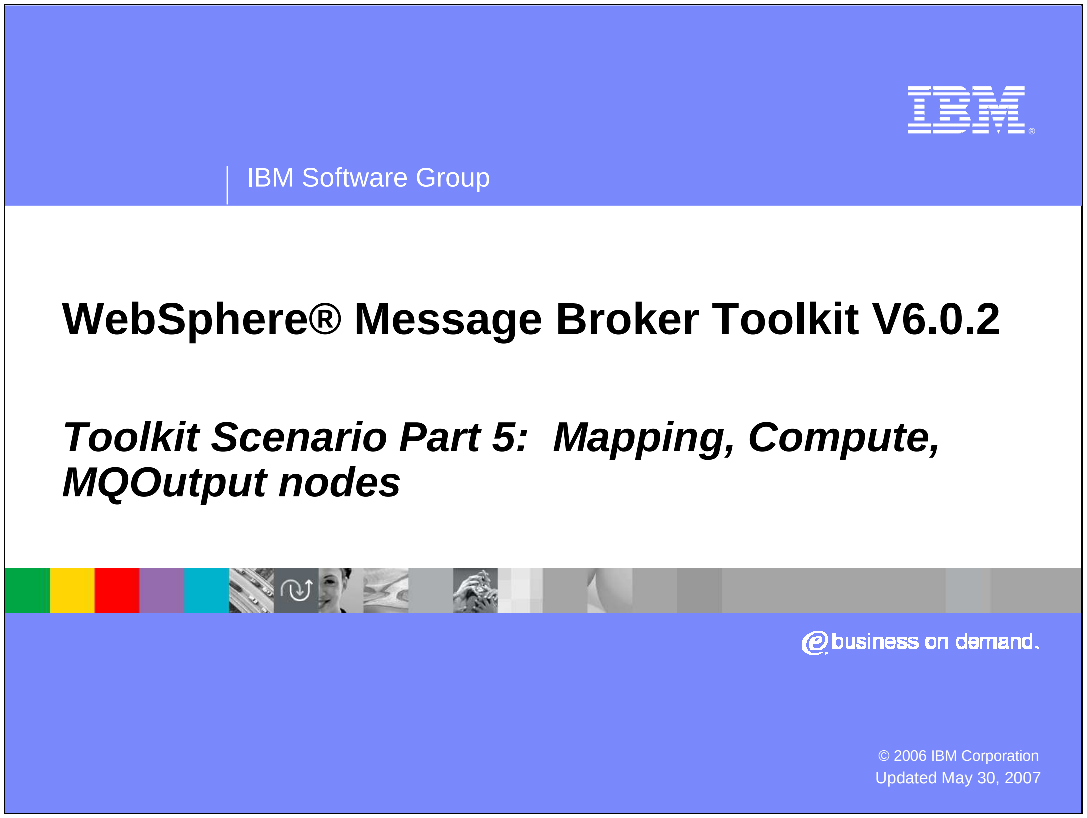


IBM Software Group

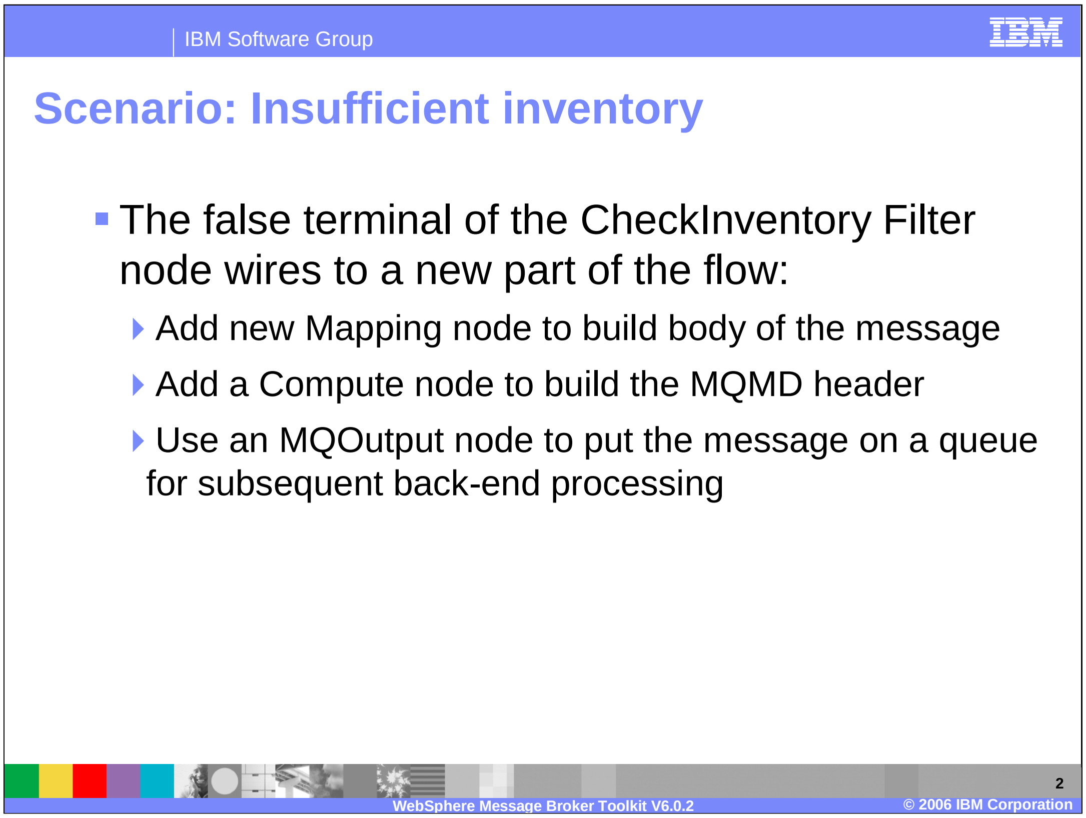
WebSphere® Message Broker Toolkit V6.0.2

Toolkit Scenario Part 5: Mapping, Compute, MQOutput nodes

business on demand.

Updated May 30, 2007

This fifth module of the scenario continues the solution using a Mapping node, a Compute node and an MQOutput node.



IBM Software Group

Scenario: Insufficient inventory

- The false terminal of the CheckInventory Filter node wires to a new part of the flow:
  - Add new Mapping node to build body of the message
  - Add a Compute node to build the MQMD header
  - Use an MQOutput node to put the message on a queue for subsequent back-end processing

