Handler.esql - /Error Handler Message Flows/Error_Handler.esql
    - ☑ Main_Flow.esql - /Error Handler Message Flows/Main_Flow.esql
  - ☑ Flows
    - ☑ Main_Flow.msgflow - /Error Handler Message Flows/Main_Flow.msgflow
  - □ Java*
    - □ Coordinated Request Reply JMS Java Application Project
  - ☑ Subflows*
    - ☑ Error_Handler.subflow - /Error Handler Message Flows/Error_Handler.subflow
  - □ XSLT*
    - □ plugin.xml - /Coordinated Request Reply JMS Java Application Project/plugin.xml

Build Options

□ Compile and in-line resources
☑ Remove contents of the archive before builc
☑ Override configurable property values
□ Add workspace project source files

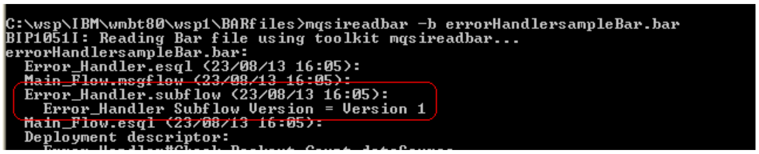
(*)-Resource types marked with * will be automatically added to the broker archive if referenced by another selected artifact.

5. Deploy the message flow into a broker runtime environment. For more information, see Deploying a broker archive file.
6. In the broker console, run the `mqsireadbar` command to check the version of the subflow.

   You run the command using the following syntax:

   ```
   mqsireadbar -b your BAR file name
   ```

   where *your BAR file name* is the name of the BAR file you have created to deploy the sample into the run time.



```
C:\wsp\IBM\wmbt80\wsp1\BARfiles>mqsireadbar -b errorHandlersampleBar.bar
BIP1051I: Reading Bar file using toolkit mqsireadbar...
errorHandlersampleBar.bar:
  Error_Handler.esql (23/08/13 16:05):
  Main_Flow.msgflow (23/08/13 16:05):
  Error_Handler.subflow (23/08/13 16:05):
    Error_Handler Subflow Version = Version 1
  Main_Flow.esql (23/08/13 16:05):
  Deployment descriptor:
    Error_Handler#Check_Packet_Count.dataSource
```

What to do next

After you set the version of a subflow, create a BAR file. For more information, see “Creating a BAR file that includes a subflow created as a `.subflow` file” on page 23 or “Creating a BAR file that includes a subflow created as a `.msgflow` file” on page 52.

Versioning a subflow by setting the Version property of the subflow

To define the version of a subflow, you can set the value of the subflow **Version** property.

Before you begin

Set up the initial configuration. See “Creating the initial configuration for the Error Handler sample” on page 7.

About this task

When you set the Version property of the subflow, you can check the value at run time in the WebSphere Message Broker Toolkit and in WebSphere Message Broker Explorer.

Procedure

You must complete the following steps in the WebSphere Message Broker Toolkit:

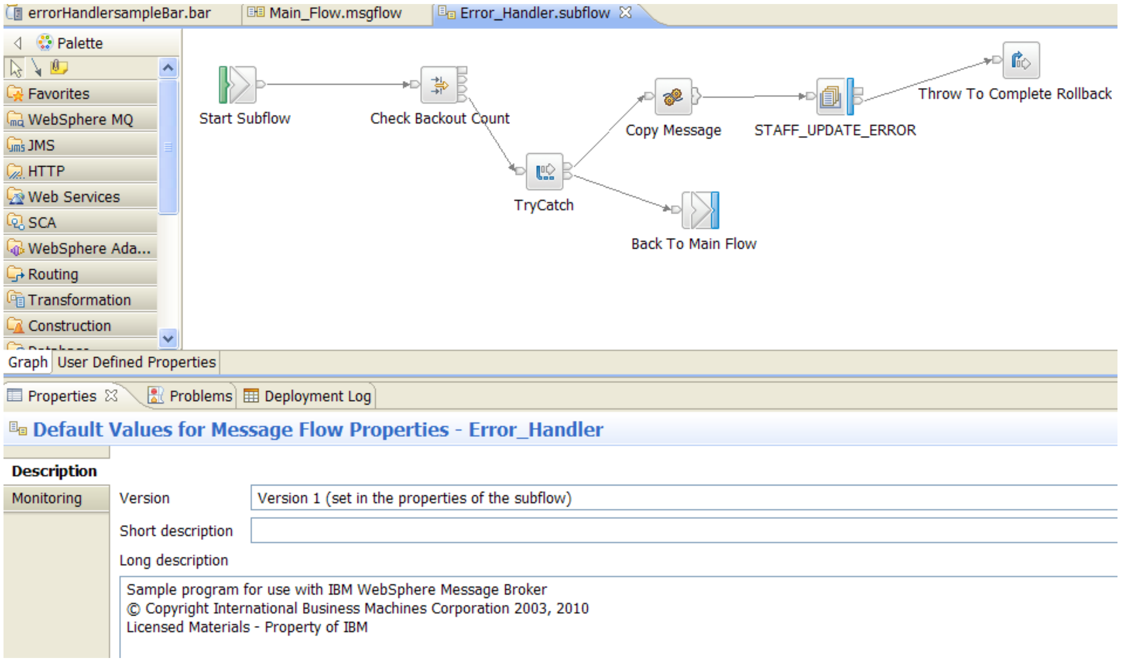
1. Open the subflow in the Message Flow editor.
2. Configure the **Version** property of the subflow.

Example

For example, in the Error Handler sample provided with WebSphere Message Broker, you can add and check the version of the subflow as follows:

1. In the WebSphere Message Broker Toolkit, open the **Error_Handler.subflow** subflow in the Message Flow editor:

   In the **Broker Development** view, right-click the name of the subflow, and then select **Open with** > **Message Flow editor**.
2. In the **Error_Handler.subflow**, configure the **Version** property of the subflow.



Default Values for Message Flow Properties - Error_Handler

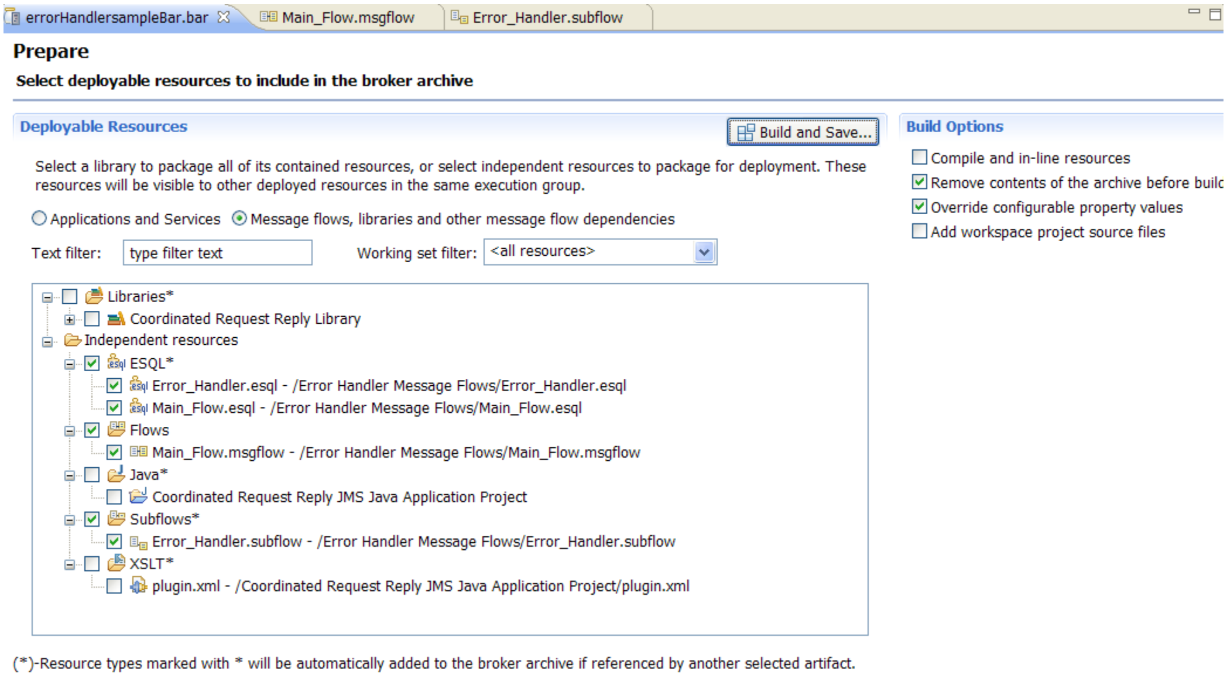
Version: Version 1 (set in the properties of the subflow)

Short description:

Long description:

Sample program for use with IBM WebSphere Message Broker
© Copyright International Business Machines Corporation 2003, 2010
Licensed Materials - Property of IBM

3. Create a BAR file that includes the **Main_flow.msgflow**.



Prepare

Select deployable resources to include in the broker archive

Deployable Resources

Select a library to package all of its contained resources, or select independent resources to package for deployment. These resources will be visible to other deployed resources in the same execution group.

○ Applications and Services ● Message flows, libraries and other message flow dependencies

Text filter: type filter text   Working set filter: <all resources>

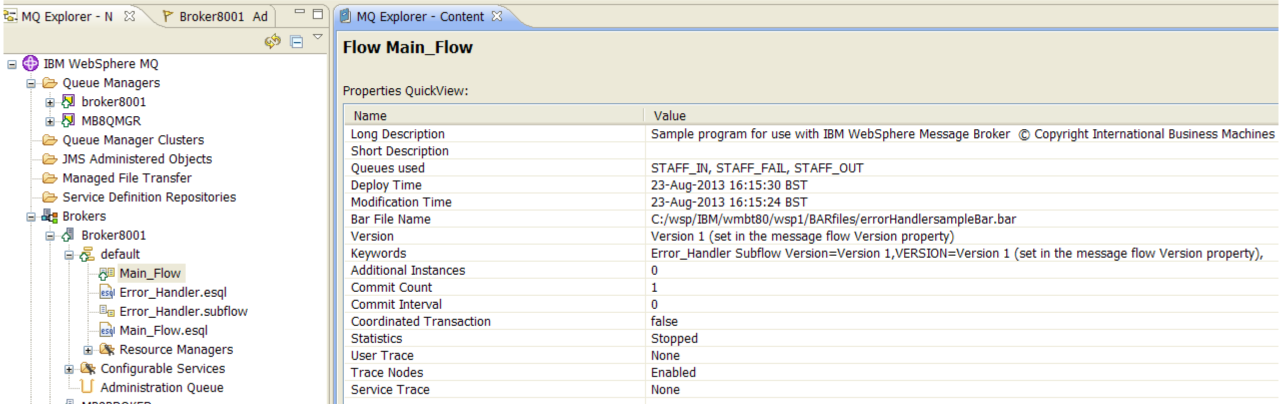
- □ Libraries*
  - □ Coordinated Request Reply Library
- Independent resources
  - ☑ ESQL*
    - ☑ Error_Handler.esql - /Error Handler Message Flows/Error_Handler.esql
    - ☑ Main_Flow.esql - /Error Handler Message Flows/Main_Flow.esql
  - ☑ Flows
    - ☑ Main_Flow.msgflow - /Error Handler Message Flows/Main_Flow.msgflow
  - □ Java*
    - □ Coordinated Request Reply JMS Java Application Project
  - ☑ Subflows*
    - ☑ Error_Handler.subflow - /Error Handler Message Flows/Error_Handler.subflow
  - □ XSLT*
    - □ plugin.xml - /Coordinated Request Reply JMS Java Application Project/plugin.xml

Build Options

□ Compile and in-line resources
☑ Remove contents of the archive before builc
☑ Override configurable property values
□ Add workspace project source files

(*)-Resource types marked with * will be automatically added to the broker archive if referenced by another selected artifact.

4. Deploy the message flow into a broker runtime environment. For more information, see Deploying a broker archive file.
5. In the WebSphere Message Broker Explorer, double-click the message flow that includes the subflow. In our example, double-click **Main_Flow**. The Properties QuickView panel opens. In this panel, you can check the version of the subflow under **Keywords**.

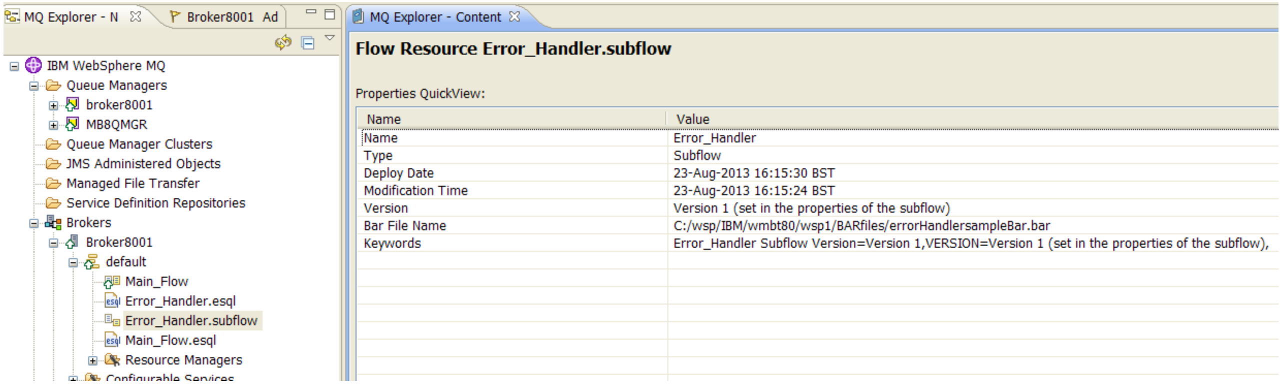


Flow Main_Flow

Properties QuickView:

Name	Value
Long Description	Sample program for use with IBM WebSphere Message Broker © Copyright International Business Machines
Short Description	
Queues used	STAFF_IN, STAFF_FAIL, STAFF_OUT
Deploy Time	23-Aug-2013 16:15:30 BST
Modification Time	23-Aug-2013 16:15:24 BST
Bar File Name	C:/wsp/IBM/wmbt80/wsp1/BARfiles/errorHandlersampleBar.bar
Version	Version 1 (set in the message flow Version property)
Keywords	Error_Handler Subflow Version=Version 1,VERSION=Version 1 (set in the message flow Version property),
Additional Instances	0
Commit Count	1
Commit Interval	0
Coordinated Transaction	false
Statistics	Stopped
User Trace	None
Trace Nodes	Enabled
Service Trace	None

6. In the WebSphere Message Broker Explorer, double-click the subflow. In our example, double-click **Error_Handler.subflow**. The Properties QuickView panel opens. In this panel, you can check the version of the subflow under **Keywords**.



Flow Resource Error_Handler.subflow

Properties QuickView:

Name	Value
Name	Error_Handler
Type	Subflow
Deploy Date	23-Aug-2013 16:15:30 BST
Modification Time	23-Aug-2013 16:15:24 BST
Version	Version 1 (set in the properties of the subflow)
Bar File Name	C:/wsp/IBM/wmbt80/wsp1/BARfiles/errorHandlersampleBar.bar
Keywords	Error_Handler Subflow Version=Version 1,VERSION=Version 1 (set in the properties of the subflow),

What to do next

After you set the version of a subflow, create a BAR file. For more information, see "Creating a BAR file that includes a subflow created as a `.subflow` file" on page 23 or "Creating a BAR file that includes a subflow created as a `.msgflow` file" on page 52.

Checking the Version of a subflow at run time

You can check the version of a subflow at run time by using the WebSphere Message Broker console or the WebSphere Message Broker Explorer.

Before you begin

Set up the initial configuration. See "Creating the initial configuration for the Error Handler sample" on page 7.

About this task

Depending on how you have defined the version of a subflow during the development phase, you have different options to check the version of a subflow at run time:

- If you defined a keyword in the Label property of a Passthrough node, you can use the WebSphere Message Broker console or the WebSphere Message Broker Explorer.
- If you added a Passthrough node to the subflow with a keyword as the node name, you can use the WebSphere Message Broker console or the WebSphere Message Broker Explorer.
- If you set the Version property of the subflow, you can use WebSphere Message Broker Explorer to see the subflow version.

Table 11. Summary of WebSphere Message Broker components where you can check the version of a subflow at run time

How you defined the version during development	**WebSphere Message Broker console**	**WebSphere Message Broker Explorer**	**WebSphere Message Broker Toolkit**
Using a keyword	x	x	It is not possible to check the version at run time.
Setting the Version property of a subflow created as a `.subflow` file	It is not possible to check the version at run time.	x	It is not possible to check the version at run time.
Setting the Version property of the subflow created as a `.msgflow` file	It is not possible to check the version at run time.	x	x

Procedure

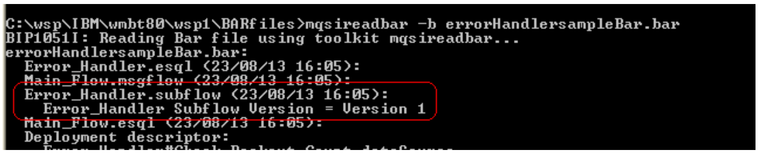
To check the version of a subflow at run time, choose one of the following options:

1. In the WebSphere Message Broker console, run the `mqsireadbar` command. This command reads the properties stored in the broker archive (BAR) file.

