



IBM Software Group

WebSphere® Message Broker Toolkit V6.0.2

Toolkit scenario part 6: Web service



@business on demand.

© 2007 IBM Corporation
Updated May 30, 2007

This presentation completes the build of the message flow for this scenario.

Scenario: No inventory response

- An additional wire from the “false” terminal of the CheckInventory Filter node connects to the flow to return a reply to the caller of this Web service
 - ▶ New Mapping node to build the response message
 - ▶ From there to the generated reply nodes for this operation which will return the no inventory message to the caller of the Web service

2

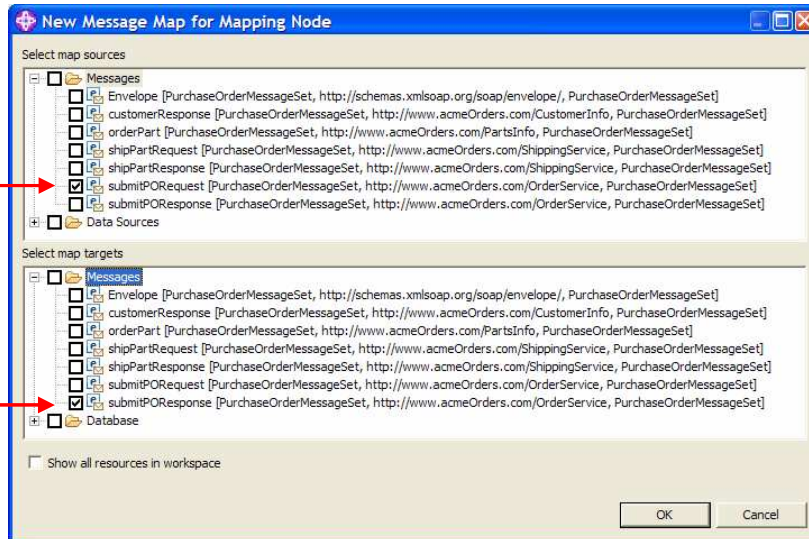
The portion of the PurchaseOrderFlow message flow that handles ordering inventory when the requested part is not available is complete. The next step is to build the flow to provide a response, notifying the requester of the no part in inventory situation. The output of this Mapping node will be wired to the generated reply nodes: the SOAPEnvelope node and HTTPReply node, which returns the message to the caller of this Web service.

Scenario: Mapping node

The screenshot displays the IBM WebSphere MessageFlow Designer interface. The main workspace shows a message flow diagram for 'PurchaseOrderFlow.msgflow'. The flow starts with a 'ws_OrderService' input terminal, which connects to a 'ws_OrderService_DeEnvelopeMsgBody' node. This node then connects to a 'CheckInventory' filter node. From the 'CheckInventory' node, the flow branches into two paths. The upper path goes through 'BuildPartOrderMessage', 'InsertMQMDHeader', and 'SUPPLIER_ORDER_Q0' to an output terminal. The lower path goes through a 'BuildPOResponseRejected' mapping node, which is highlighted with a red box. The output of this mapping node connects to 'ws_OrderService_EnvelopeMsgRespBody' and 'ws_OrderService_Reply' output terminals. A 'Palette' on the left lists various nodes, including 'Mapping'. Below the diagram, the 'Mapping Node Properties - BuildPOResponseRejected' dialog is open, with the 'Mapping routine' field set to 'BuildPOResponseRejected' and the 'Mapping mode' set to 'Message', both highlighted with red boxes. The bottom of the window shows 'Toolkit scenario part 6: Web service' and '© 2007 IBM Corporation'.

Drag a new Mapping node to the message flow. From the false terminal of the CheckInventory Filter node, add a wire to the input terminal of this new Mapping node. Set the node name and mapping routine property to BuildPOResponseRejected. The output terminal of the Mapping node is then wired to the remaining generated flow. The SOAPEnvelope node and HTTPReply node will return the message to the caller of this Web service.

Scenario: Open map



Open the Mapping editor for the BuildPOResponseRejected Mapping node. Select the submitPOResponse messages in the input and targets as shown here.

