IBM Software Group | WebSphere Software

WebSphere Message Broker
What's New in Toolkit 6.0.2

David Hardcastle

IBM Hursley Lab

(david_hardcastle@uk.ibm.com)

SOA on your terms and our expertise

ON DEMAND BUSINESS™

Trademarks

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

AIX	AS/400	DB2
IBM	iSeries	MQSeries
MVS	OS/390	SupportPac
System/390	WebSphere	xSeries
z/OS	400	

- Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc, in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Other company, product, and service names may be trademarks or service marks of others.

Agenda

- Why a 6.0.2 toolkit release?
- Key AD improvements
  - Illustrated using a Demo !
- Testing
- Other Improvements
- Looking ahead

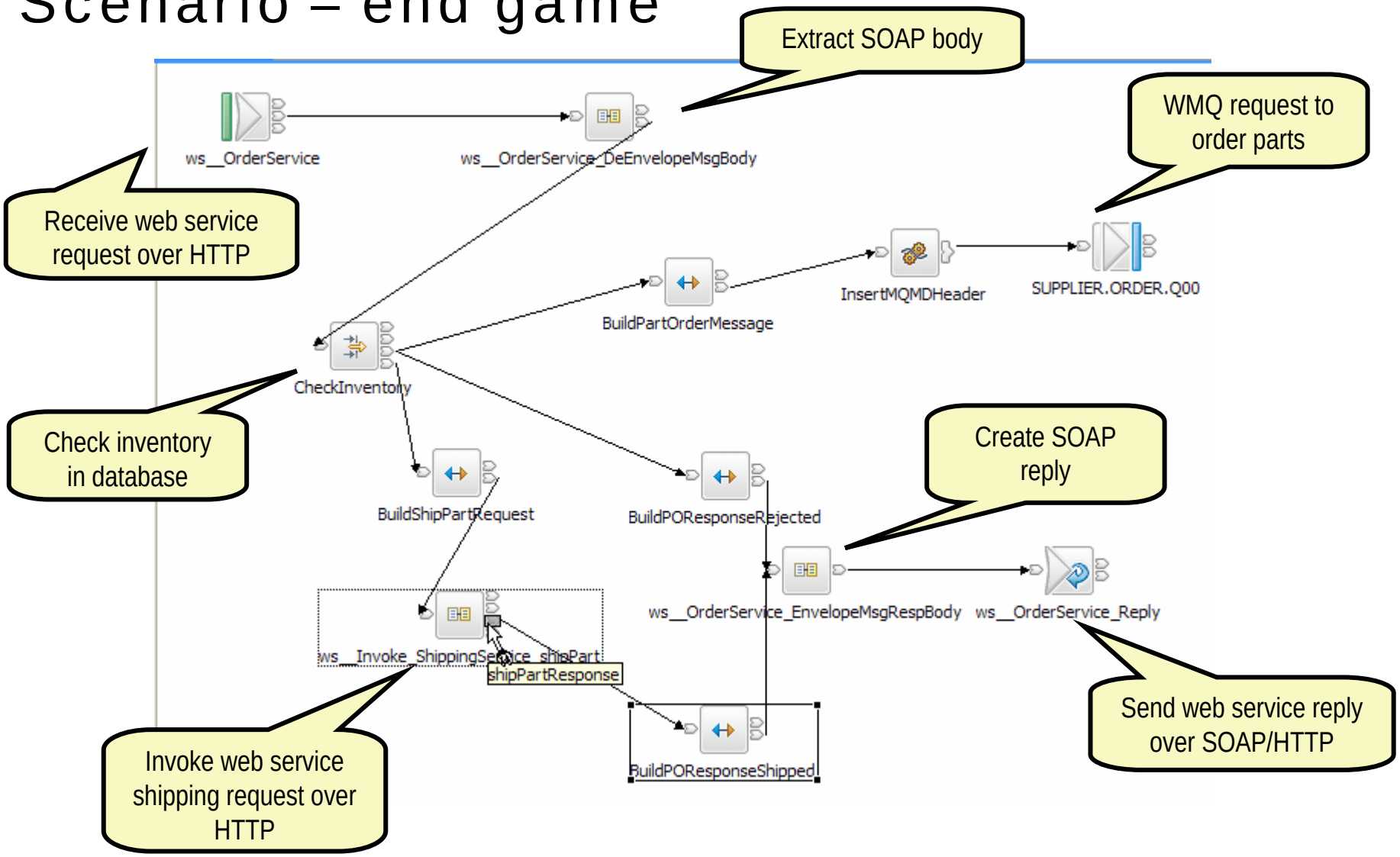
About the 6.0.2 toolkit

- **Why**
  - There is opportunity for improvement in the helping new users get up to speed quickly with the Toolkit
- **What**
  - Aimed primarily at simplifying out of box experience
  - Primarily targeting core ESB-like scenarios
    - But not strictly limited to that
  - Improving “time to value”
    - Hopefully changes benefit all existing users too
- **When**
  - Some improvements in Fix Pack 2 (6.0.0.2)
  - New **6.0.2** release of Toolkit in December 2006
  - Via RPU Upgrade

Key AD improvements

- Illustrated using a simple scenario
- Order Processing
  - Expose a web service (SOAP/HTTP) for placing purchase order
  - Flow will check for sufficient inventory (stock)
  - Insufficient stock for order…
    - Place order from supplier via MQ message to CICS (say)
    - ‘Out of stock’ response to caller
  - Sufficient stock for order…
    - Invoke different web service to place shipping order
    - Positive response to caller

Scenario – end game

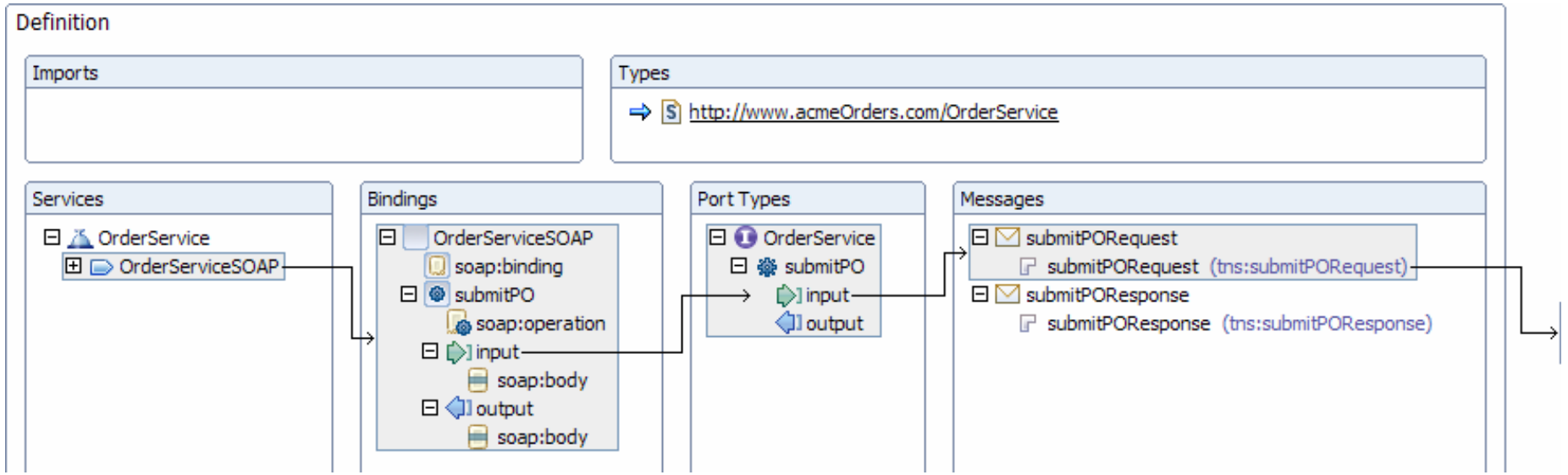


Scenario - challenges

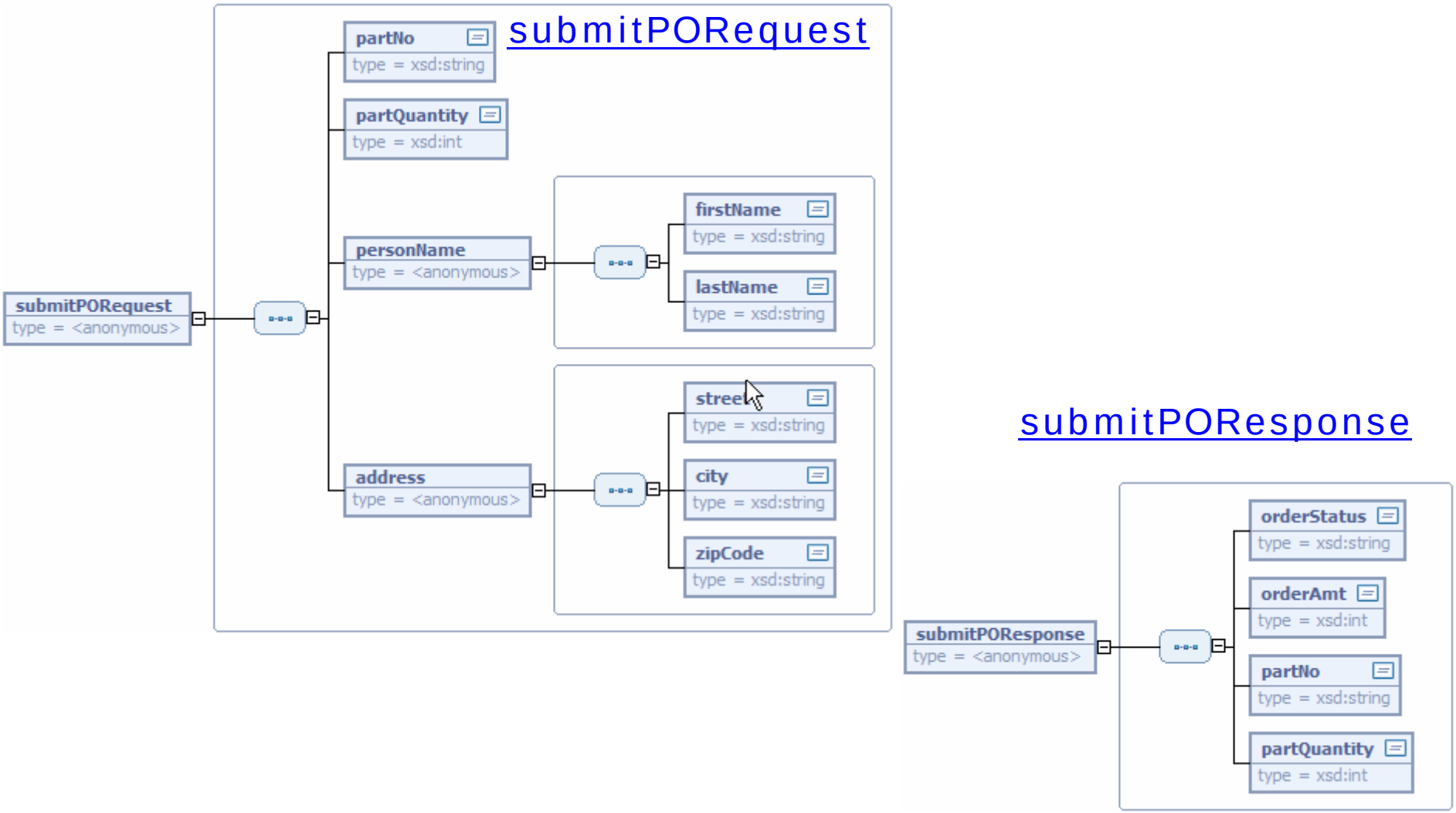
- Creating message set from WSDL and XSD files
  - WSDL for the web service purchase order operation to be exposed
  - WSDL for the web service shipping operation to be invoked
- Configuring and using HTTP nodes
  - Input and reply for the web service purchase order operation to be exposed
  - Request for the web service shipping operation to be invoked
- Dealing with the SOAP messages
  - Body has to be extracted from SOAP envelope for easy mapping
- Dealing with mixed protocols
  - Body has to be placed in a SOAP envelope to invoke web service
  - WMQ headers have to be created to use MQ output
- Choosing/finding the appropriate nodes to use for each task
- Testing … how to send a SOAP message to HTTP Input node and see the reply from the HTTP Reply node?

Scenario: WSDL for Purchase Order

OrderService.wsdl

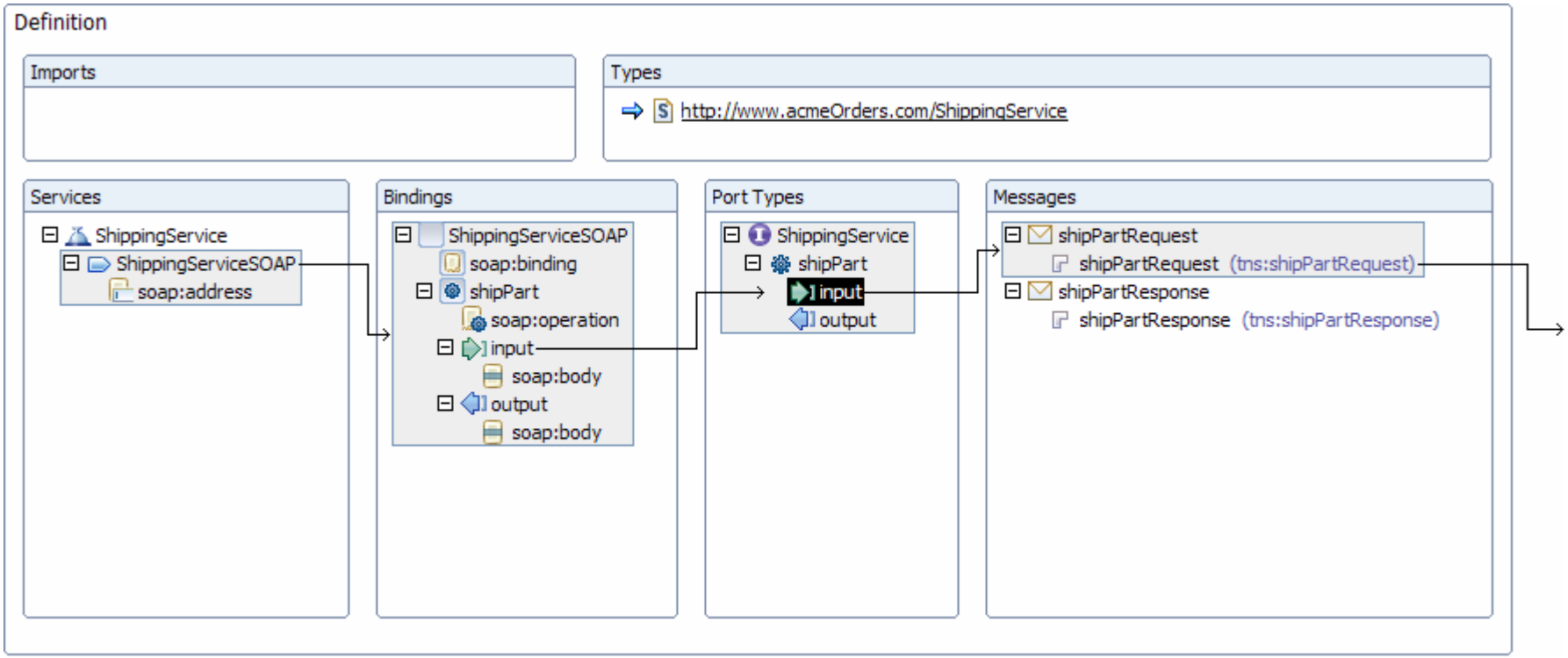


Scenario: XSDs for Purchase Order

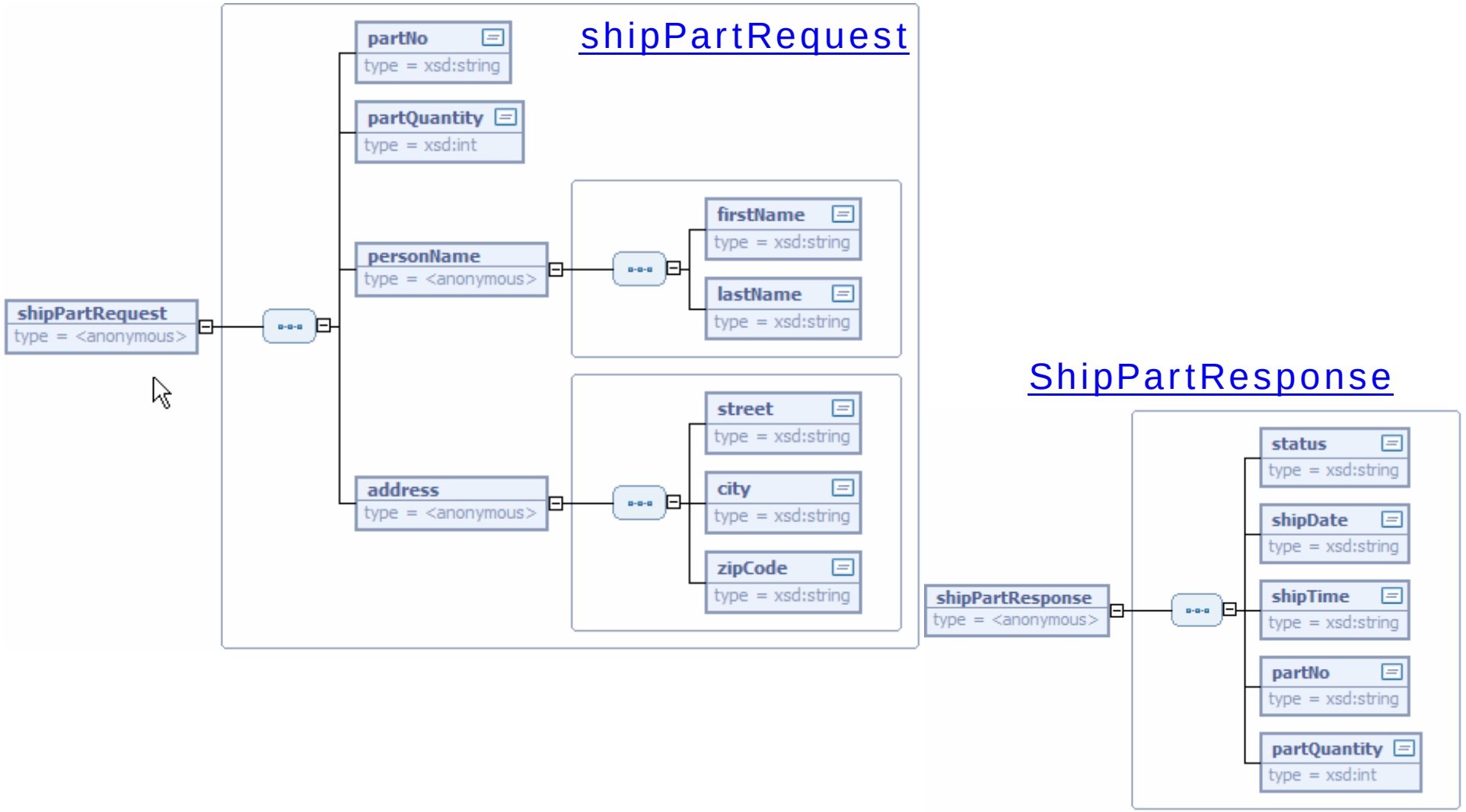


Scenario: WSDL for Shipping Service

ShippingService.wsdl

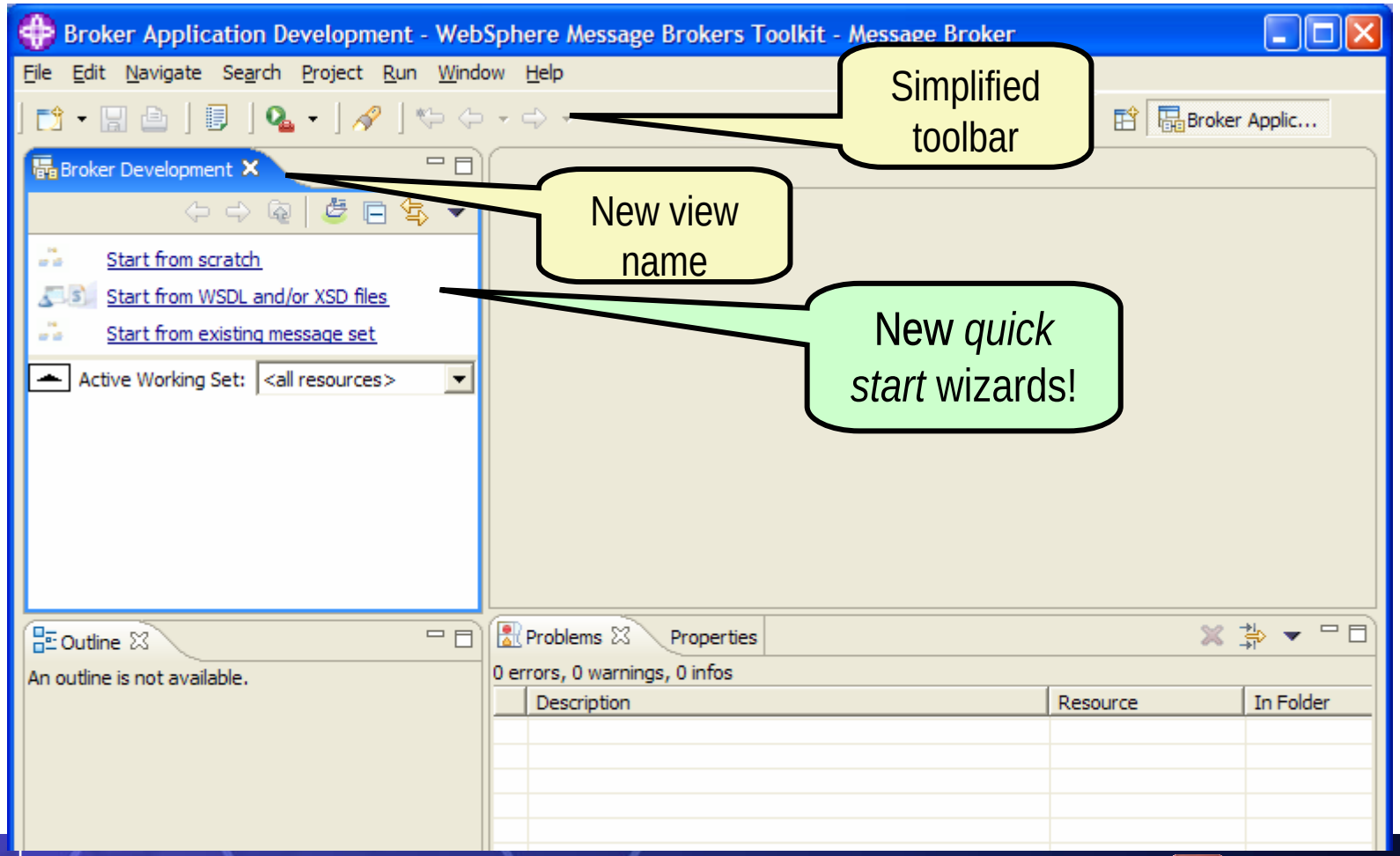


Scenario: XSDs for Shipping Service



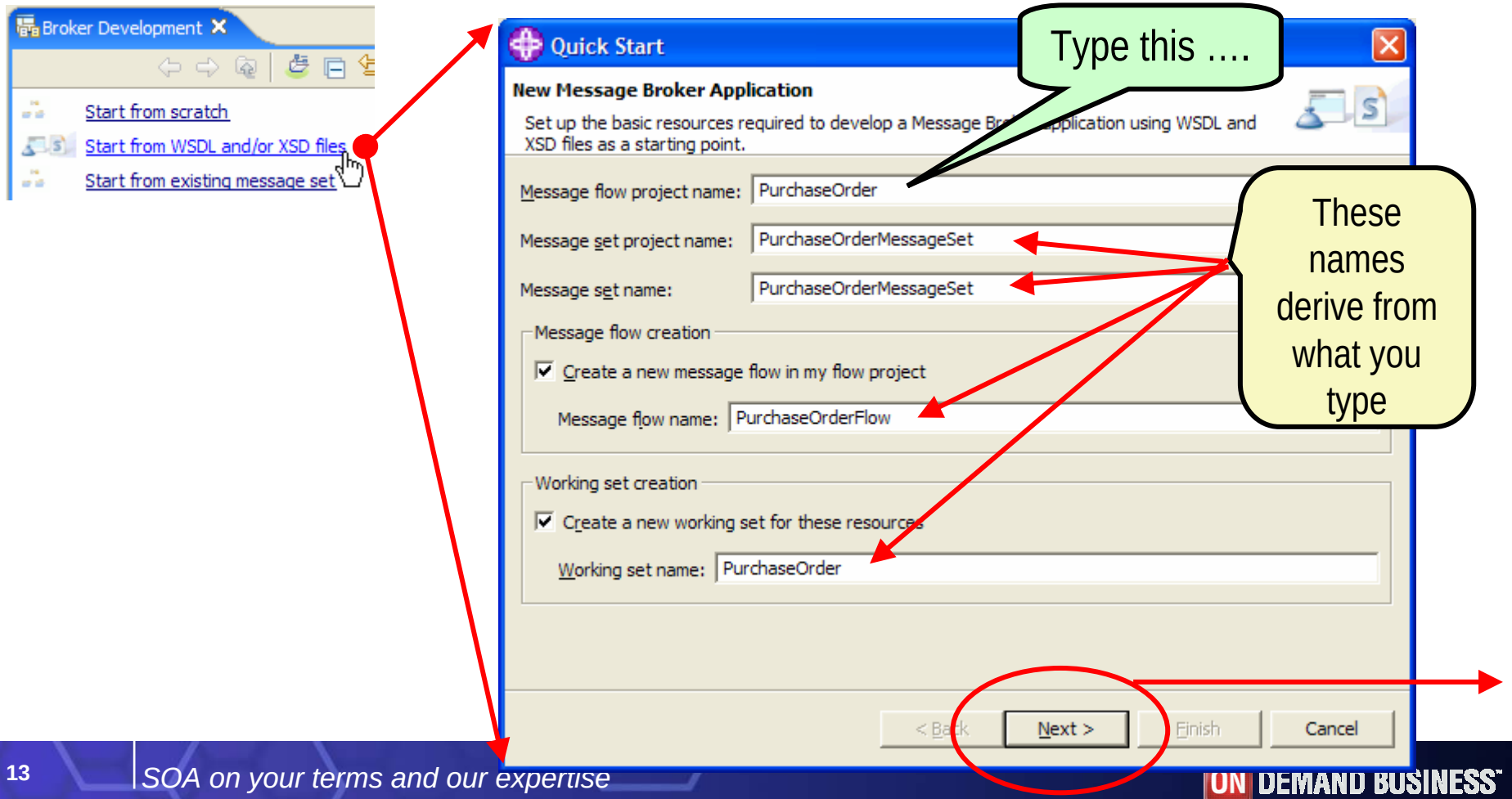
Scenario: Step 1

- Where do I begin? Fire up Toolkit 6.0.2!