   You run the command using the following syntax:

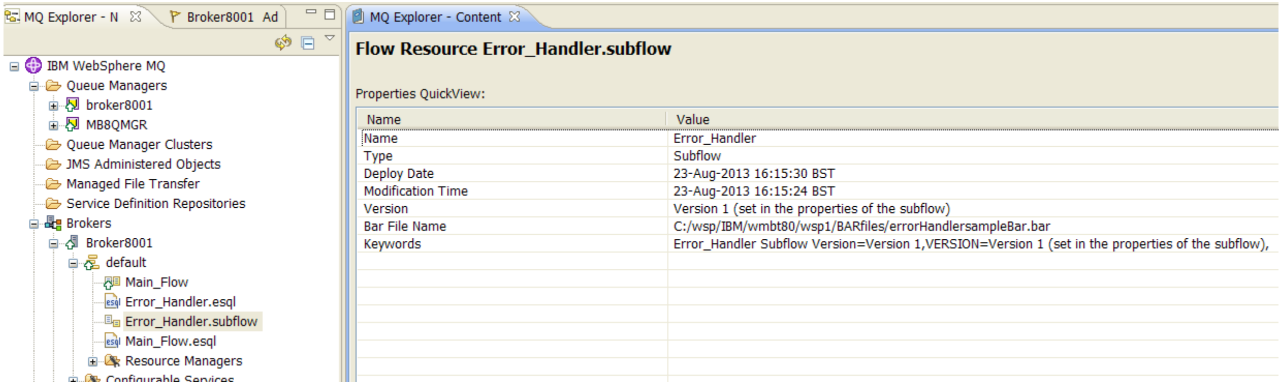
   `mqsireadbar -b` *`your BAR file name`*

   where *your BAR file name* is the name of the BAR file you have created to deploy into the run time.



```
C:\wsp\IBM\wmbt80\wsp1\BARfiles>mqsireadbar -b errorHandlersampleBar.bar
BIP1051I: Reading Bar file using toolkit mqsireadbar...
errorHandlersampleBar.bar:
  Error_Handler.esql (23/08/13 16:05):
  Main_Flow.msgflow (23/08/13 16:05):
  Error_Handler.subflow (23/08/13 16:05):
    Error_Handler Subflow Version = Version 1
  Main_Flow.esql (23/08/13 16:05):
  Deployment descriptor:
```

2. In the WebSphere Message Broker Explorer, double-click a subflow. The MQ Explorer - Content tab opens where you can see the Properties QuickView.



Flow Resource Error_Handler.subflow

Properties QuickView:

Name	Value
Name	Error_Handler
Type	Subflow
Deploy Date	23-Aug-2013 16:15:30 BST
Modification Time	23-Aug-2013 16:15:24 BST
Version	Version 1 (set in the properties of the subflow)
Bar File Name	C:/wsp/IBM/wmbt80/wsp1/BARfiles/errorHandlersampleBar.bar
Keywords	Error_Handler Subflow Version=Version 1,VERSION=Version 1 (set in the properties of the subflow),

What to do next

Return to “Updating a subflow” on page 31.

Creating a BAR file that includes a subflow created as a `.msgflow` file

When deploying a subflow created as a `.msgflow` file, you need to understand how WebSphere Message Broker packages subflows into BAR files, and what are the configurations that you can create to deploy into the broker run time. Beware of the differences that exist between packaging subflows into a BAR file in WebSphere Message Broker Version 8.0.0.0 and WebSphere Message Broker Version 8.0.0.1.

Before you begin

Set up the initial configuration. See “Creating the initial configuration for the Error Handler sample” on page 7.

Procedure

To create a BAR file in WebSphere Message Broker Version 8.0.0.1, complete the following steps:

1. Create a broker archive (BAR file) for each configuration that you want to deploy to the run time. For more information, see Creating a broker archive.
2. Add files to the BAR file. For more information, see Adding files to a broker archive.
3. Configure the BAR file and its resources. For more information, see Configuring the start mode of flows and applications at development time.
4. Choose whether you want to deploy the resources in the BAR file in their compile mode or as objects. For more information, see “BAR builder behavior for subflows created as `.msgflow` files” on page 53.
5. Save and build the BAR file in the BAR builder.

What to do next

Deploy a subflow created as a `.msgflow` file. For more information, see “Deploying a subflow created as a `.msgflow` file” on page 57.

BAR builder behavior for subflows created as `.msgflow` files

Learn how a BAR file is created based on your message flow design, the subflows that you created as `.msgflow` files, and the build options that you choose. Notice the differences between WebSphere Message Broker Version 8.0.0.0 and WebSphere Message Broker Version 8.0.0.1.

Packaging considerations for WebSphere Message Broker Version 8.0.0.0

When you build a BAR file in WebSphere Message Broker Toolkit, you must consider the following BAR builder behavior:

- When you add a message flow to a BAR file, the BAR builder will automatically include any required subflows.
- Subflows that are created as `.msgflow` files are not displayed in the BAR file as separate items, and are added automatically. To include these subflows, you have to add only the parent message flow.
- Message flows are included in the BAR file as compiled objects, that is, as `.cmf` files.

  Note: The BAR file creation option **Compile and in-line resources** is not available in WebSphere Message Broker Version 8.0.0.0, which means that a message flow is always added to a BAR file in its compiled form.
- For each message flow that you add to a BAR file, there will be one compiled message flow (CMF), which will include the message flow and any embedded subflows created as `.msgflow` files, and as many subflow source files as subflows are added to the message flow.
- When you add a subflow to a BAR file, the subflow is created in a Message Broker project, and the message flow where the subflow is being used is packaged in a different BAR file, the BAR builder behaviour depends on the subflow type:
  - If the subflow is created as a `.msgflow` file, it adds the subflow as a compiled object.

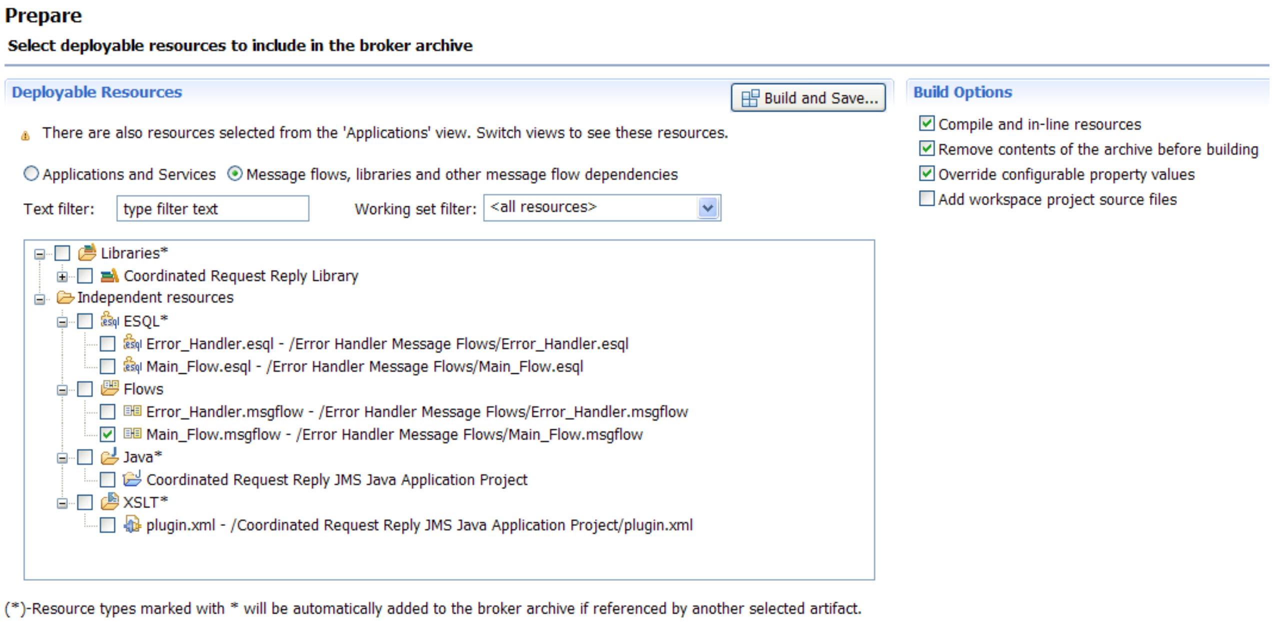
    Note: The subflow is only available for selection if it has an Input node, such as the MQInput node. Subflows that have only generic input nodes cannot be deployed independently.
  - If the subflow is created as a `.subflow` file, it adds the subflow source file.

Packaging considerations for WebSphere Message Broker Version 8.0.0.1 and later versions

When you build a BAR file in WebSphere Message broker toolkit, you must consider the following BAR builder behavior:

- When you add a message flow defined in a Message Broker project to a BAR file, the BAR builder will automatically include any required subflows. Subflows that are created as `.msgflow` files are not displayed in the BAR file as separate items, and are added automatically. To include these subflows, you have to add only the parent message flow.

  For example, in the following figure, the subflow **Error_handler.msgflow** is included in the message flow **Main_Flow.msgflow**.



Prepare

Select deployable resources to include in the broker archive

Deployable Resources Build and Save...

There are also resources selected from the 'Applications' view. Switch views to see these resources.

○ Applications and Services ● Message flows, libraries and other message flow dependencies

Text filter: type filter text Working set filter: <all resources>

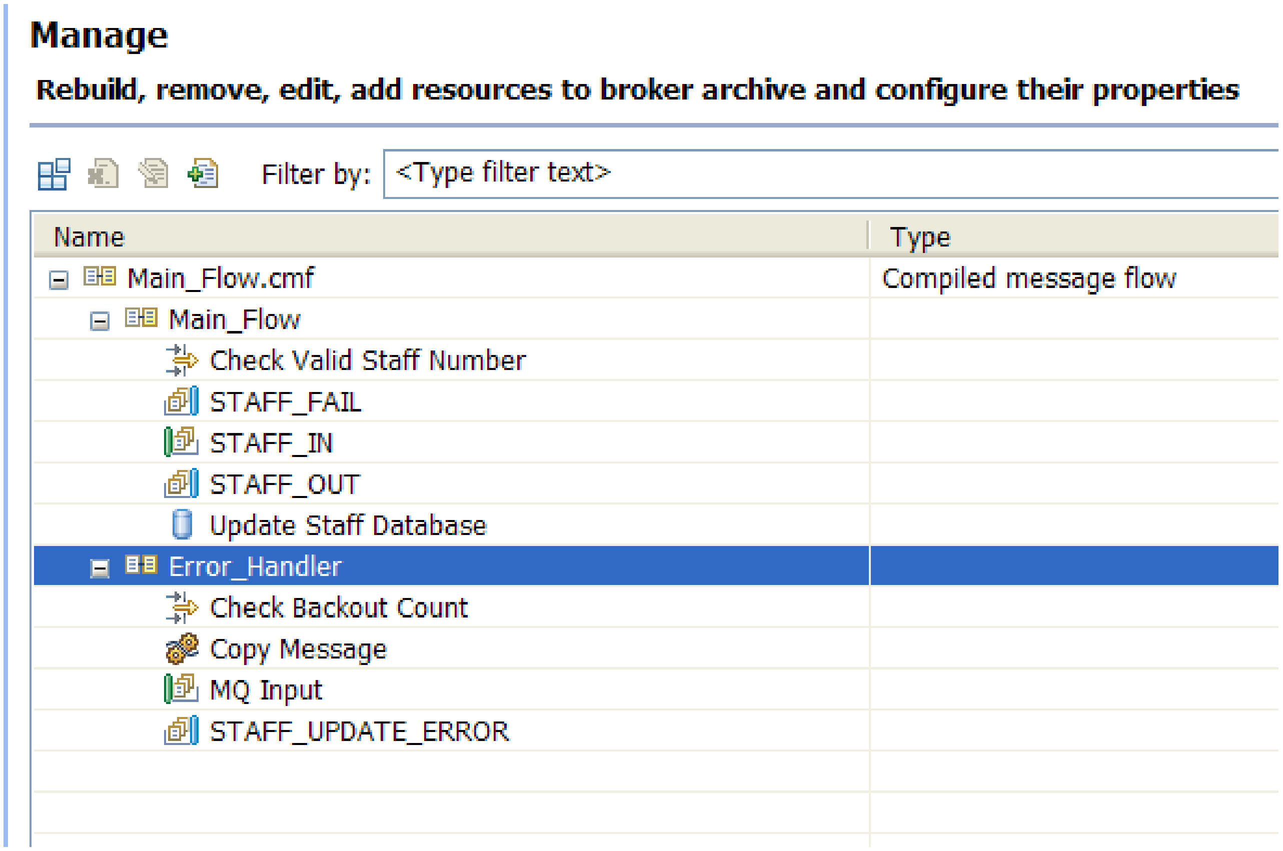
- ☐ Libraries*
  - ☐ Coordinated Request Reply Library
- Independent resources
  - ☐ ESQL*
    - ☐ Error_Handler.esql - /Error Handler Message Flows/Error_Handler.esql
    - ☐ Main_Flow.esql - /Error Handler Message Flows/Main_Flow.esql
  - ☐ Flows
    - ☐ Error_Handler.msgflow - /Error Handler Message Flows/Error_Handler.msgflow
    - ☑ Main_Flow.msgflow - /Error Handler Message Flows/Main_Flow.msgflow
  - ☐ Java*
    - ☐ Coordinated Request Reply JMS Java Application Project
  - ☐ XSLT*
    - ☐ plugin.xml - /Coordinated Request Reply JMS Java Application Project/plugin.xml

Build Options

- ☑ Compile and in-line resources
- ☑ Remove contents of the archive before building
- ☑ Override configurable property values
- ☐ Add workspace project source files

(*)-Resource types marked with * will be automatically added to the broker archive if referenced by another selected artifact.

When you select the message flow to include in the BAR file, the subflow does not get selected automatically. However, the subflow is included in the compiled message flow included in the BAR file.



Manage

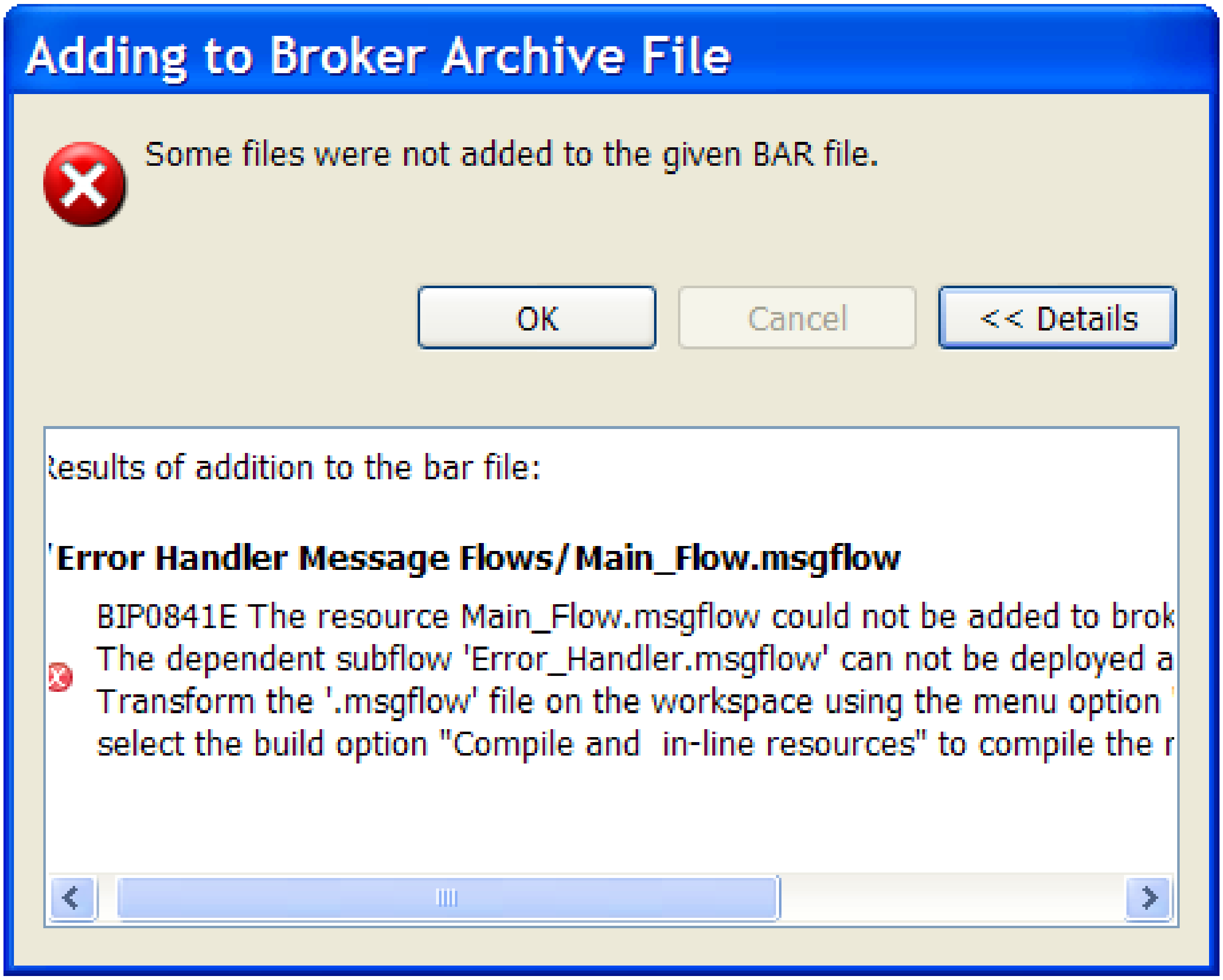
Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type
Main_Flow.cmf	Compiled message flow
Main_Flow	
Check Valid Staff Number	
STAFF_FAIL	
STAFF_IN	
STAFF_OUT	
Update Staff Database	
Error_Handler	
Check Backout Count	
Copy Message	
MQ Input	
STAFF_UPDATE_ERROR	

Note: Subflows created as `.msgflow` files that do not have an Input node such as MQinput node are not available for selection when you add resources to a BAR file.