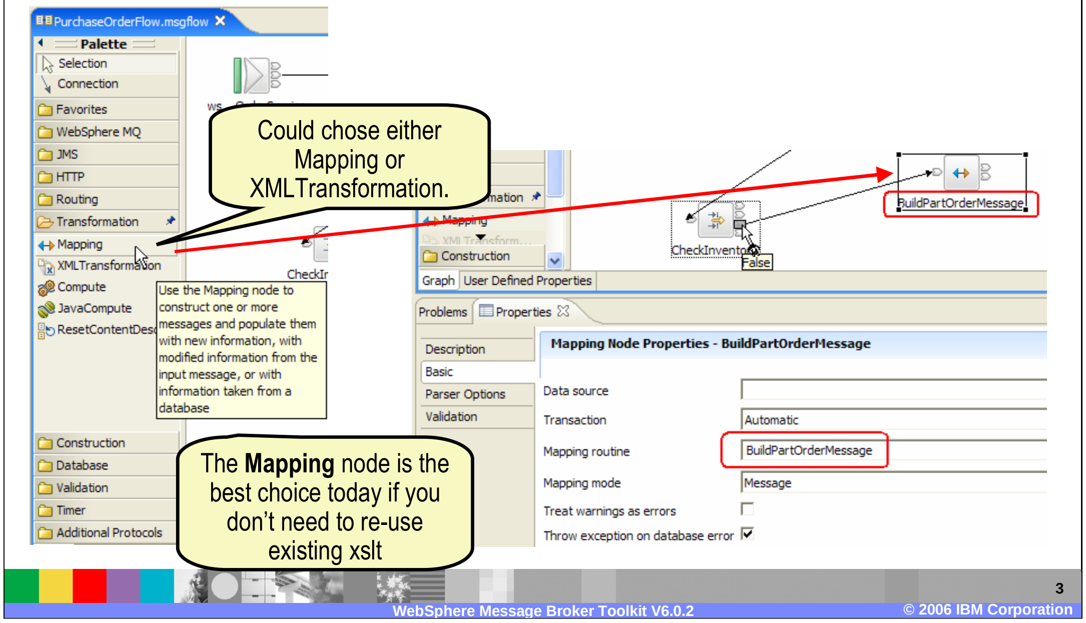
2

WebSphere Message Broker Toolkit V6.0.2

When the CheckInventory Filter node determines that there is not sufficient inventory to fulfill the order, the false terminal of the filter node is wired to the flow to request additional inventory. A Mapping node is needed to build the body of the WebSphere message that will be read to order additional inventory, a Compute node is needed to build the WebSphere MQ MQMD header for the message, and an MQOutput node puts the message on the queue for subsequent processing.

IBM Software Group

Scenario: Out of inventory mapping



Could chose either Mapping or XMLTransformation.

The **Mapping** node is the best choice today if you don't need to re-use existing xslt

Use the Mapping node to construct one or more messages and populate them with new information, with modified information from the input message, or with information taken from a database

Mapping Node Properties - BuildPartOrderMessage

Data source

Transaction Automatic

Mapping routine BuildPartOrderMessage

Mapping mode Message

Treat warnings as errors ☐

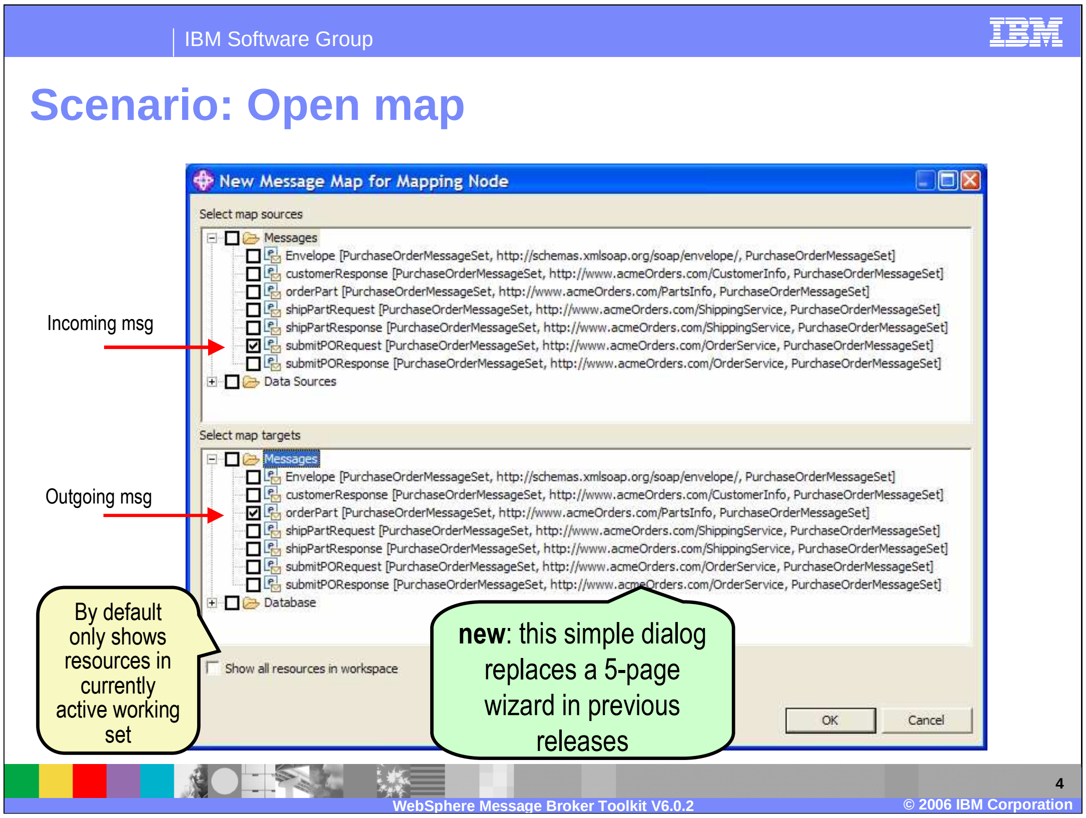
Throw exception on database error ☑

3

WebSphere Message Broker Toolkit V6.0.2

© 2006 IBM Corporation

A node is needed to create a map when the requested parts are out of inventory. The Mapping node is selected from the Transformation category based on the text explanation of that node. Wire the false output terminal of the Filter node (named CheckInventory) to the Mapping node. Set the properties and name BuildPartOrderMessage. This map will layout the information needed to order the part.



IBM Software Group

Scenario: Open map

New Message Map for Mapping Node

Select map sources

- □ Messages
  - □ Envelope [PurchaseOrderMessageSet, http://schemas.xmlsoap.org/soap/envelope/, PurchaseOrderMessageSet]
  - □ customerResponse [PurchaseOrderMessageSet, http://www.acmeOrders.com/CustomerInfo, PurchaseOrderMessageSet]
  - □ orderPart [PurchaseOrderMessageSet, http://www.acmeOrders.com/PartsInfo, PurchaseOrderMessageSet]
  - □ shipPartRequest [PurchaseOrderMessageSet, http://www.acmeOrders.com/ShippingService, PurchaseOrderMessageSet]
  - □ shipPartResponse [PurchaseOrderMessageSet, http://www.acmeOrders.com/ShippingService, PurchaseOrderMessageSet]
  - ☑ submitPORequest [PurchaseOrderMessageSet, http://www.acmeOrders.com/OrderService, PurchaseOrderMessageSet]
  - □ submitPOResponse [PurchaseOrderMessageSet, http://www.acmeOrders.com/OrderService, PurchaseOrderMessageSet]
- □ Data Sources

Select map targets

- □ Messages
  - □ Envelope [PurchaseOrderMessageSet, http://schemas.xmlsoap.org/soap/envelope/, PurchaseOrderMessageSet]
  - □ customerResponse [PurchaseOrderMessageSet, http://www.acmeOrders.com/CustomerInfo, PurchaseOrderMessageSet]
  - ☑ orderPart [PurchaseOrderMessageSet, http://www.acmeOrders.com/PartsInfo, PurchaseOrderMessageSet]
  - □ shipPartRequest [PurchaseOrderMessageSet, http://www.acmeOrders.com/ShippingService, PurchaseOrderMessageSet]
  - □ shipPartResponse [PurchaseOrderMessageSet, http://www.acmeOrders.com/ShippingService, PurchaseOrderMessageSet]
  - □ submitPORequest [PurchaseOrderMessageSet, http://www.acmeOrders.com/OrderService, PurchaseOrderMessageSet]
  - □ submitPOResponse [PurchaseOrderMessageSet, http://www.acmeOrders.com/OrderService, PurchaseOrderMessageSet]
- □ Database