Scenario: Perform mapping

Map Script	Value
BuildResponseRejected	
Parameters	
Target	
Properties	
tns1:submitPOResponse	
orderStatus	'Rejected'
orderAmt	0
partNo	\$source/tns1:submitPORequest/partNo
partQuantity	\$source/tns1:submitPORequest/partQuantity

Perform the following mapping operations:

1. Map Properties by name
2. Map partNo and partQuantity by drag-and-drop
3. Set orderStatus to 'Rejected' (use right click "Enter Expression")
4. Set orderAmt to 0 (use right click "Enter Expression")

Scenario: Sufficient inventory

- The true terminal of the CheckInventory Filter node wires to a new part of the flow:
 - ▶ Add new Mapping node
 - ▶ Build the ship request message
 - ▶ Use as input to the shipping Web service

When the CheckInventory Filter node determines that there is sufficient inventory to fulfill the order, the true terminal of the filter node is wired to the flow to ship the part to the requester.

To build this flow, a Mapping node is needed to build the ship request message, which is input to the shipping Web service.

IBM Software Group IBM

Scenario: Create Mapping node

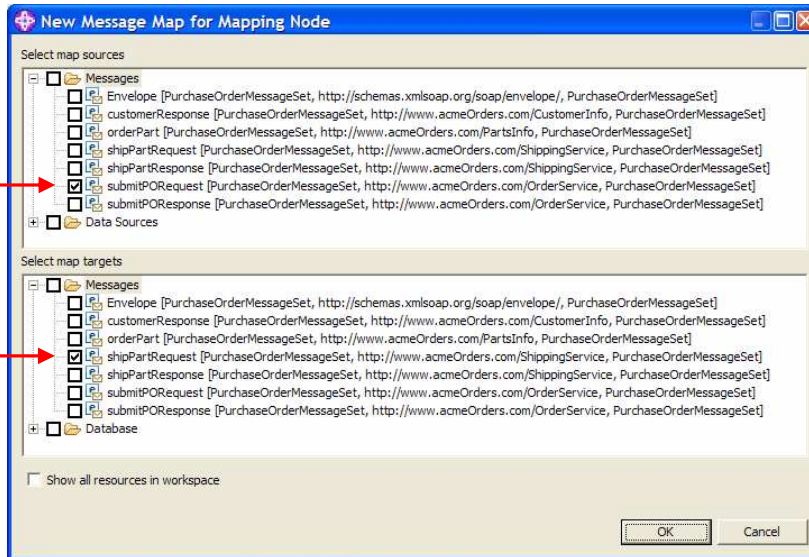
The screenshot displays the IBM WebSphere Message Toolkit interface. The main workspace shows a message flow diagram for 'PurchaseOrderFlow.msgflow'. A 'CheckInventory' filter node is highlighted with a red box, and a 'BuildShipPartRequest' mapping node is also highlighted with a red box. The 'BuildShipPartRequest' node is connected to the 'true' terminal of the 'CheckInventory' node. The 'Properties' window for the 'BuildShipPartRequest' node is open, showing the following settings:

Mapping Node Properties - BuildShipPartRequest	
Basic	
Parser Options	
Validation	
Data source	
Transaction	Automatic
Mapping routine	BuildShipPartRequest
Mapping mode	Message
Treat warnings as errors	<input type="checkbox"/>
Throw exception on database error	<input checked="" type="checkbox"/>

Toolkit scenario part 6: Web service © 2007 IBM Corporation

Drag a Mapping node and wire it from the true terminal of the CheckInventory Filter node. Set the Mapping node name and mapping routine properties to BuildShipPartRequest.

Scenario: Open map



Open the map and select the source, purchase order request, and the target ship purchase order request.

Scenario: Map fields

Map Script	Value
BuildShipPartRequest	
Parameters	
Target	
tns1:shipPartRequest	
partNo	\$source/tns2:submitPORrequest/partNo
partQuantity	\$source/tns2:submitPORrequest/partQuantity
personName	
firstName	\$source/tns2:submitPORrequest/personName/firstName
lastName	\$source/tns2:submitPORrequest/personName/lastName
address	
street	\$source/tns2:submitPORrequest/address/street
city	\$source/tns2:submitPORrequest/address/city
zipCode	\$source/tns2:submitPORrequest/address/zipCode