- You can choose to select the BAR file creation option **Compile and in-line resources** so that a message flow is added to a BAR file in its compiled form. Message flows are included in the BAR file as compiled objects, that is, as `.cmf` files, and subflows created as `.msgflow` files are included inline in the compiled object.
- If you try to create a BAR file with the option **Compile and in-line resources** not selected, and the message flow has embedded subflows created as `.msgflow` files, you get the following error:



Some files were not added to the given BAR file.

Results of addition to the bar file:

'Error Handler Message Flows/Main_Flow.msgflow

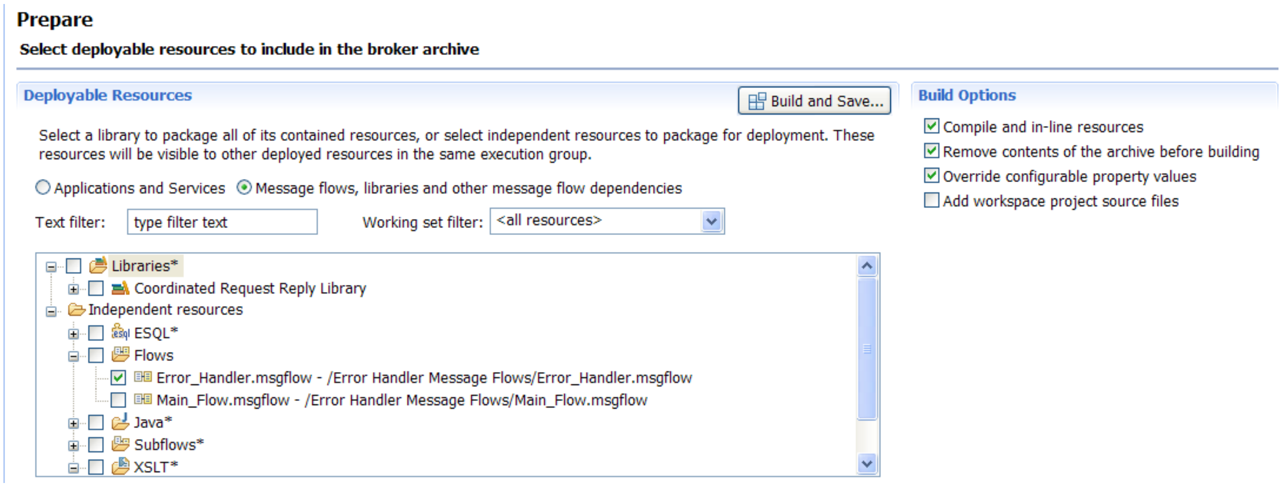
BIP0841E The resource Main_Flow.msgflow could not be added to brok
The dependent subflow 'Error_Handler.msgflow' can not be deployed a
Transform the '.msgflow' file on the workspace using the menu option '
select the build option "Compile and in-line resources" to compile the r

If you see this error message, open the BAR file properties and select the option **Compile and in-line resources**.

- When you add a subflow to a BAR file; the subflow is created in a Message Broker project; and the message flow where the subflow is being used is packaged in a different BAR file, the BAR builder behaviour depends on the subflow type:
  - If the subflow is created as a `.msgflow` file, WebSphere Message Broker adds the subflow as a compiled object if it has an Input node (such as the MQInput node).

    Note: Subflows that only have generic input nodes cannot be deployed independently.

For example, in the following figure, the **Error_Handler.msgflow** is a subflow created as a `.msgflow` file:



Prepare

Select deployable resources to include in the broker archive

Deployable Resources

Select a library to package all of its contained resources, or select independent resources to package for deployment. These resources will be visible to other deployed resources in the same execution group.

○ Applications and Services ● Message flows, libraries and other message flow dependencies

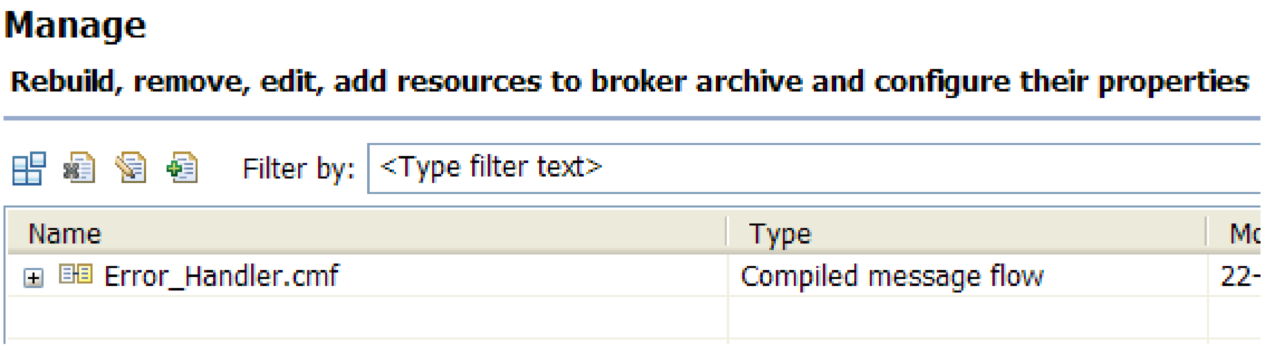
Text filter: type filter text Working set filter: <all resources>

- □ Libraries*
  - □ Coordinated Request Reply Library
- Independent resources
  - □ ESQL*
  - □ Flows
    - ☑ Error_Handler.msgflow - /Error Handler Message Flows/Error_Handler.msgflow
    - □ Main_Flow.msgflow - /Error Handler Message Flows/Main_Flow.msgflow
  - □ Java*
  - □ Subflows*
  - □ XSLT*

Build Options

☑ Compile and in-line resources
☑ Remove contents of the archive before building
☑ Override configurable property values
□ Add workspace project source files

When you create the BAR file and the option **Compile and in-line resources** is selected, the subflow is added to the BAR file as a compiled object.



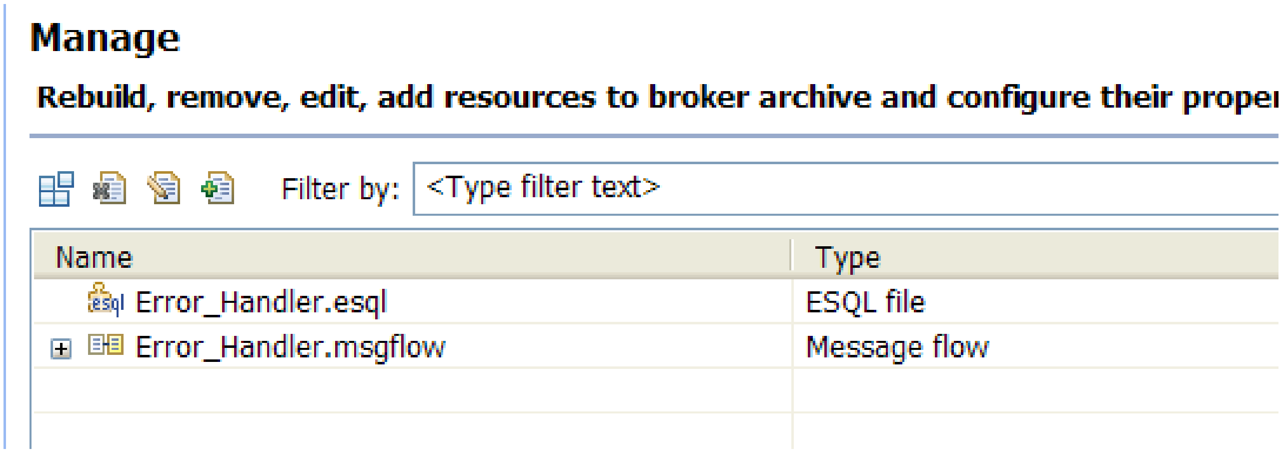
Manage

Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type	Mc
Error_Handler.cmf	Compiled message flow	22-

When you create the BAR file and the option **Compile and in-line resources** is not selected, the subflow is added to the BAR file as a source object.



Manage

Rebuild, remove, edit, add resources to broker archive and configure their prope

Filter by: <Type filter text>

Name	Type
Error_Handler.esql	ESQL file
Error_Handler.msgflow	Message flow

– If the subflow is created as a `.subflow` file, the subflow source file is added.

Related tasks:

"Creating a BAR file that includes a subflow created as a `.msgflow` file" on page 52
When deploying a subflow created as a `.msgflow` file, you need to understand how WebSphere Message Broker packages subflows into BAR files, and what are the configurations that you can create to deploy into the broker run time. Beware of the differences that exist between packaging subflows into a BAR file in WebSphere Message Broker Version 8.0.0.0 and WebSphere Message Broker Version 8.0.0.1.

Deploying a subflow created as a `.msgflow` file

You deploy a subflow created as a `.msgflow` file to an execution group by sending a broker archive (BAR) file to an execution group in a broker, which unpacks and stores the contents ready for when your message flows are started. Beware of the differences that exist deploying a subflow in WebSphere Message Broker Version 8.0.0.0, Version 8.0.0.1 and later versions.

About this task

From a deployment perspective, in WebSphere Message Broker, subflows created as a `.msgflow` file can be deployed in any of the following ways:

- As part of a message flow defined within a Message Broker project.
- As part of an application or a service.

Note: If you deploy a subflow created as a `.msgflow` file as an independent resource, you will encounter processing inconsistencies in the broker runtime environment. This subflow deployment configuration is not valid. The following information explains why it is not valid:

- When you deploy a subflow separately from any of the message flows defined within a Message Broker project that use this subflow, the subflow created as a `.msgflow` file must contain an Input node such as MQInput node. When your subflow includes an Input node and optionally an Output node, and you add this subflow into your message flow, you are effectively invoking a message flow.
- The subflow can be deployed directly into an execution group in a broker, or as part of a library. The subflow and the message flows that include this subflow must be deployed in the same execution group in a broker.
- You create a BAR file containing just a `.msgflow` file acting as a subflow with an Input node. However, you also must create a BAR file to deploy the message flow that includes the subflow. The BAR file that includes the message flow also includes a copy of the subflow.
- When you run your solution, you have active in the broker runtime environment two copies of the subflow that can pick up messages for processing by its Input node. Not all messages are processed equally.

Procedure

To deploy a subflow created as a `.msgflow` file, complete the following steps:

1. Decide whether you plan to deploy a subflow as part of an application, an integration service, or an Message Broker project.
2. Verify that you have a valid deployment configuration. For more information, see "Deployment configurations for subflows created as `.msgflow` files" on page 58.

3. Deploy the BAR file into the WebSphere Message Broker run time. For more information, see Deploying a broker archive file.

What to do next

After you have deployed a version of a subflow to the run time, you might want update the subflow. For more information, see “Updating a subflow” on page 31.

Deployment configurations for subflows created as `.msgflow` files

You have different deployment configurations, that is combinations of resources that are supported in WebSphere Message Broker run time, when you deploy applications, integration services, and independent resources into the runtime environment. Deployment configurations depend on the configuration choices you make when you build a BAR file and how you have designed your message flow and subflows.

Deployment configurations for independent resources in WebSphere Message Broker Version 8.0.0.0

The following table summarizes the different combinations of resources that are supported in Websphere Message Broker Version 8.0.0.0 when you deploy independent resources into the broker run time:

Table 12. Summary of supported deployment configurations for independent resources in WebSphere Message Broker Version 8.0.0.0

Configurations (These configurations are available for compatibility with previous releases of WebSphere Message Broker.)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Message flow (.msgflow)	x	x	x	x	x				
Subflow (.msgflow) with generic input node		x				x			
Subflow (.msgflow) with an Input node			x				x		
Subflow (.subflow) with generic input node				x				x	
Subflow (.subflow) with an Input node					x				x
Bar file resources	.cmf file	.cmf file	.cmf file	.cmf file and .subflow source file	.cmf file and .subflow source file			.subflow source file	.subflow source file

Table 12. Summary of supported deployment configurations for independent resources in WebSphere Message Broker Version 8.0.0.0 (continued)

Configurations (These configurations are available for compatibility with previous releases of WebSphere Message Broker.)	1	2	3	4	5	6	7	8	9
Configuration	valid	valid	valid	valid	valid	NOT valid	NOT valid	valid	valid

Deployment configurations for independent resources in WebSphere Message Broker Version 8.0.0.1 and later versions

The following table summarizes the different combinations of resources that are supported in Websphere Message Broker Version 8.0.0.1 and later versions when you deploy independent resources into the broker run time and the **Compile and in-line resources** option to build a BAR file is selected:

Table 13. Summary of supported deployment configurations for independent resources in WebSphere Message Broker Version 8.0.0.1 and later versions where the **Compile and in-line resources** *option to build a BAR file is selected*

Configurations when the option Compile and in-line resources is selected. (These configurations are available for compatibility with previous releases of WebSphere Message Broker.)	1	2	3	4	5	6	7	8	9
Message flow (.msgflow)	x	x	x	x	x				
Subflow (.msgflow) with generic input node		x				x			
Subflow (.msgflow) with an Input node			x				x		
Subflow (.subflow) with generic input node				x				x	
Subflow (.subflow) with an Input node					x				x
Bar file resources	1 .cmf file	1 .cmf file	1 .cmf file	1 .cmf file and 1 .subflow source file	1 .cmf file and 1 .subflow source file			1 .subflow source file	1 .subflow source file
Configuration	valid	valid	valid	valid, but it is not recommended	valid, but it is not recommended	NOT valid	NOT valid	valid	valid

The following table summarizes the different combinations of resources that are supported in Websphere Message Broker Version 8.0.0.1 and later versions when you deploy independent resources into the broker run time and the **Compile and in-line resources** option to build a BAR file is not selected:

Table 14. Summary of supported deployment configurations for independent resources in WebSphere Message Broker version 8.0.0.1 and later versions where the **Compile and in-line resources** *option to build a BAR file is not selected*

Configurations when the option Compile and in-line resources is not selected. (These configurations are available for compatibility with previous releases of WebSphere Message Broker.)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Message flow (.msgflow)	x	x	x	x	x				
Subflow (.msgflow) with generic input node		x				x			
Subflow (.msgflow) with an Input node			x				x		
Subflow (.subflow) with generic input node				x				x	
Subflow (.subflow) with an Input node					x				x
Bar file resources	1 .msgflow file			1 .msgflow file and 1 .subflow source file	1 .msgflow file and 1 .subflow source file			1 .subflow source file	1 .subflow source file
Configuration	valid	NOT valid	NOT valid	valid	valid	NOT valid	NOT valid	valid	valid

Related tasks:

"Deploying a subflow created as a .msgflow file" on page 57
You deploy a subflow created as a .msgflow file to an execution group by sending a broker archive (BAR) file to an execution group in a broker, which unpacks and stores the contents ready for when your message flows are started. Beware of the differences that exist deploying a subflow in WebSphere Message Broker Version 8.0.0.0, Version 8.0.0.1 and later versions.

Updating a subflow

When updating a subflow, you need to check the subflow version, and understand how you can deploy a new version of the subflow to the WebSphere Message Broker run time. For example, if a subflow is deployed as part of an application or a service, then you need to update your applications or services to include the new subflow version, and then redeploy the application or service.

Procedure

To update a subflow, complete the following steps:

1. Check the version of a subflow. For more information, see “Checking the Version of a subflow at run time” on page 22.
2. Update the value of the subflow version.

   You update the version of a subflow manually in the WebSphere Message Broker Toolkit. You can modify the **Label** property of the Passthrough node, the **Name** property of the Passthrough node, or the **Version** property of a subflow. For more information on how to define the version of a subflow, see “Versioning a subflow” on page 13.
3. Update the subflow implementation to include the new requirements, for example, add new nodes or modify the logic of a Compute node.
4. Update the BAR file and refresh its contents. For more information, see Refreshing the contents of a broker archive.
5. Deploy the BAR file into the WebSphere Message Broker run time. For more information, see Deploying a broker archive file.

   For subflows created as a `.subflow` file, you must consider the following behaviour when deploying a new version of a subflow:
   - If the subflow is deployed as part of an application, a service, or a library, then you need to update your applications, services, and libraries to include the new subflow version, and redeploy them.

     Note: If the application, service, or library has been deployed before, redeploying it removes the currently deployed application, service, or library and its contents, and replaces it with the deployed application, service, or library.
   - If the subflow is deployed separately from any of the message flows that use this subflow and you deploy a new version of the subflow, then all the message flows are updated automatically.

   For subflows created as a `.msgflow` file, you must consider the following behaviour when deploying a new version of a subflow:
   - You need to update your applications, services, and independent resources that use the subflow to include the new subflow version, and redeploy them.

     Note: Legacy subflows can only be updated at run time when the parent message flow is recompiled and deployed.