□ Show all resources in workspace

Incoming msg

Outgoing msg

By default only shows resources in currently active working set

new: this simple dialog replaces a 5-page wizard in previous releases

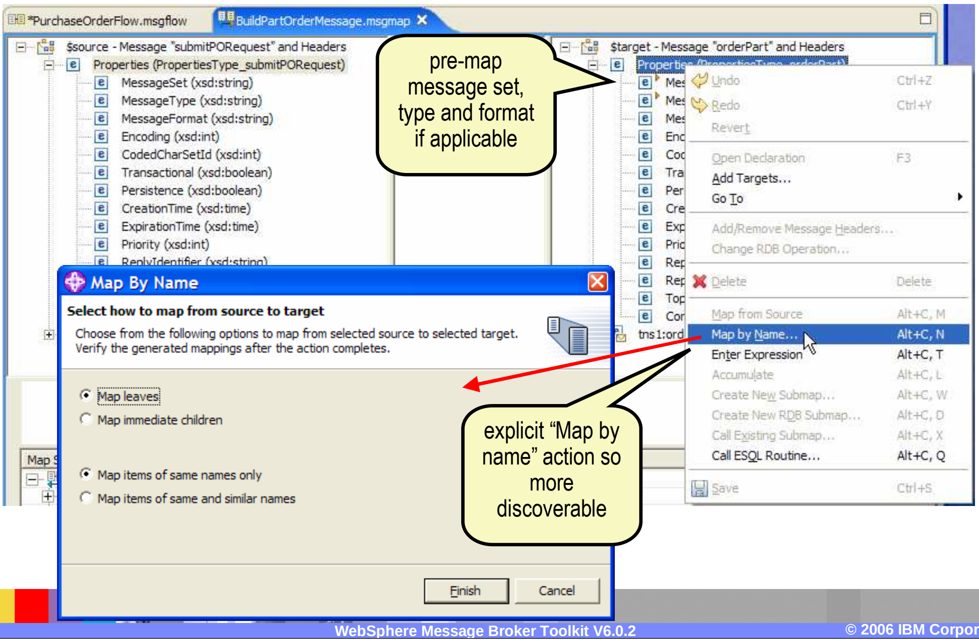
4

WebSphere Message Broker Toolkit V6.0.2

© 2006 IBM Corporation

After wiring the False terminal of the Filter node (CheckInventory) to the Mapping node (BuildPartOrderMessage), double click on the Mapping node to open the Map wizard.

Scenario: Map the properties


*PurchaseOrderFlow.msgflow
BuildPartOrderMessage.msgmap
$source - Message "submitPORequest" and Headers
Properties (PropertiesType_submitPORequest)
MessageSet (xsd:string)
MessageType (xsd:string)
MessageFormat (xsd:string)
Encoding (xsd:int)
CodedCharSetId (xsd:int)
Transactional (xsd:boolean)
Persistence (xsd:boolean)
CreationTime (xsd:time)
ExpirationTime (xsd:time)
Priority (xsd:int)
ReplyIdentifier (xsd:string)
$target - Message "orderPart" and Headers
Properties (PropertiesType_orderPart)
tns1:ord
Undo Ctrl+Z
Redo Ctrl+Y
Revert
Open Declaration F3
Add Targets…
Go To
Add/Remove Message Headers…
Change RDB Operation…
Delete Delete
Map from Source Alt+C, M
Map by Name… Alt+C, N
Enter Expression Alt+C, T
Accumulate Alt+C, L
Create New Submap… Alt+C, W
Create New RDB Submap… Alt+C, D
Call Existing Submap… Alt+C, X
Call ESQL Routine… Alt+C, Q
Save Ctrl+S
Map By Name
Finish
Cancel

Select how to map from source to target

Choose from the following options to map from selected source to selected target.
Verify the generated mappings after the action completes.

● Map leaves
○ Map immediate children

● Map items of same names only
○ Map items of same and similar names

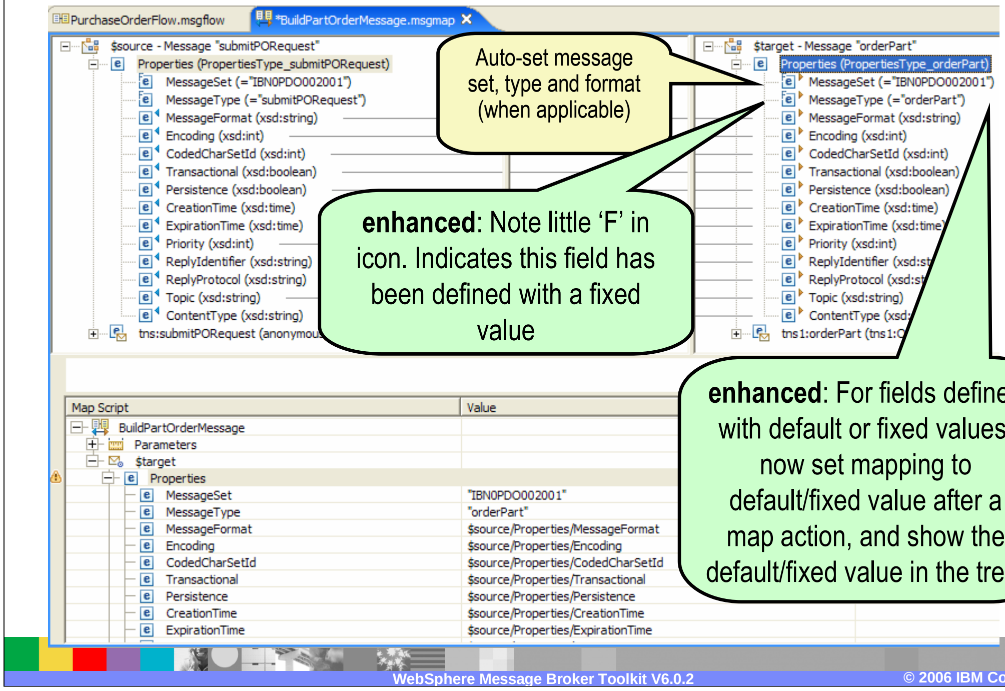
pre-map message set, type and format if applicable

explicit “Map by name” action so more discoverable

In mapping editor, select Properties on both sides, and then Map By Name from right click.