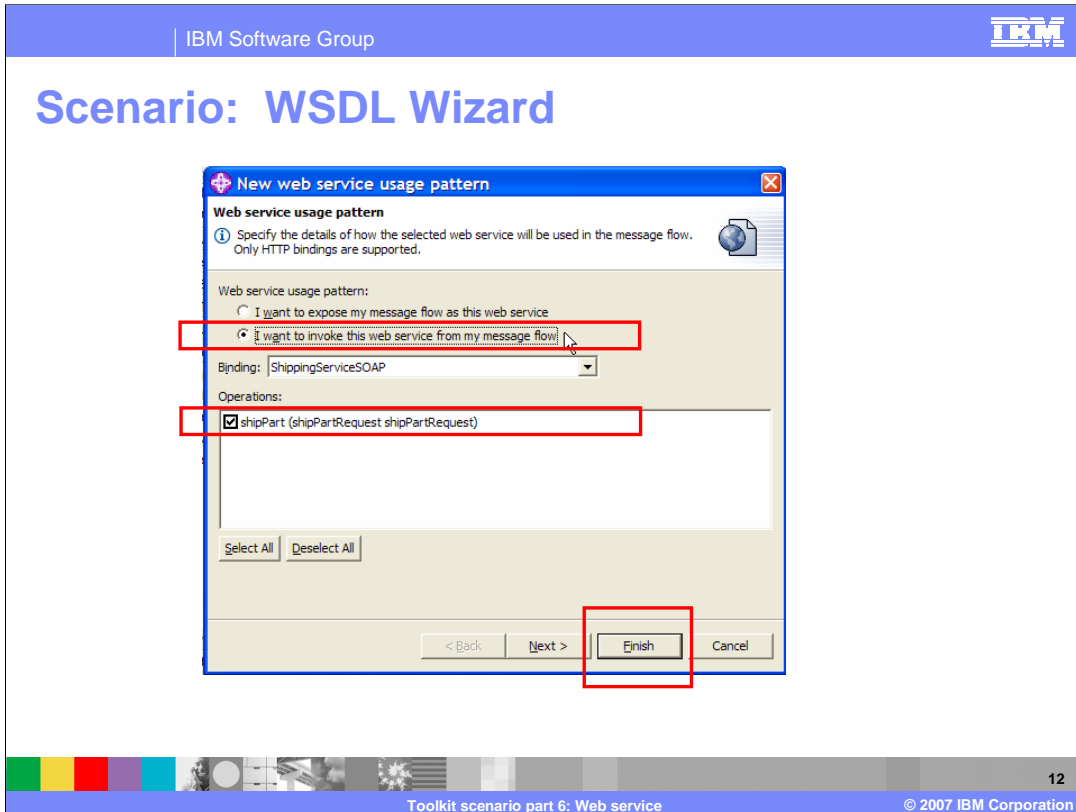
Select \$source and \$target roots, then right click on \$target and select Map By Name, and press Enter to complete the mapping to build the ship part request message.

Scenario: Drag WSDL

The screenshot displays the IBM Business Process Manager interface. On the left, the 'Project Explorer' shows a tree view of the 'PurchaseOrder' project. Under the 'importFiles' folder, the file 'ShippingService.wsdl' is highlighted. A red arrow points from this file to the 'CheckInventory' activity in the message flow diagram on the right. The diagram shows a sequence of activities: 'ws__OrderService_DeEnvelopeMsgBody' leads to 'BuildPartOrderMessage', which leads to 'BuildPOResponseRejected', which leads to 'ws__OrderService_EnvelopeMsgResp'. The 'CheckInventory' activity is also connected to 'BuildShipParRequest'.

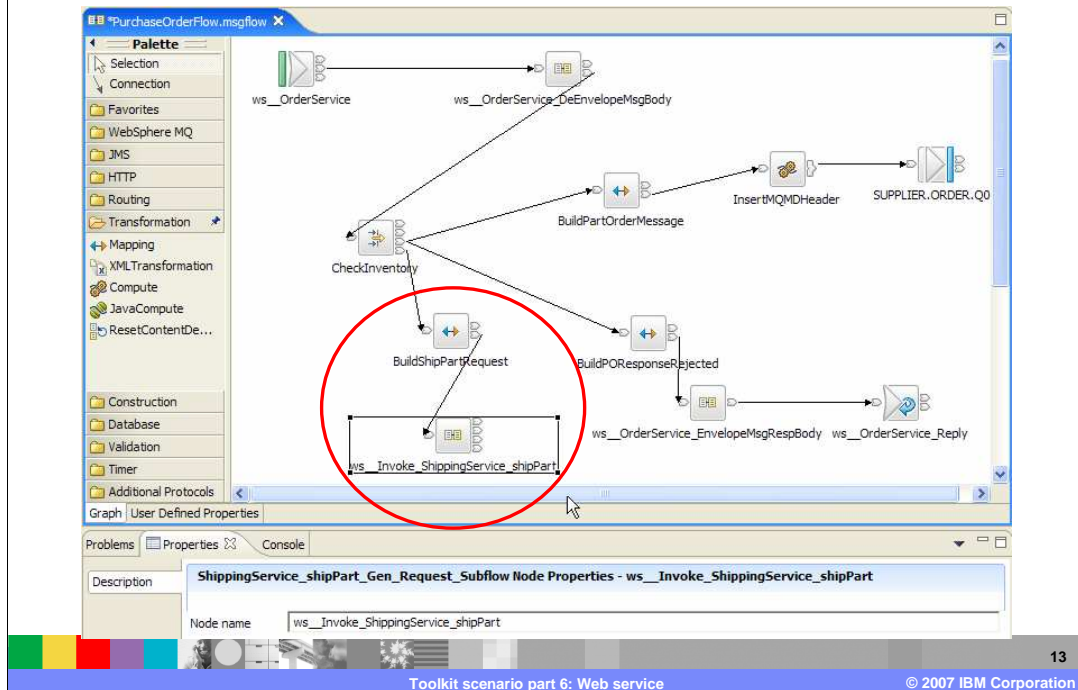
Drag ShippingService.wsdl from the message set "importFiles" folder.