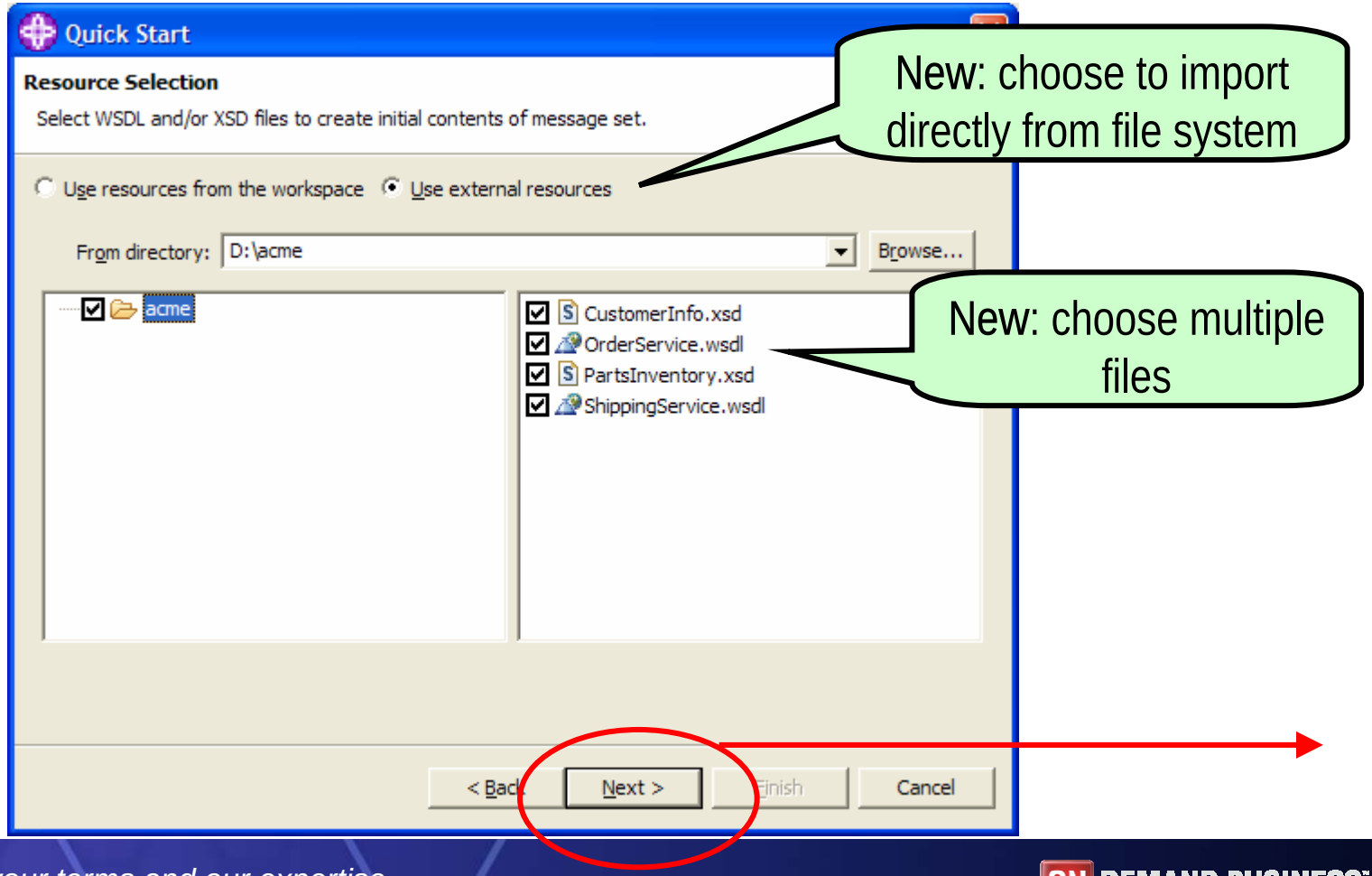
Scenario: Step 2a – page 1

- Choose root name for new “application”



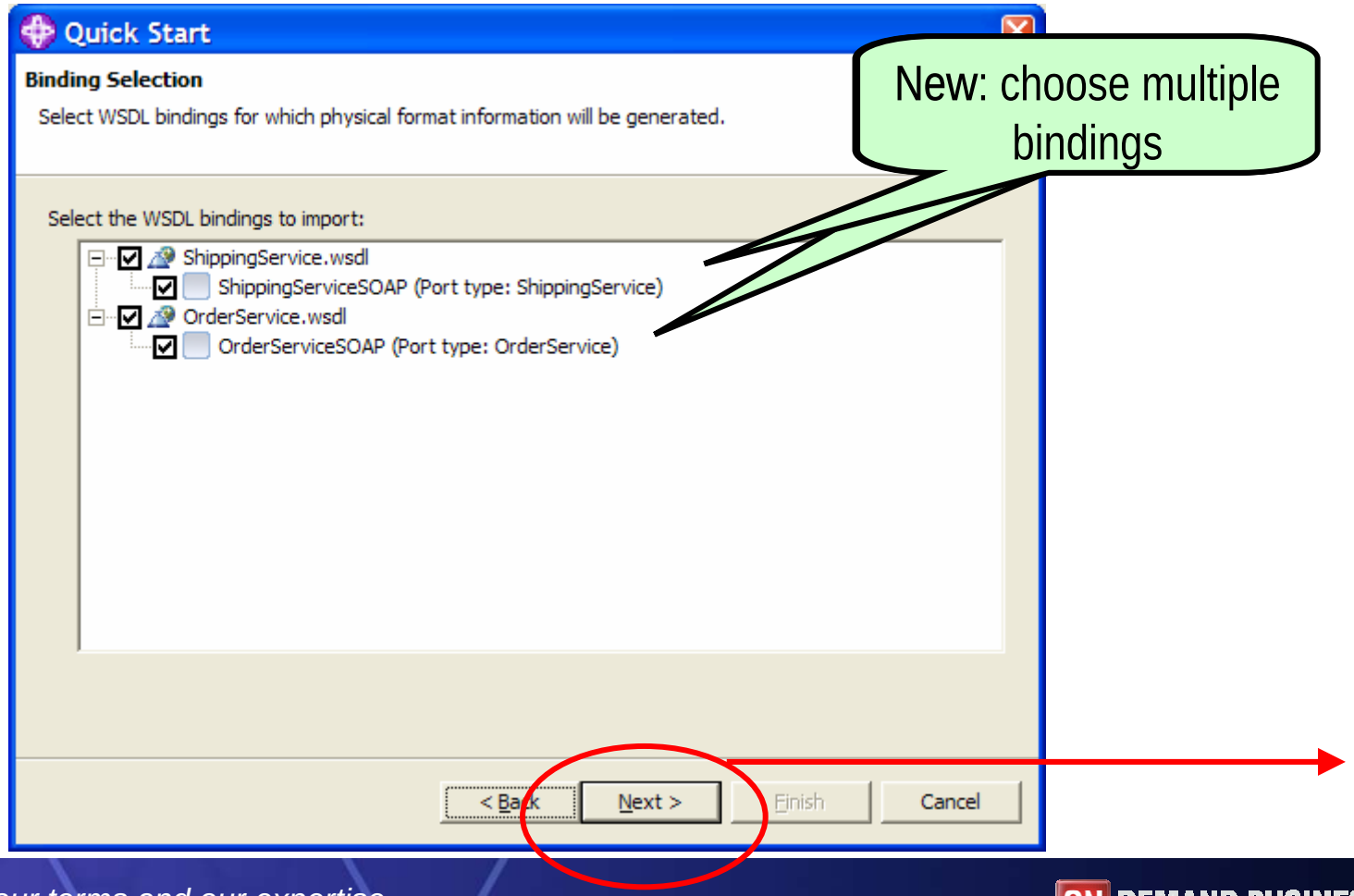
Scenario: Step 2b – page 2

- Choose WSDL and XSD files from file system



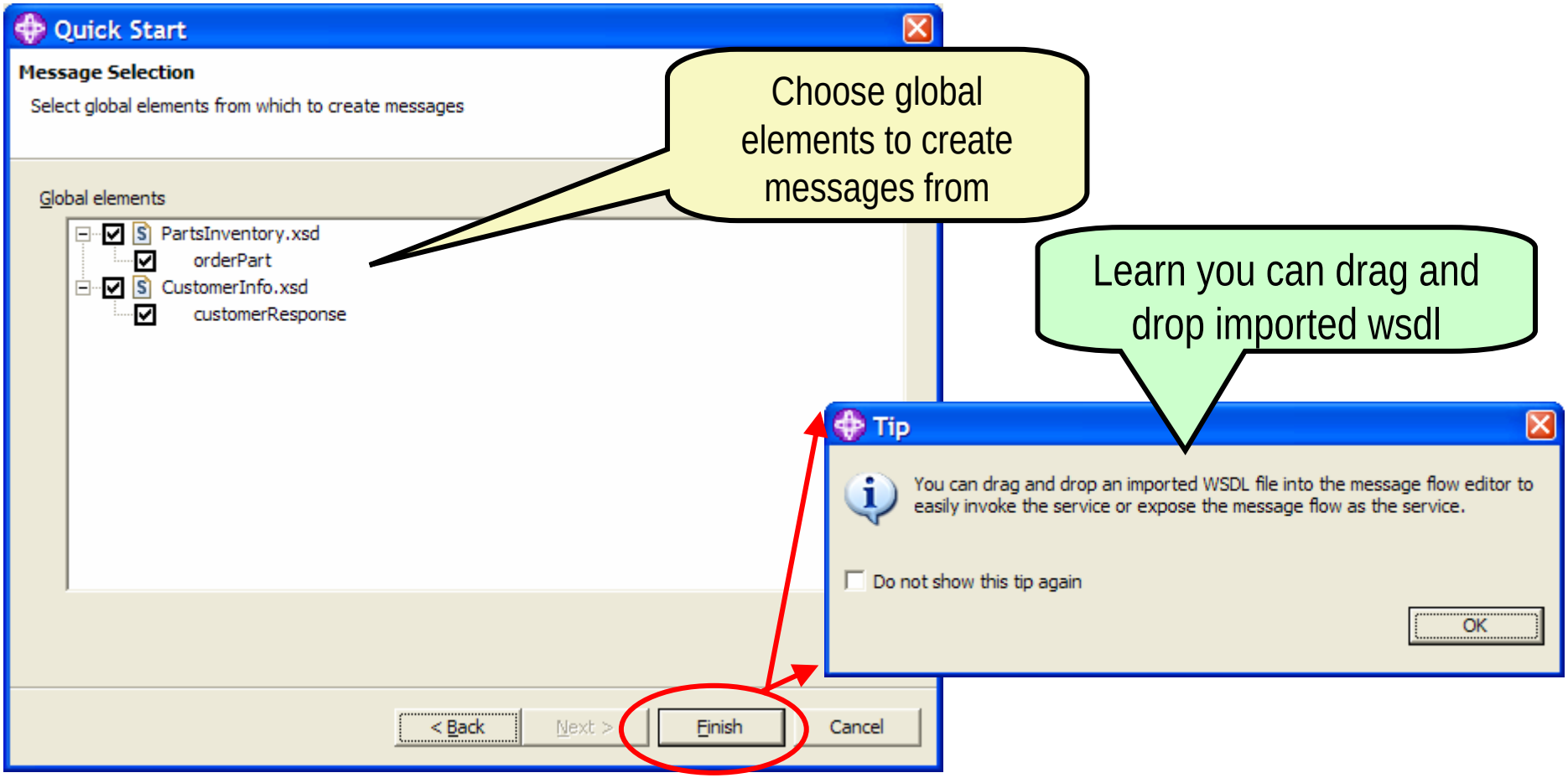
Scenario: Step 2c – page 3

- Choose bindings from which to model messages

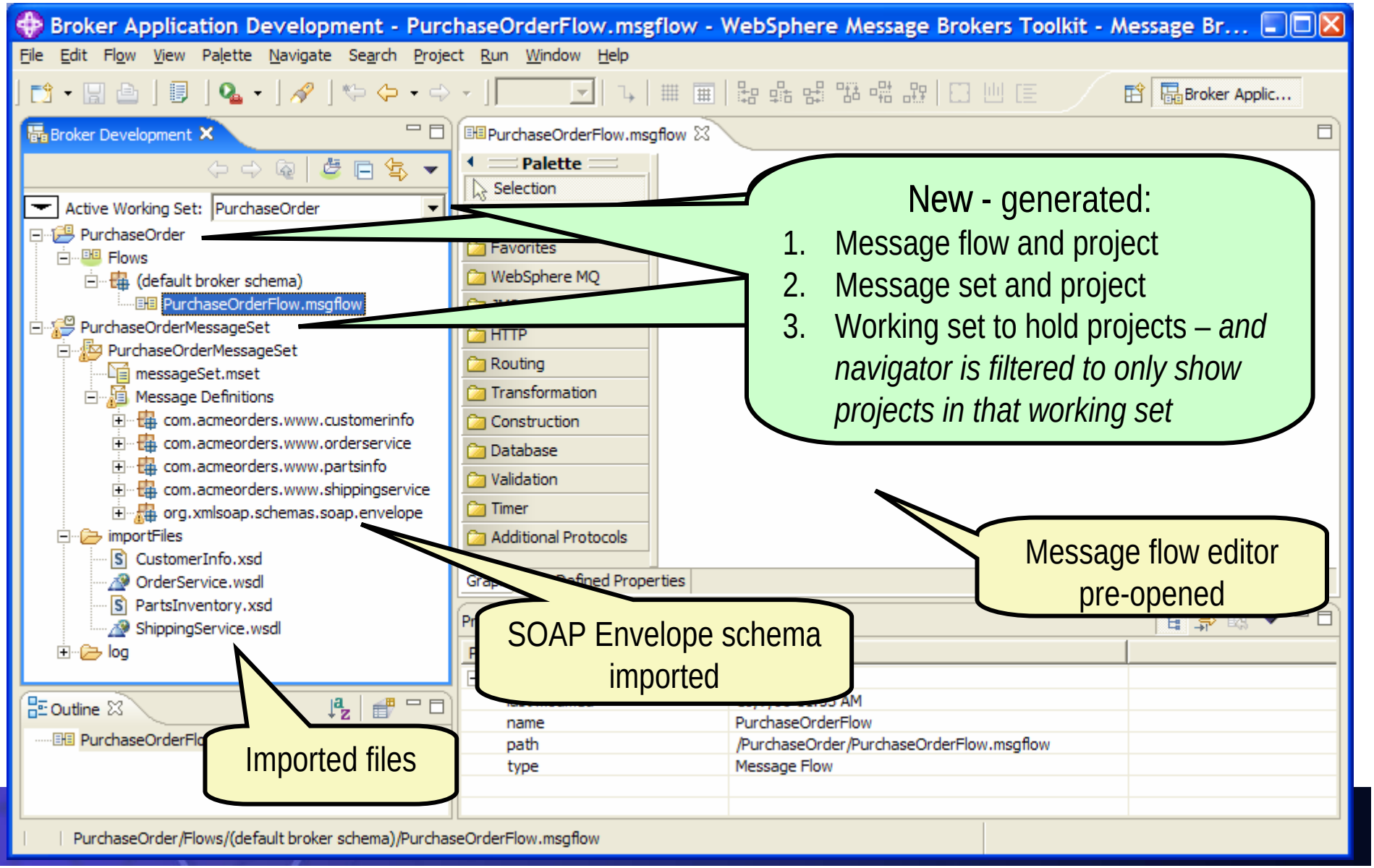


Scenario: Step 2d – page 4

- Choose which XSD messages to model

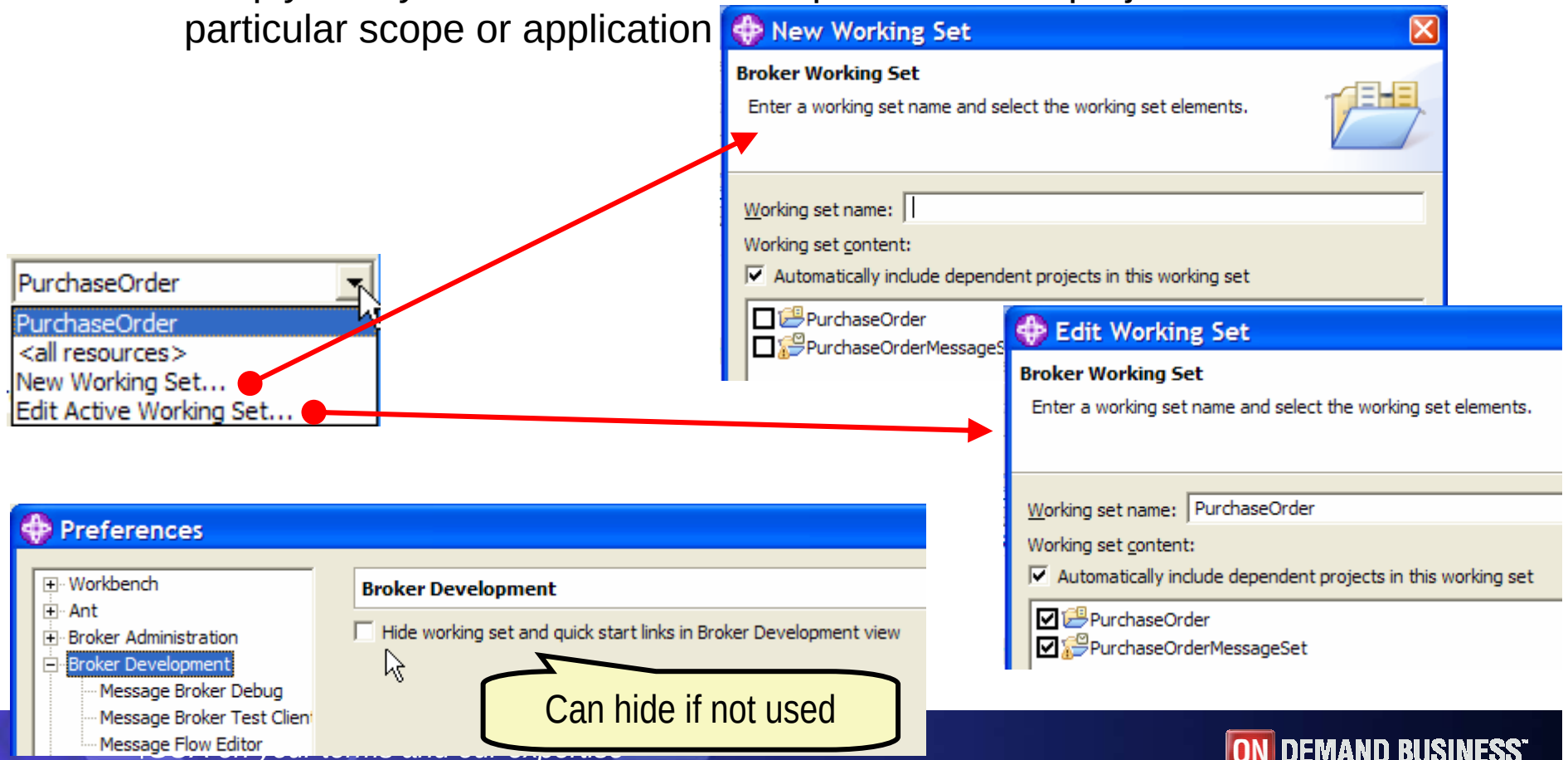


Scenario: Step 2e – after wizard



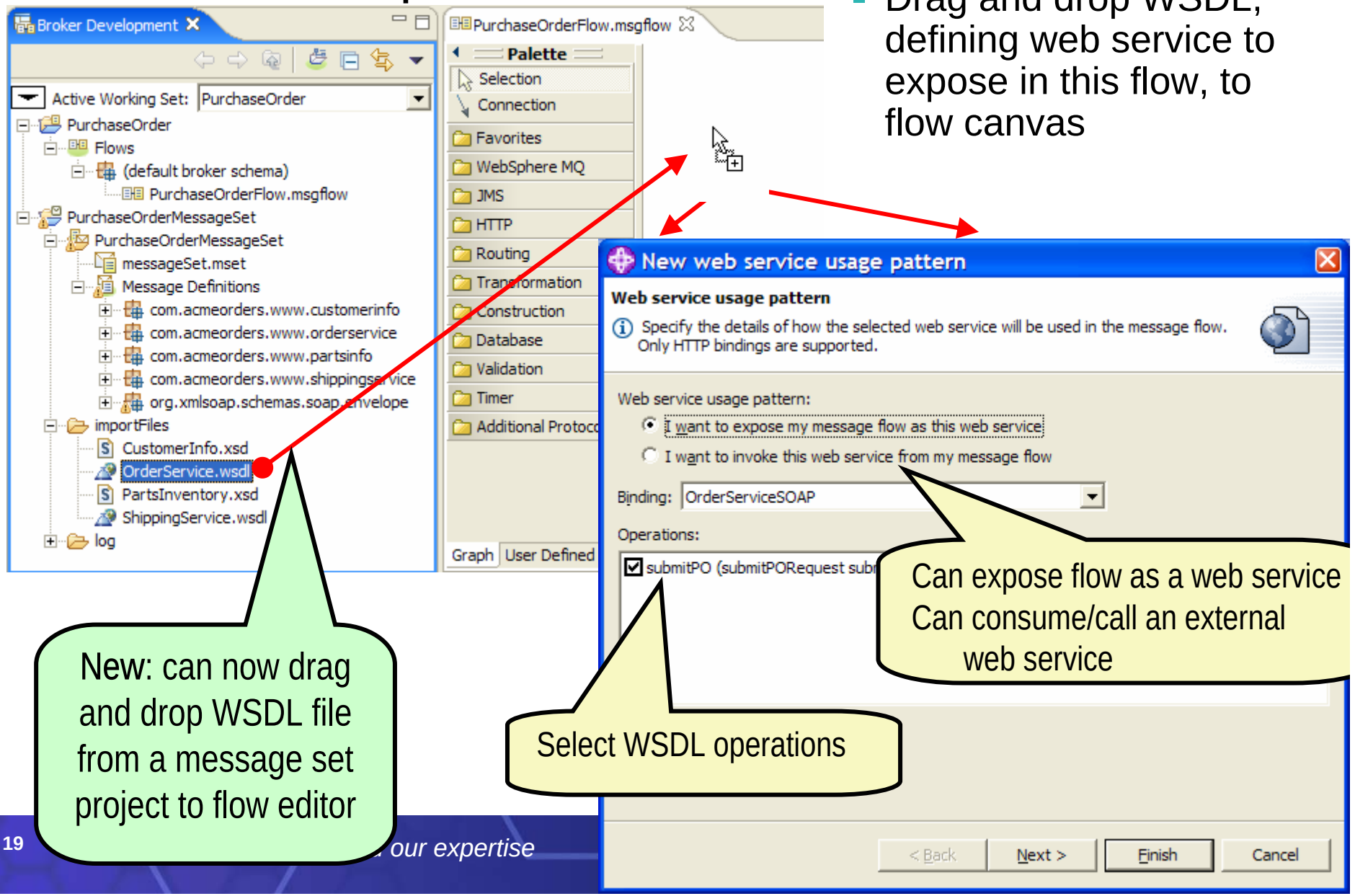
About Working Sets

- New for Broker Development perspective in 6.0.2 … but an existing eclipse concept. A named group containing one or more projects.
  - Simply a way to filter Broker Development view to projects related to particular scope or application