IBM Software Group

Scenario: Properties mapped



PurchaseOrderFlow.msgflow | *BuildPartOrderMessage.msgmap

```
$source - Message "submitPORequest"
  Properties (PropertiesType_submitPORequest)
    MessageSet (="IBN0PDO002001")
    MessageType (="submitPORequest")
    MessageFormat (xsd:string)
    Encoding (xsd:int)
    CodedCharSetId (xsd:int)
    Transactional (xsd:boolean)
    Persistence (xsd:boolean)
    CreationTime (xsd:time)
    ExpirationTime (xsd:time)
    Priority (xsd:int)
    ReplyIdentifier (xsd:string)
    ReplyProtocol (xsd:string)
    Topic (xsd:string)
    ContentType (xsd:string)
  tns:submitPORequest (anonymou
```

```
$target - Message "orderPart"
  Properties (PropertiesType_orderPart)
    MessageSet (="IBN0PDO002001")
    MessageType (="orderPart")
    MessageFormat (xsd:string)
    Encoding (xsd:int)
    CodedCharSetId (xsd:int)
    Transactional (xsd:boolean)
    Persistence (xsd:boolean)
    CreationTime (xsd:time)
    ExpirationTime (xsd:time)
    Priority (xsd:int)
    ReplyIdentifier (xsd:st
    ReplyProtocol (xsd:st
    Topic (xsd:string)
    ContentType (xsd:
  tns1:orderPart (tns1:C
```

Map Script	Value
BuildPartOrderMessage	
Parameters	
$target	
Properties	
MessageSet	"IBN0PDO002001"
MessageType	"orderPart"
MessageFormat	$source/Properties/MessageFormat
Encoding	$source/Properties/Encoding
CodedCharSetId	$source/Properties/CodedCharSetId
Transactional	$source/Properties/Transactional
Persistence	$source/Properties/Persistence
CreationTime	$source/Properties/CreationTime
ExpirationTime	$source/Properties/ExpirationTime

Auto-set message set, type and format (when applicable)

enhanced: Note little ‘F’ in icon. Indicates this field has been defined with a fixed value

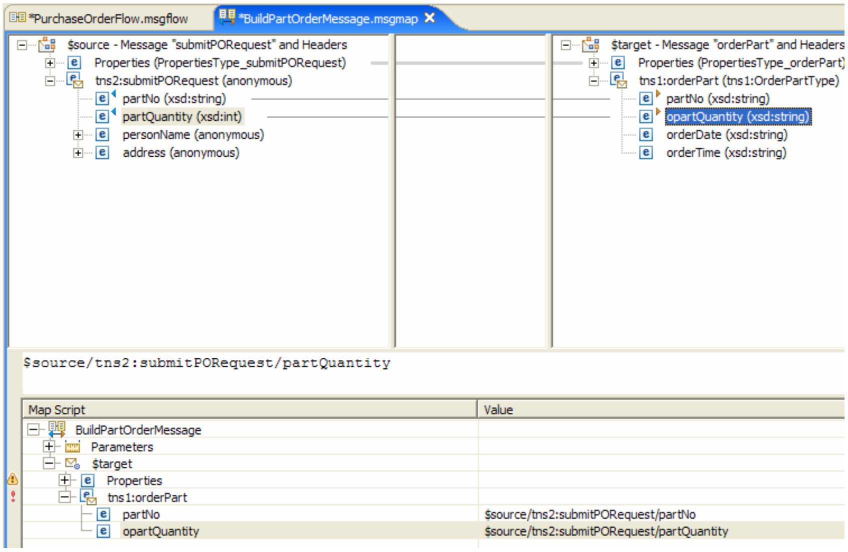
enhanced: For fields defined with default or fixed values, now set mapping to default/fixed value after a map action, and show the default/fixed value in the tree.

6

WebSphere Message Broker Toolkit V6.0.2

A mapping is generated for all source and target properties of the same name.

Scenario: Map the fields

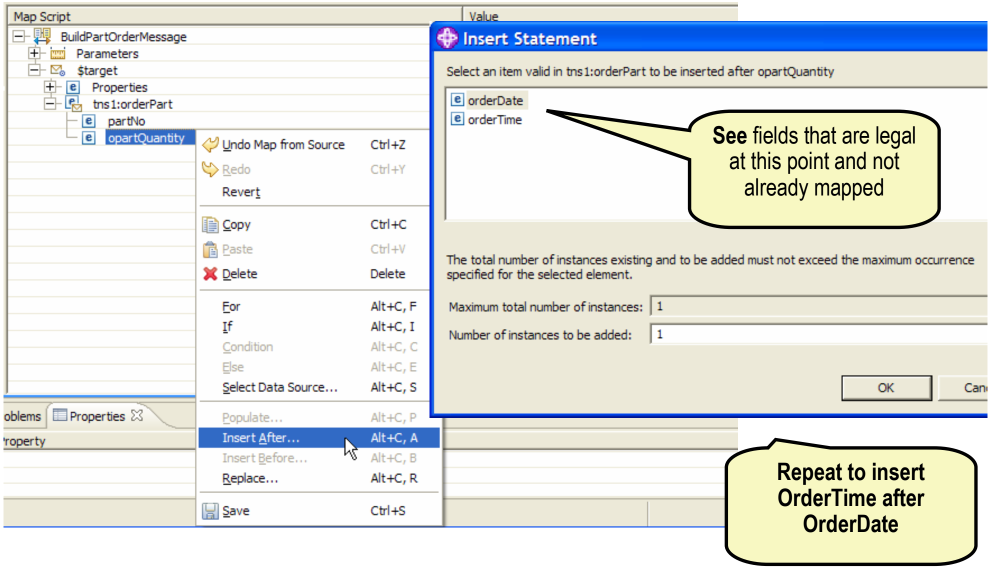

*PurchaseOrderFlow.msgflow
*BuildPartOrderMessage.msgmap
$source - Message "submitPORequest" and Headers
Properties (PropertiesType_submitPORequest)
tns2:submitPORequest (anonymous)
partNo (xsd:string)
partQuantity (xsd:int)
personName (anonymous)
address (anonymous)
$target - Message "orderPart" and Headers
Properties (PropertiesType_orderPart)
tns1:orderPart (tns1:OrderPartType)
partNo (xsd:string)
opartQuantity (xsd:string)
orderDate (xsd:string)
orderTime (xsd:string)

```
$source/tns2:submitPORequest/partQuantity
```

Map Script	Value
BuildPartOrderMessage	
Parameters	
$target	
Properties	
tns1:orderPart	
partNo	$source/tns2:submitPORequest/partNo
opartQuantity	$source/tns2:submitPORequest/partQuantity

After expanding the body portion, drag `partNo` and `partQuantity` from the source to the target.

Scenario: Add fields



Select an item valid in tns1:orderPart to be inserted after opartQuantity

The total number of instances existing and to be added must not exceed the maximum occurrence specified for the selected element.

Maximum total number of instances: 1

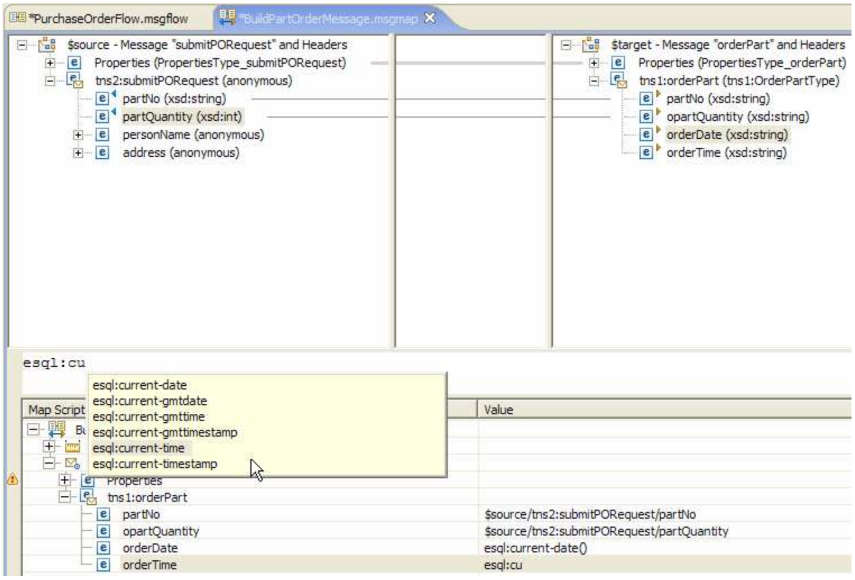
Number of instances to be added: 1

See fields that are legal at this point and not already mapped

Repeat to insert OrderTime after OrderDate

In the Map Script, insert orderDate and orderTime.