Scenario: WSDL Wizard



A new Web service usage pattern is presented. Select ShippingServiceSOAP binding; and select response operation to invoke the Web service from your message flow.

Scenario: Subflow node



Wire the output terminal of the BuildShipPartRequest Mapping node to the ShippingService Subflow node that you just created.

IBM Software Group IBM

Scenario: Generated sub flow

Generates a sub flow to put message in SOAP envelope, invoke Web service, and extract result from SOAP envelope. Per operation

New SOAPEnvelope node. Surrounds body with SOAP envelope

SOAPExtract node. extract response message body from SOAP envelope

One response output node per WSDL operation (selected in wizard) and fault

Toolkit scenario part 6: Web service © 2007 IBM Corporation

From the working set navigator, double click the generated ShippingService message flow to open it in the Message flow editor. The generated sub flow puts the message in a SOAP envelope, invokes a Web service and extracts the result from the SOAP extract. As explained earlier, the SOAPExtract and SOAPEnvelope nodes are available on the Web as a category 3 SupportPac IA90.

Scenario: Send reply

- Finally, after successfully invoking the Web service to ship the parts to the requester, you can reply to the caller who invoked the Web service:
 - ▶ Use a map node to create the response message
 - ▶ Wire from generated sub flow to the map
 - ▶ Wire from the map to generated reply nodes

After invoking the Web service to ship the parts, generate a response message to the requester.

Scenario: Mapping node

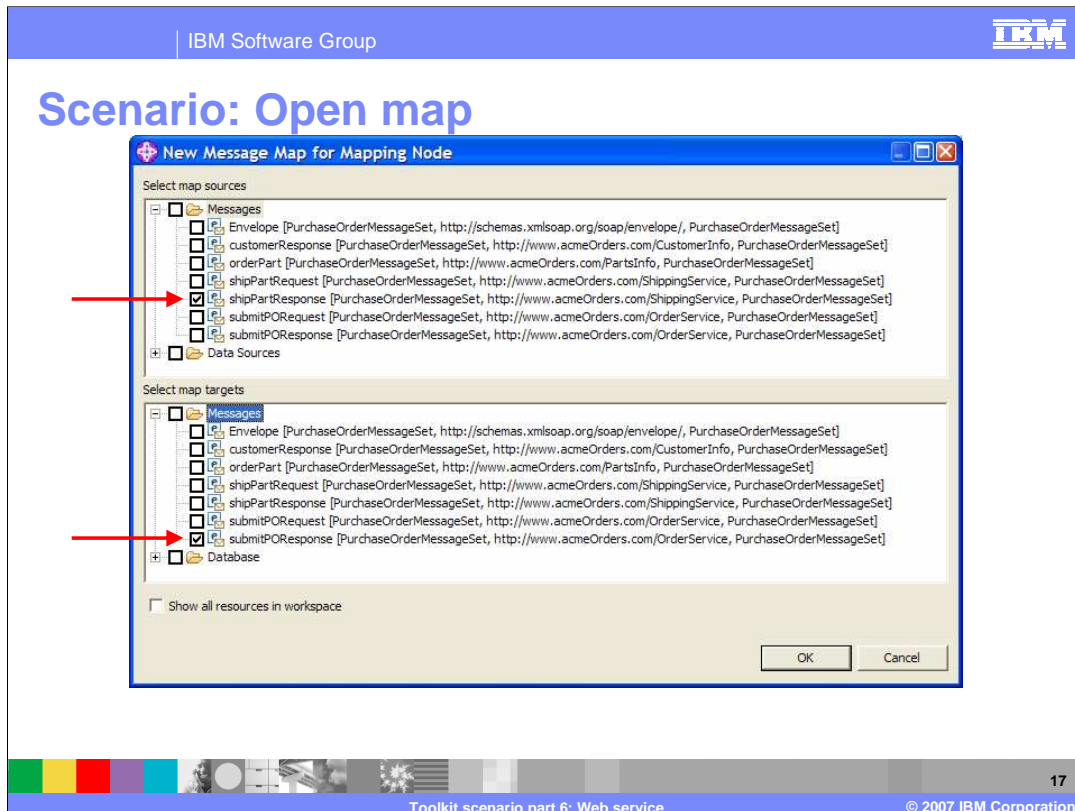
The screenshot displays the IBM WebSphere Studio interface. At the top, the title bar reads "IBM Software Group" and "IBM". Below the title bar, the main heading is "Scenario: Mapping node".