What to do next

Configure a message flow that includes subflows created as `.subflow` files. For more information, see “Configuring a message flow that includes subflows” on page 32.

Configuring a message flow that includes subflows

Depending on your subflow design, that is, the number of Input nodes that you have modeled, and how you configure the overall message flow BAR file deployment properties and the Input node deployment properties, WebSphere Message Broker handles threads at run time differently. This can impact your solution performance and your message parallel processing behavior at run time.

About this task

When you design a message flow or a subflow, you connect message flow nodes and subflows to define the integration solution logic. In a message flow, you use one or more Input nodes, such as the MQinput node, to model how a message is received by the message flow. If the first node in a message flow is a subflow, the subflow must contain one or more Input nodes. You can also have subflows in the middle of a message flow that include Input nodes.

When you design a subflow that is included in a message flow and you configure the BAR file that contains your integration solution, consider the following questions:

1. How many Input nodes, such as MQInput node, your message flow has? How many Input nodes your subflows have?

   Note: The number of Input nodes determines the default number of threads allocated by WebSphere Message Broker to process messages.
2. Have you calculated how many threads are required by WebSphere Message Broker to run your message flow?
3. Are you planning to start all instances at message flow start time? If yes, consider how many threads are allocated at start up and what other solutions are competing for resources in your system.
4. Do you have any order requirements when processing messages? If yes, consider that an Input node can receive messages out of order, and it will wait for the correct message to arrive before it continues processing any messages. By adding additional instances you can cause resource bottlenecks and not improve your solution performance.
5. Does your message flow design and subflows have multiple Input nodes? Do you need to receive messages through these Input nodes in parallel before continuing the message flow logic? Does one of them take longer to receive a message?

The answers to these questions will help you decide how you want to configure your resources.

To decide which option to choose to create additional instances, consider the following rules:

- In order to create instances which are shared between multiple Input nodes, you must set the **Additional instances** property at the message flow level.
- In order to create instances which are allocated to a specific Input node, you must set the **Additional instances** property at the Input node level.

In addition, you must consider the following behavior based on how you configure additional instances:

- When you set additional instances on an Input node, the additional instances or threads only run against that specific Input node. For example, if you have Input node A with 15 additional instances and Input node B with 0 additional instances, then if there is work waiting on Queue B, it will only be processed one message at a time. The spare instances belonging to Input node A cannot be used to process messages received from Input node B.
- The disadvantage of setting thread pools at message flow level is that one busy Input node could starve other input nodes.

You must decide how you want to set additional instances based on the priority of work on different Input nodes and your subflows and message flow design.

Procedure

When you design a message flow or a subflow that includes one or more Input nodes, you must consider the impact on resource allocation and performance:

- To learn how you can add additional instances at message flow level so that WebSphere Message Broker allocates additional threads to run messages at run time, see "Adding additional instances at message flow level" on page 34.
- To learn how you can add additional instances at Input node level so that WebSphere Message Broker allocates additional threads to run messages at run time, see "Adding additional instances at message flow node level" on page 36.

What to do next

Allocate additional instances at message flow level or at Input node level. For more information, see "Adding additional instances at message flow level" on page 34 and "Adding additional instances at message flow node level" on page 36.

Adding additional instances at message flow level

To change the number of threads that a message flow can use at run time to process messages, you can configure the **Additional instances** property of a message flow in a BAR file. These instances can be started as you need them or you can choose to start all of them at once when the message flow starts by selecting the **Start additional instances when the flow starts** property.

About this task

The default value of the **Additional instances** property is 0 threads. The maximum value of the **Additional instances** property is 256 threads.

The following conditions apply when WebSphere Message Broker allocates threads to a message flow based on your configuration options:

- By default, one thread is allocated per Input node. The Input node can be defined in the main message flow or within a subflow.
- You can add more instances or threads by setting the **Additional instances** property of the message flow that is included in the BAR file to a value greater than 0 and up to 256 threads.

  Note: If you set the value to a number greater than 256, the BAR file will fail to deploy.
- If you have more than one Input node, the additional instances that you configure can be used by any Input node that receives messages. An additional instance is allocated to an Input node that has its default thread busy processing a previous message and needs to process an incoming message.
- The number of threads available to process messages when you start a message flow is by default equal to the number of Input nodes. WebSphere Message Broker will start additional instances as required.
- If you want to start all of the threads configured at once, including the ones configured through the **Additional instances** property, you must select the **Start additional instances when the flow starts** property associated with the message flow.
- Additional message flow instances are configured at flow level. If you have a subflow included in a message flow, you cannot configure additional instances for the subflow independently. This condition applies whether the subflow is

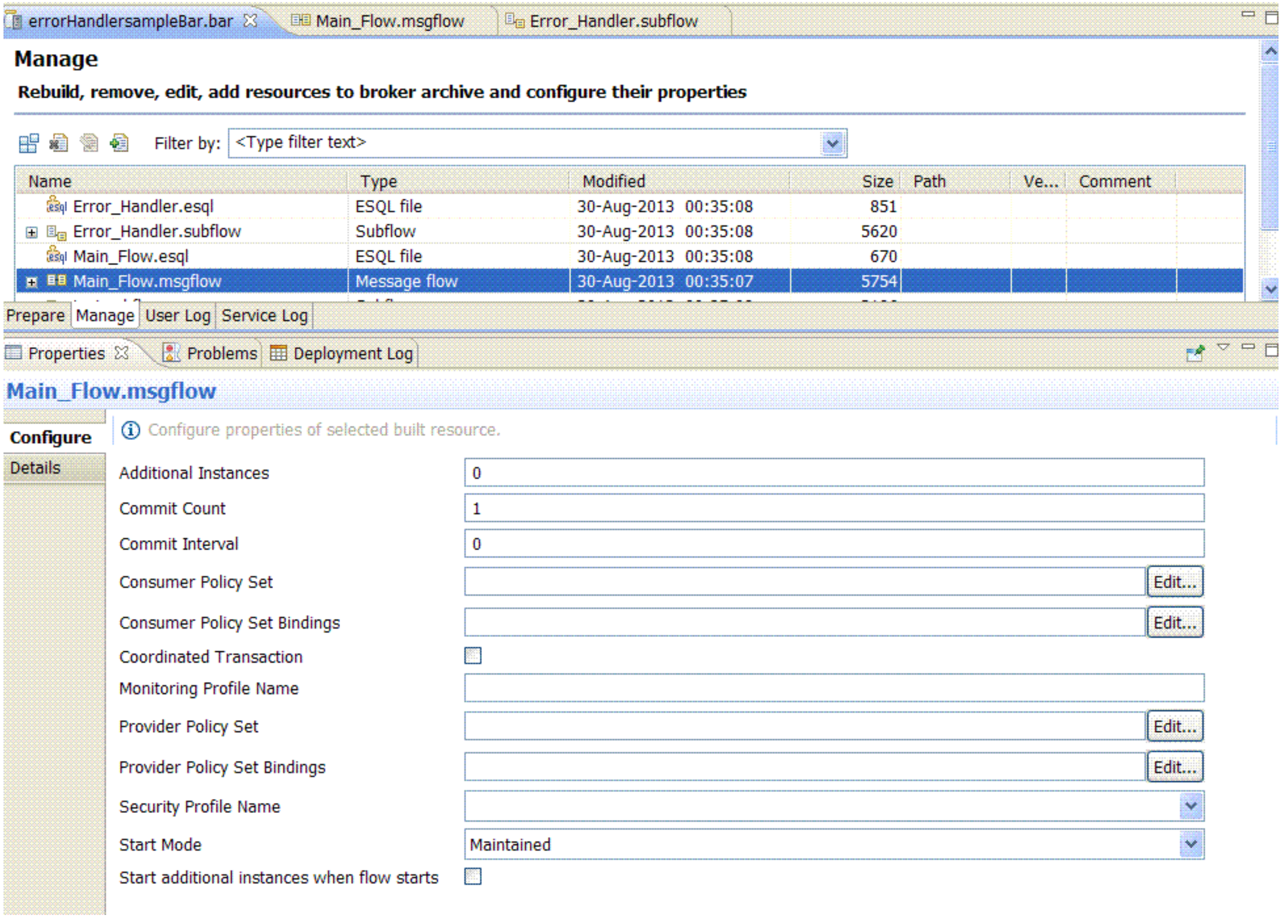
used as the starting node in a message flow, as a middle node in a message flow, or as the last node in a message flow.

Procedure

To configure the **Additional instances** property of a message flow in a BAR file, complete the following steps:

1. Open the BAR file that contains the message flow and subflows for which you want to set the **Additional instances** property. In the **Broker Development** view, identify the project that contains the BAR file, and double-click the file to open it.
2. Set the **Additional instances** property of a message flow that is included in the BAR file to a value greater than 0 and up to 256.

   The following figure outlines the configuration properties of a message flow:



Manage

Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type	Modified	Size	Path	Ve...	Comment
Error_Handler.esql	ESQL file	30-Aug-2013 00:35:08	851			
Error_Handler.subflow	Subflow	30-Aug-2013 00:35:08	5620			
Main_Flow.esql	ESQL file	30-Aug-2013 00:35:08	670			
Main_Flow.msgflow	Message flow	30-Aug-2013 00:35:07	5754			

Main_Flow.msgflow

Configure properties of selected built resource.

Additional Instances	0
Commit Count	1
Commit Interval	0
Consumer Policy Set	Edit...
Consumer Policy Set Bindings	Edit...
Coordinated Transaction	□
Monitoring Profile Name	
Provider Policy Set	Edit...
Provider Policy Set Bindings	Edit...
Security Profile Name	
Start Mode	Maintained
Start additional instances when flow starts	□

   Note: If you set the value to a number greater than 256, the BAR file will fail to deploy.

Results

Based on your message flow design and configuration, the following tables summarize the number of threads available when running the message flow in WebSphere Message Broker:

Table 15. Sample summary table of parallel threads available when running a message flow in a broker

Number of Input nodes in a message flow	Number of default instances or threads per message flow (1 thread per input node in a message flow)	Additional instances property (set at message flow level)	Maximum number of threads that can process messages in parallel
1	1	0	1
1	1	1	2
1	1	2	3
2	2	0	2
2	2	1	3
2	2	2	4
3	3	2	5

Table 16. Sample summary table of parallel threads available when running a message flow in a broker that contains a subflow with Input nodes

Number of Input nodes in a message flow (not included in a subflow)	Number of Input nodes modeled in a subflow that is included in a message flow	Number of default instances or threads per message flow (1 thread per input node in a message flow)	Additional instances property (set at message flow level)	Maximum number of threads that can process messages in parallel
0	1	1	0	1
1	1	2	0	2
0	1	1	1	2
1	1	2	1	3
0	1	1	2	3
1	1	2	2	4
0	2	2	0	2
2	2	4	2	6
0	2	2	1	3
0	2	2	2	4
0	3	3	2	5

What to do next

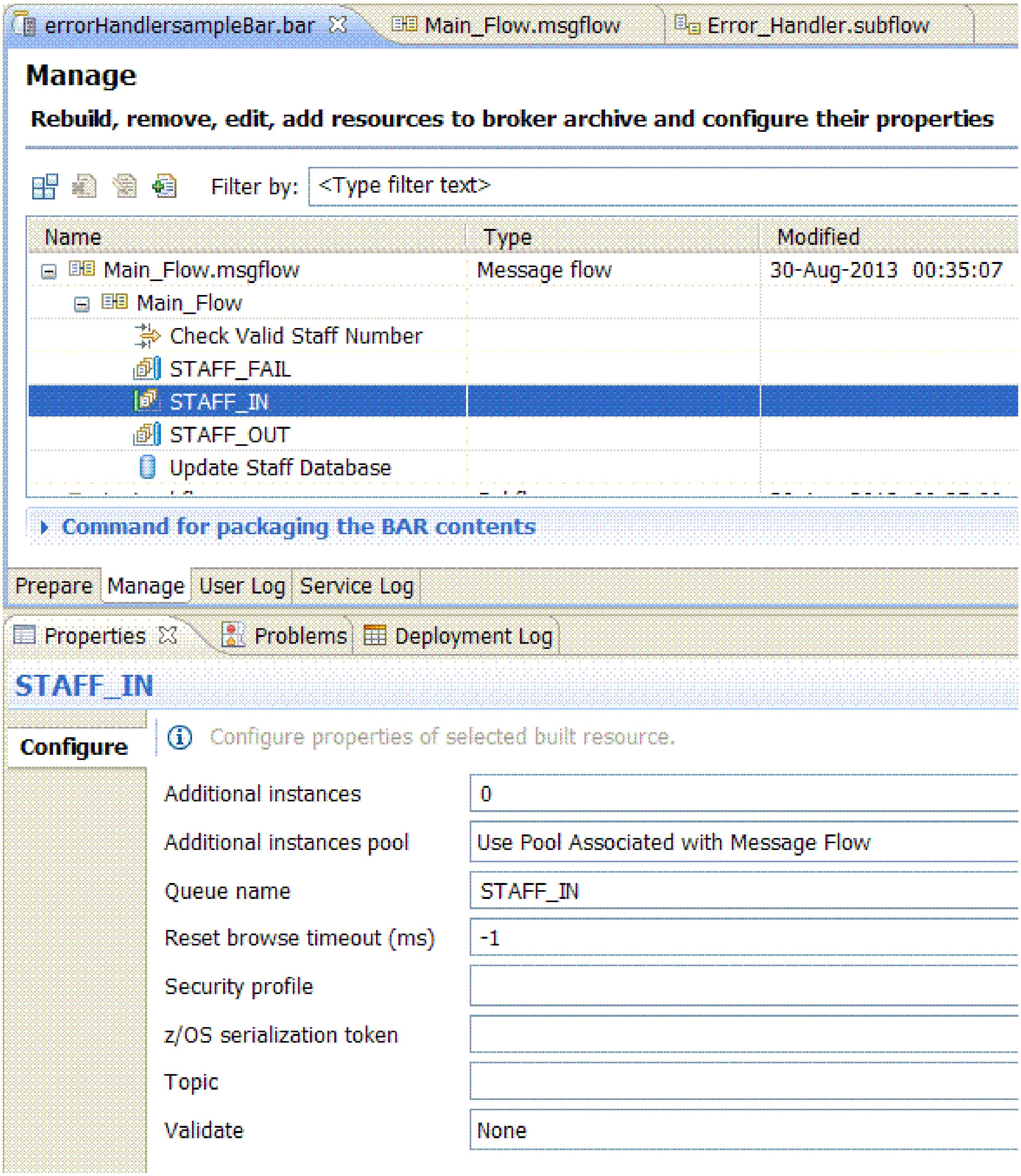
Return to “Configuring a message flow that includes subflows” on page 32.

Adding additional instances at message flow node level

To change the number of threads that a message flow node can use at run time to process messages, you can configure the **Additional instances** property of an Input node that is included in a message flow or subflow added to a BAR file.

About this task

The following figure outlines the configuration properties of a message flow node:



Manage

Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type	Modified
Main_Flow.msgflow	Message flow	30-Aug-2013 00:35:07
Main_Flow		
Check Valid Staff Number		
STAFF_FAIL		
STAFF_IN		
STAFF_OUT		
Update Staff Database		

Command for packaging the BAR contents

STAFF_IN

Configure properties of selected built resource.

Additional instances	0
Additional instances pool	Use Pool Associated with Message Flow
Queue name	STAFF_IN
Reset browse timeout (ms)	-1
Security profile	
z/OS serialization token	
Topic	
Validate	None

WebSphere Message Broker allocates threads to a message flow as follows:

- By default, one thread is allocated per Input node. The Input node can be defined in the main message flow or within a subflow.
- The default value of the **Additional instances** property is 0 threads. The maximum value of the **Additional instances** property is 256 threads.
- All instances defined in the **Additional instances** property for an Input node are started by WebSphere Message Broker as required.

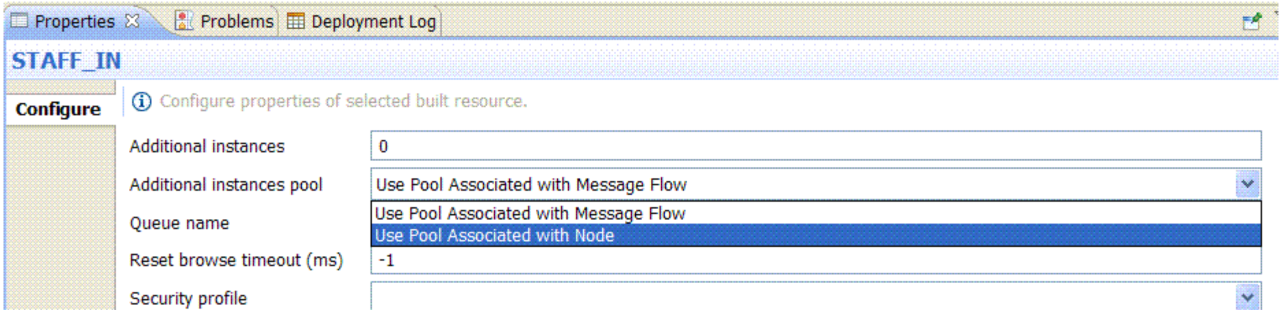
- The number of threads available to process messages when you start a message flow is by default equal to the number of Input nodes plus the number of additional instances configured for each Input node in the BAR file manage section.
- If you have a subflow included in a message flow, you can configure additional instances for each Input node modeled within the subflow. This condition applies whether the subflow is used as the starting node in a message flow, as a middle node in a message flow, or as the last node in a message flow.