Scenario: Set value to added fields



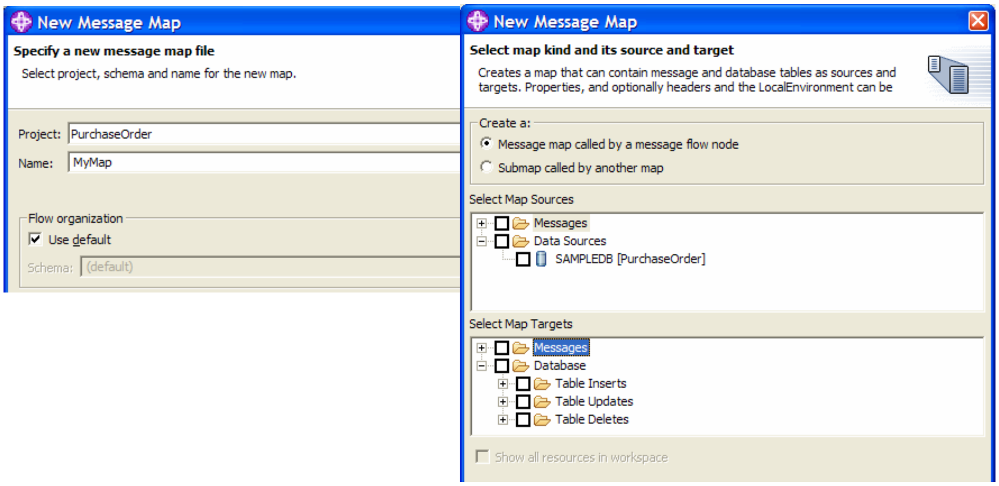
*PurchaseOrderFlow.msgflow
*BuildPartOrderMessage.msgmap
$source - Message "submitPORequest" and Headers
Properties (PropertiesType_submitPORequest)
tns2:submitPORequest (anonymous)
partNo (xsd:string)
partQuantity (xsd:int)
personName (anonymous)
address (anonymous)
$target - Message "orderPart" and Headers
Properties (PropertiesType_orderPart)
tns1:orderPart (tns1:OrderPartType)
partNo (xsd:string)
opartQuantity (xsd:string)
orderDate (xsd:string)
orderTime (xsd:string)
esql:current-date
esql:current-gmtdate
esql:current-gmttime
esql:current-gmttimestamp
esql:current-time
esql:current-timestamp
Bu
Properties

```
esql:cu
```

Map Script	Value
tns1:orderPart	
partNo	$source/tns2:submitPORequest/partNo
opartQuantity	$source/tns2:submitPORequest/partQuantity
orderDate	esql:current-date()
orderTime	esql:cu

In the Map Script, set orderDate and orderTime to esql:current-date and esql:current-time.

Notes: Mapping editor enhancements 1



New Message Map

Specify a new message map file

Select project, schema and name for the new map.

Project: PurchaseOrder

Name: MyMap

Flow organization

☑ Use default

Schema: (default)

New Message Map

Select map kind and its source and target

Creates a map that can contain message and database tables as sources and targets. Properties, and optionally headers and the LocalEnvironment can be

Create a:

● Message map called by a message flow node

○ Submap called by another map

Select Map Sources

- □ Messages
- □ Data Sources
  - □ SAMPLEDB [PurchaseOrder]

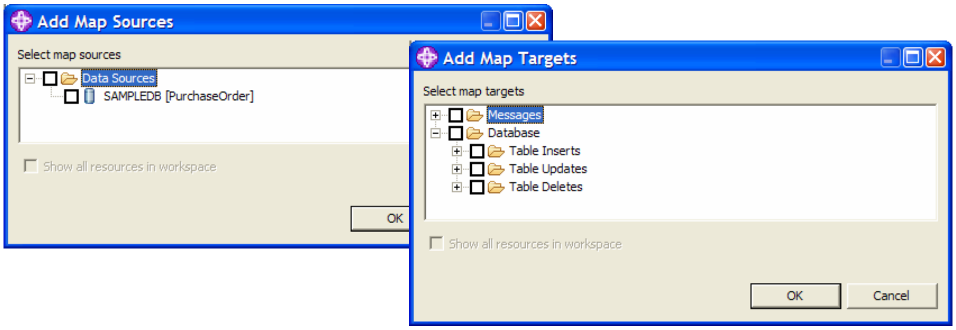
Select Map Targets

- □ Messages
- □ Database
  - □ Table Inserts
  - □ Table Updates
  - □ Table Deletes

□ Show all resources in workspace

In WebSphere Message Broker Toolkit V6.0.2, there are a number of enhancements to the Mapping editor. First, the wizard has been simplified. Shown here are the wizard pages when opened from File > New > Message Map.

Notes: Mapping editor enhancements 2



Add Map Sources

Select map sources

□ Data Sources
□ SAMPLEDB [PurchaseOrder]

□ Show all resources in workspace

Add Map Targets

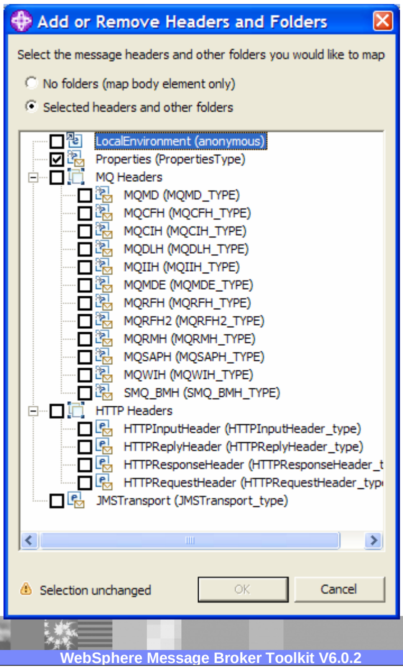
Select map targets

□ Messages
□ Database
□ Table Inserts
□ Table Updates
□ Table Deletes

□ Show all resources in workspace

In the Mapping editor, you can add sources and targets by right clicking actions on $source and $target roots in trees. You can also drag roots from the navigator to add sources and targets. For example you could drag a message, a database table, or a global element (in submaps).