The central part of the image shows a diagram with several nodes and connections. A node labeled "BuildShipPartRequest" is connected to "ws__Invoke_ShippingService_shipPart". Another node, "ws__OrderService_EnvelopeMsgRespBodyws__OrderService_Reply", is connected to a node labeled "BuildPOResponseShipped". The "BuildPOResponseShipped" node is highlighted with a red rectangular box.

Below the diagram, the "Mapping Node Properties - BuildPOResponseShipped" window is open. The "Mapping routine" field is highlighted with a red rectangular box and contains the text "BuildPOResponseShipped". Other visible properties include "Transaction" set to "Automatic", "Mapping mode" set to "Message", and "Throw exception on database error" checked.

At the bottom of the screenshot, the status bar shows "Toolkit scenario part 6: Web service" and "© 2007 IBM Corporation". The page number "16" is also visible in the bottom right corner.

Drag a Mapping node and set its name and mapping routine property to BuildPOResponseShipped



Open the map and select source, shipPartResponse, and target, then submitPOResponse.

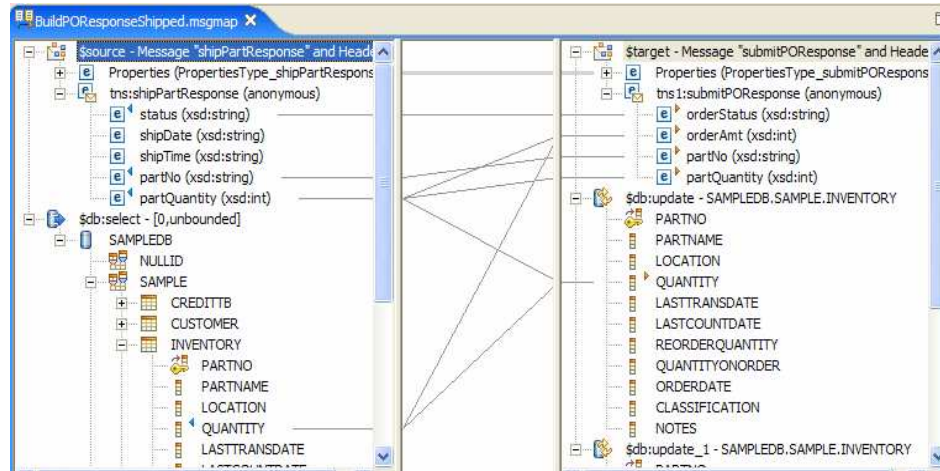
Scenario: Map fields

The screenshot displays the Message Mapping Editor interface. The source message tree on the left includes properties like `status (xsd:string)`, `shipDate (xsd:string)`, `shipTime (xsd:string)`, `partNo (xsd:string)`, and `partQuantity (xsd:int)`. The target message tree on the right includes `orderStatus (xsd:string)`, `orderAmt (xsd:int)`, `partNo (xsd:string)`, and `partQuantity (xsd:int)`. A mapping table at the bottom shows the following configuration:

Map Script	Value
BuildPOResponseShipped	
Parameters	
\$target	
Properties	
tns2:submitPOResponse	
orderStatus	<code>\$source/tns:shipPartResponse/status</code>
orderAmt	50.0
partNo	<code>\$source/tns:shipPartResponse/partNo</code>
partQuantity	<code>\$source/tns:shipPartResponse/partQuantity</code>