Procedure

To add more threads to process messages by a specific Input node, you must complete the following steps:

1. Open the BAR file that contains the message flow and subflows for which you want to set the **Additional instances** property. In the **Broker Development** view, identify the project that contains the BAR file, and double-click the file to open it.
2. Set the **Additional instances** property of the Input node that is included in the BAR file to a value greater than 0 and up to 256.

   Note: If you set the value to a number greater than 256, the BAR file will fail to deploy.
3. Change the **Additional instances pool** to `Use Pool associated with Node`.



STAFF_IN

Configure

Configure properties of selected built resource.

Additional instances: 0

Additional instances pool: Use Pool Associated with Message Flow

Use Pool Associated with Message Flow

Use Pool Associated with Node

Queue name

Reset browse timeout (ms): -1

Security profile

Results

Based on your message flow design and configuration, the following tables summarize the number of threads available when running the message flow in WebSphere Message Broker:

Table 17. Sample summary table of parallel threads available when running a message flow in a broker

Number of Input nodes in a message flow or subflow	Number of default instances or threads per message flow (1 thread per input node in a message flow)	Additional instances property (set at input node level)	Maximum number of threads that can process messages in parallel
1	1	0	(1+0) = 1
1	1	1	(1+1) = 2
1	1	2	(1+2) = 3
2	2	0	(1+0) + (1+0) = 2
2	2	Additional instances = 1 for all Input nodes	(1+1) + (1+1) = 4

Table 17. Sample summary table of parallel threads available when running a message flow in a broker (continued)

Number of Input nodes in a message flow or subflow	Number of default instances or threads per message flow (1 thread per input node in a message flow)	Additional instances property (set at input node level)	Maximum number of threads that can process messages in parallel
3	3	Additional instances = 2 for all Input nodes	(1 +2) + (1 +2) + (1 +2) = 9
3	3	Additional instances = 1 for one Input node; Additional instances = 2 for another Input node; Additional instances = 3 for another Input node	(1+1) + (1+2) + (1 + 3) = 9

What to do next

Return to "Configuring a message flow that includes subflows" on page 32.

Implementing the solution: Converting legacy subflows

To convert legacy subflows into subflows created as `.subflow` files, you use the **Convert to subflow** function in WebSphere Message Broker.

About this task

The following conditions apply when you convert a subflow created as a `.msgflow` file into a subflow created as a `.subflow` file:

- If the subflow contains subflows that are defined as **`.msgflow`** files, you must convert these subflows to **`.subflow`** files.
- If the subflow is used by a message flow, you must update the parent message flow so that it references the new **`.subflow`** file.
- If the subflow contains other legacy resources such as the WebSphere Message Broker Version 7.0 Mapping node, you must convert these resources before you can run the **Convert to subflow** function.

Note: Subflows created as `.subflow` files are the recommended type of subflow in Websphere Message Broker. You should convert your subflows created as `.msgflows` files into `.subflow` files.

Procedure

Complete the following steps to convert a legacy subflow:

1. Learn how WebSphere Message Broker Version 8 handles your legacy subflows at deployment and run time. For more information, see "Implementing the solution: Creating a subflow as a `.msgflow` file" on page 39.
2. Learn how WebSphere Message Broker Version 8 handles subflows created as `.subflow` files at deployment and run time. For more information, see "Implementing the solution: Creating a subflow as a `.subflow` file" on page 10.
3. Identify other resources included in your subflow that require conversion:

- WebSphere Message Broker Version 7.0 Mapping nodes: You must convert a message map from a previous version of WebSphere Message Broker to a graphical data map. For more information, see Converting a message map from a .msgmap file to a .map file.
- User-defined nodes created from subflows in WebSphere Message Broker Version 7: If you have a user defined node that was created from a subflow, you cannot use it in a subflow created as a `.subflow` file. There is no automatic conversion for these nodes. You must implement a new subflow to replace the user defined node or convert the original subflow that you used to create a user-defined node. For more information, see Using a subflow as a user-defined node.
- MQOptimizedFlow nodes: These are deprecated nodes in WebSphere Message Broker Version 8.
- Nodes representing subflows that are created as `.msgflow` files: You must convert all the subflows included in your subflow.

4. Convert your legacy subflows created as `.msgflow` files to `.subflow` files. For more information, see "Converting subflows."
5. Update existing solutions to use the converted subflow instead of the legacy subflow.

   When you convert a subflow created as a `.msgflow` file into a `.subflow` file, all the subflow references that you have in parent message flows and other subflows to this subflow are automatically converted.

Results

Next steps:

- To learn how to convert a subflow that does not include other legacy resources, see "Converting subflows."
- To learn how to convert a subflow that does not include other legacy resources, see "Converting subflows that include other legacy resources" on page 72.

Converting subflows

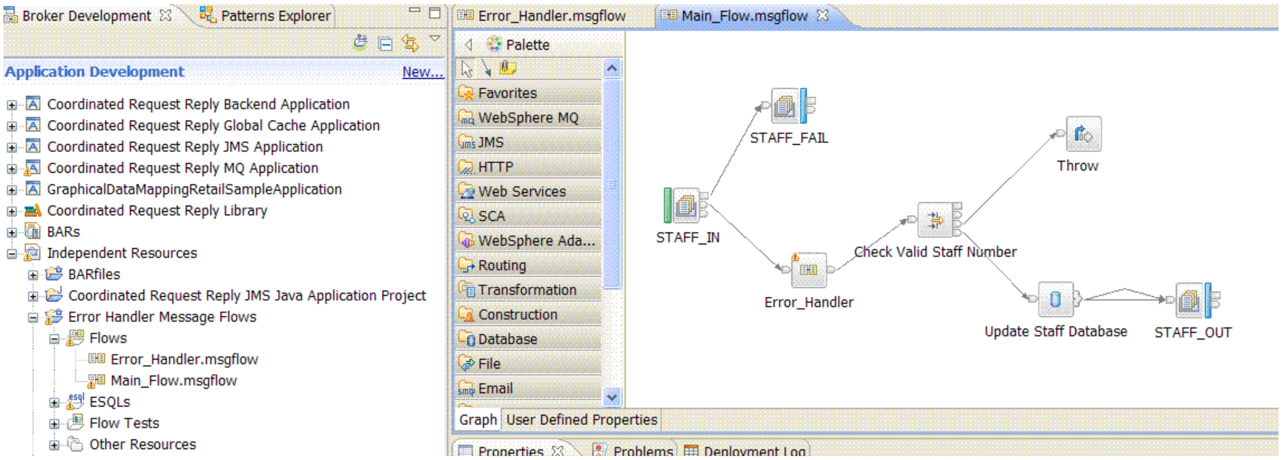
Use the Error Handler sample to convert a subflow created as a **`.msgflow`** file to a **`.subflow`** file by using the **Convert to subflow** function in WebSphere Message Broker.

Before you begin

Set up the initial configuration. See "Creating the initial configuration for the Error Handler sample" on page 7.

About this task

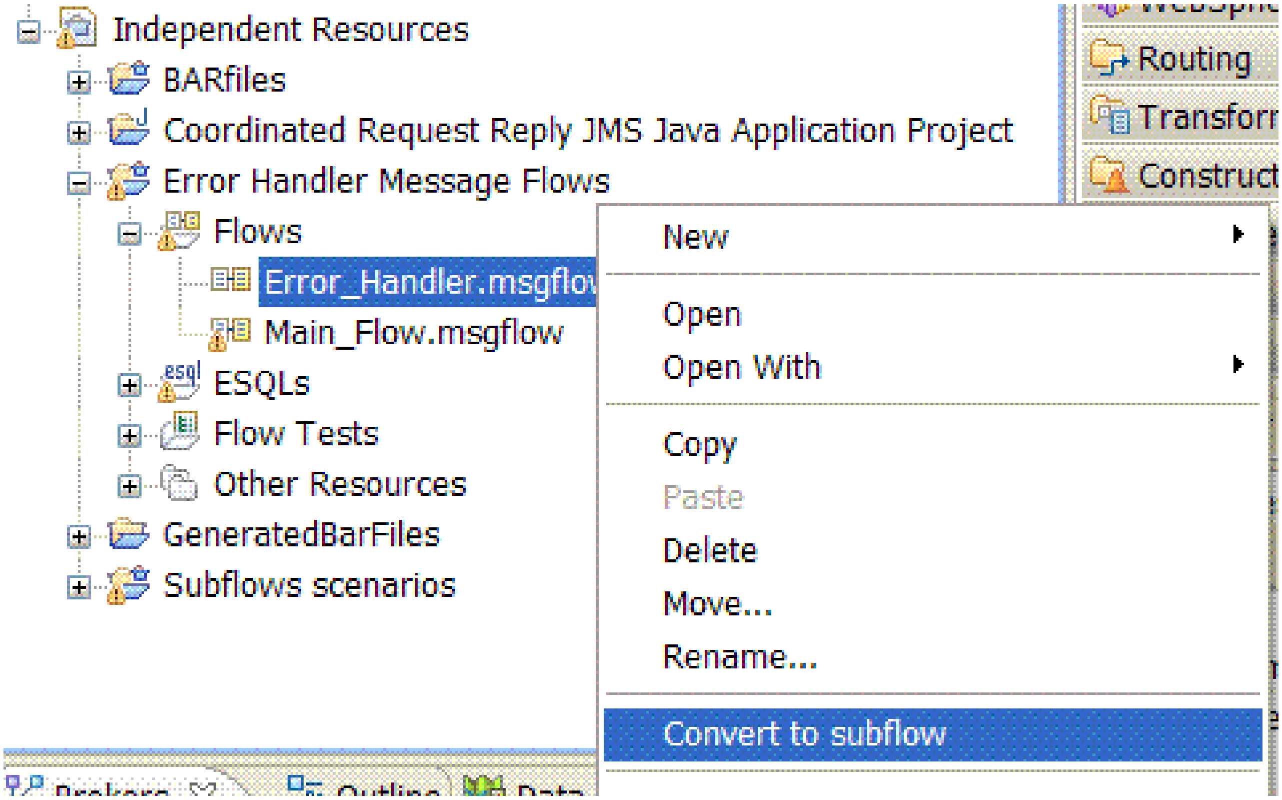
The Error Handler sample has been migrated from WebSphere Message Broker Version 7. It includes a legacy subflow that needs to be converted to a subflow created as a `.subflow` file. This subflow does not contain additional legacy resources that might require conversion before you can convert it. The following figure illustrates the starting point before conversion where your starting point is a message flow named **Main_Flow.msgflow** and a subflow named **Error_Handler.msgflow**:



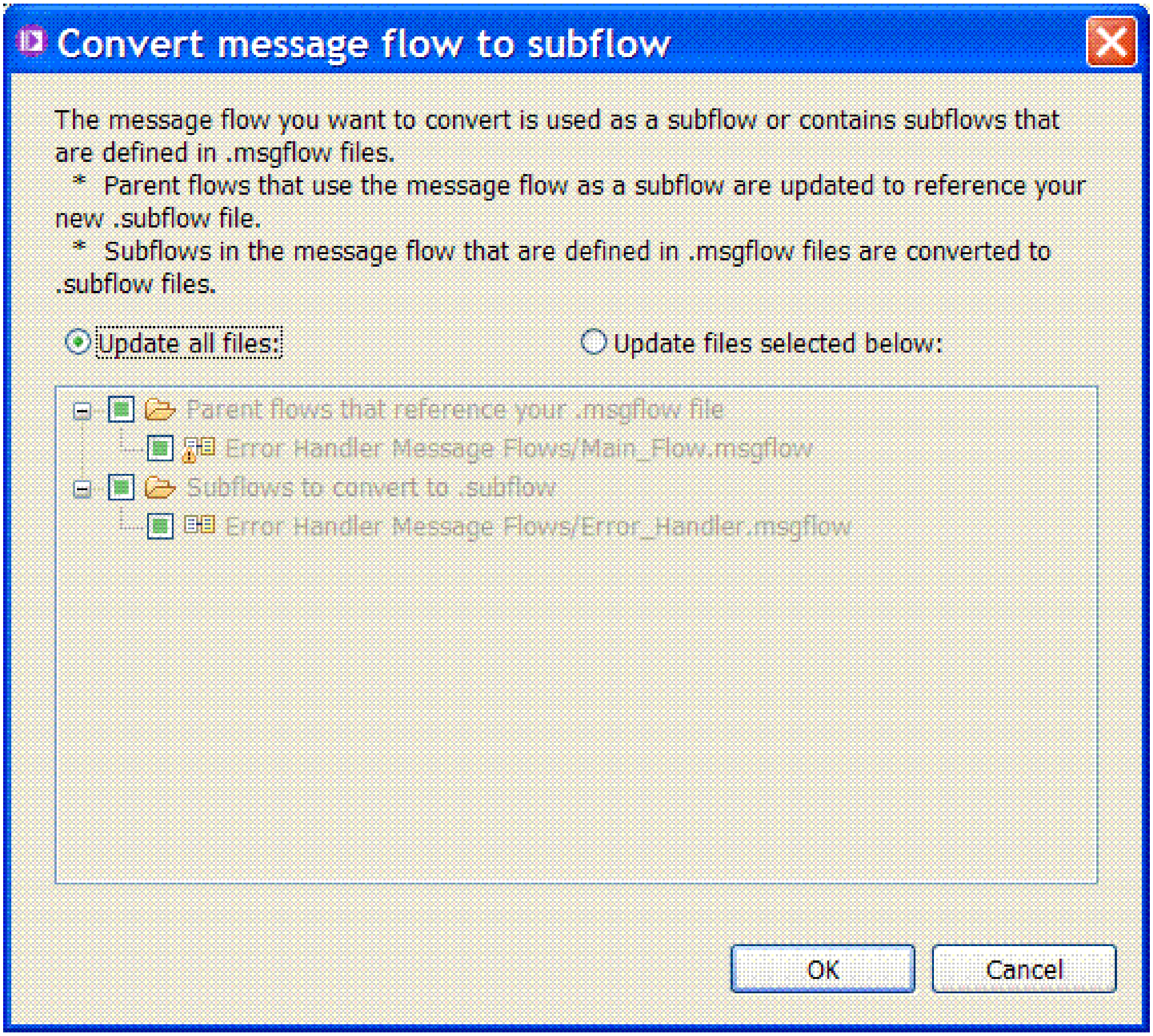
Procedure

You must complete the following steps to convert the **Error_Handler.msgflow** subflow into a subflow created as a `.subflow` file:

1. In the **Broker Development** view, identify the Message Broker project located under the **Independendent resources** folder that contains the subflow that you want to convert. Select the subflow, in our example, **Error_Handler.msgflow**. Right-click the subflow and select **Convert to subflow**.



2. In the Convert message flow to subflow window, select **Update all files**, and click **Ok**. The conversion starts.



The message flow you want to convert is used as a subflow or contains subflows that are defined in .msgflow files.

* Parent flows that use the message flow as a subflow are updated to reference your new .subflow file.
* Subflows in the message flow that are defined in .msgflow files are converted to .subflow files.