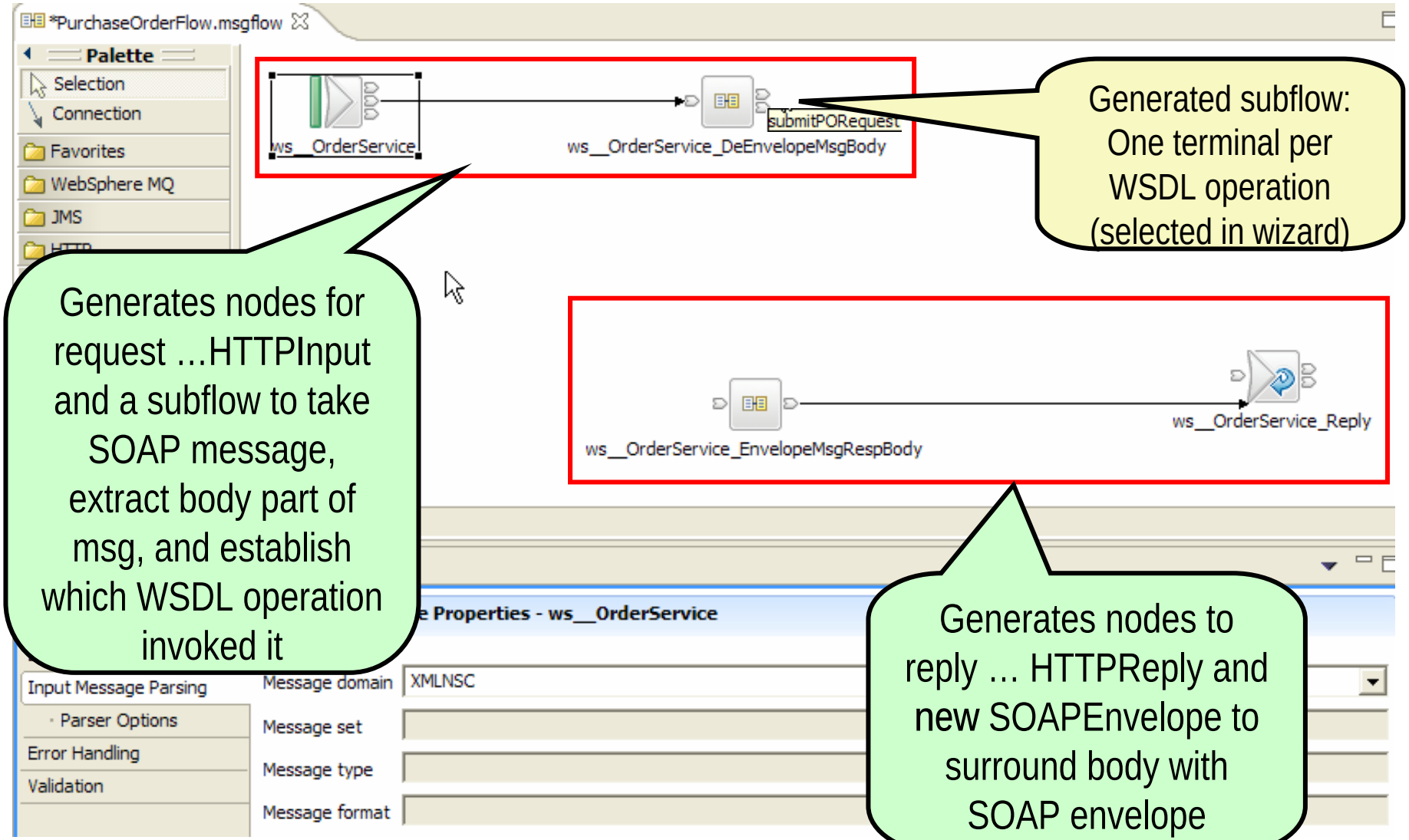


Scenario: Step 3

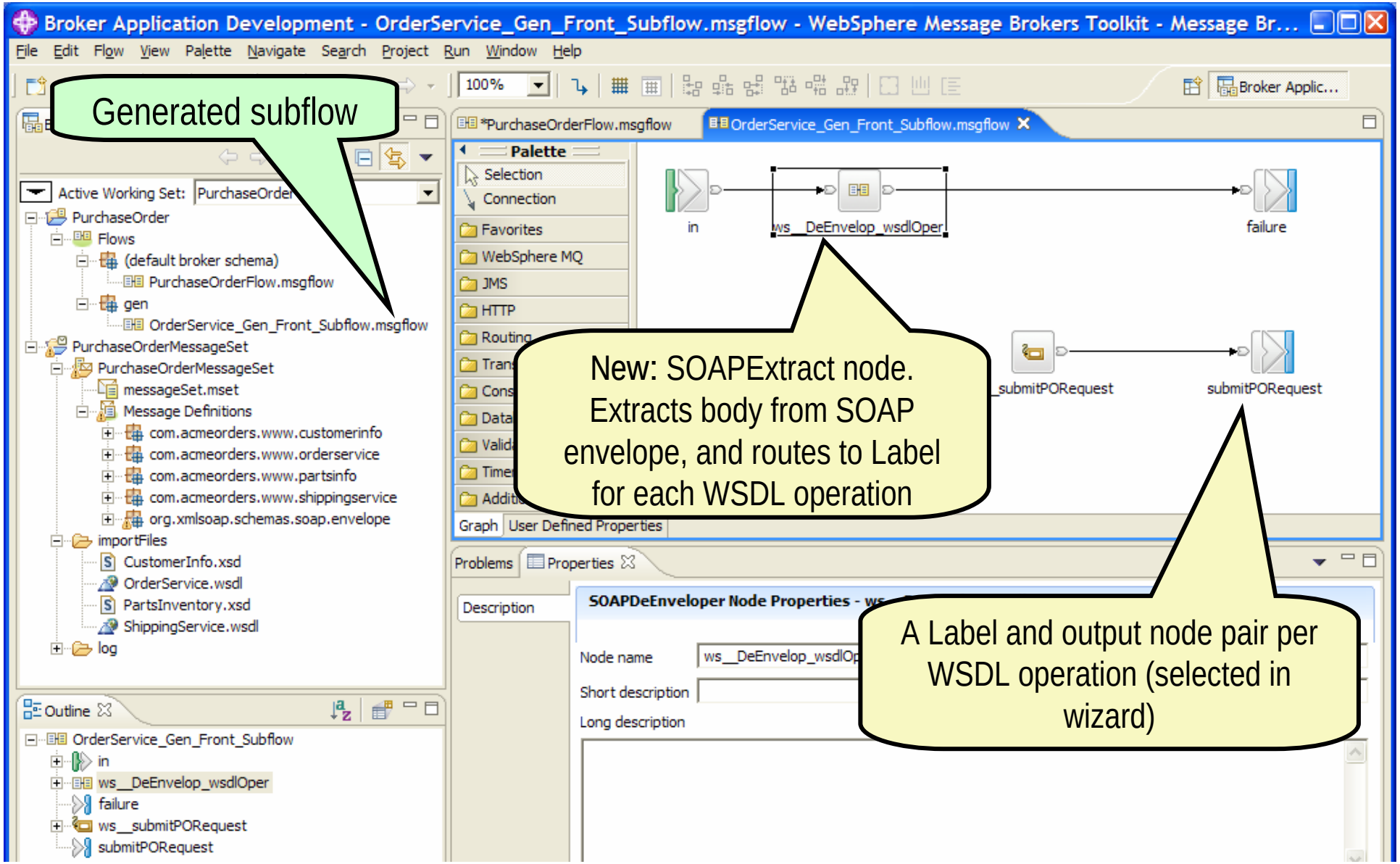
- Drag and drop WSDL, defining web service to expose in this flow, to flow canvas



Scenario: Step 3 – after drag and drop



Scenario: step 3 – Inside the subflow

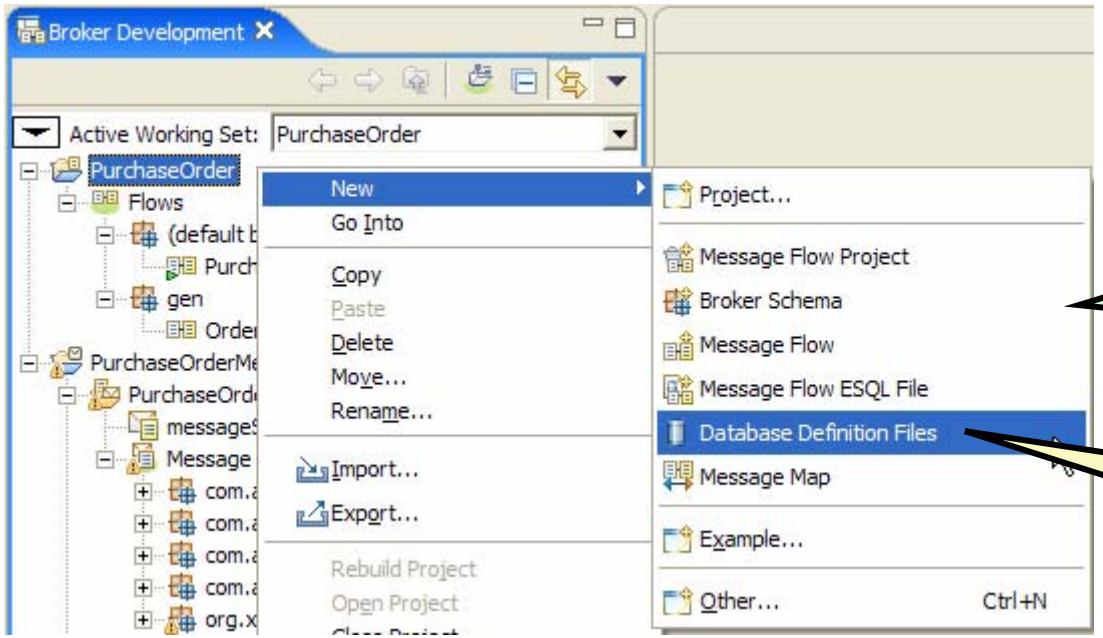


About Drag and Drop support

- We are increasing drag and drop support over time.
- As of 6.0.2, you can now drag and drop the following onto the flow editor:
  - **WSDL** files from message sets, as we have seen.
  - **XSL** files to create and/or configure XMLTransformation nodes
  - **Java** classes to create JavaCompute node and configure its class name property
  - **ESQL** modules to create Compute node and configure its ESQL module name property
  - Sub-flows, as before

Scenario: Step 4

- Create database connection for inventory table

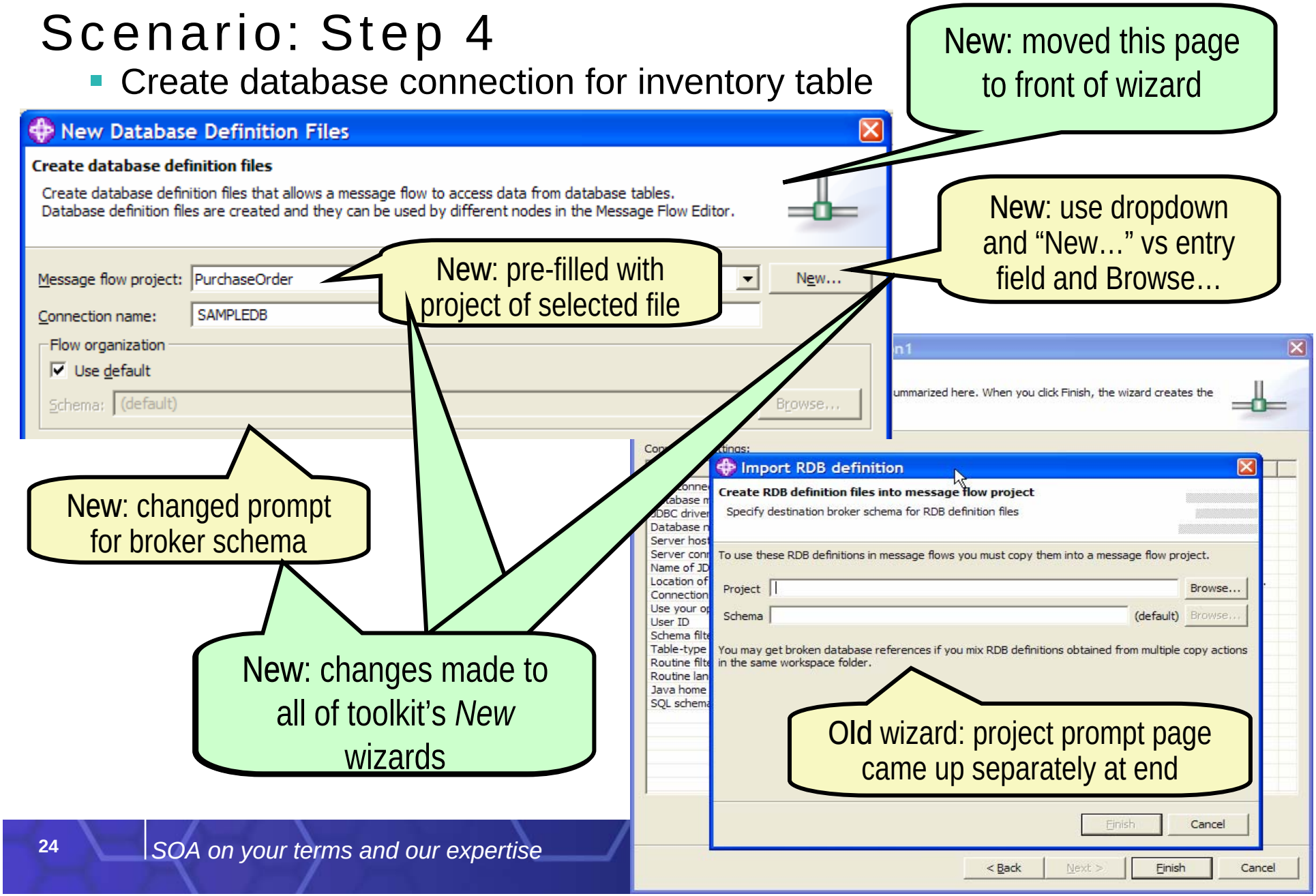


Simplified New menus: only show what is applicable for current selection

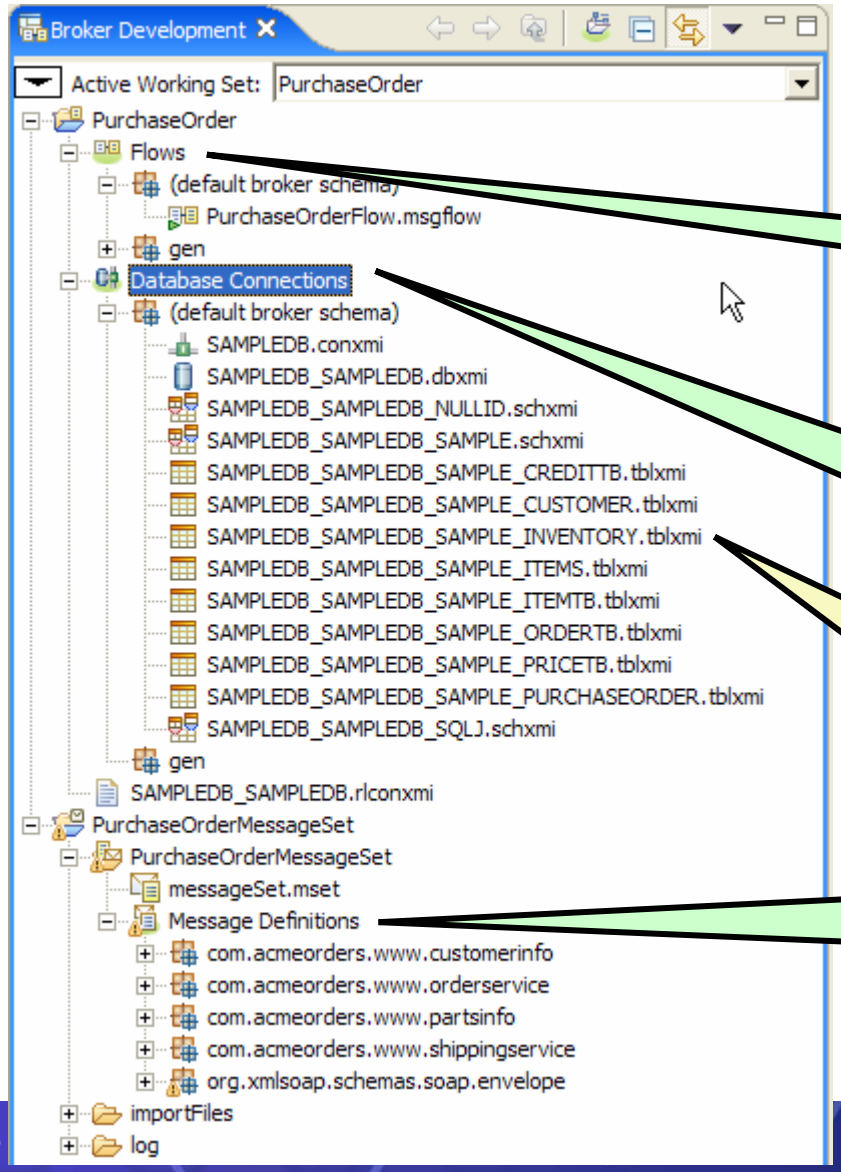
Renamed from “RDB Definition Files”

Scenario: Step 4

- Create database connection for inventory table



Scenario: Step 4



- The navigator after database connection created…

New: Flows category

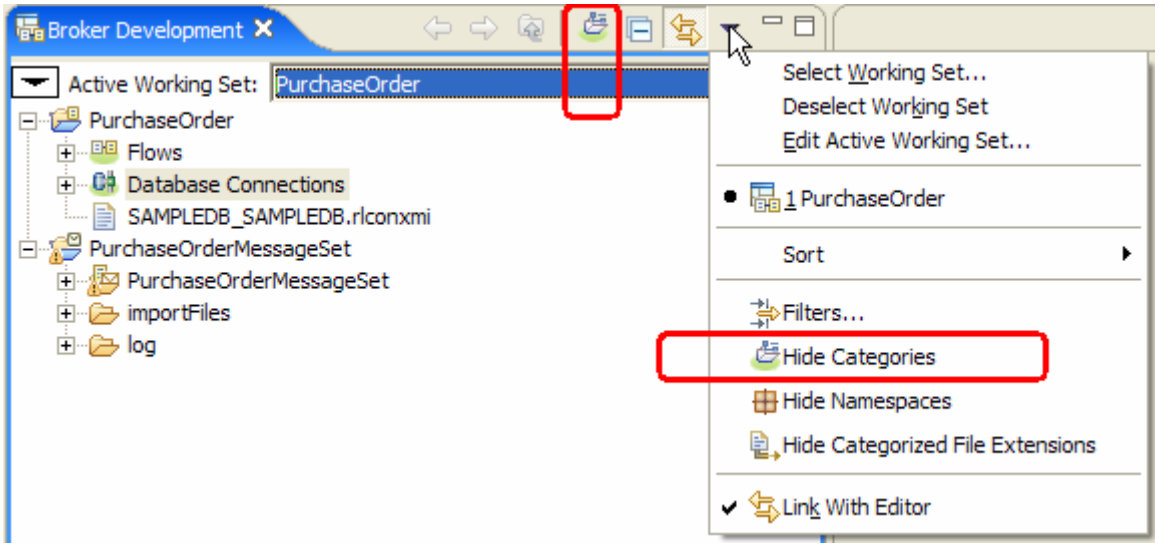
New: Database Connections category

The table we are interested in

New: Message Definitions category

About Categories

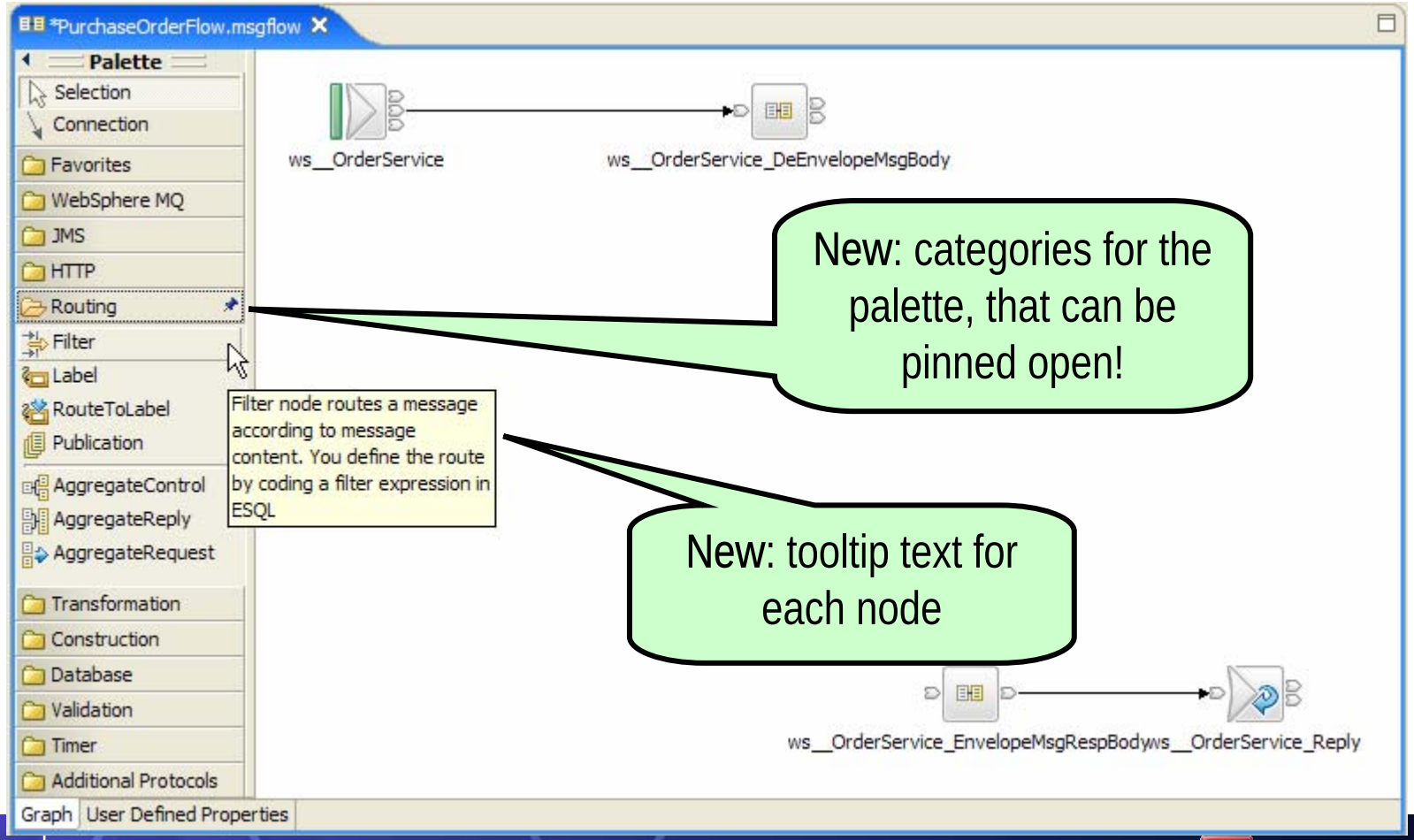
- New for 6.0.2 are categories shown in the navigator, to help users understand at a glance what they are looking at:
  - Message Flow projects:
    - Flows
    - Maps
    - ESQLs
    - Database Connections
    - Enqueues
  - Message Set projects:
    - Message Definitions
    - Message Categories



- You can turn this off
  - Can also hide:
    - Namespaces
    - Broker schemas
    - File extensions

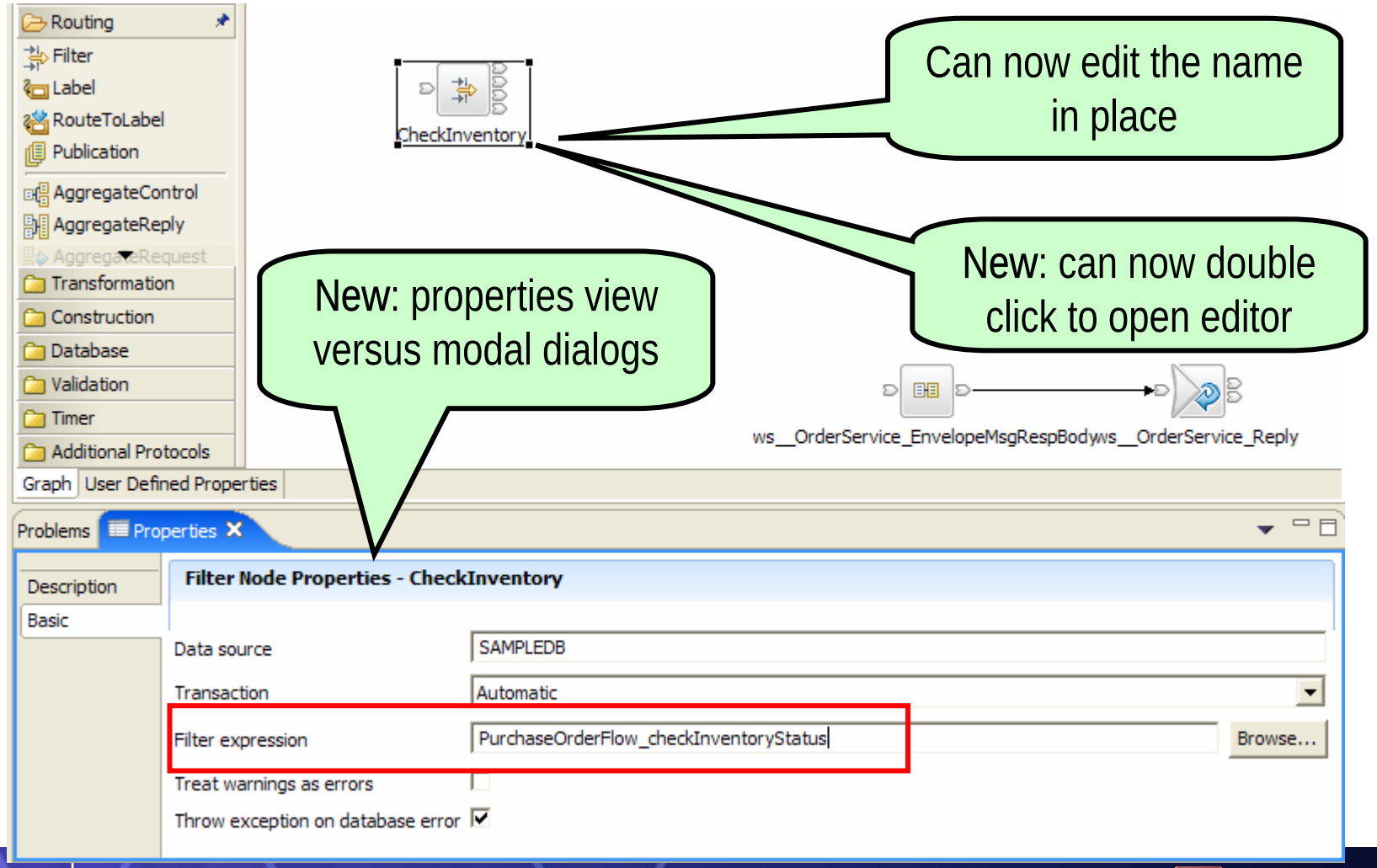
Scenario: Step 5

- Look for a node to help us route message differently depending on inventory status

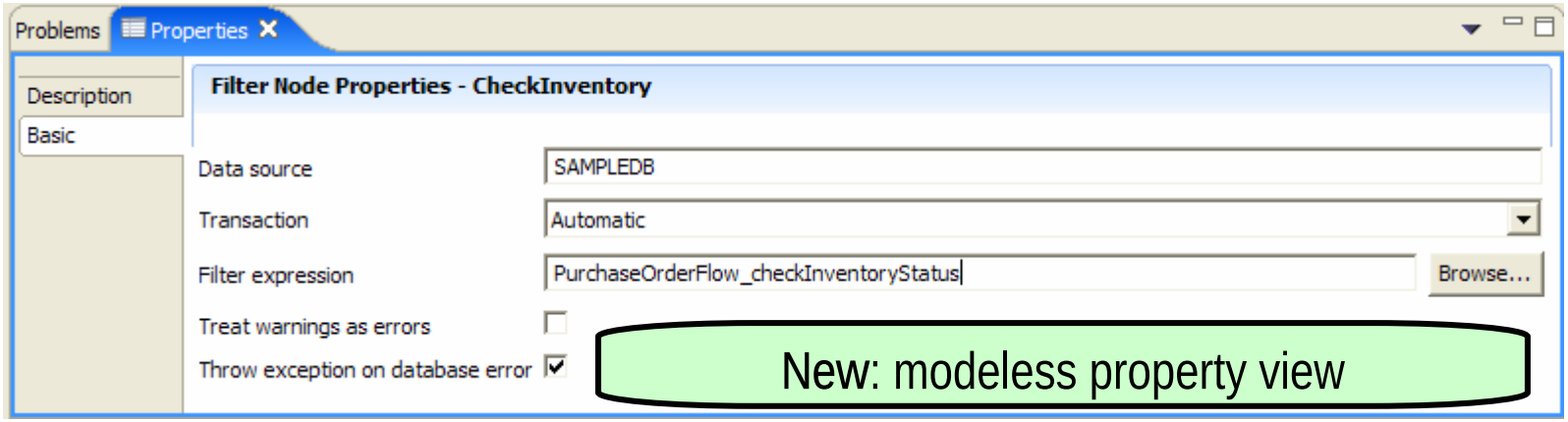


Scenario: Step 5a

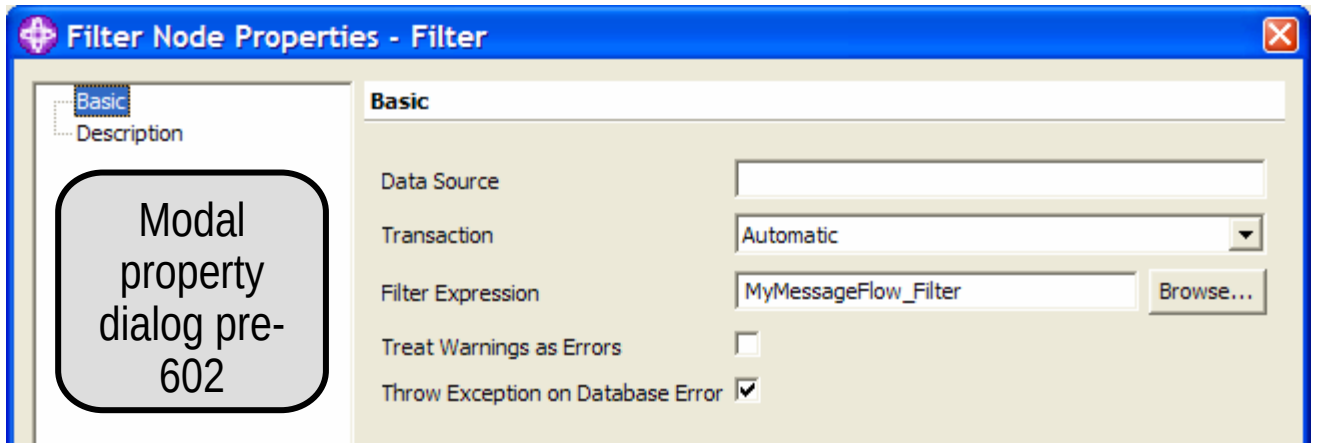
- Drop a Filter node, rename it, and set expression property