Select `$source` and `$target` roots, and right click on `$target` and select **Map By Name**. Press Enter, then drag `status` to `OrderStatus`. Select `OrderAmt` in target tree, right click, then select **Enter Expression** and enter `50.0`

Note: About the “orderAmt” value



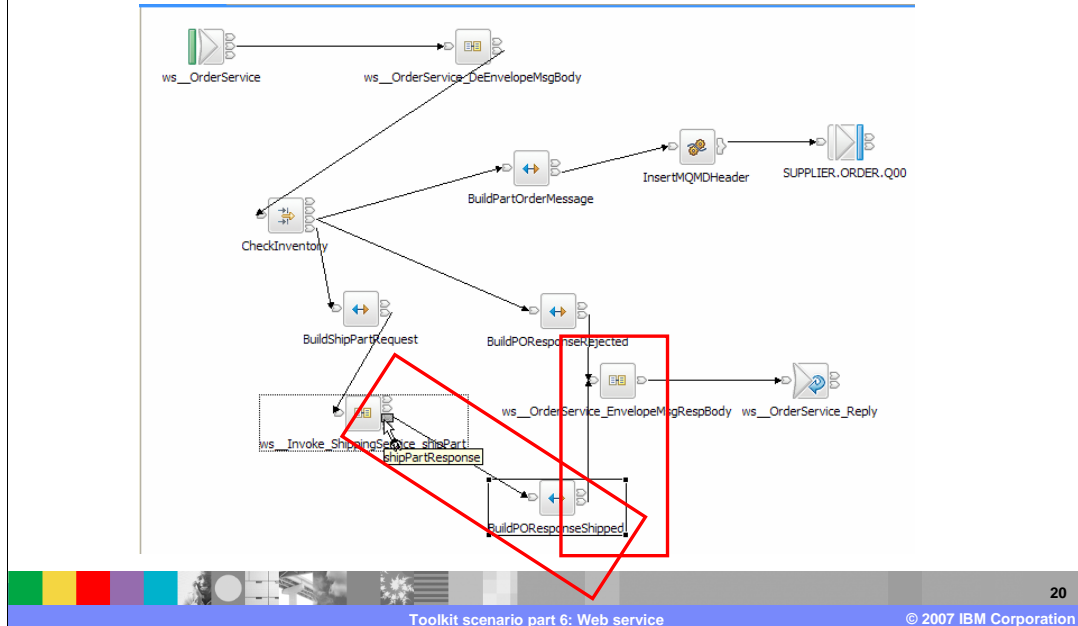
19

Toolkit scenario part 6: Web service

© 2007 IBM Corporation

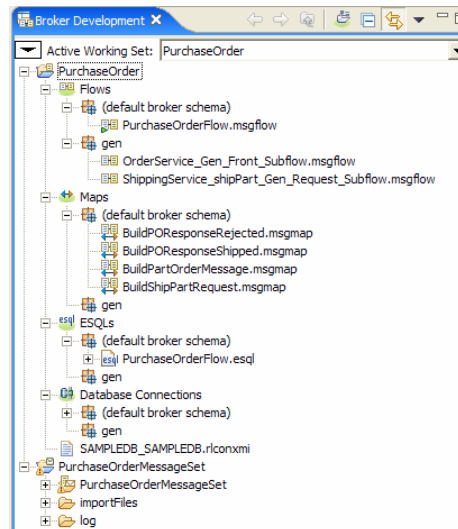
The `orderAmt` value should be set to the quantity multiplied by the unit price, however, in the interest of brevity, this has not been done here. You could check the inventory, limit the `orderAmt` to what is actually available, update the inventory, and compute the `orderAmt` all in one map, and this is shown in the screen capture here.

Scenario: Wire map node



Wire the nodes as shown here, by first wiring from the submitResponse terminal of the ShippingService sub flow node to the input terminal of the BuildPOResponseShipped Mapping node. Next, wire the Mapping node out terminal to the generated reply SOAPEnvelope node and HTTPReply node which will return the message to the caller of this Web service.

Authoring steps are done!



The PurchaseOrder design is now finished, and the navigator view for the PurchaseOrder project shows all the components. This concludes part 6 of the WebSphere Message Broker V6.0.2 Toolkit scenario.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject= Feedback about Toolkit.Scenarios.Part6_WebService.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20Toolkit.Scenarios.Part6_WebService.ppt)



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM WebSphere

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

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

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