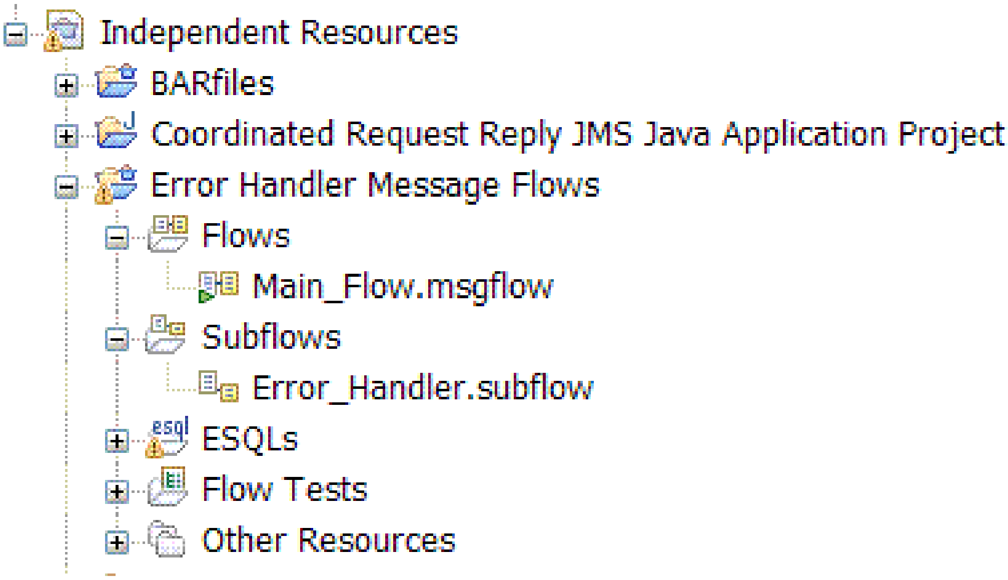
● Update all files: ○ Update files selected below:

- ☑ Parent flows that reference your .msgflow file
  - ☑ Error Handler Message Flows/Main_Flow.msgflow
- ☑ Subflows to convert to .subflow
  - ☑ Error Handler Message Flows/Error_Handler.msgflow

What to do next

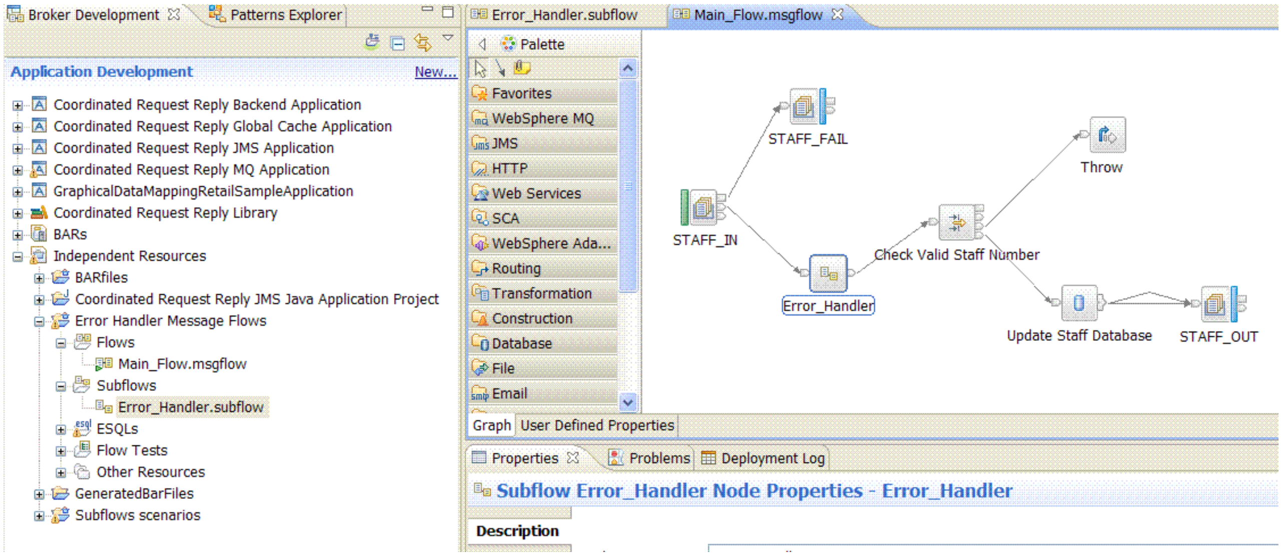
You must verify that the converted subflow is created and available:

1. After the conversion completes, verify that the initial subflow disappears from the **Flows** folder, and the converted subflow appears under the **Subflows** folder.



- Independent Resources
  - BARfiles
  - Coordinated Request Reply JMS Java Application Project
  - Error Handler Message Flows
    - Flows
      - Main_Flow.msgflow
    - Subflows
      - Error_Handler.subflow
    - ESQLs
    - Flow Tests
    - Other Resources

2. Verify that the message flow includes the subflow as a node.



Converting subflows that include other legacy resources

You can convert a subflow created as a **.msgflow** file to a **.subflow** file by using the **Convert to subflow** function in WebSphere Message Broker. If the subflow includes other legacy resources, you must convert them before you can convert your subflow.

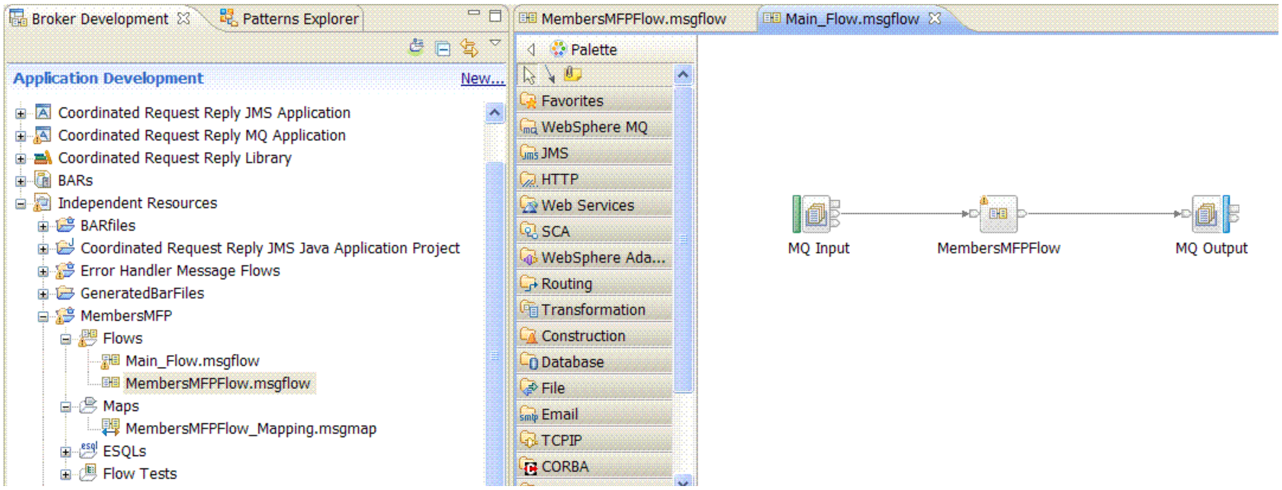
Before you begin

Set up the initial configuration "Creating the initial configuration for the MembersMFP sample" on page 9.

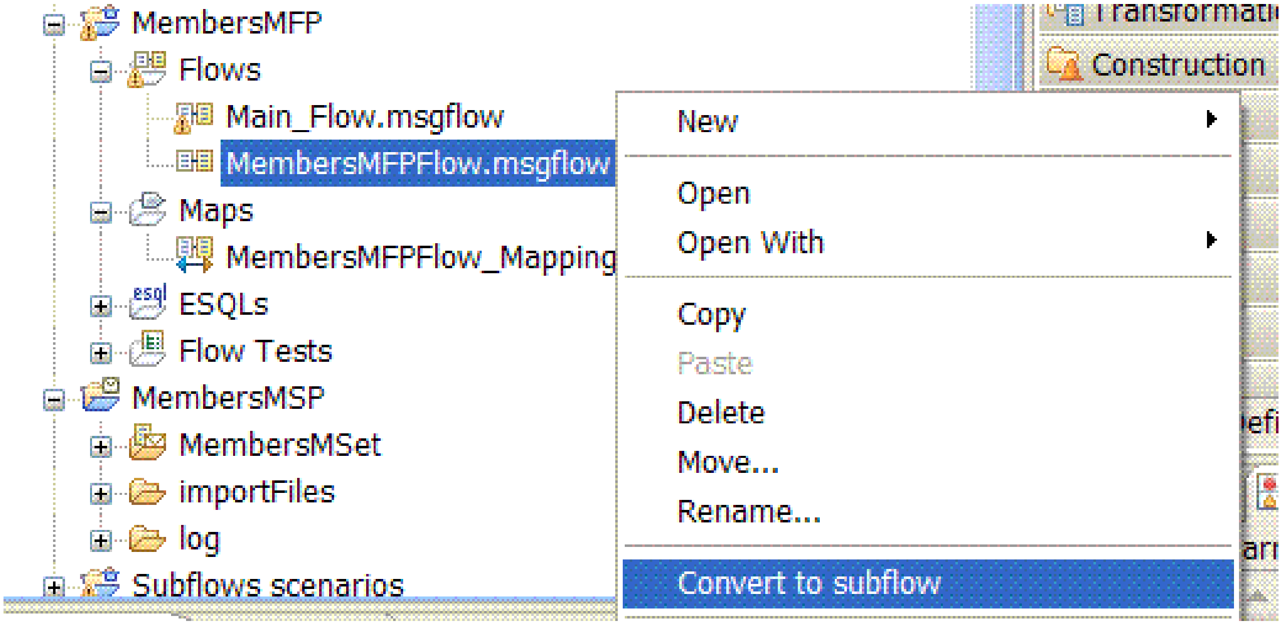
Procedure

You must complete the following steps to convert the **MembersMFPFlow.msgflow** subflow sample into a subflow created as a `.subflow` file:

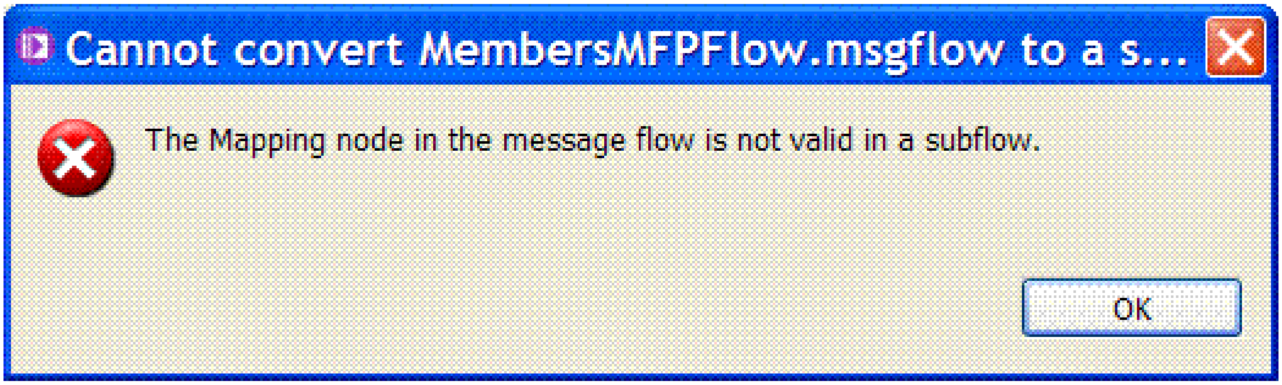
1. In the **Broker Development** view, identify the Message Broker project located under the **Independent resources** folder that contains the subflow that you want to convert. Select the subflow, in our example, **MembersMFPFlow.msgflow**.



2. Right-click the subflow and select **Convert to subflow**.



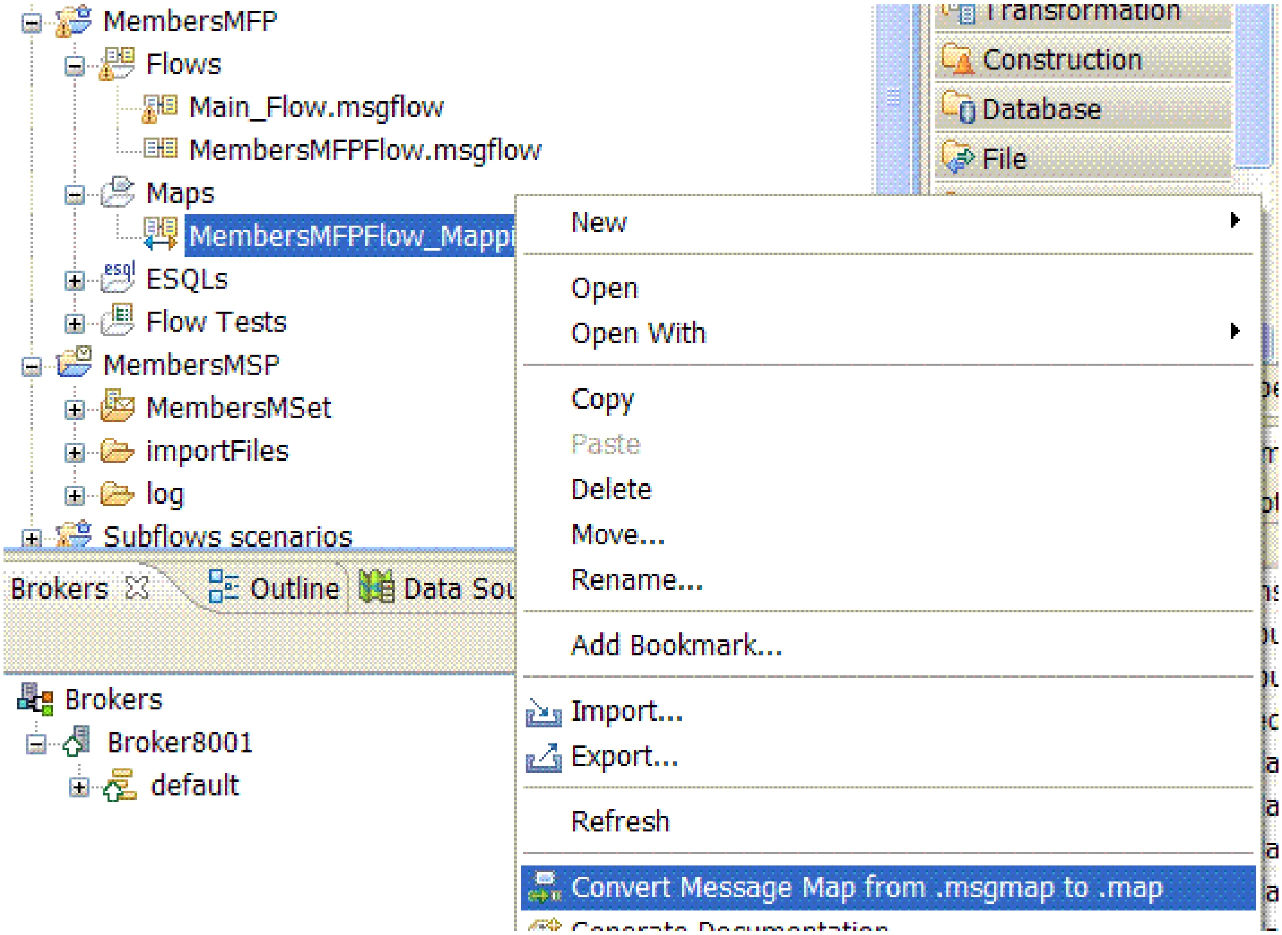
3. An error window opens that tells you that you cannot convert the subflow. The subflow contains a legacy Mapping node that you must convert before you can select **Convert to subflow**.



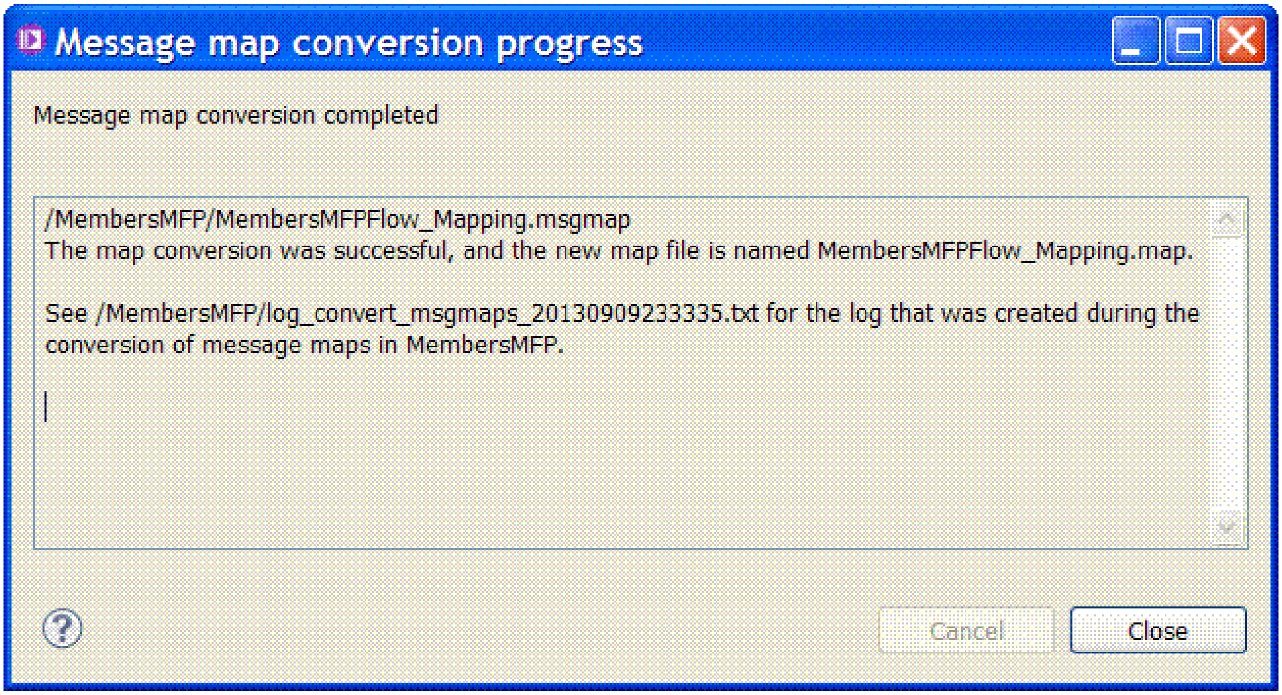
Cannot convert MembersMFPFlow.msgflow to a s...

The Mapping node in the message flow is not valid in a subflow.

4. In the **Broker Development** view, identify the Message Broker project located under the **Independent resources** folder that contains the subflow that you want to convert. Identify the map that needs to be converted. In our example, it is the **MembersMFPFlow_Mapping.msgmap**. Select **Convert Mesage Map from .msgmap to .map**.