Notes: Mapping editor enhancements 3



Add or Remove Headers and Folders

Select the message headers and other folders you would like to map

○ No folders (map body element only)

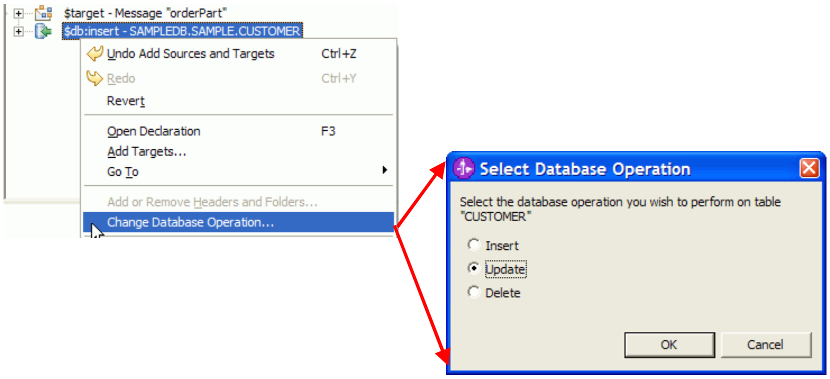
● Selected headers and other folders

- □ LocalEnvironment (anonymous)
- ☑ Properties (PropertiesType)
- □ MQ Headers
  - □ MQMD (MQMD_TYPE)
  - □ MQCFH (MQCFH_TYPE)
  - □ MQCIH (MQCIH_TYPE)
  - □ MQDLH (MQDLH_TYPE)
  - □ MQIIH (MQIIH_TYPE)
  - □ MQMDE (MQMDE_TYPE)
  - □ MQRFH (MQRFH_TYPE)
  - □ MQRFH2 (MQRFH2_TYPE)
  - □ MQRMH (MQRMH_TYPE)
  - □ MQSAPH (MQSAPH_TYPE)
  - □ MQWIH (MQWIH_TYPE)
  - □ SMQ_BMH (SMQ_BMH_TYPE)
- □ HTTP Headers
  - □ HTTPInputHeader (HTTPInputHeader_type)
  - □ HTTPReplyHeader (HTTPReplyHeader_type)
  - □ HTTPResponseHeader (HTTPResponseHeader_t
  - □ HTTPRequestHeader (HTTPRequestHeader_typ
- □ JMSTransport (JMSTransport_type)

Selection unchanged

By default, the Properties folder is shown in the Mapping editor to enable explicit mapping, and any headers are automatically copied from input message. The Local Environment is not mapped by default, however, within the Mapping editor you can right click on Source or Target and select "*Add/Remove Message Headers…*" The results of doing this are shown in this screen capture.

Notes: Mapping editor enhancements 4


$target - Message "orderPart"
$db:insert - SAMPLEDB.SAMPLE.CUSTOMER
Undo Add Sources and Targets Ctrl+Z
Redo Ctrl+Y
Revert
Open Declaration F3
Add Targets...
Go To
Add or Remove Headers and Folders...
Change Database Operation...
Select Database Operation
OK
Cancel

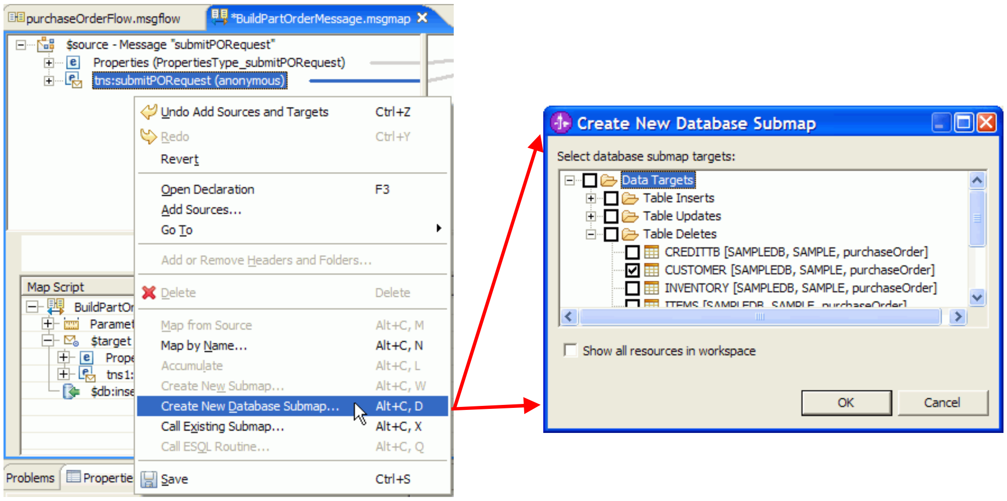
Select the database operation you wish to perform on table "CUSTOMER"

○ Insert
● Update
○ Delete

In V6.0.2, there is a new Select Database Operation, which is shown here. Additionally, you can now map from database to database, whereas previously you could only map from database to message or message to database.

IBM Software Group

Notes: Mapping editor enhancements 5


purchaseOrderFlow.msgflow
*BuildPartOrderMessage.msgmap
$source - Message "submitPORequest"
Properties (PropertiesType_submitPORequest)
tns:submitPORequest (anonymous)
Map Script
BuildPartOr
Paramet
$target
Prope
tns1:
$db:inse
Undo Add Sources and Targets Ctrl+Z
Redo Ctrl+Y
Revert
Open Declaration F3
Add Sources...
Go To
Add or Remove Headers and Folders...
Delete Delete
Map from Source Alt+C, M
Map by Name... Alt+C, N
Accumulate Alt+C, L
Create New Submap... Alt+C, W
Create New Database Submap... Alt+C, D
Call Existing Submap... Alt+C, X
Call ESQL Routine... Alt+C, Q
Save Ctrl+S
Problems
Propertie
Create New Database Submap
Select database submap targets:
Data Targets
Table Inserts
Table Updates
Table Deletes
CREDITTB [SAMPLEDB, SAMPLE, purchaseOrder]
CUSTOMER [SAMPLEDB, SAMPLE, purchaseOrder]
INVENTORY [SAMPLEDB, SAMPLE, purchaseOrder]
ITEMS [SAMPLEDB, SAMPLE, purchaseOrder]
Show all resources in workspace
OK
Cancel

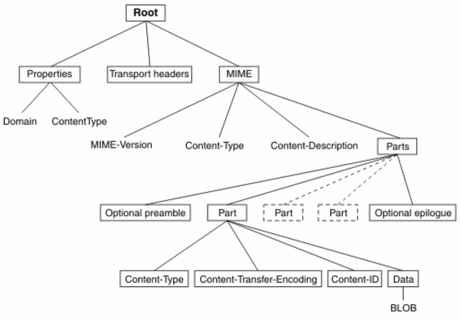
14

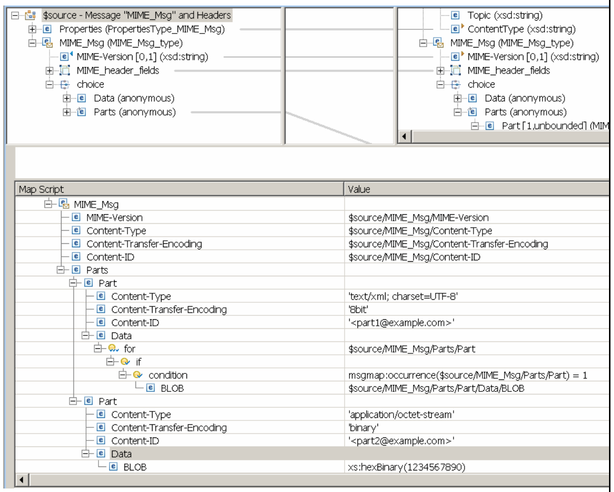
WebSphere Message Broker Toolkit V6.0.2

© 2006 IBM Corporation

You can Create a New Database Submap using a right click action as shown here.