About properties view

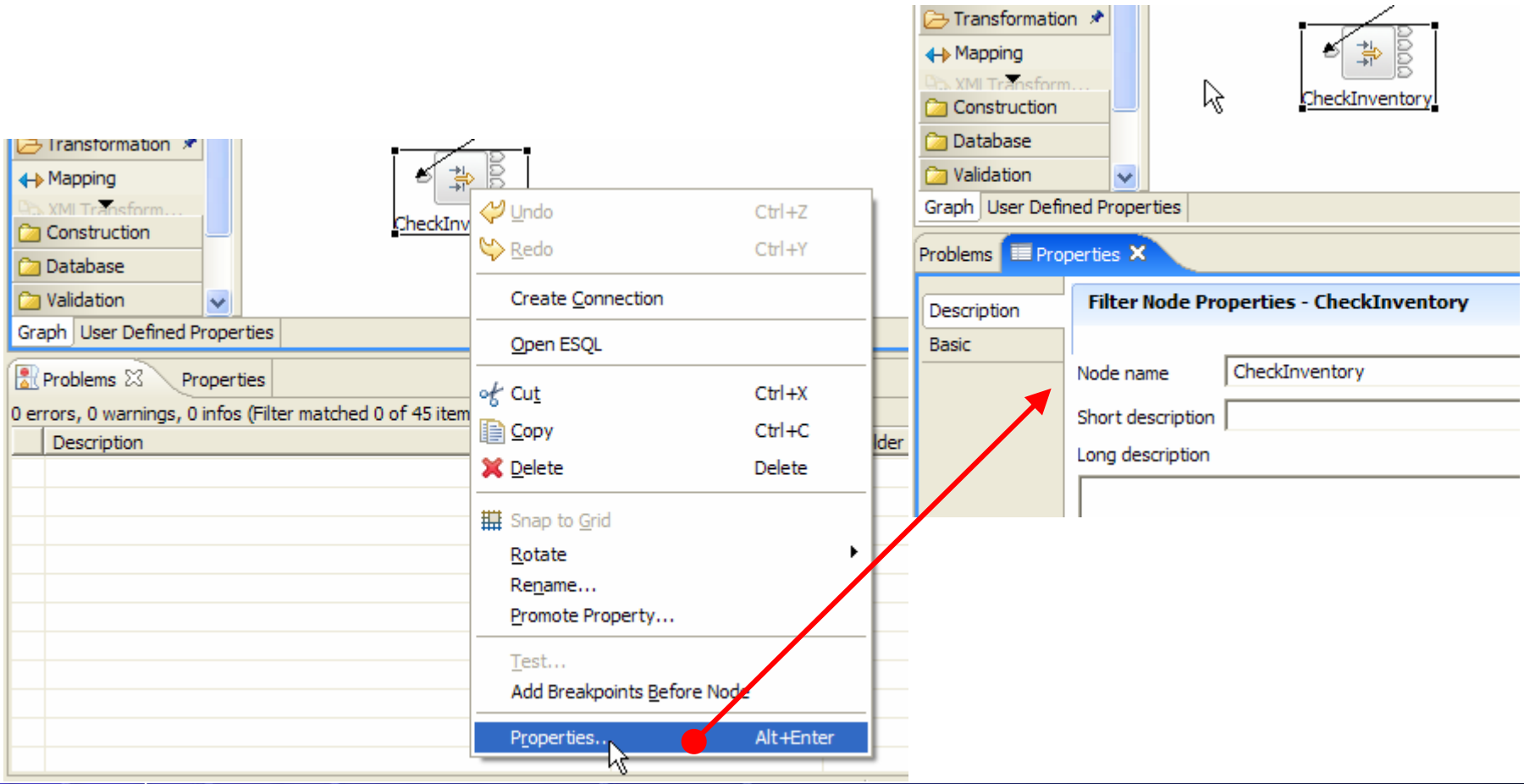


versus



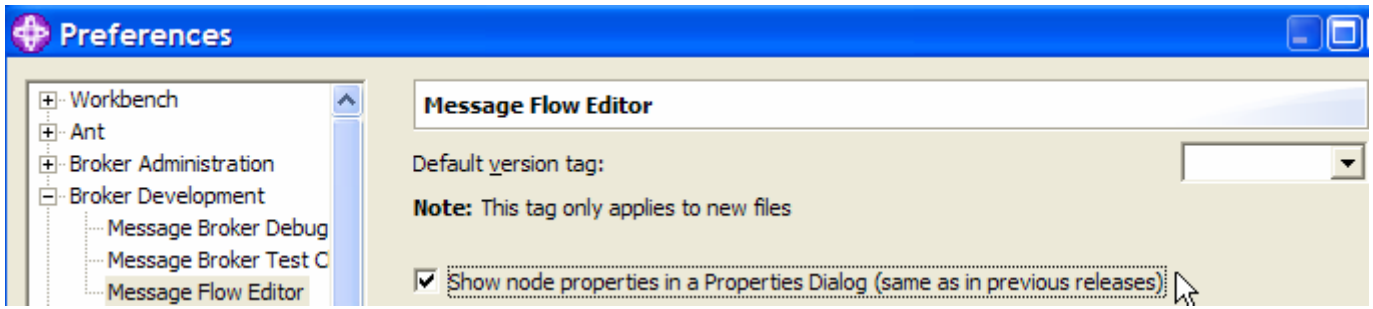
Launching properties view

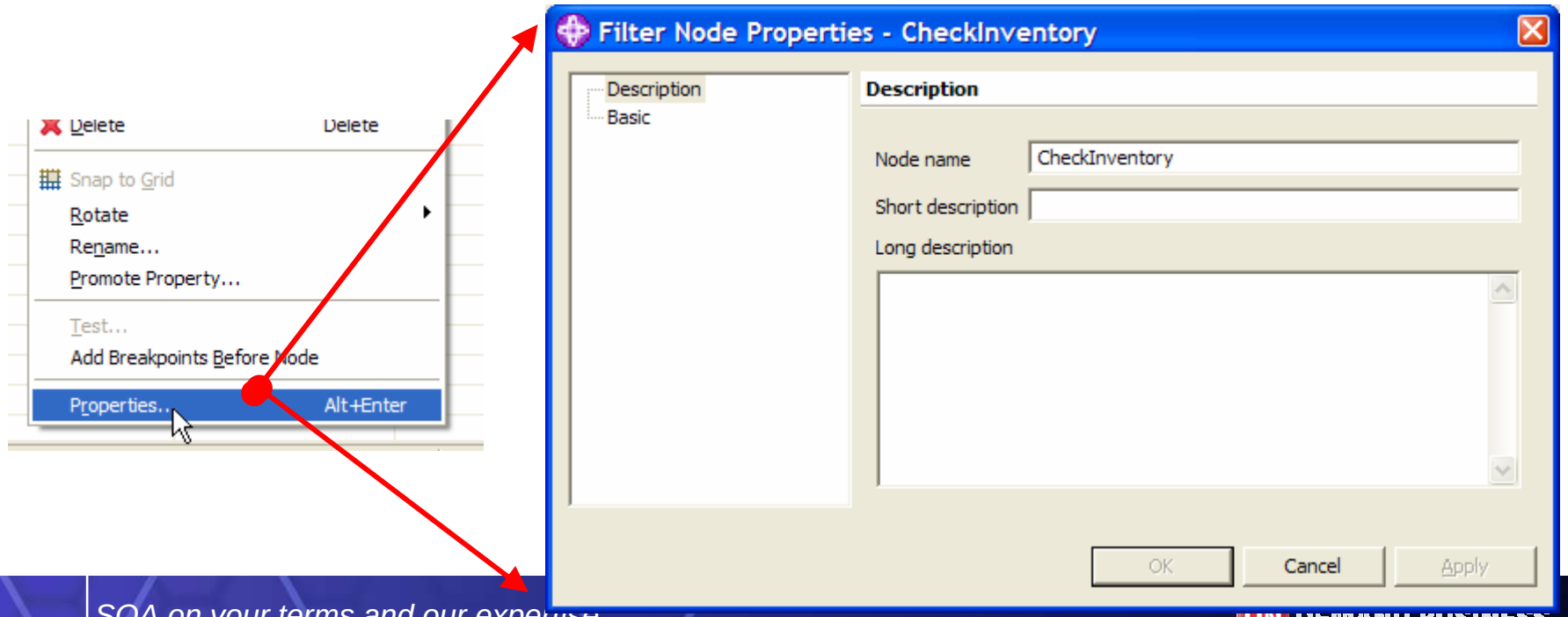
- Right-click Properties action just gives Properties view focus



Restoring Properties Dialog

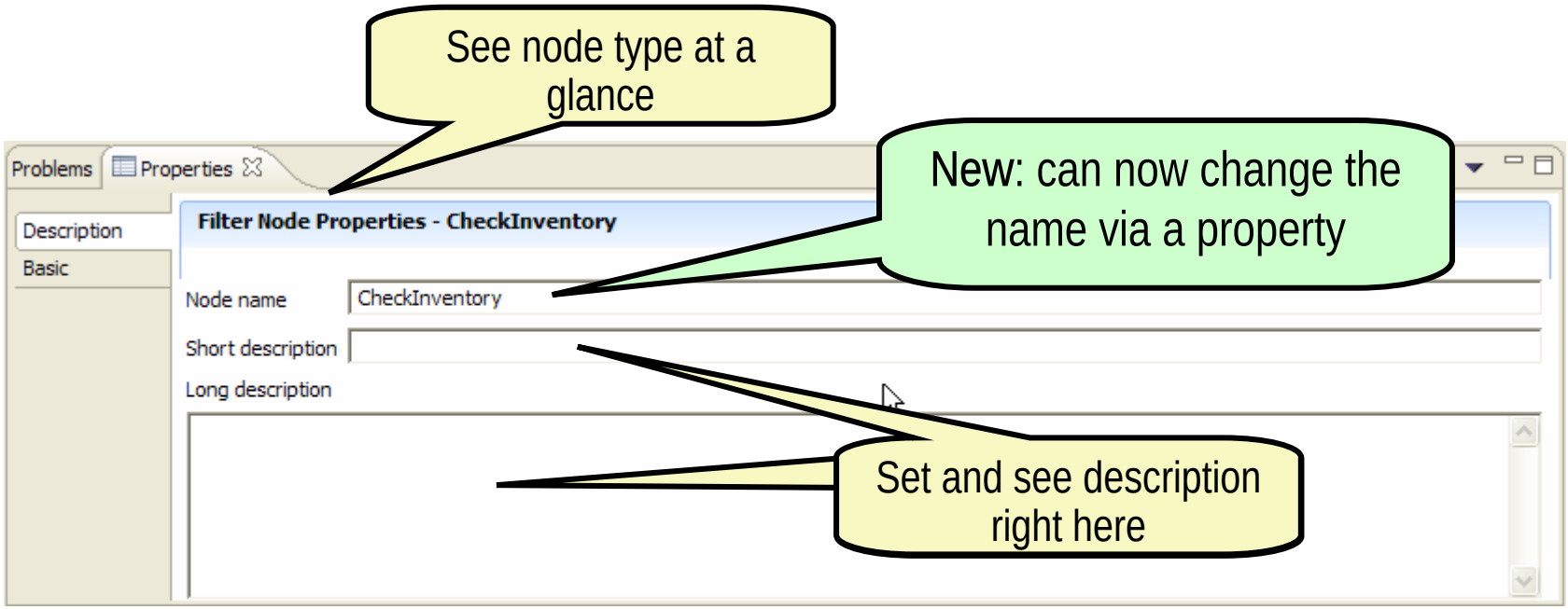
- You can change default behavior to launch Properties dialog if you prefer



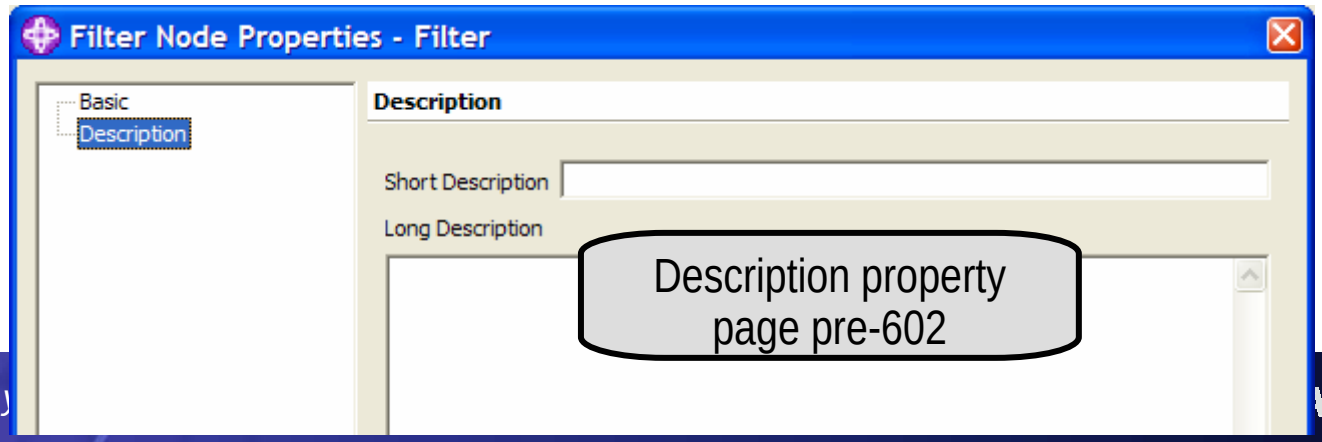


About the Description property tab

- All nodes support the same “Description” as *first* property tab



versus



Scenario: Step 5b

- Double-click on Filter node (**new!**) to open ESQL editor.
- Write ESQL logic to check inventory database (as before)

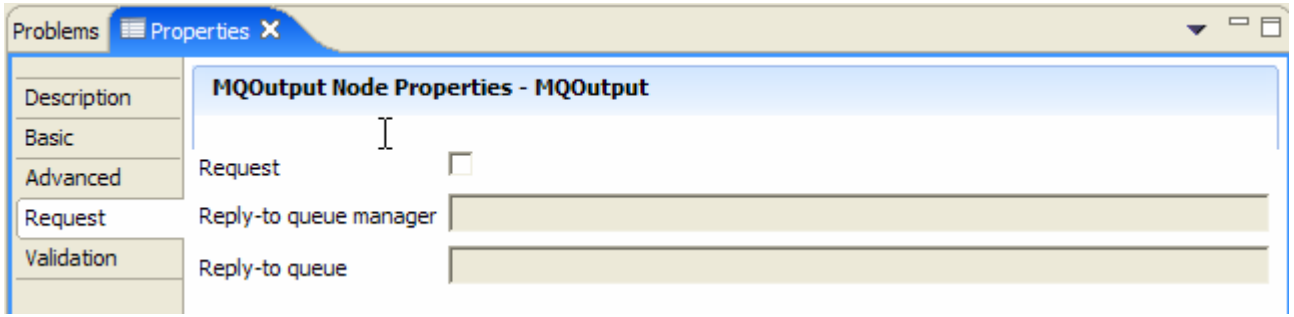
PurchaseOrderFlow.msgflow | PurchaseOrderFlow.esql

```
DECLARE ns NAMESPACE 'http://www.acmeOrders.com/OrderService';


CREATE FILTER MODULE PurchaseOrderFlow_checkInventoryStatus
    CREATE FUNCTION Main() RETURNS BOOLEAN
    BEGIN

        DECLARE partNo REFERENCE TO Root.XMLNSC.ns:submitPORequest.partNo;

        DECLARE partQty INTEGER;
        DECLARE partQtyAvailable INTEGER;

        -- extract the cpartNo from message

        SET partQty = Root.XMLNSC.ns:submitPORequest.partQuantity;

        --if the customerID cannot be found in the message then we return unknown
         IF (partNo is NULL) THEN
         RETURN UNKNOWN;
         END IF;

        -- search the inventory table for the part No
        SET partQtyAvailable = THE (SELECT ITEM T.QUANTITY FROM Database.SAMPLE.INVENTORY AS T WHERE T.PARTNO = partNo);

        --if part quantity requested is less than part quantity available return true
        IF (partQty  >= partQtyAvailable ) THEN
            RETURN FALSE;
        ELSE
            RETURN TRUE;
        END IF;


    END;


END MODULE;
```

New: improved validation and completion for reference fields

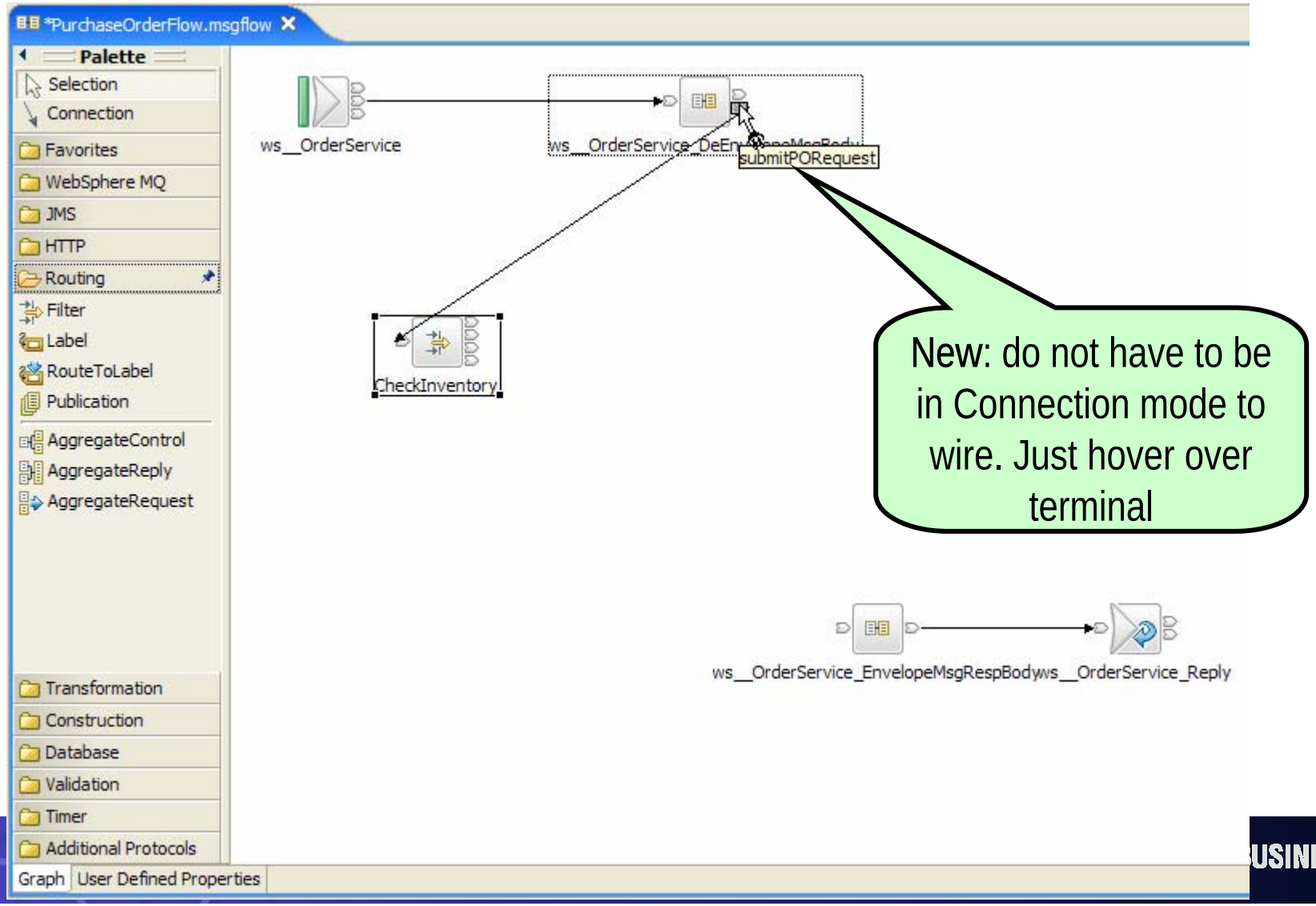
Other Properties Enhancements

- Re-organized/improved properties for following nodes:
  - HTTPInput
  - HTTPRequest
  - JMSOutput
  - XMLTransformation



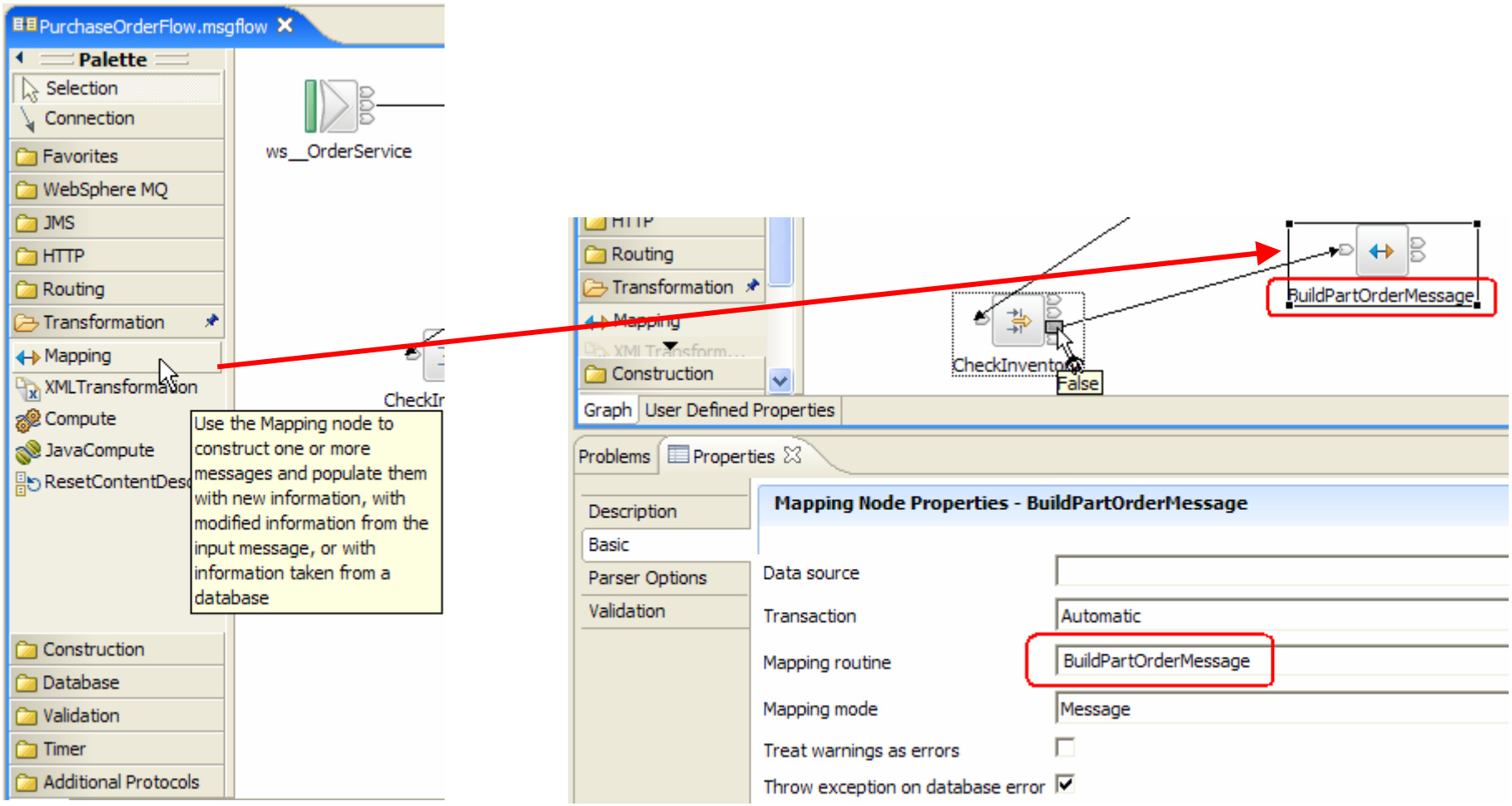
Scenario: Step 5b

- Wire request subflow, for submitPORequest operation, to filter node



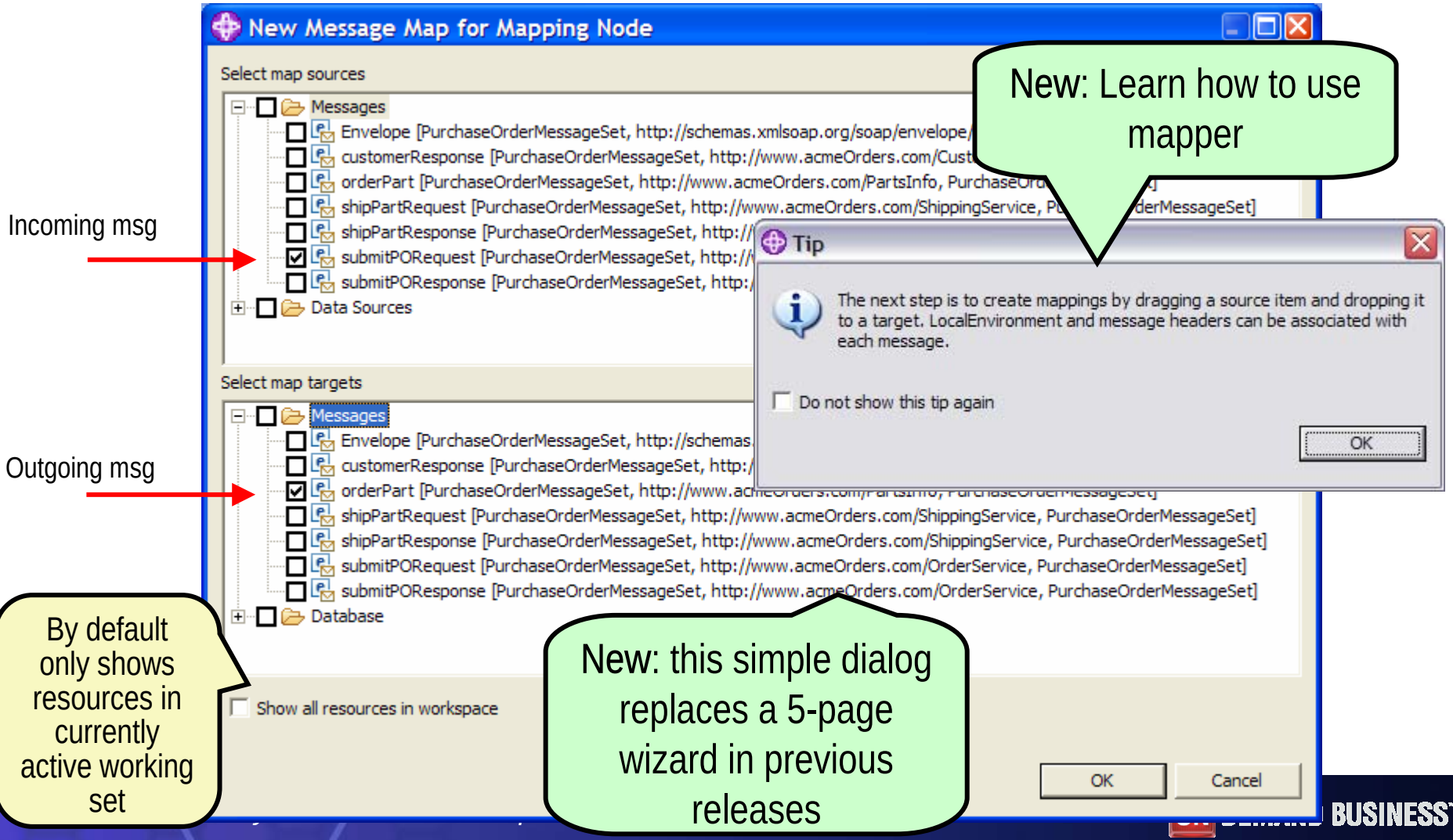
Scenario: Step 6a ... Mapping Node

- Need to create a map, when out of inventory, for ordering parts. What node to use? We search Transformation category and choose first one.