5. The Message map conversion progress window opens and details the results from converting your map. Click **Close**.

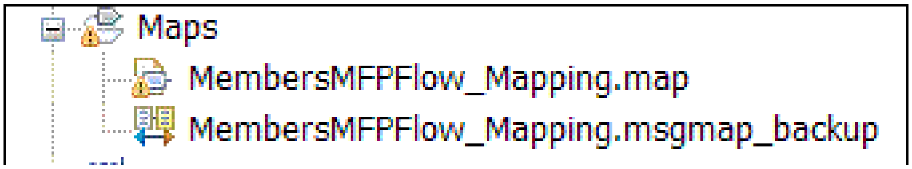


Message map conversion completed

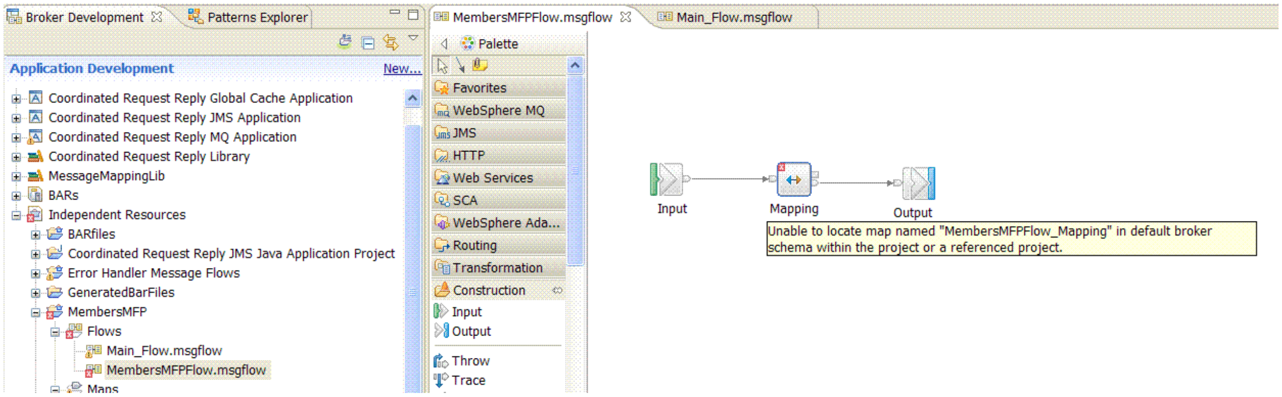
```
/MembersMFP/MembersMFPFlow_Mapping.msgmap
The map conversion was successful, and the new map file is named MembersMFPFlow_Mapping.map.

See /MembersMFP/log_convert_msgmaps_20130909233335.txt for the log that was created during the
conversion of message maps in MembersMFP.
```

6. After you click **Close**, the **Convert Mesage Map from .msgmap to .map** function completes. You can see your converted map in the **Broker Development** view, in the **Maps** folder under your Message Broker project. The converted map has a `.map` file extension.

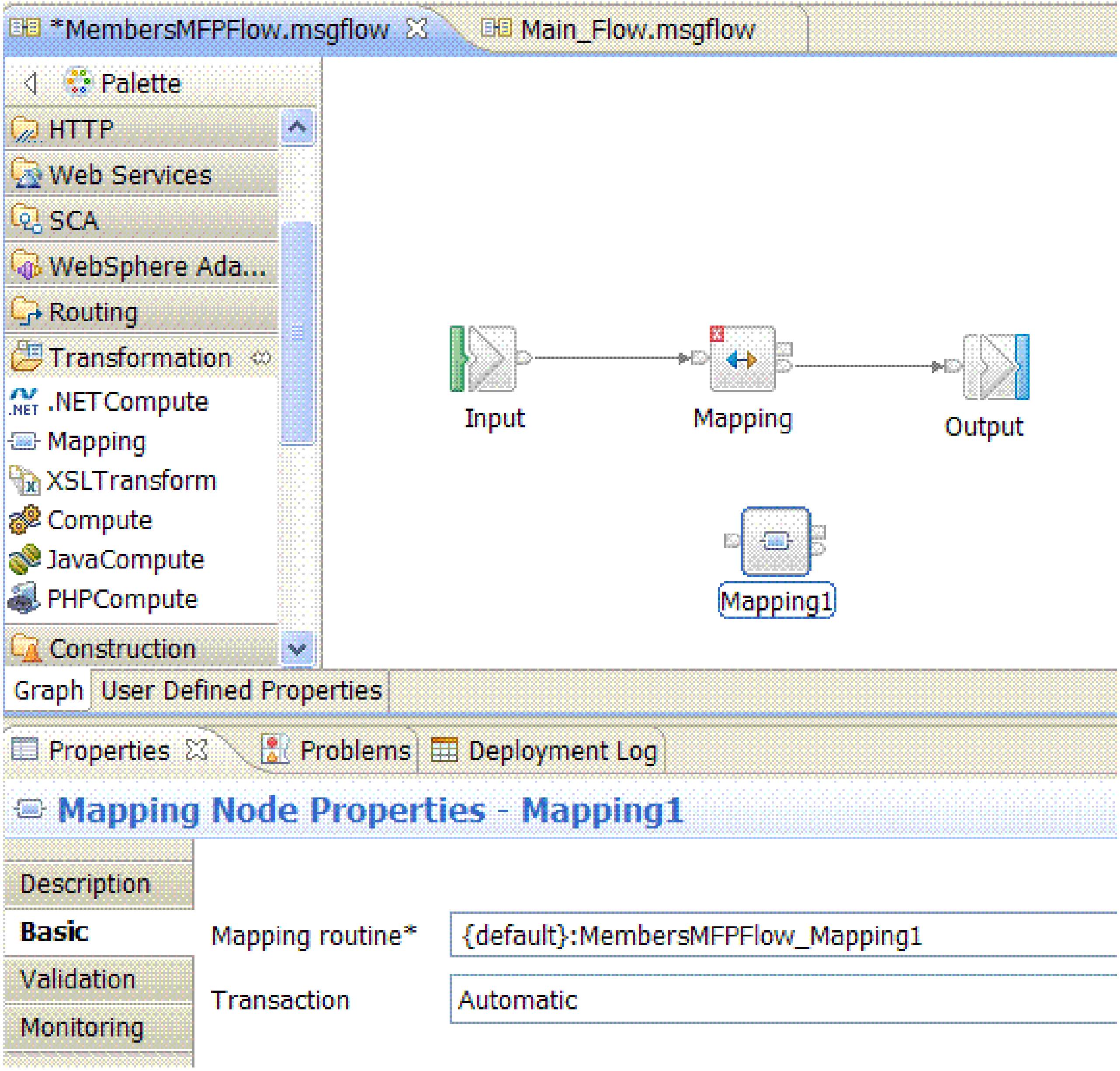

Maps
MembersMFPFlow_Mapping.map
MembersMFPFlow_Mapping.msgmap_backup

7. After you convert the map, you need to fix the following errors:
   - Errors related with transformation functions that require manual conversion steps to finalize its conversion. For more information, see Scenario: Converting a message map to a graphical data map.
   - The error related to the Mapping node that had the legacy map configured. The node is a legacy node, and needs to be replaced by the WebSphere Message Broker Version 8 Mapping node.



Unable to locate map named "MembersMFPFlow_Mapping" in default broker schema within the project or a referenced project.

8. To replaced the legacy Mapping node with the WebSphere Message Broker Version 8 Mapping node, complete the following steps:
   - Identify the new Mapping node in the Palette.
   - Drag and drop the Mapping node onto the message flow canvas.

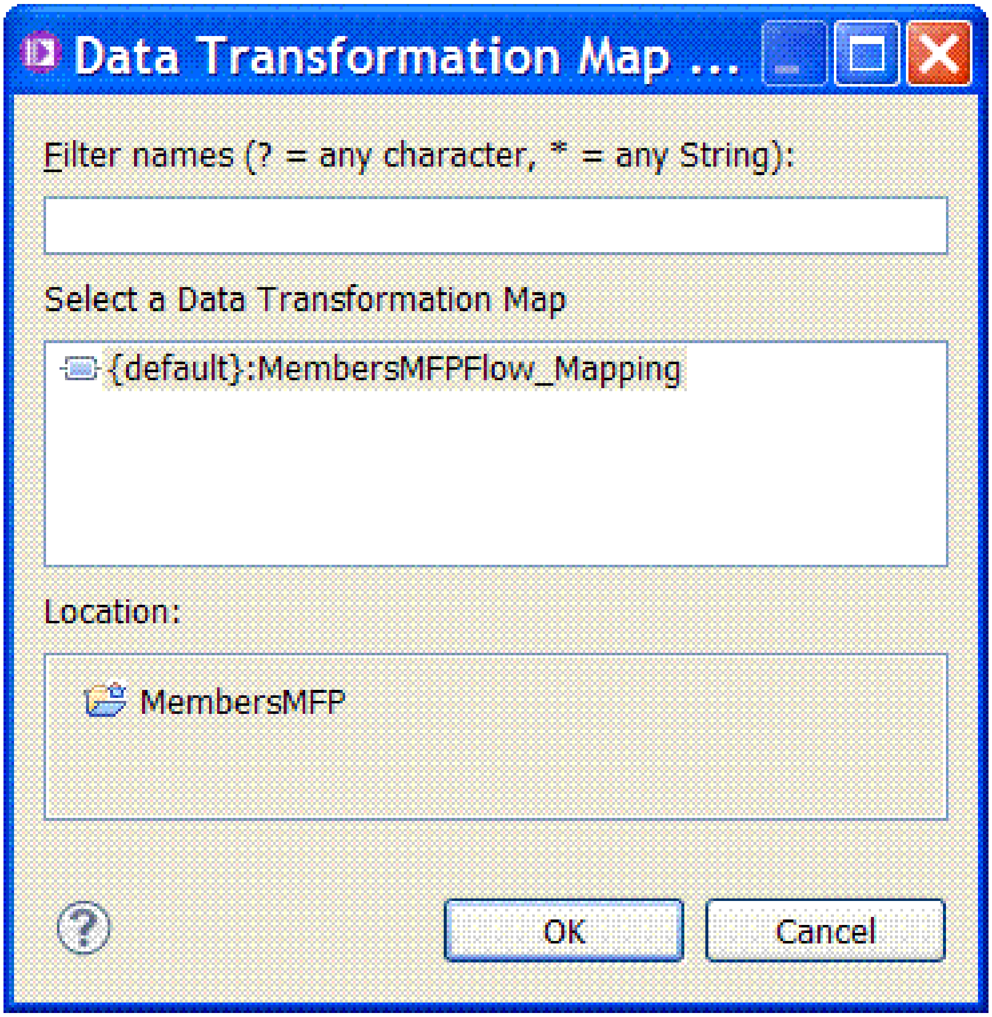


Mapping Node Properties - Mapping1

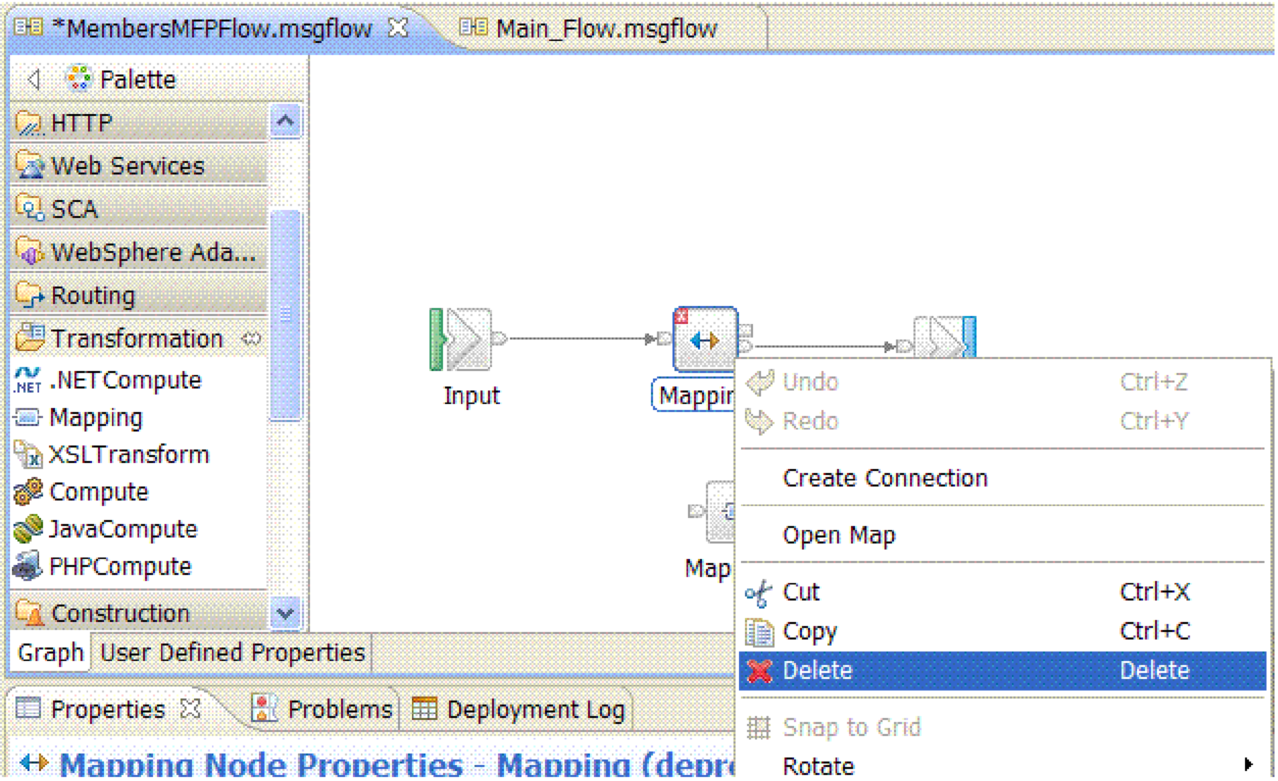
Mapping routine* {default}:MembersMFPFlow_Mapping1

Transaction Automatic

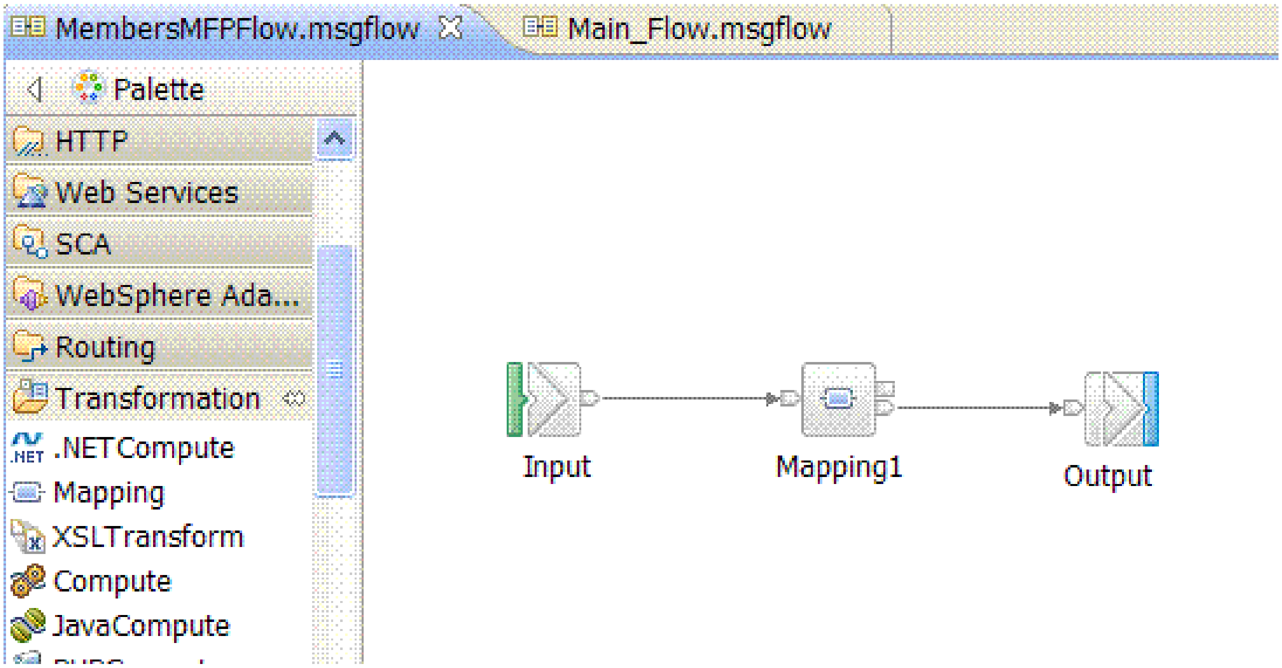
- Select the map that you want to associate to the Mapping node. This is the map you just finished converting.