Notes: Mapping editor enhancements 6



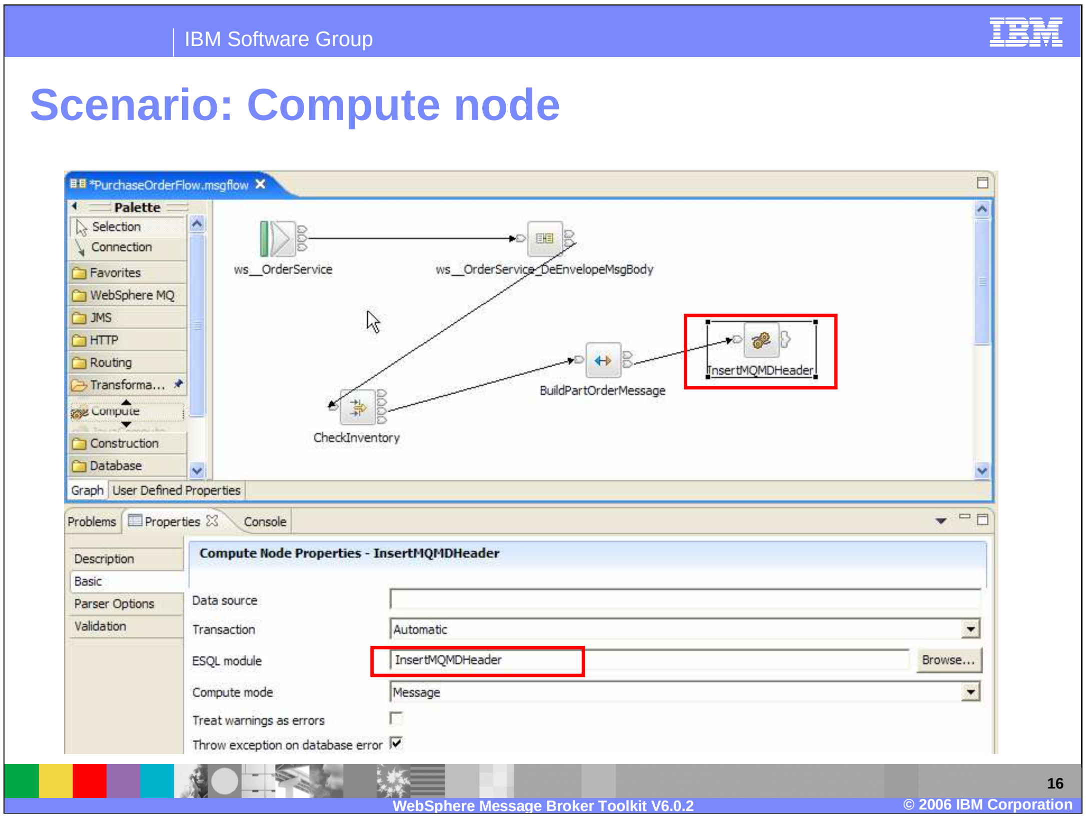


```
$source - Message "MIME_Msg" and Headers
  Properties (PropertiesType_MIME_Msg)
  MIME_Msg (MIME_Msg_type)
    MIME-Version [0,1] (xsd:string)
    MIME_header_fields
    choice
      Data (anonymous)
      Parts (anonymous)
```

```
      Topic (xsd:string)
      ContentType (xsd:string)
  MIME_Msg (MIME_Msg_type)
    MIME-Version [0,1] (xsd:string)
    MIME_header_fields
    choice
      Data (anonymous)
      Parts (anonymous)
        Part [1,unbounded] (MIM
```

Map Script	Value
MIME_Msg	
MIME-Version	$source/MIME_Msg/MIME-Version
Content-Type	$source/MIME_Msg/Content-Type
Content-Transfer-Encoding	$source/MIME_Msg/Content-Transfer-Encoding
Content-ID	$source/MIME_Msg/Content-ID
Parts	
Part	
Content-Type	'text/xml; charset=UTF-8'
Content-Transfer-Encoding	'8bit'
Content-ID	'<part1@example.com>'
Data	
for	$source/MIME_Msg/Parts/Part
if	
condition	msgmap:occurrence($source/MIME_Msg/Parts/Part) = 1
BLOB	$source/MIME_Msg/Parts/Part/Data/BLOB
Part	
Content-Type	'application/octet-stream'
Content-Transfer-Encoding	'binary'
Content-ID	'<part2@example.com>'
Data	
BLOB	xs:hexBinary(1234567890)

You can now use the Mapping editor to map messages in a MIME domain. Domains supported are MRM, MIME, XML, XMLNS, XMLNSC, JMSMap, and JMSStream.



IBM Software Group

Scenario: Compute node

Compute Node Properties - InsertMQMDHeader

Data source

Transaction: Automatic

ESQL module: InsertMQMDHeader

Compute mode: Message

Treat warnings as errors ☐

Throw exception on database error ☑

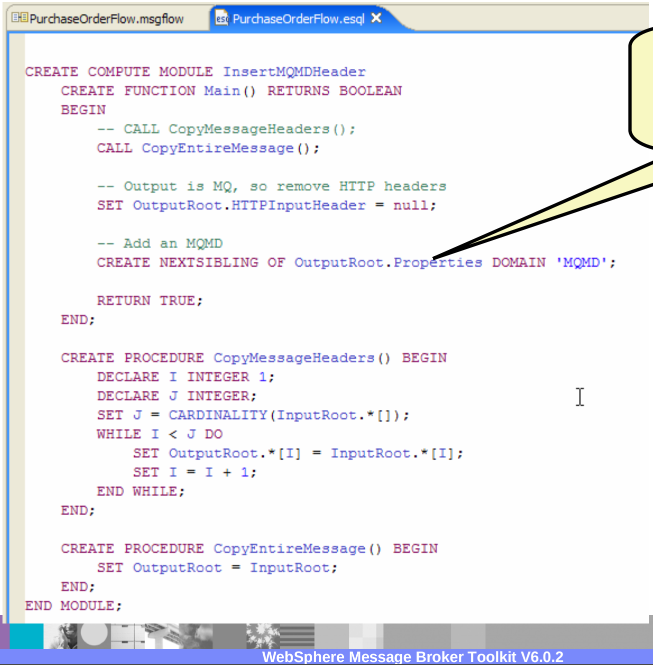
16

WebSphere Message Broker Toolkit V6.0.2

© 2006 IBM Corporation

In the flow editor, drop a Compute node, wire it to the output of the Mapping node, and set its name and ESQL Module property to `InsertMQMDHeader`. This Compute node will be used to create the MQMD header, which is needed for the message that will be sent to the MQOutput node. The message will be subsequently processed by a back-end system to order the needed inventory.