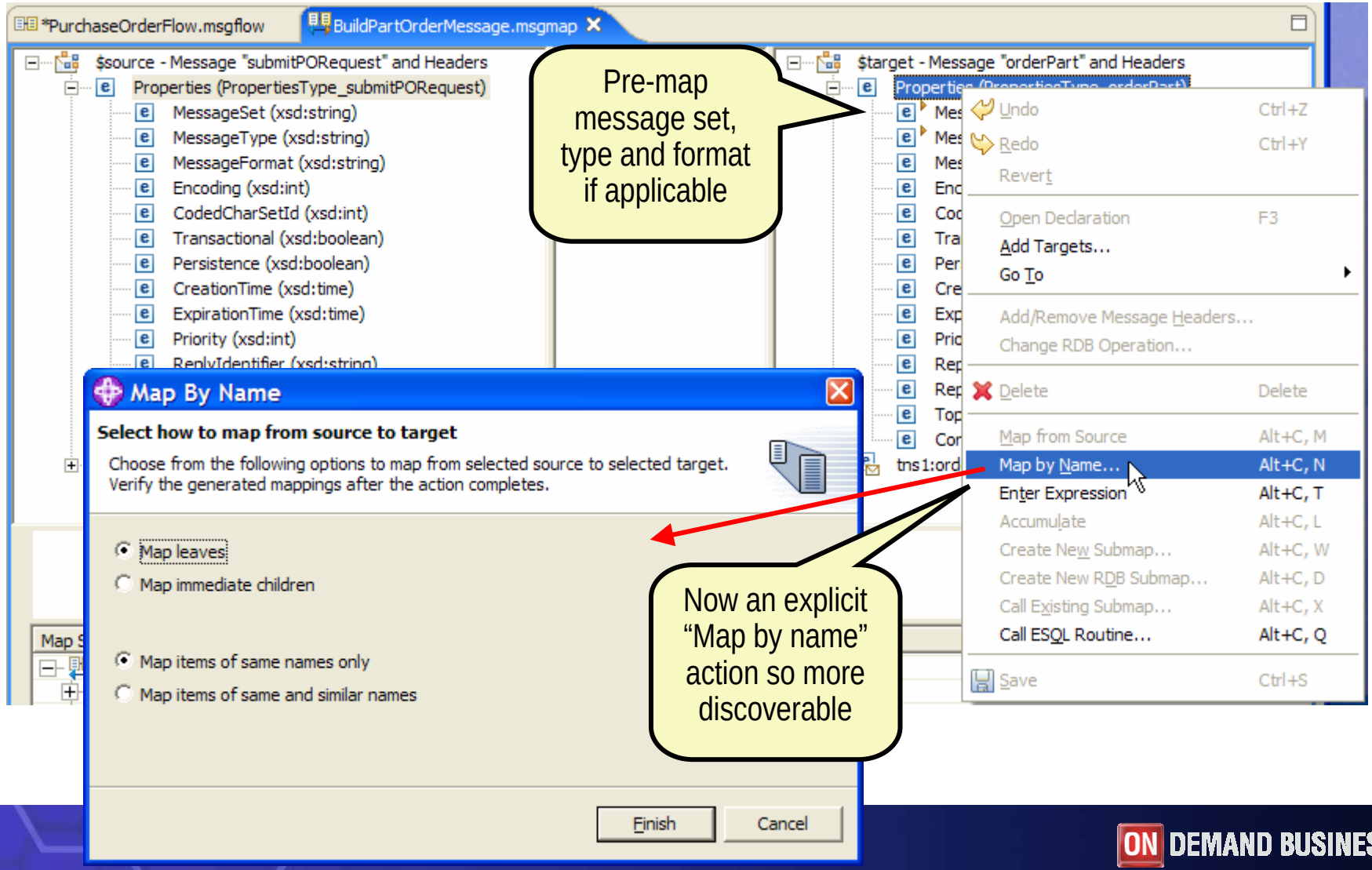
Scenario: Step 6b ... Open Map

- After setting node name and routine name property, and wiring to the Mapper node, double click on it to open mapper wizard

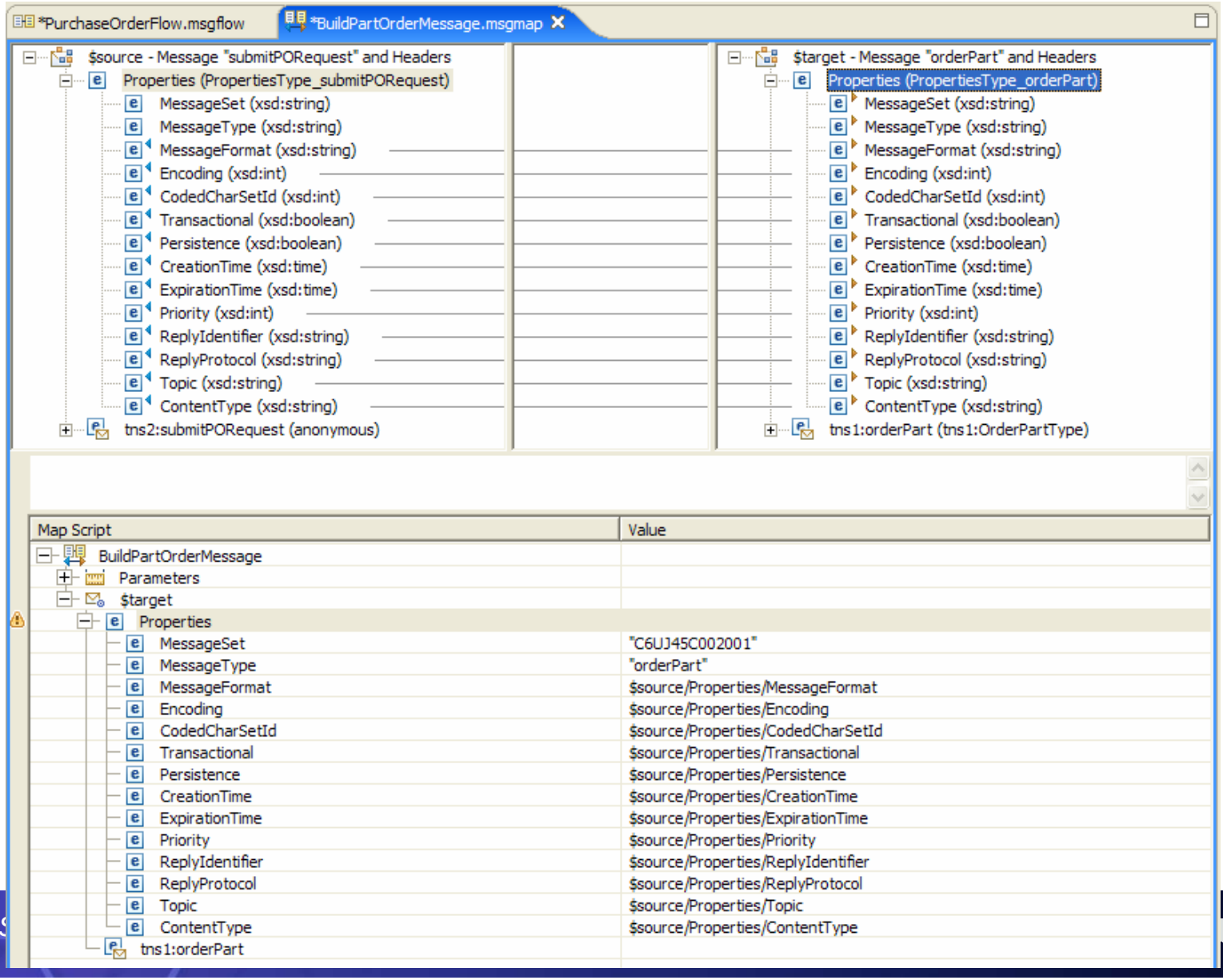


Scenario: Step 6c ... map the properties

- In mapping editor, select Properties on both sides, and then Map By Name

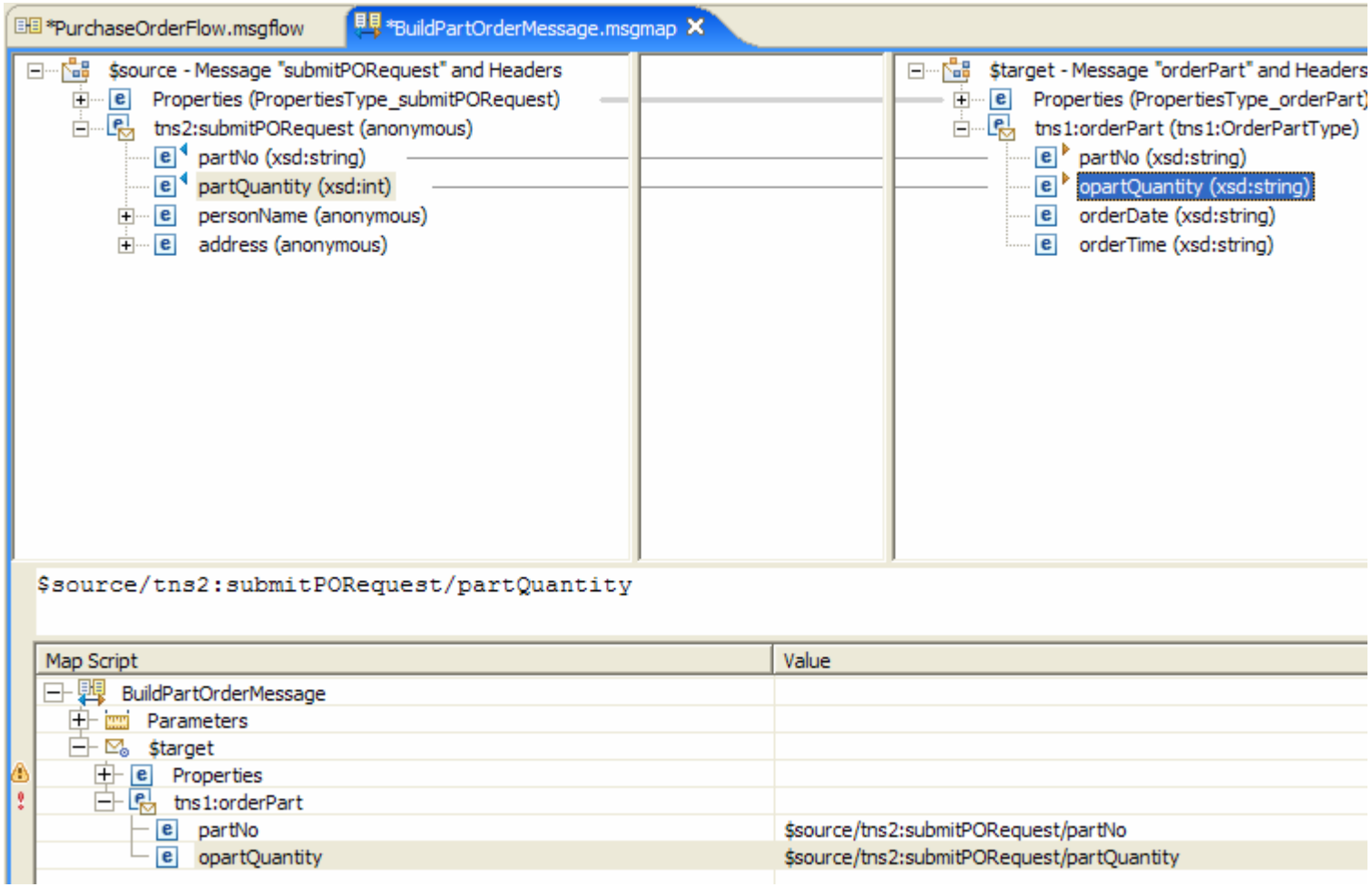


Scenario: Step 6c ... mapped properties

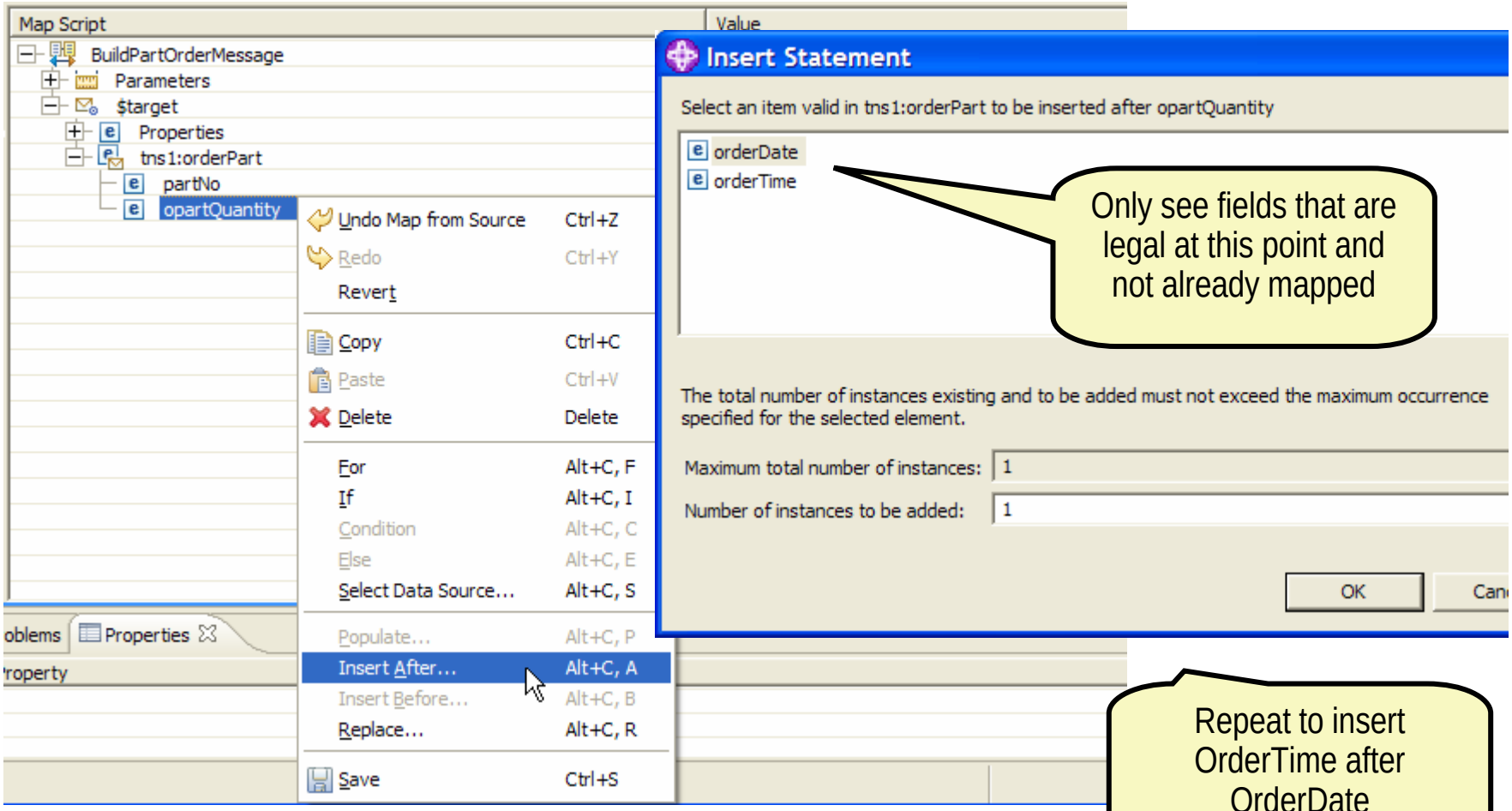


Scenario: Step 6d ... map the fields

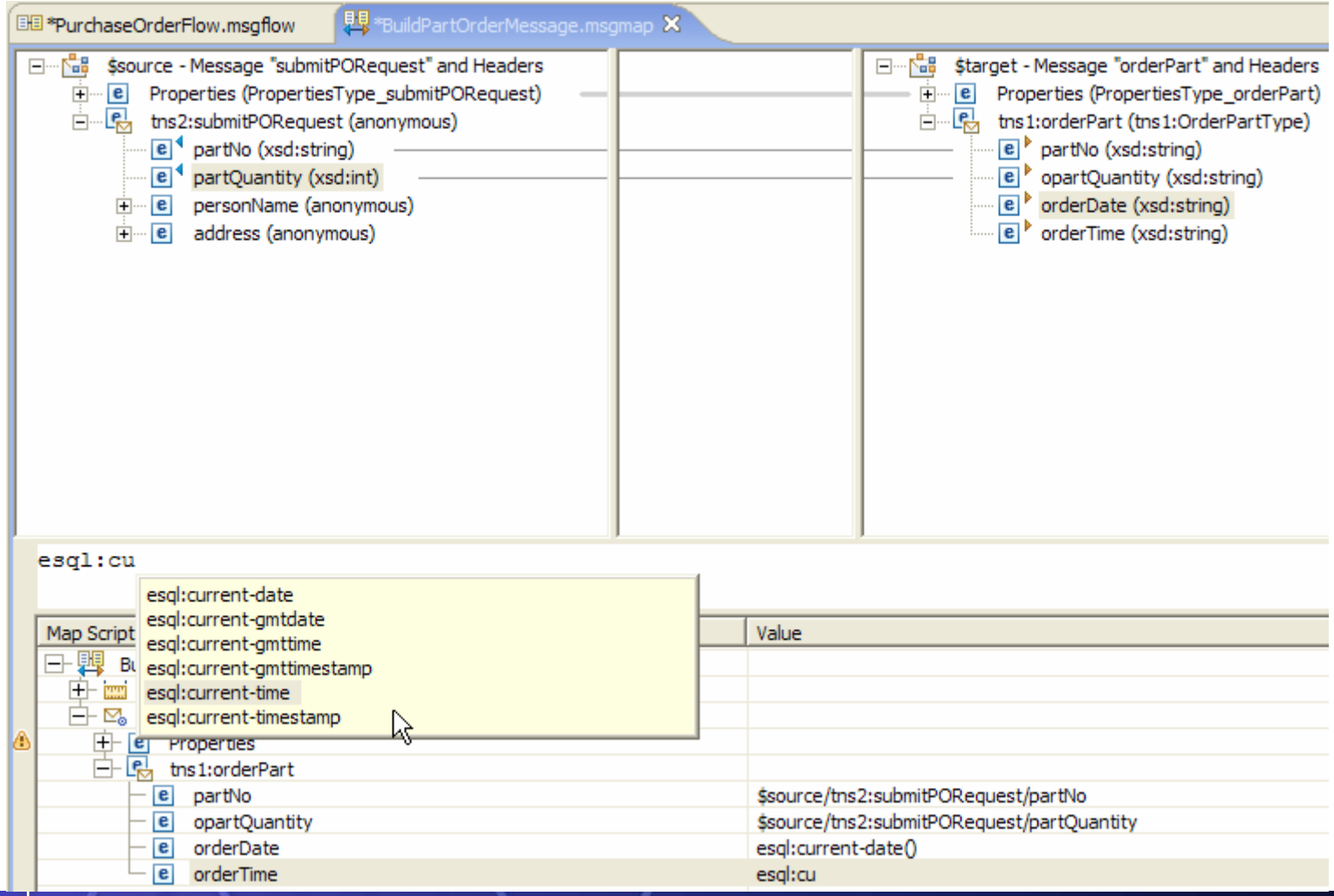
- In mapping editor, drag and drop `partNo` and `partQuantity`



Scenario: Step 6e ... insert orderDate & orderTime

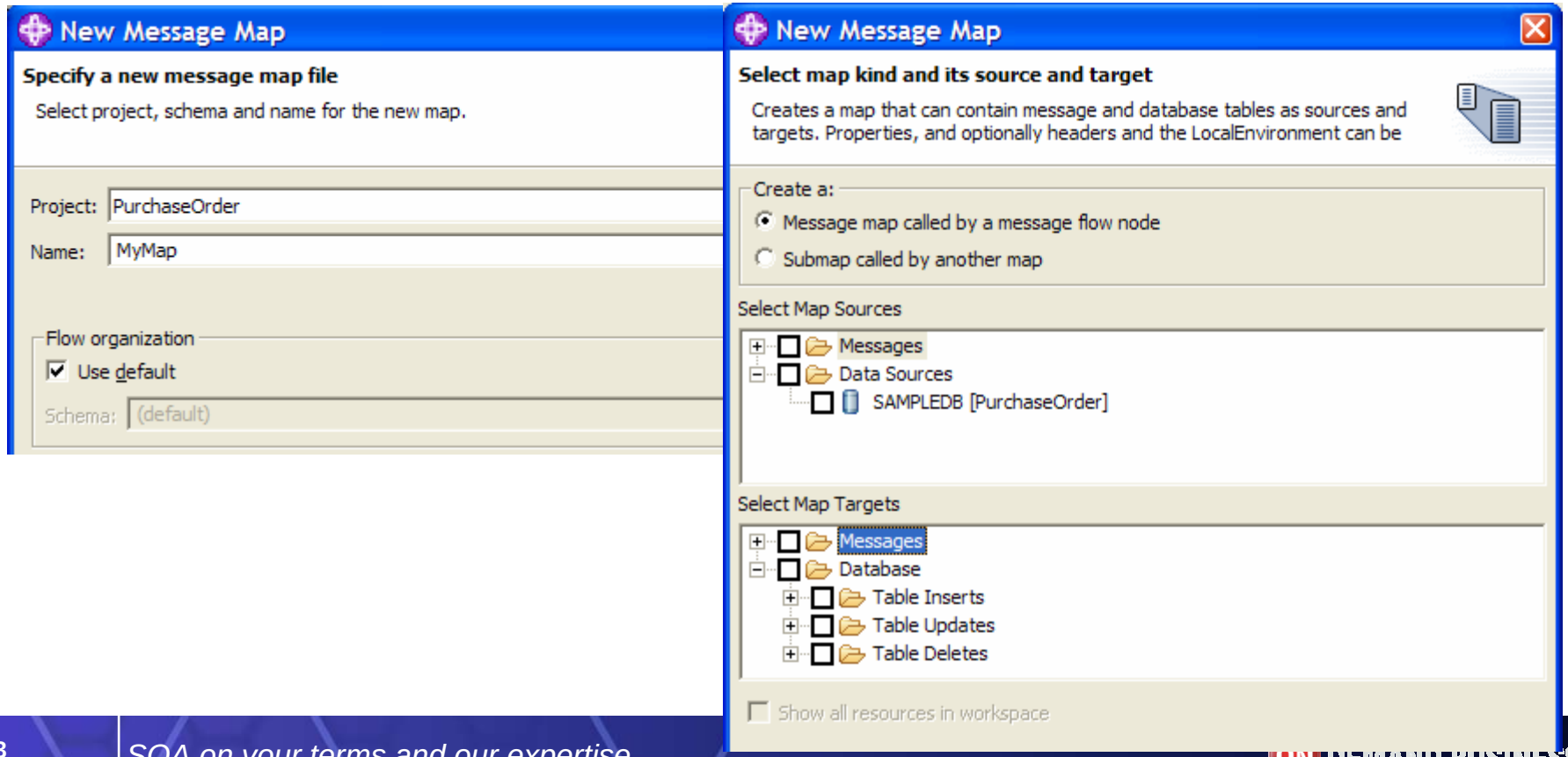


Scenario: Step 6e ... set orderDate and orderTime to esql:current-date() and esql:current-time()



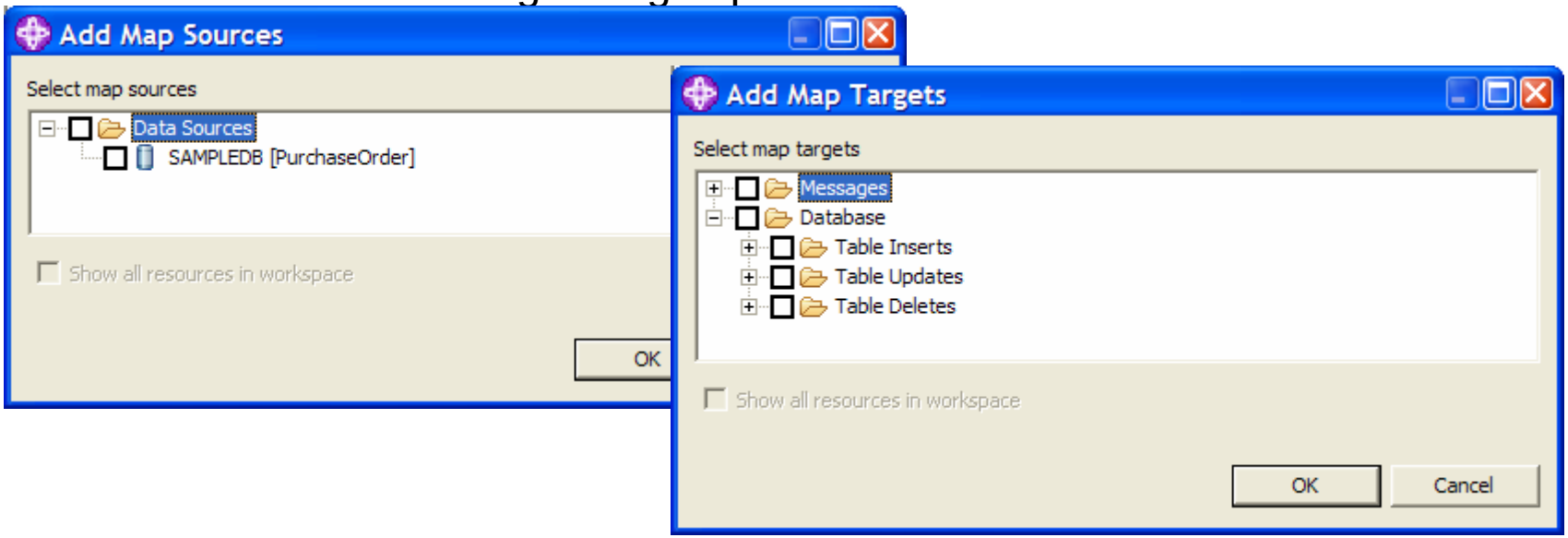
Additional Mapper Enhancements 1 of 6

- There are a number of enhancements to the mapper:
  - 1. Vastly simplified wizard, as we have seen when opened from the Flow editor. Here are the wizard pages when opened from File->New:



Additional Mapper Enhancements 2 of 6

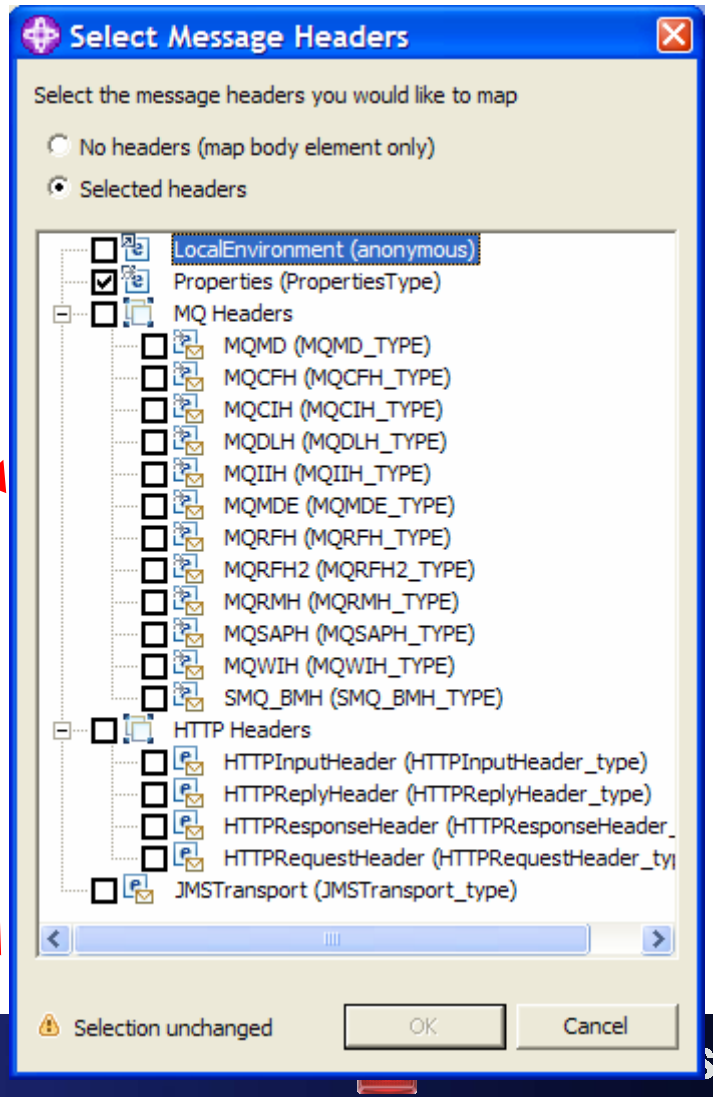
- Additional enhancements to the mapper:
  - 2. Changed “Add Sources and Targets” to become “Add Sources…” and “Add Targets…” right click actions on $source and $target roots in trees. Resulting dialogs updated to be consistent with new wizard:



- 3. You can also drag and drop roots from the nav to add src/tgt
  - Eg: message or database table or global element (for submaps)

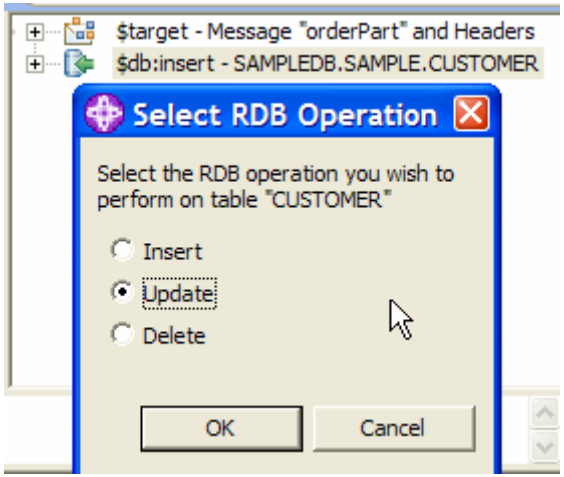
Additional Mapper Enhancements 3 of 6

- Additional enhancements to the mapper:
  - 4. In dealing with headers by default:
    - The Properties folder is shown in the mapper to enable explicit mapping
    - Any headers are automatically copied from input message by default
    - The Local Environment is not mapped by default
  - 5. But within the Map editor the user can:
    - Right click on Source or Target and select "*Add/Remove Message Headers…*" resulting in this:



Additional Mapper Enhancements 4 of 6

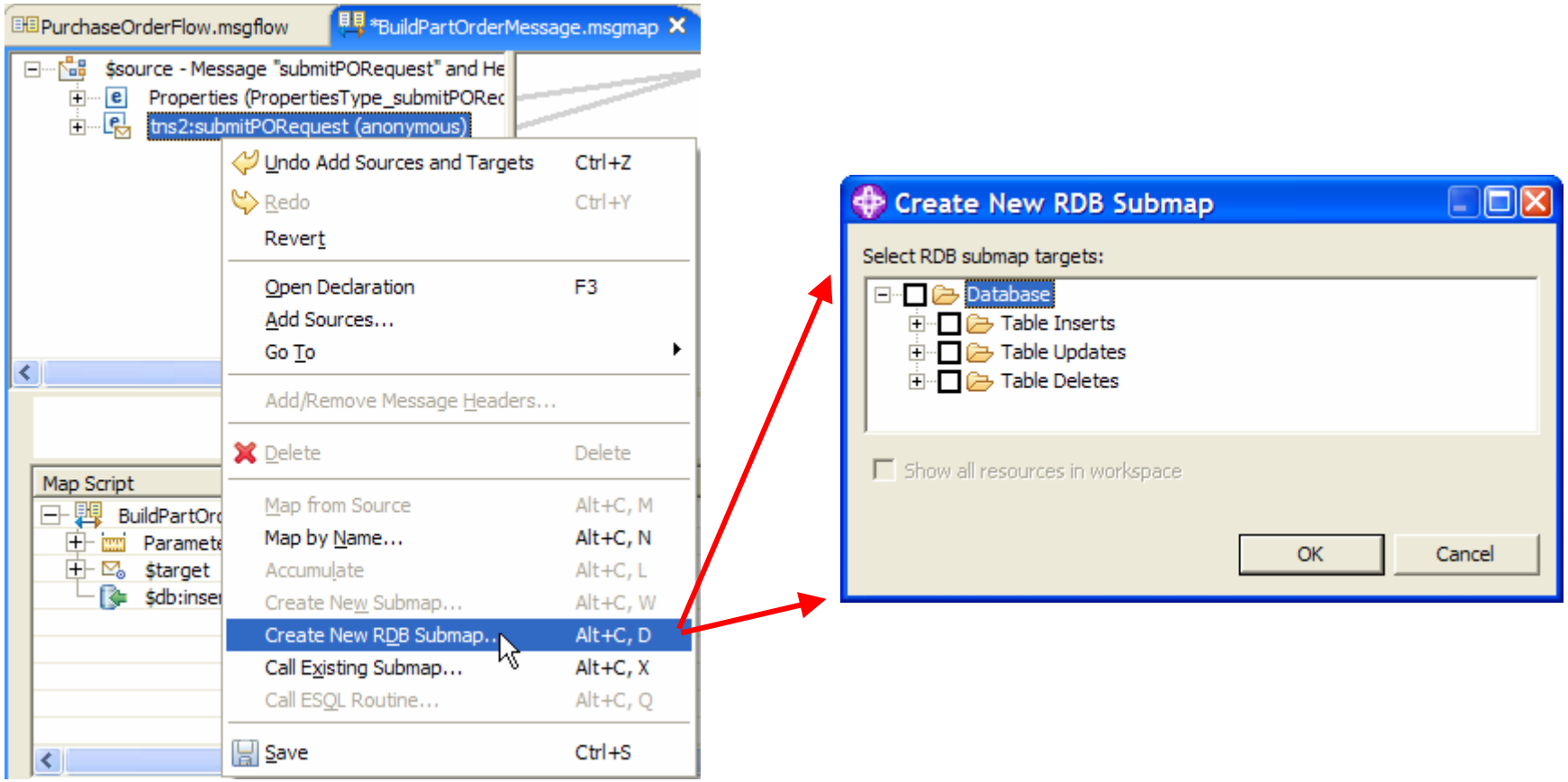
- Additional enhancements to the mapper:
  - 6. New "*Change RDB operation…*" right click action:



- Further,
  - 7. You can now map from database to database. Previously, you could only map from database to message or message to database.

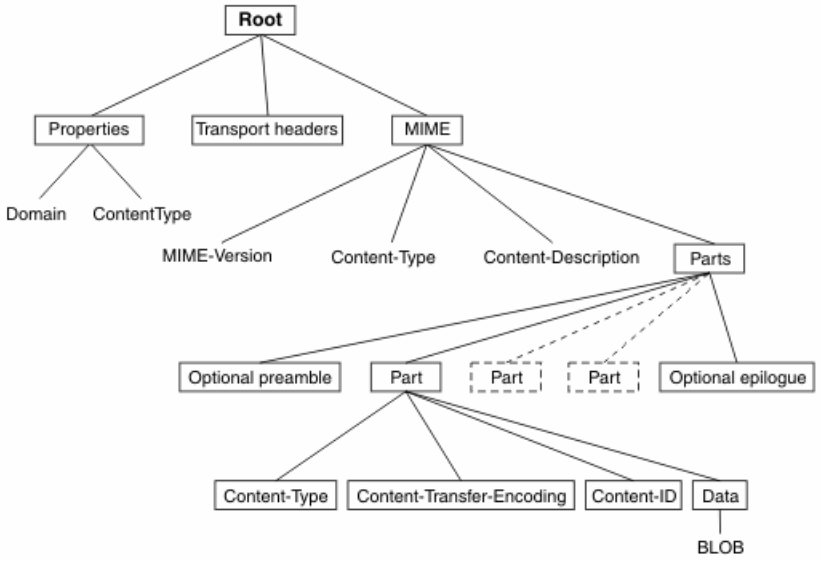
Additional Mapper Enhancements 5 of 6

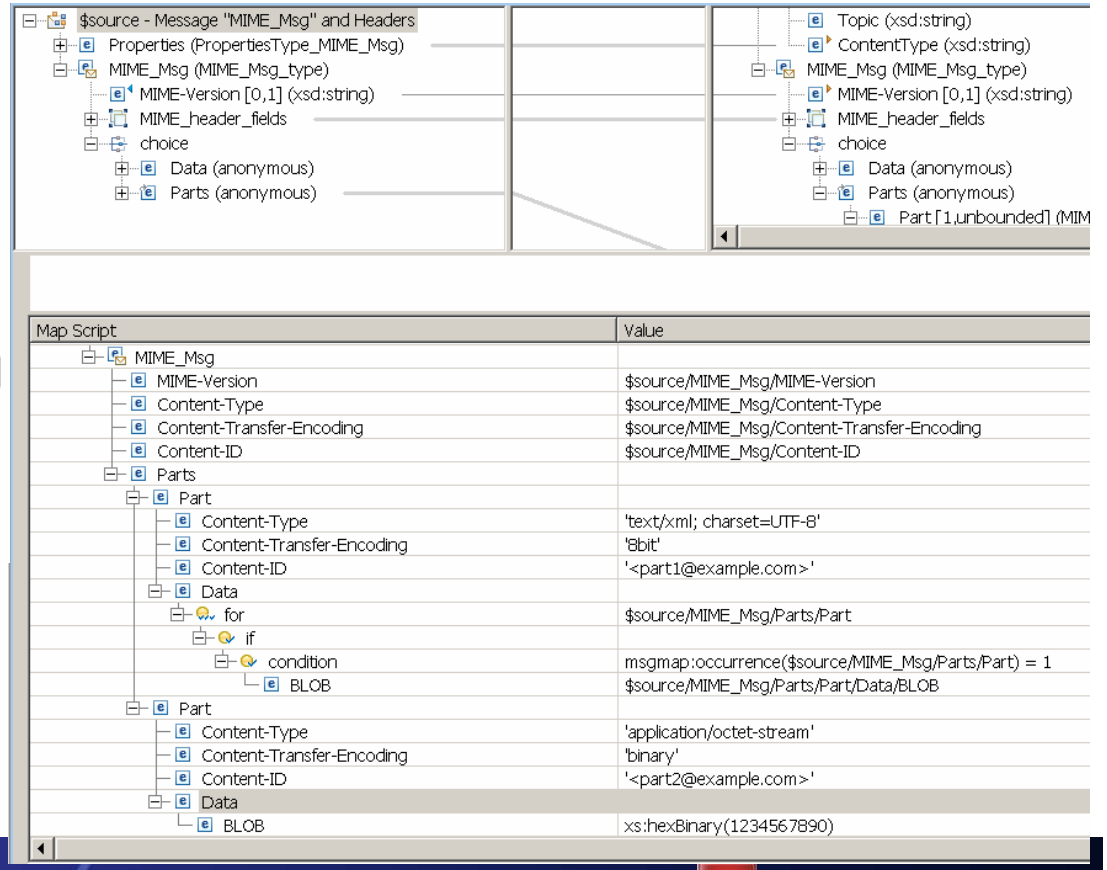
- Additional enhancements to the mapper:
  - 8. New "*Create RDB Submap…*" right click action (moved from wizard):



Additional Mapper Enhancements 6 of 6

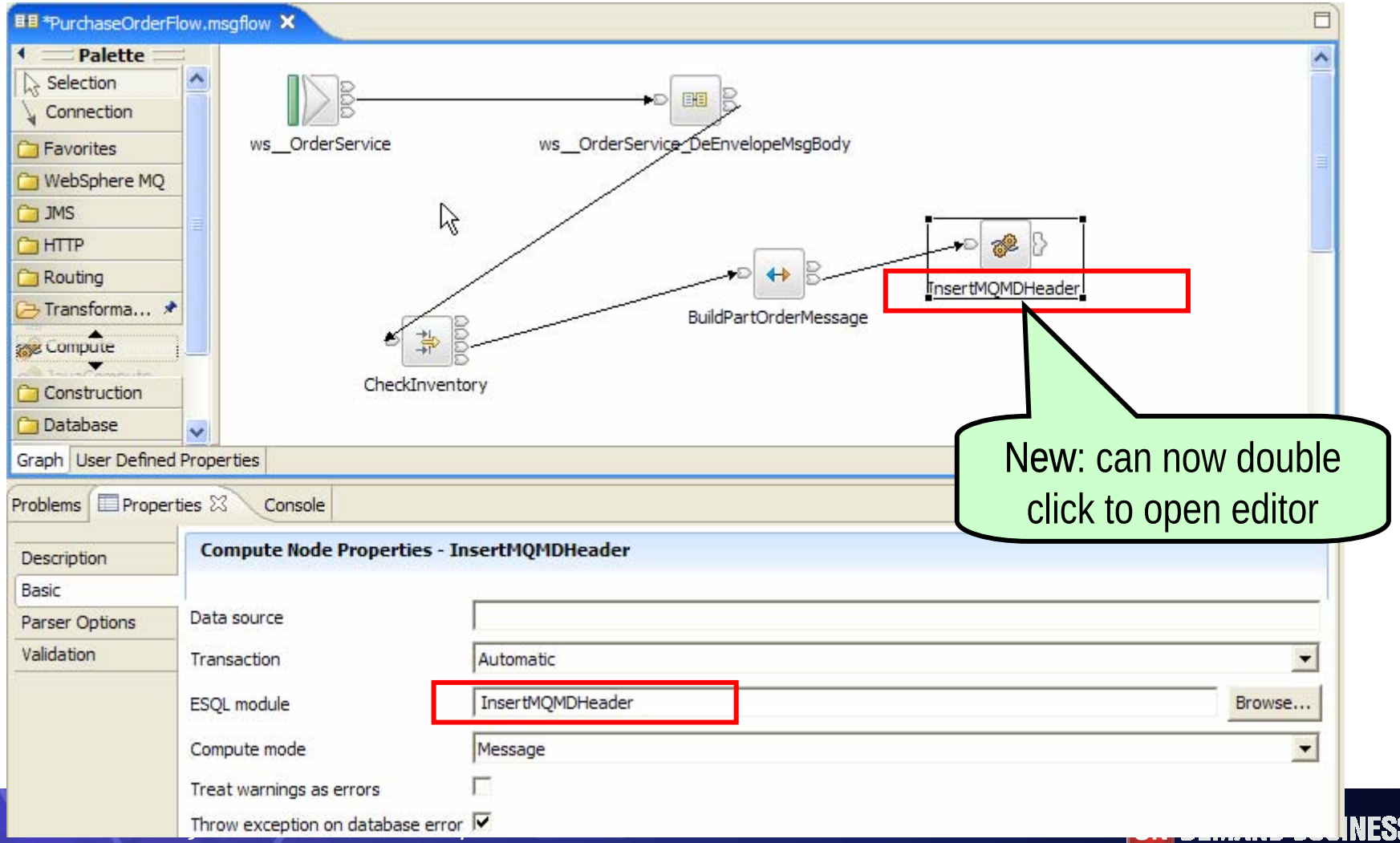
- Additional enhancements to the mapper:
  - 9. You can now use the mapper to map messages in a MIME domain
  - Domains supported: MRM, MIME, XML, XMLNS, XMLNSC, JMSMap, JMSStream





Scenario: Step 7a ... Compute node

- In flow editor, drop a Compute node, wire to it, and set its name and ESQL Module property to `InsertMQMDHeader` . Press Ctrl+S to save.



Scenario: Step 7b ... Compute node

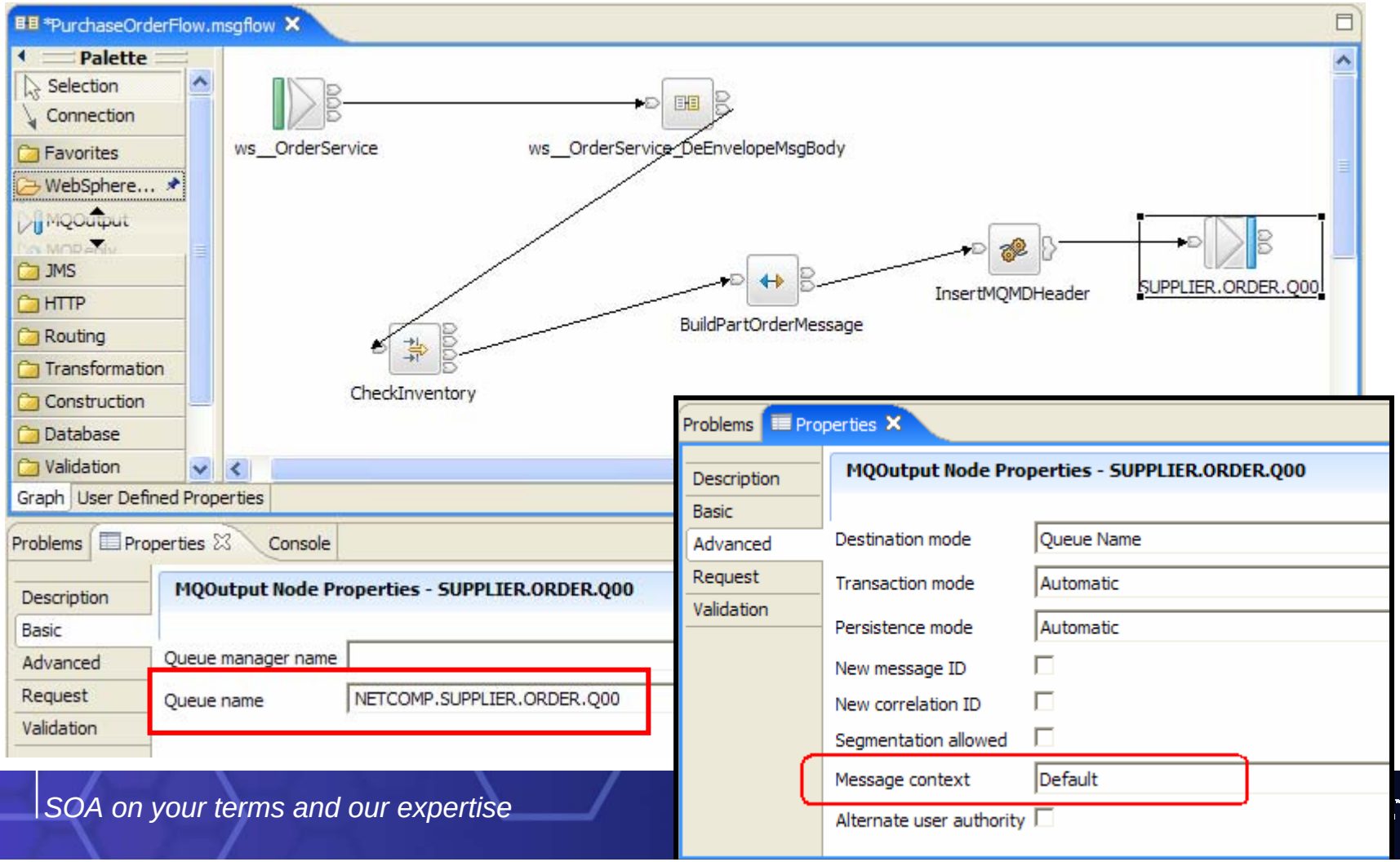
- Double click (new!) the InsertMQMDHeader node to enter ESQL editor
- Change the default generated Main() function as shown below….

PurchaseOrderFlow.msgflow | PurchaseOrderFlow.esql

```
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;
```

Scenario: Step 7c ... MQOutput node

- In flow editor, drop MQOutput node, set its name and Queue name property as shown, and wire to it from the out terminal of the compute node

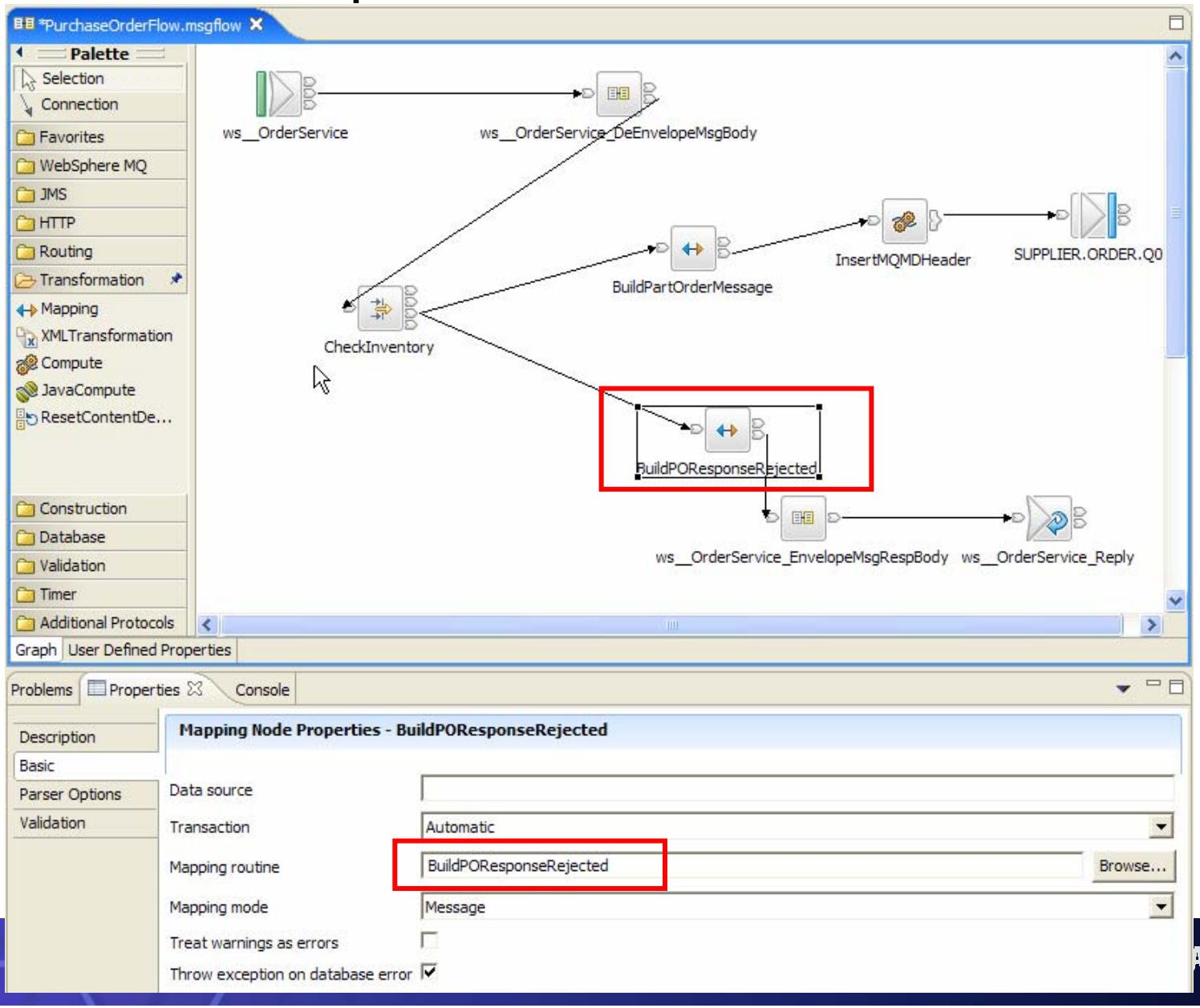


Scenario: Step 8 - Send Response When No Inventory

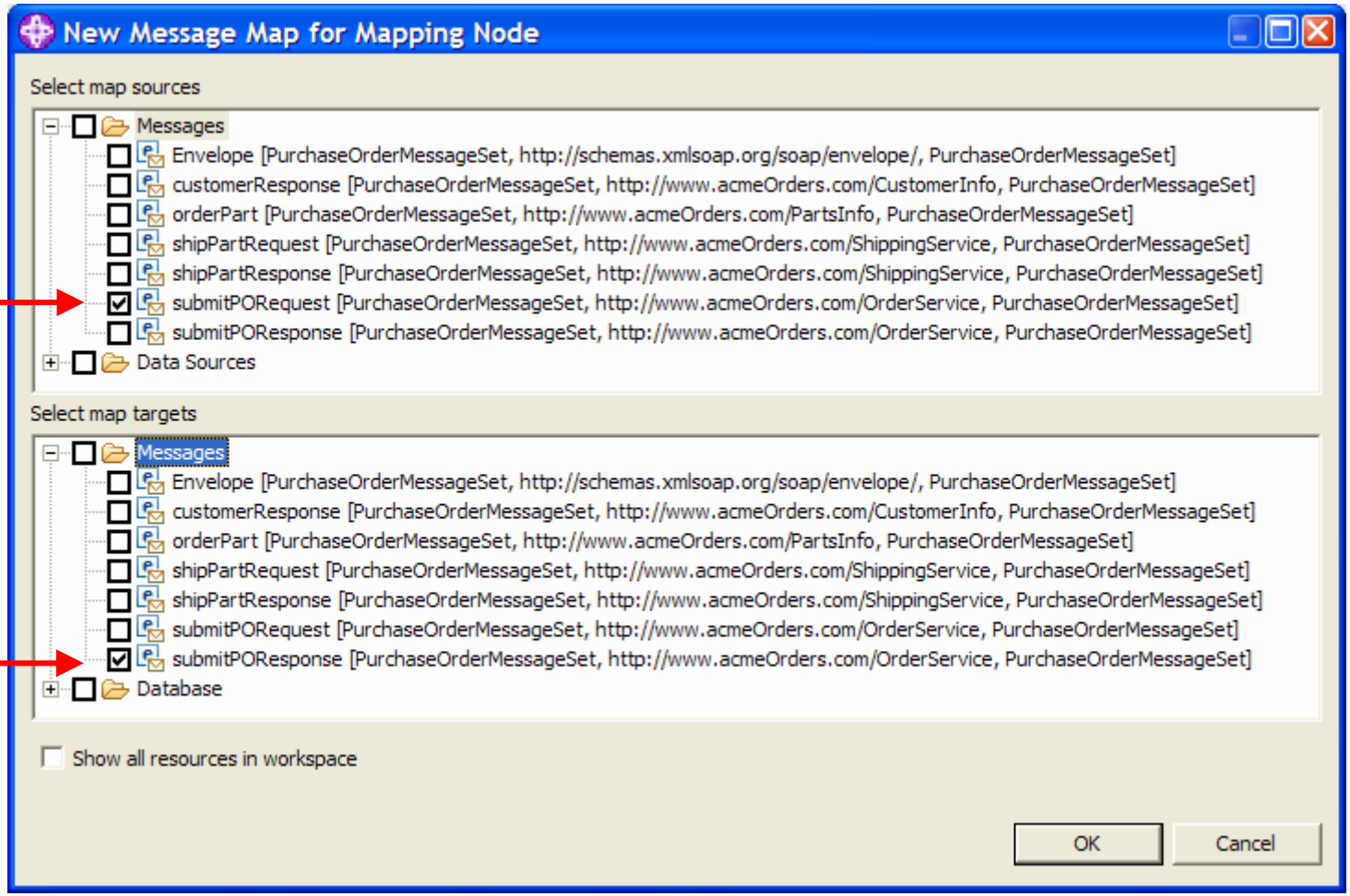
- Here we will again wire from the “false” terminal of the checkInventory filter node, this time to return an HTTP reply to the caller of this web service
  - To a map node to build the response message
  - From there to the generated reply nodes for this operation

- **8a**: Drag and drop a map node and wire it. Also set its node name and mapping routine property to `BuildPOResponseRejected`.