Data Transformation Map ...
Filter names (? = any character, * = any String):
Select a Data Transformation Map
{default}:MembersMFPFlow_Mapping
Location:
MembersMFP
OK
Cancel

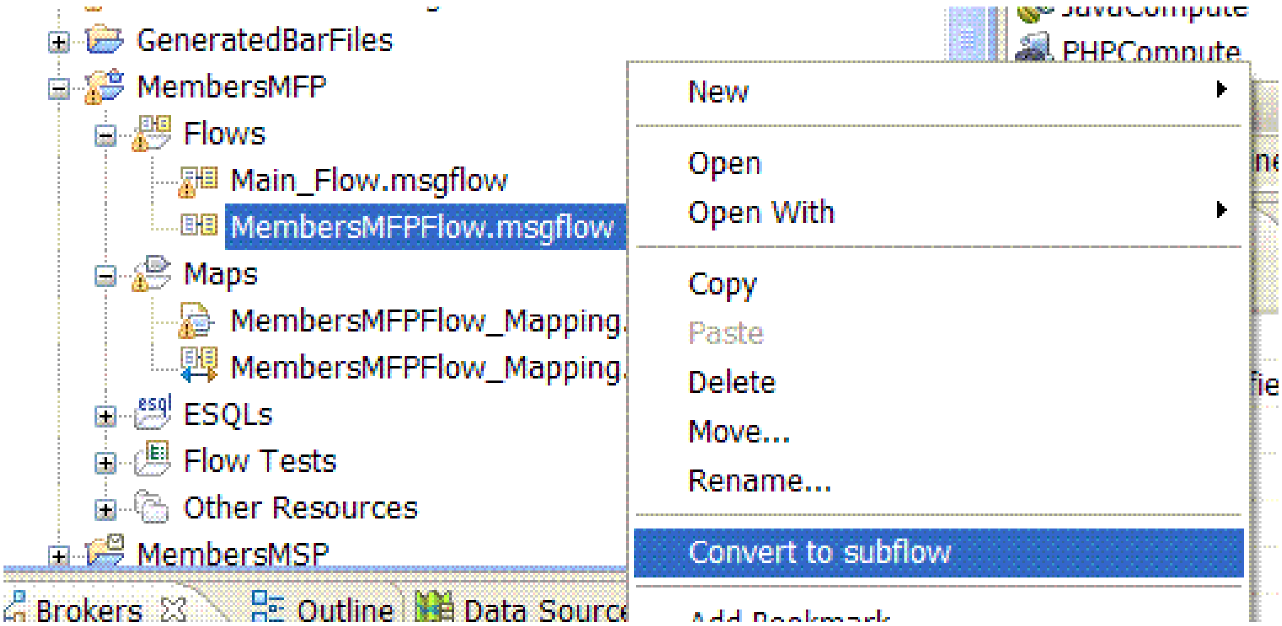
- Delete the old Mapping node.



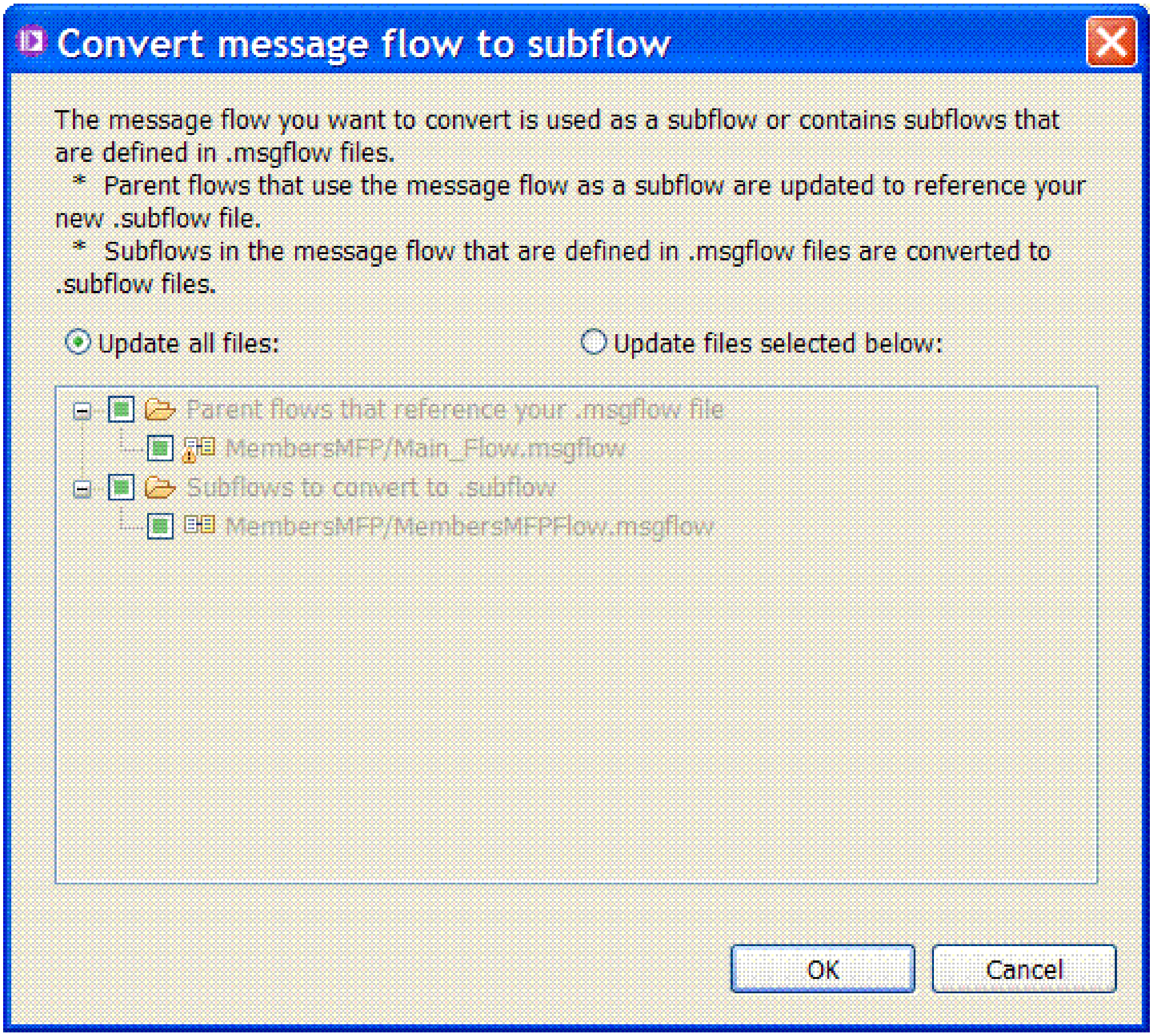
- Connect the new Mapping node where you just removed the old Mapping node.



9. After you have converted your legacy map and replaced the old Mapping node with the new one, select the **MembersMFPFlow.msgflow** subflow. Right-click the subflow and select **Convert to subflow**.



10. In the Convert message flow to subflow window, select **Update all files**, and click **Ok**. The conversion starts.



Convert message flow to subflow

The message flow you want to convert is used as a subflow or contains subflows that are defined in .msgflow files.

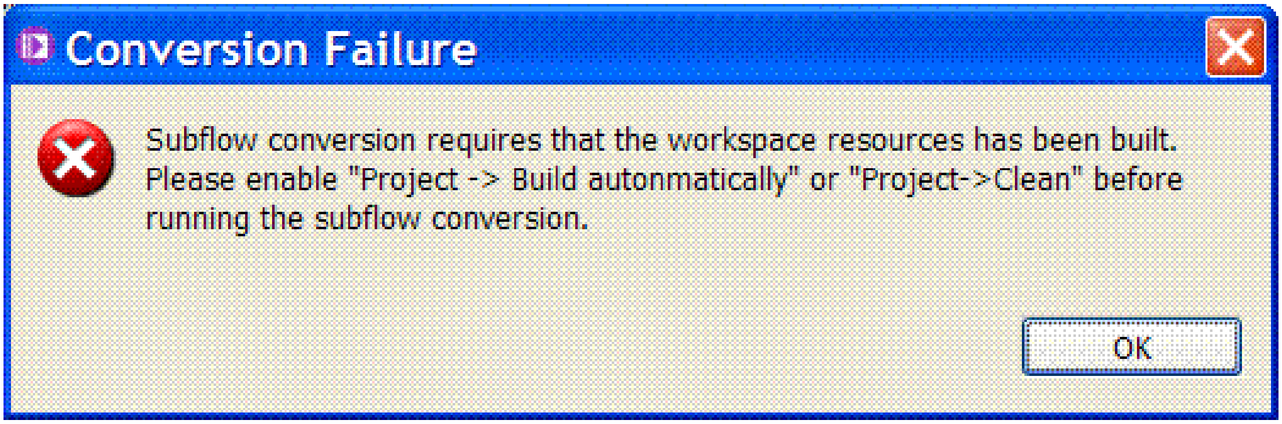
* Parent flows that use the message flow as a subflow are updated to reference your new .subflow file.

* Subflows in the message flow that are defined in .msgflow files are converted to .subflow files.

● Update all files: ○ Update files selected below:

- ☑ Parent flows that reference your .msgflow file
  - ☑ MembersMFP/Main_Flow.msgflow
- ☑ Subflows to convert to .subflow
  - ☑ MembersMFP/MembersMFPFlow.msgflow

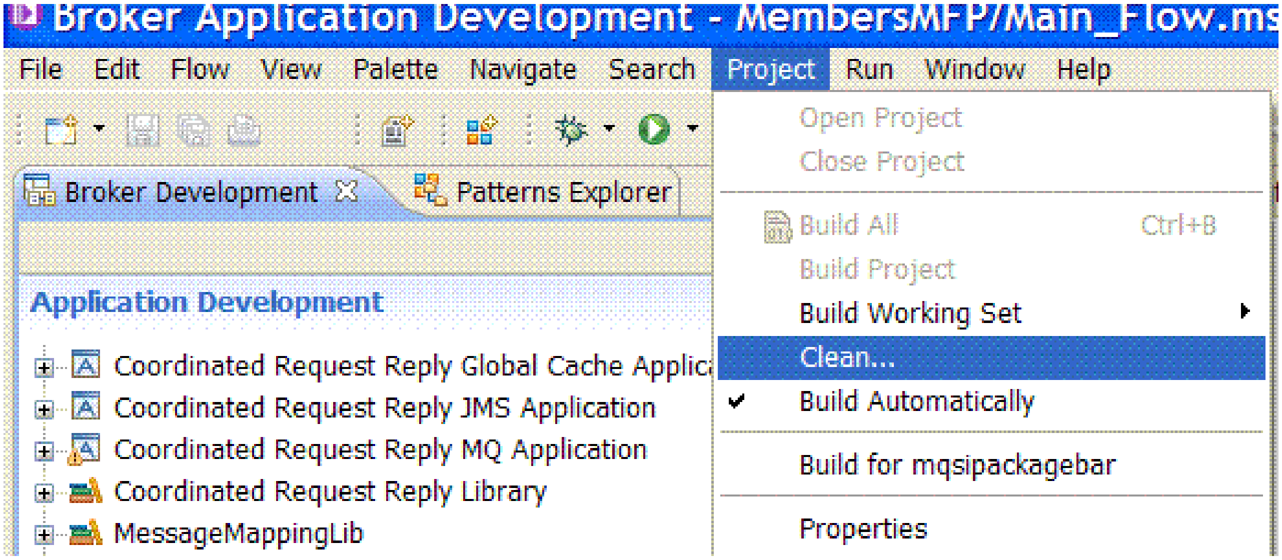
11. Optional: If the Conversion Failure window opens, click **Ok**.



Conversion Failure

Subflow conversion requires that the workspace resources has been built. Please enable "Project -> Build autonmatically" or "Project->Clean" before running the subflow conversion.

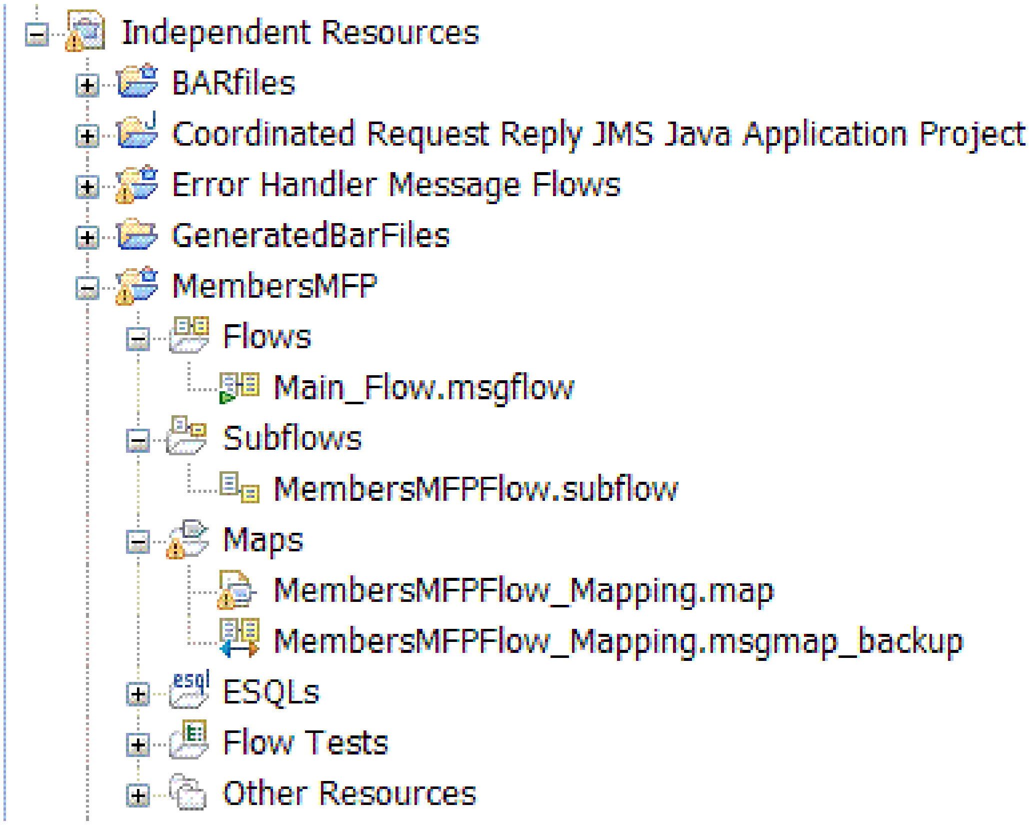
Then, select **Project** > **Clean**.


Broker Application Development - MembersMFP/Main_Flow.ms
File Edit Flow View Palette Navigate Search Project Run Window Help
Broker Development
Patterns Explorer
Application Development
Coordinated Request Reply Global Cache Applic
Coordinated Request Reply JMS Application
Coordinated Request Reply MQ Application
Coordinated Request Reply Library
MessageMappingLib
Open Project
Close Project
Build All Ctrl+B
Build Project
Build Working Set
Clean...
Build Automatically
Build for mqsipackagebar
Properties

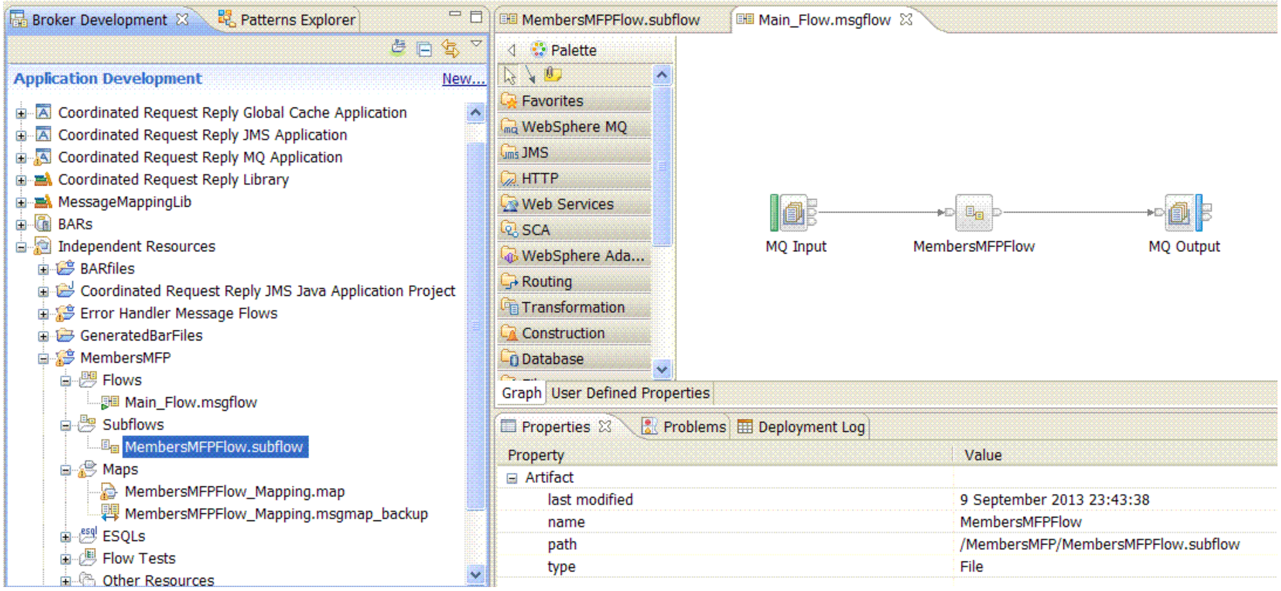
What to do next

You must verify that the converted subflow is created and available:

1. After the conversion completes, verify that the initial subflow disappears from the **Flows** folder, and the converted subflow appears under the **Subflows** folder.


Independent Resources
BARfiles
Coordinated Request Reply JMS Java Application Project
Error Handler Message Flows
GeneratedBarFiles
MembersMFP
Flows
Main_Flow.msgflow
Subflows
MembersMFPFlow.subflow
Maps
MembersMFPFlow_Mapping.map
MembersMFPFlow_Mapping.msgmap_backup
ESQLs
Flow Tests
Other Resources

2. Verify that the message flow includes the subflow as a node.



Property	Value
Artifact	
last modified	9 September 2013 23:43:38
name	MembersMFPFlow
path	/MembersMFP/MembersMFPFlow.subflow
type	File

Related information:

Scenario: Converting a message map to a graphical data map
In previous versions of WebSphere Message Broker, you might have created a message map. To edit your maps in WebSphere Message Broker Version 8.0 or later, you must convert message maps to graphical data maps.

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