Scenario: ESQL MQMD header

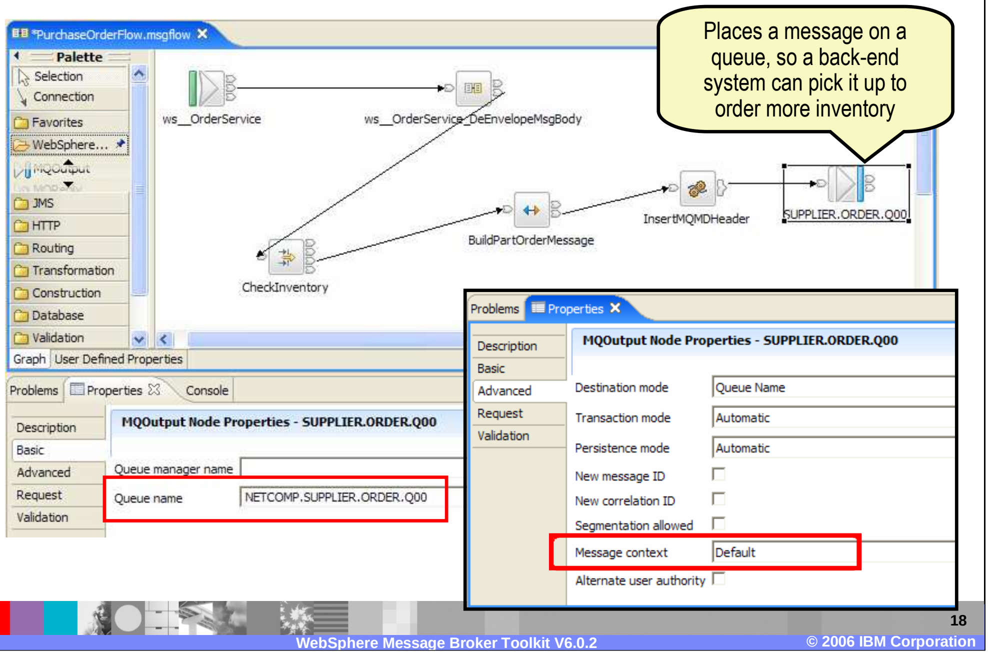


```
CREATE COMPUTE MODULE InsertMQMDHeader
    CREATE FUNCTION Main() RETURNS BOOLEAN
    BEGIN
        -- CALL CopyMessageHeaders();
        CALL CopyEntireMessage();

        -- Output is MQ, so remove HTTP headers
        SET OutputRoot.HTTPInputHeader = null;

        -- Add an MQMD
        CREATE NEXTSIBLING OF OutputRoot.Properties DOMAIN 'MQMD';

        RETURN TRUE;
    END;

    CREATE PROCEDURE CopyMessageHeaders() BEGIN
        DECLARE I INTEGER 1;
        DECLARE J INTEGER;
        SET J = CARDINALITY(InputRoot.*[]);
        WHILE I < J DO
            SET OutputRoot.*[I] = InputRoot.*[I];
            SET I = I + 1;
        END WHILE;
    END;

    CREATE PROCEDURE CopyEntireMessage() BEGIN
        SET OutputRoot = InputRoot;
    END;
END MODULE;
```

Need to put a message on an WebSphere MQ queue, but first must create an MQMD header

Double click the Compute node (InsertMQMDHeader) to enter ESQL editor. Since the message is to be put to a WebSphere MQ queue, an MQMD message header must be created. Shown here is the code to create this header.

Scenario: MQOutput node



Places a message on a queue, so a back-end system can pick it up to order more inventory

MQOutput Node Properties - SUPPLIER.ORDER.Q00

Queue manager name

Queue name: NETCOMP.SUPPLIER.ORDER.Q00

MQOutput Node Properties - SUPPLIER.ORDER.Q00

Destination mode: Queue Name

Transaction mode: Automatic

Persistence mode: Automatic

New message ID □

New correlation ID □

Segmentation allowed □

Message context: Default

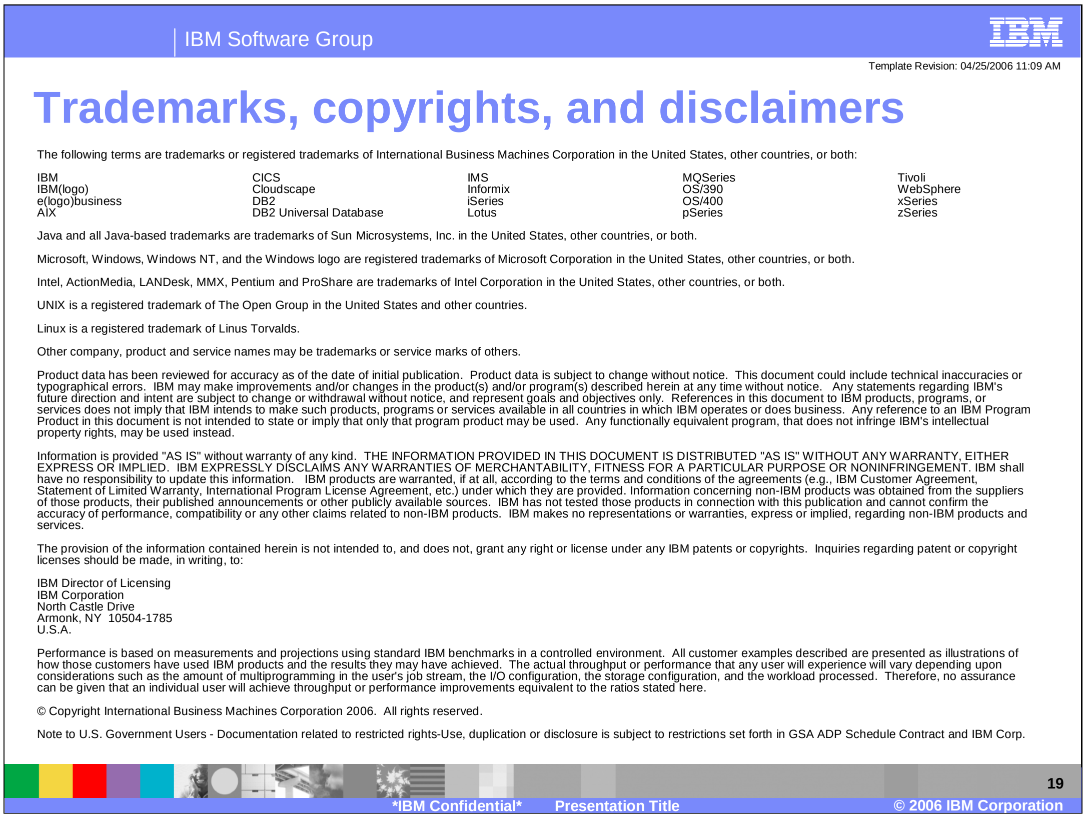
Alternate user authority □

18

WebSphere Message Broker Toolkit V6.0.2

© 2006 IBM Corporation

In flow editor, drop MQOutput node, set its name and Queue name property as shown, and wire it from the out terminal of the Compute node. This places a message on the queue so that a back-end system can process the message to order more inventory. This concludes part 5 of the WebSphere Message Broker V6.0.2 Toolkit scenario.



Template Revision: 04/25/2006 11:09 AM

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
IBM(logo)
e(logo)business
AIX
CICS
Cloudscape
DB2
DB2 Universal Database
IMS
Informix
iSeries
Lotus
MQSeries
OS/390
OS/400
pSeries
Tivoli
WebSphere
xSeries
zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

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 and/or changes in the product(s) and/or program(s) 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 NONINFRINGEMENT. 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 (e.g., 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.

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.