Scenario: Step 8a – Create and Wire Mapping node

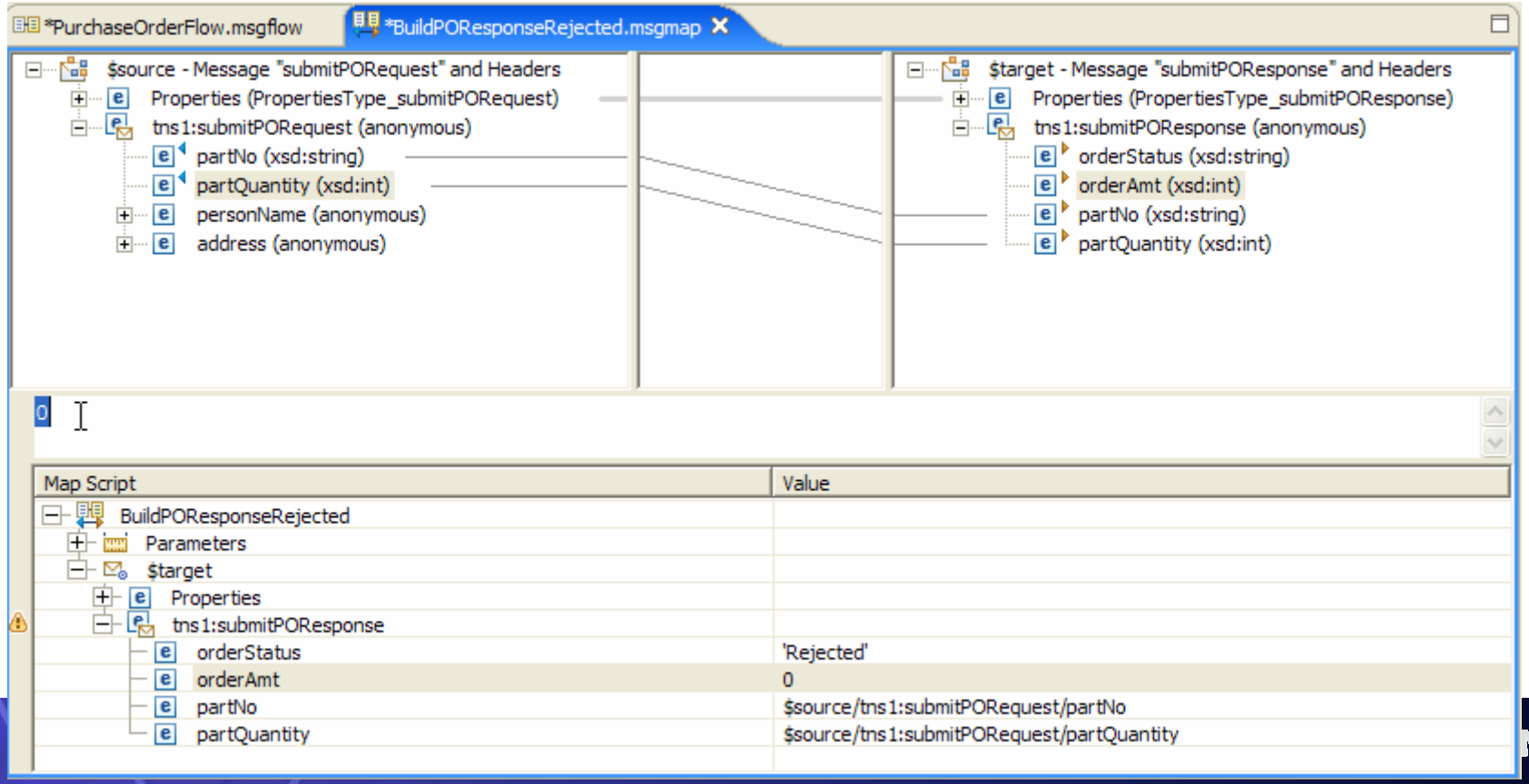


Scenario: Step 8b – Open Map



Scenario: Step 8c – Do Mapping

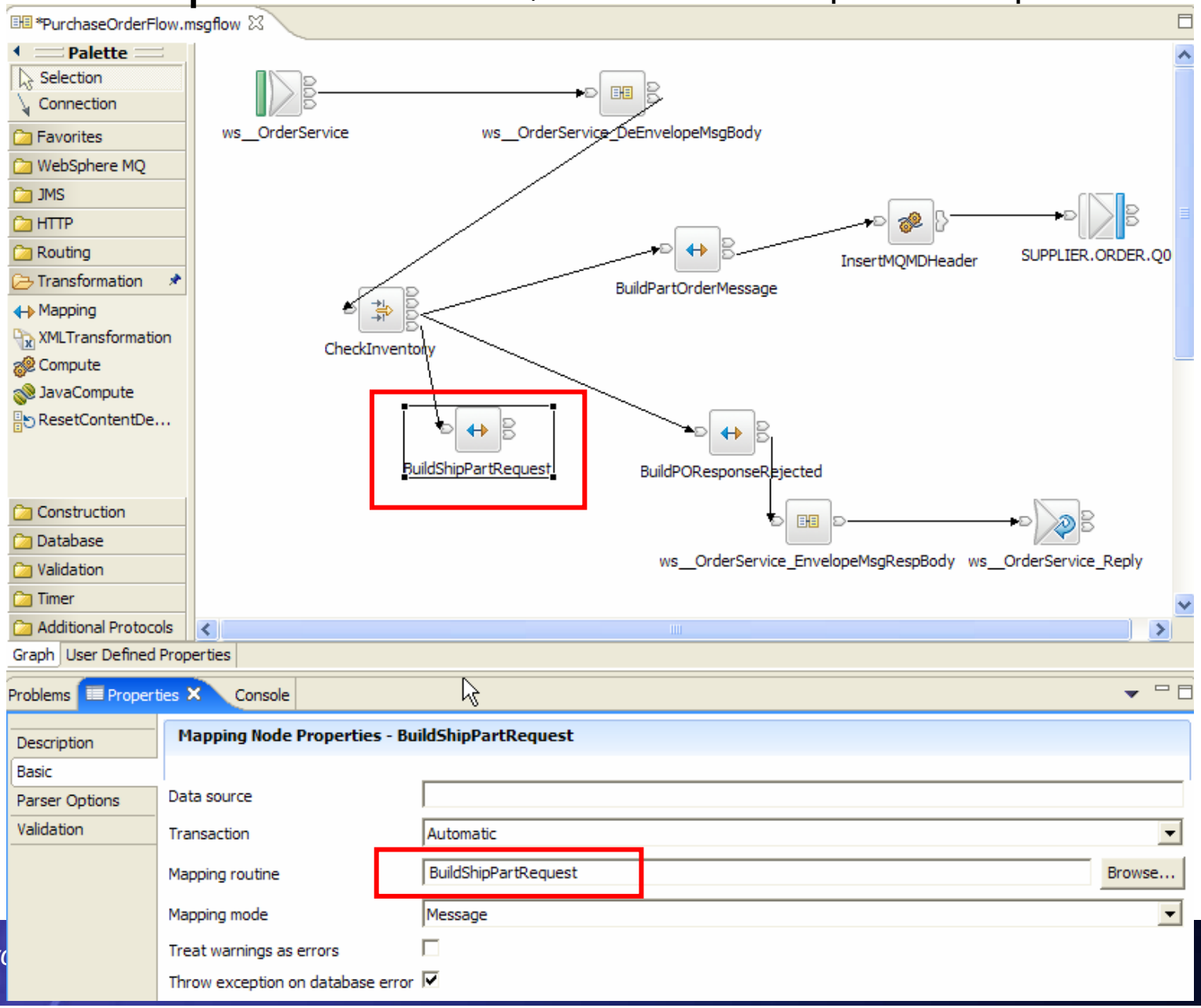
- Map Properties by name
- Map partNo and partQuantity by drag and drop
- Set orderStatus to ‘Rejected’ (use right click “Enter Expression”)
- Set orderAmt to 0 (use right click “Enter Expression”)



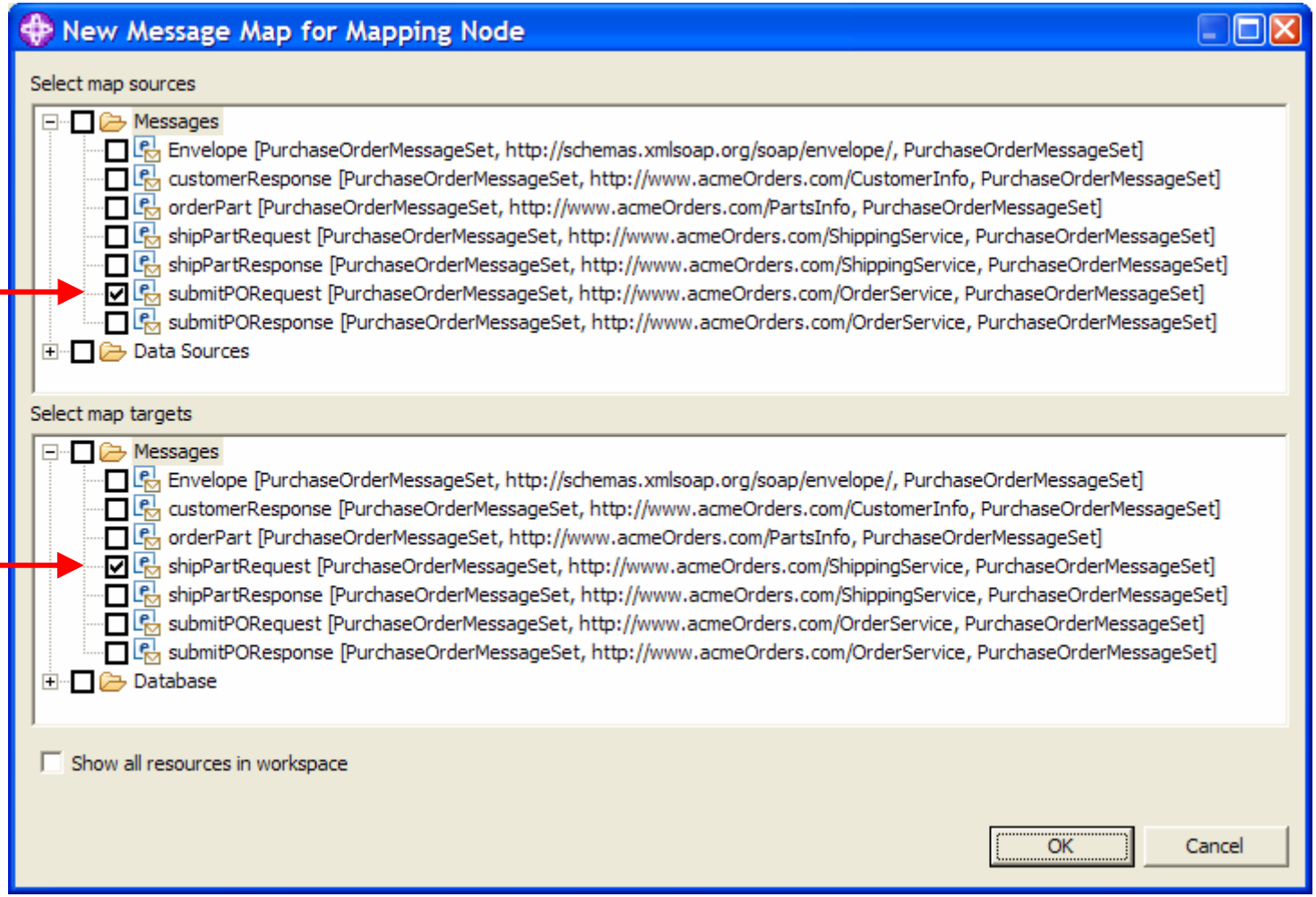
Scenario: Step 9 – Process “sufficient inventory” situation

- Here we will now wire from the “true” terminal of the checkInventory filter node:
  - To a map node to build the ship request message, which we later use as input to the shipping web service
- 9a: Drag and drop a map node and wire to it. Also set its name and mapping routine property to `BuildShipPartRequest`

Scenario: Step 9a – Create, Wire BuildShipPart Map node

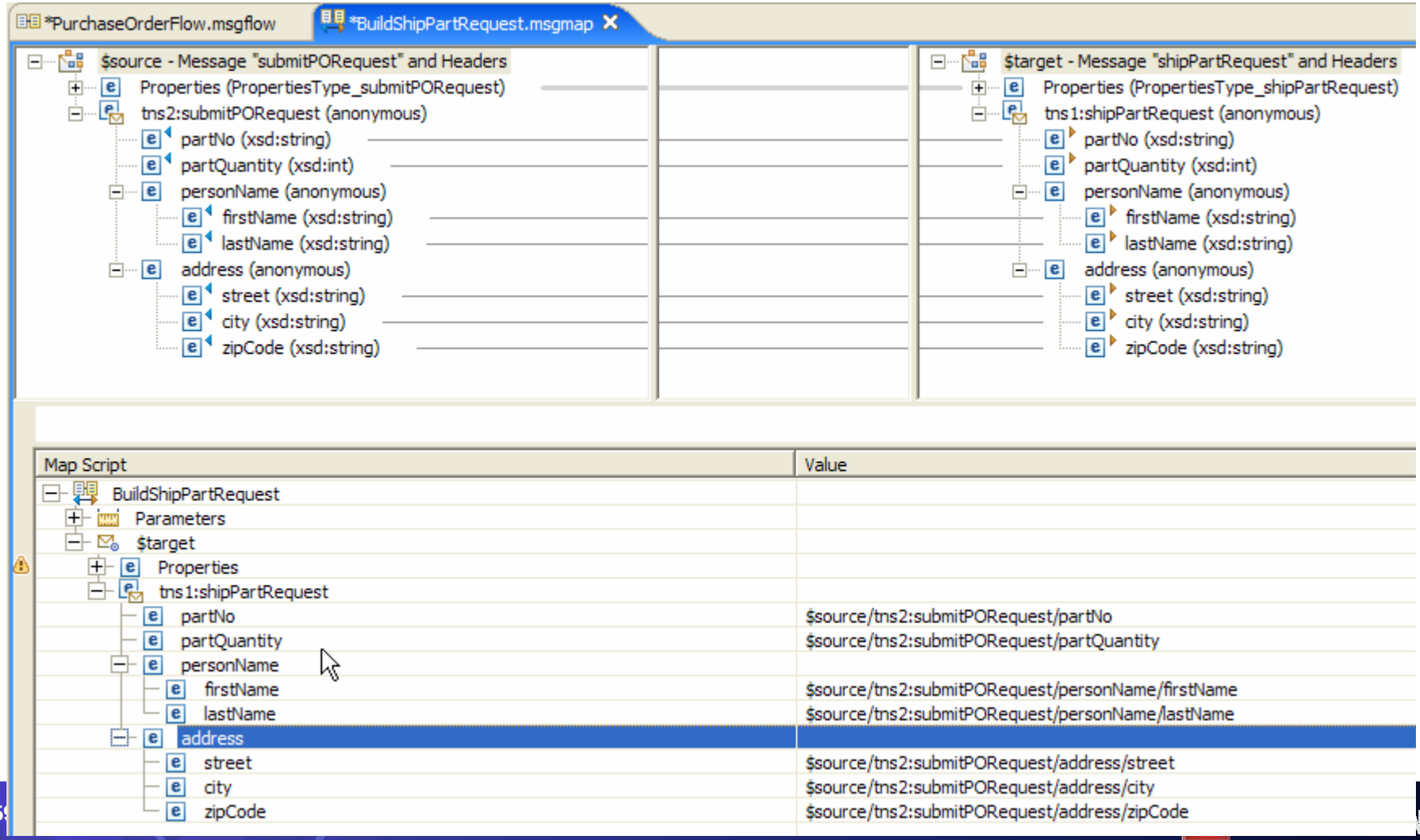


Scenario: Step 9b – Open Map



Scenario: Step 9c – Do Mapping

- Select $source and $target roots
- Right click on $target and select Map By Name. Press Enter. Done!

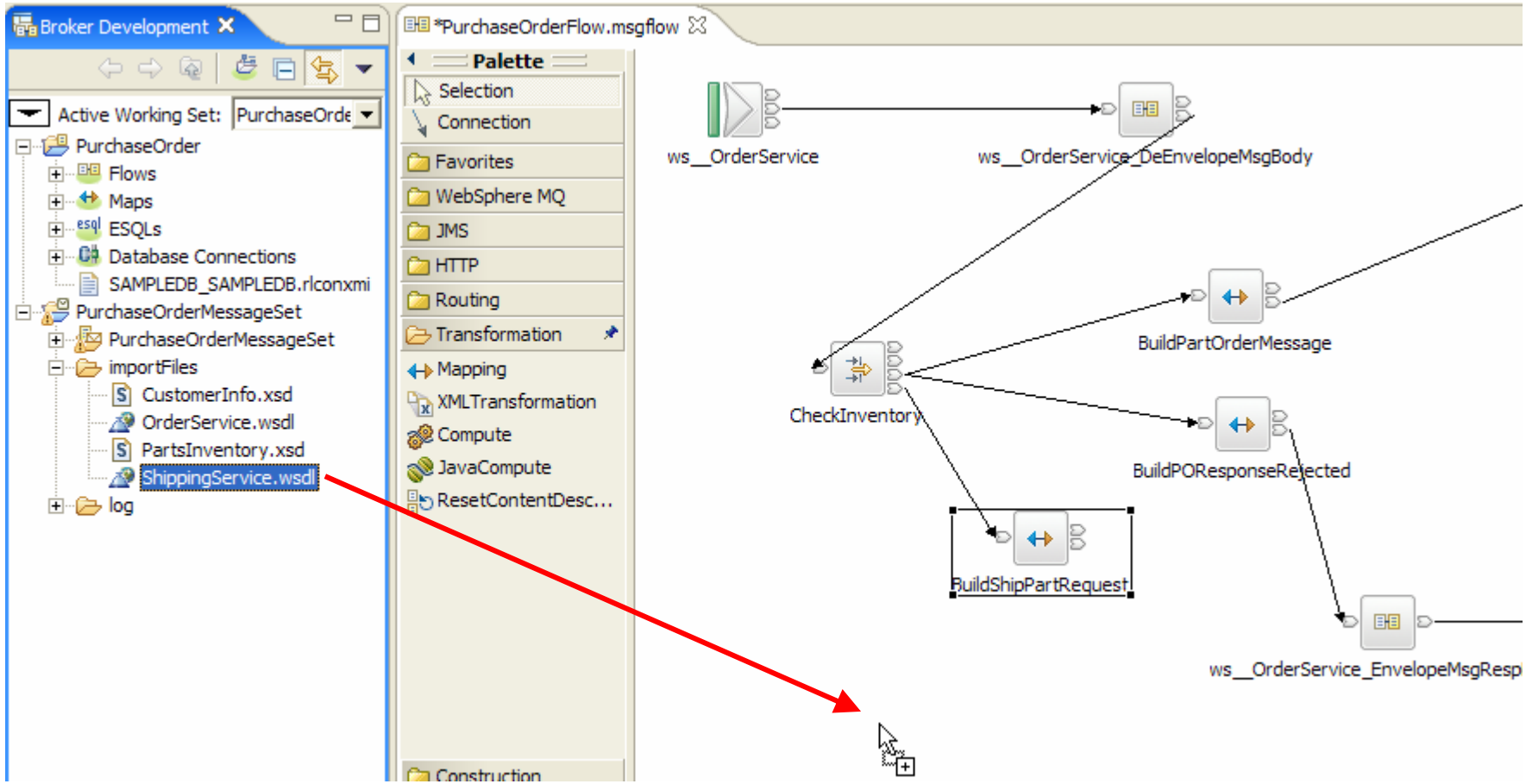


Scenario: Step 10 – Call Shipping Web Service

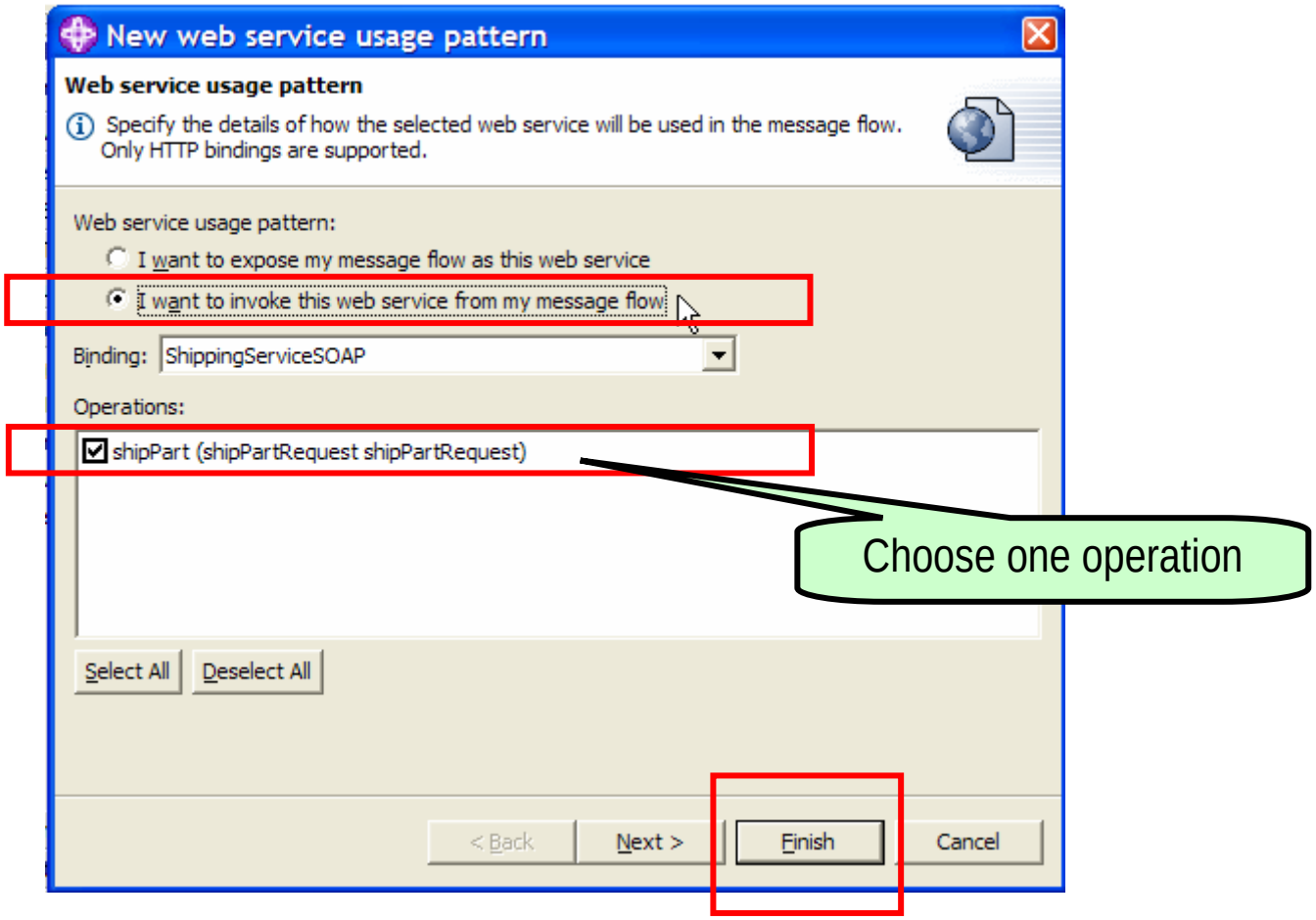
- Now that we have built up the message for invoking the web service, its time to actually do that:
  - Drag and drop the target web service's WSDL to canvas
  - Choose to invoke the web service
  - Choose the operation to invoke
  - Wire to generated nodes

- **10a** – drag and drop `ShippingService.wsdl` from message set's "`importFiles`" folder …

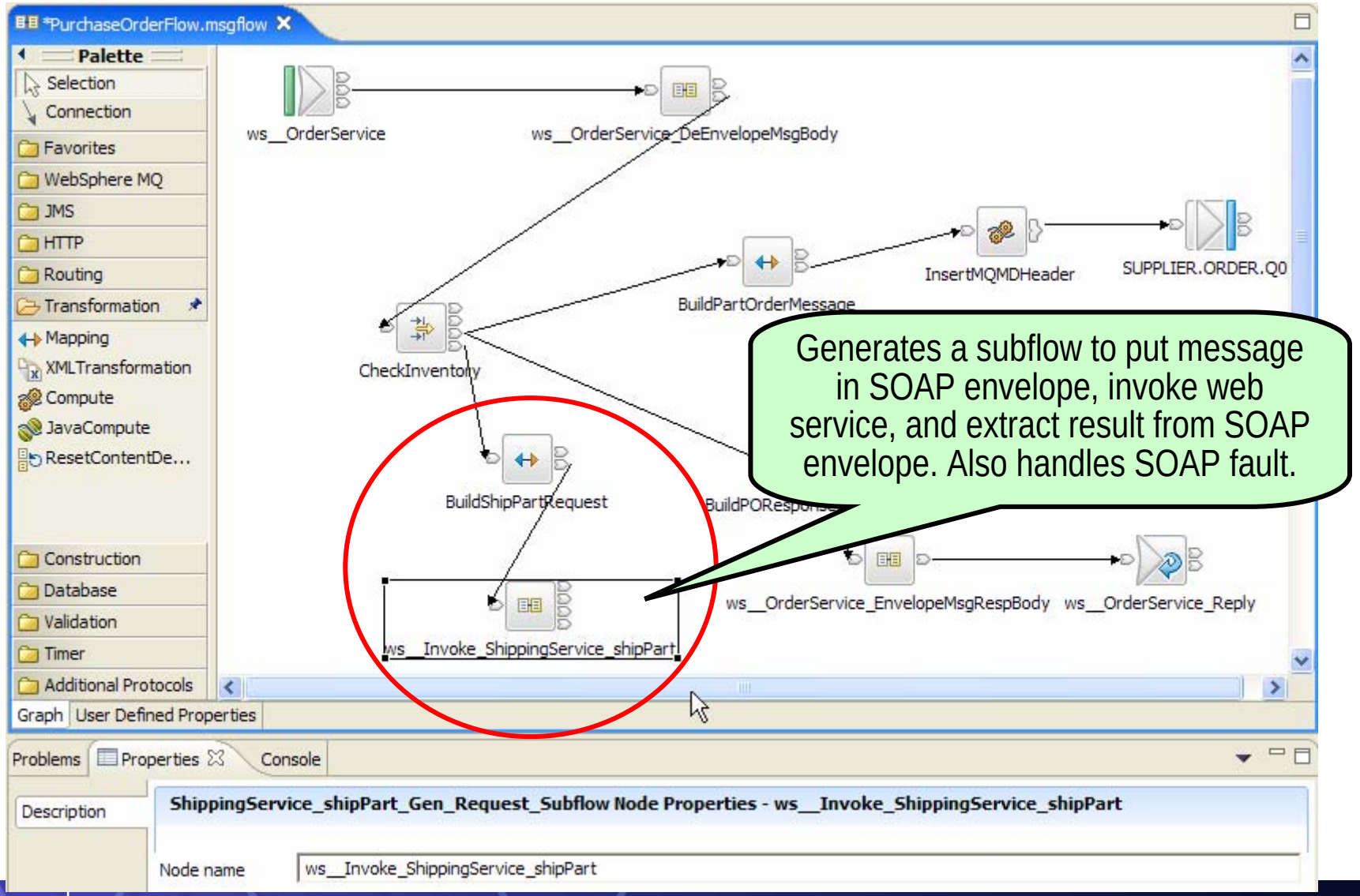
Scenario: Step 10a – Drag and Drop WSDL



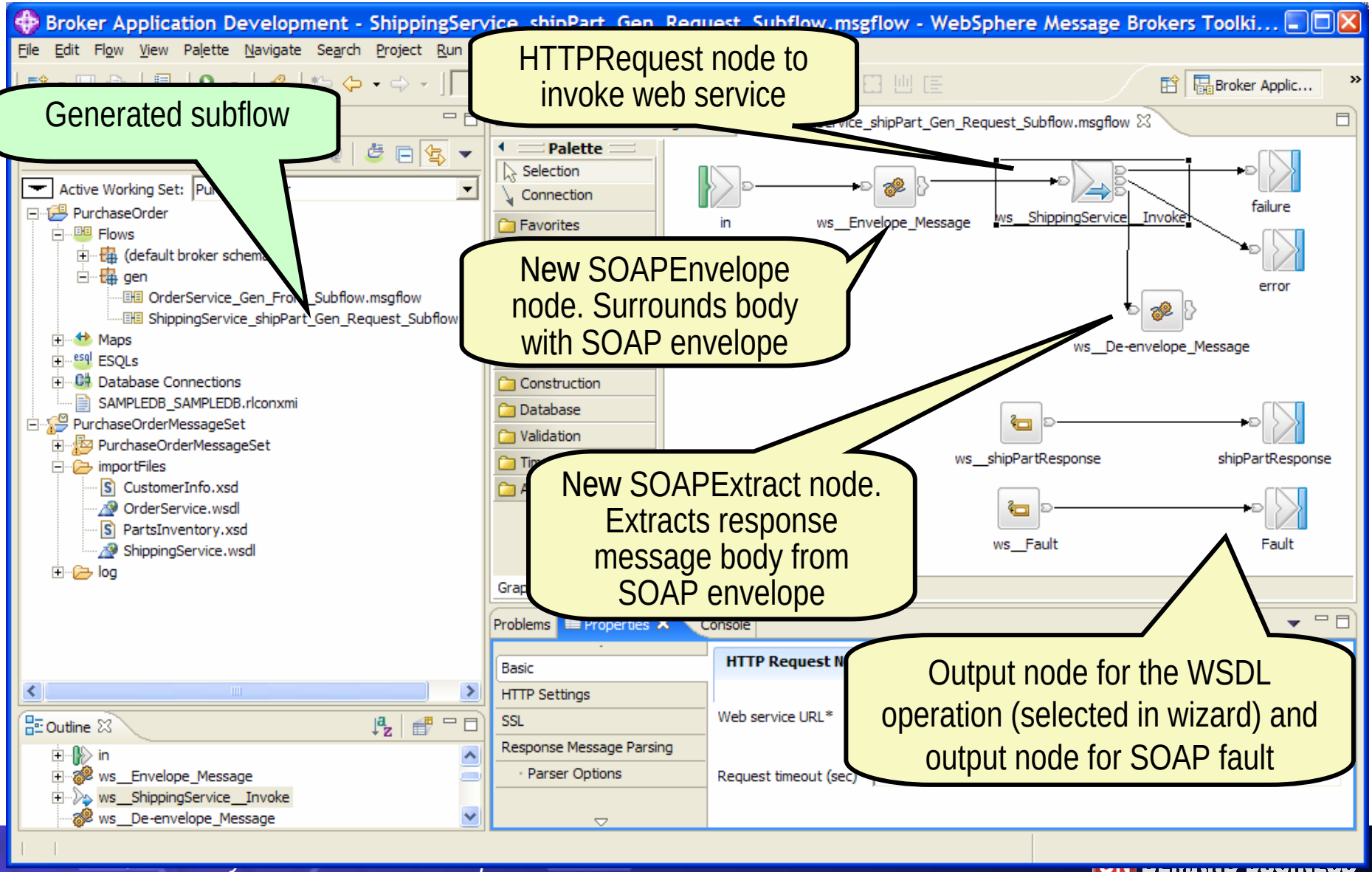
Scenario: Step 10b – WSDL Drag and Drop Wizard



Scenario: Step 10c – Generated Subflow



Scenario: Step 10c – Inside the Subflow

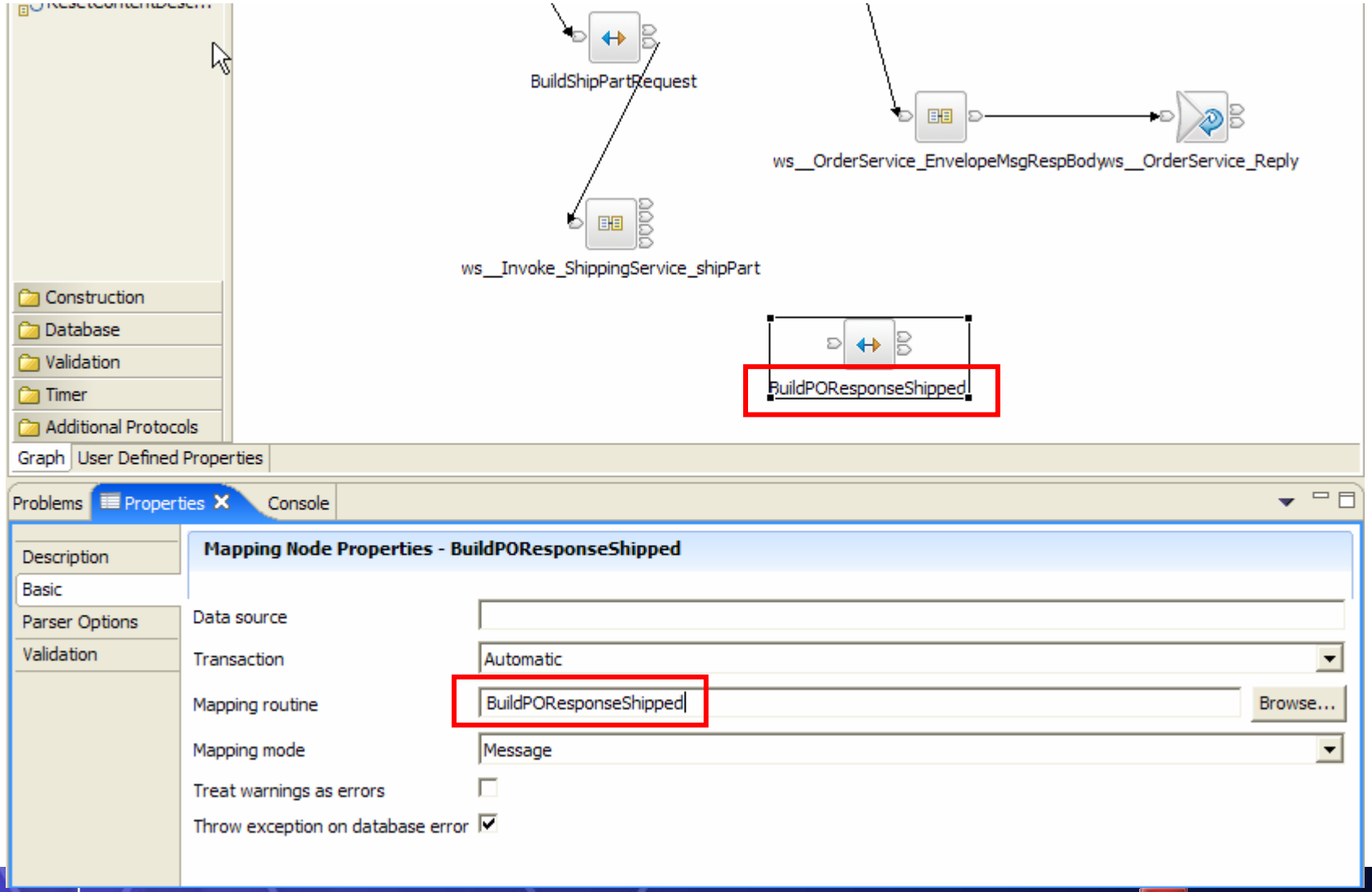


Scenario: Step 11 – Prepare and Send Reply

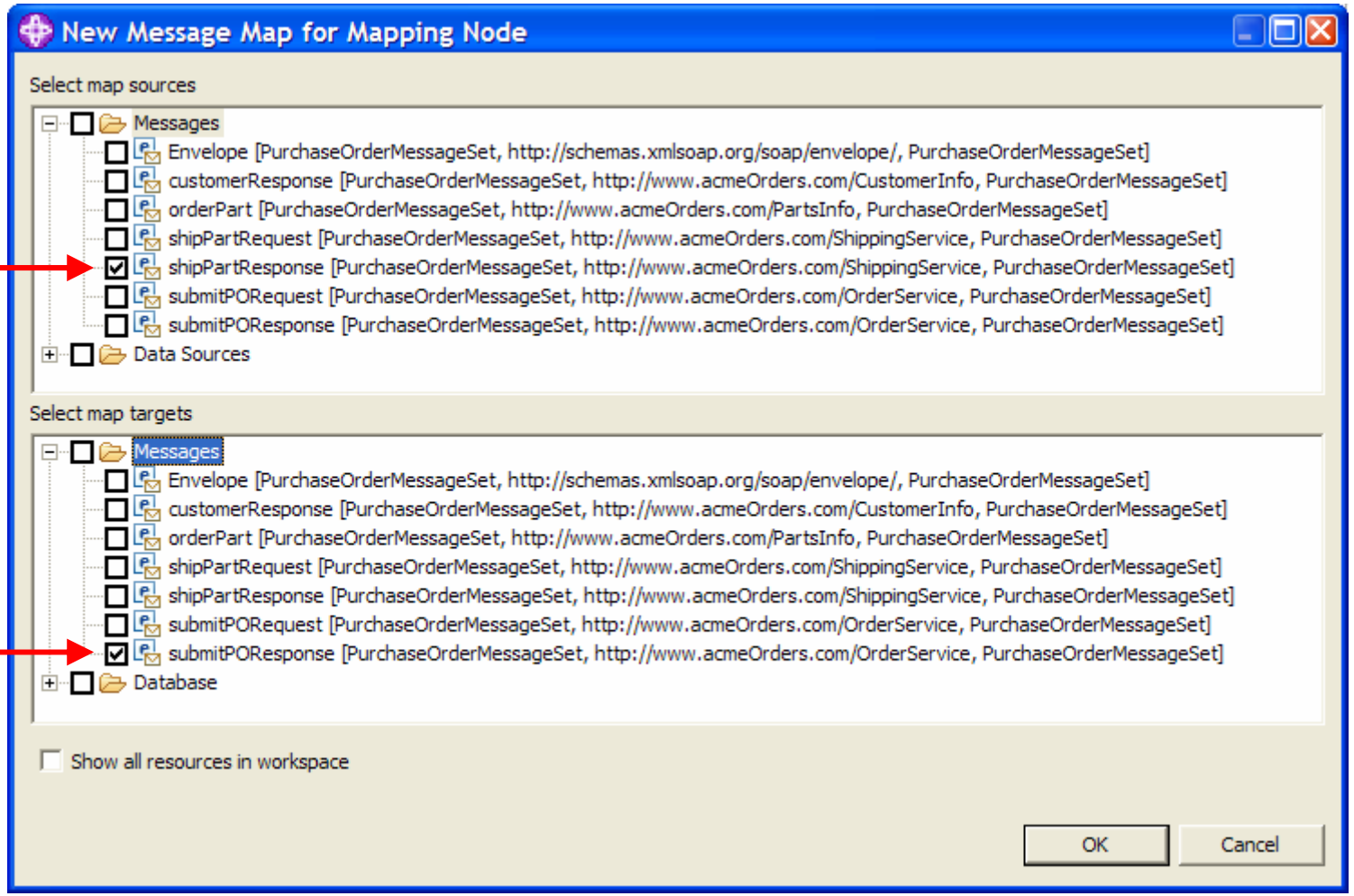
- Finally, after successfully invoking the web service to ship the parts to the customer, we can reply to the caller who invoked our web service:
  - Use a map node to create the response message
  - Wire from generated subflow to the map
  - Wire from the map to generated reply nodes

- **11a** – drag and drop a map node and set its name and mapping routine property to `BuildPOResponseShipped`

Scenario: Step 11a – Create Mapping Node

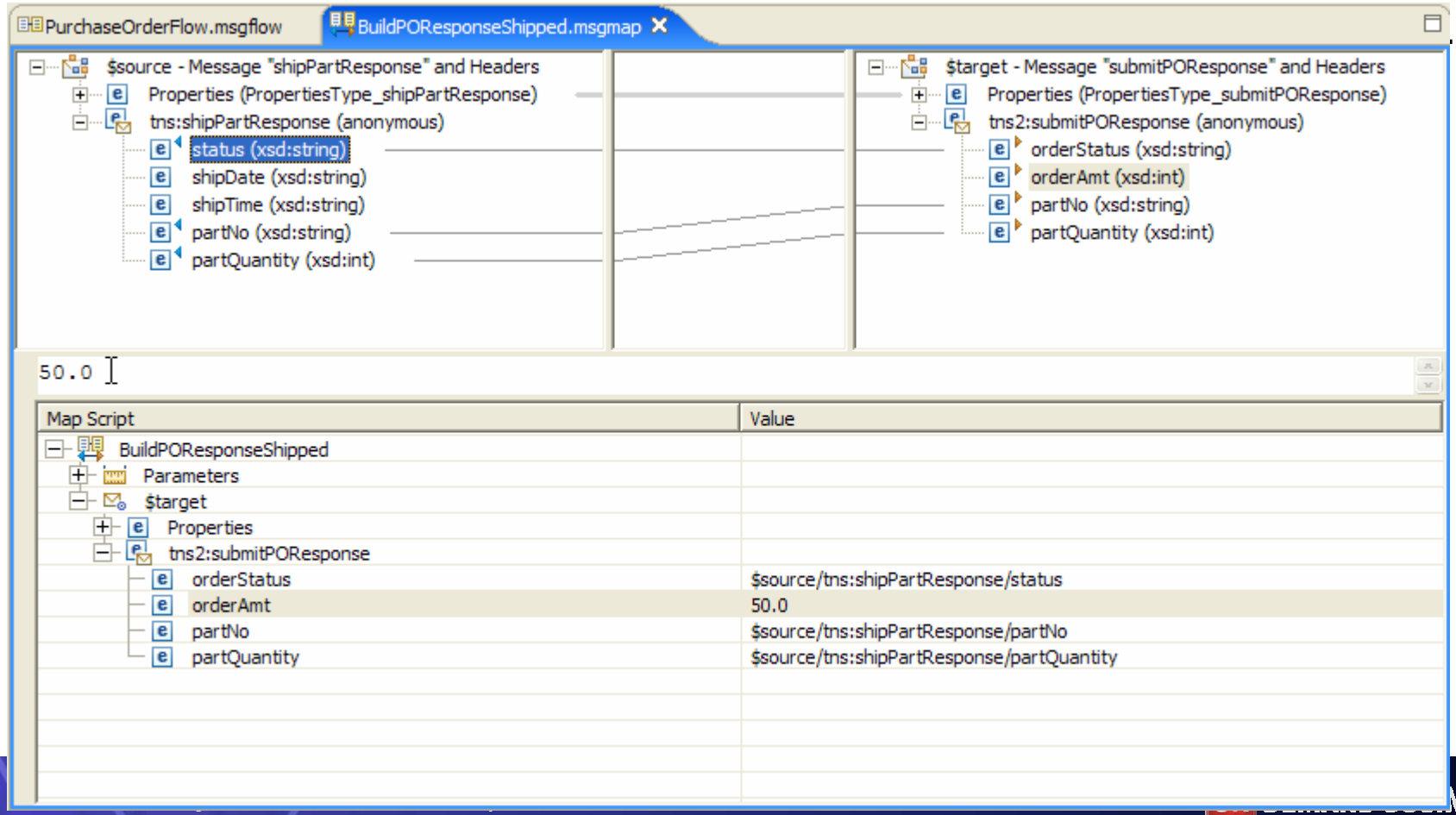


Scenario: Step 11b – Open Map



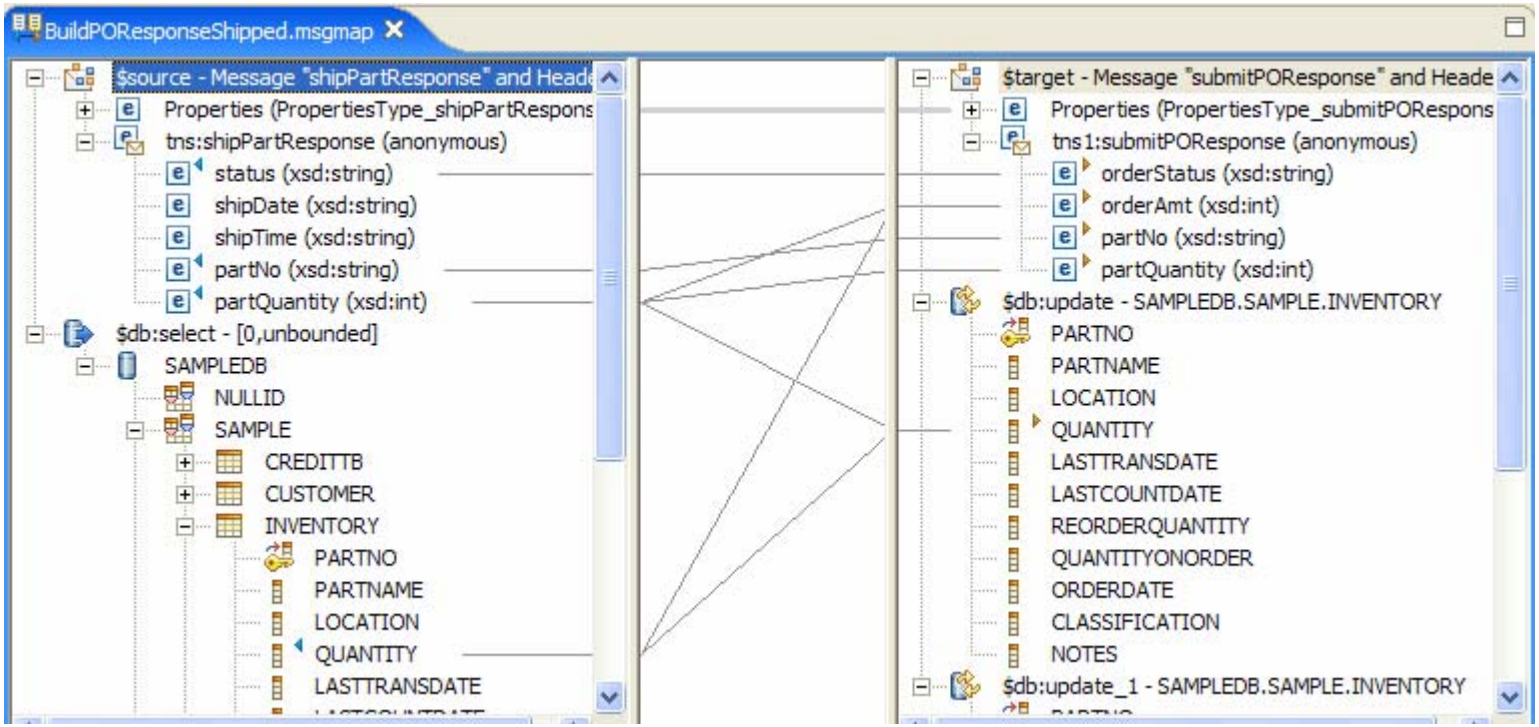
Scenario: Step 11c – Do Mapping

- Select $source and $target roots
- Right click on $target and select Map By Name. Press Enter.
- Drag and drop `status` to `OrderStatus`.



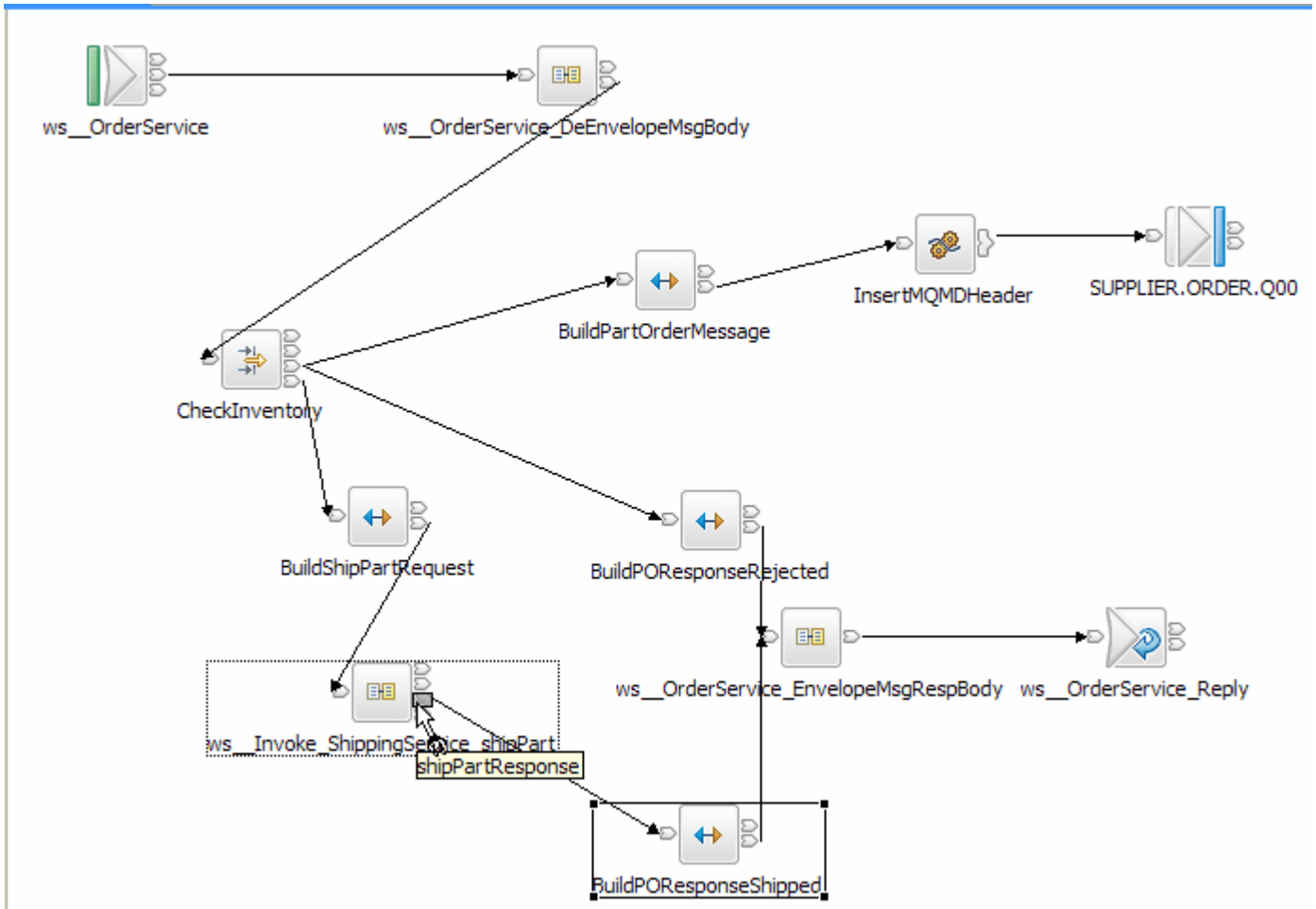
About that “orderAmt” value

- The orderAmt value should be set to the quantity times the part price, but in interest of time aren’t doing that here.
- But it is doable. In fact, you could in one map, check the inventory, limit the orderAmt to what is actually available, update the inventory, and compute the orderAmt! Here it is:

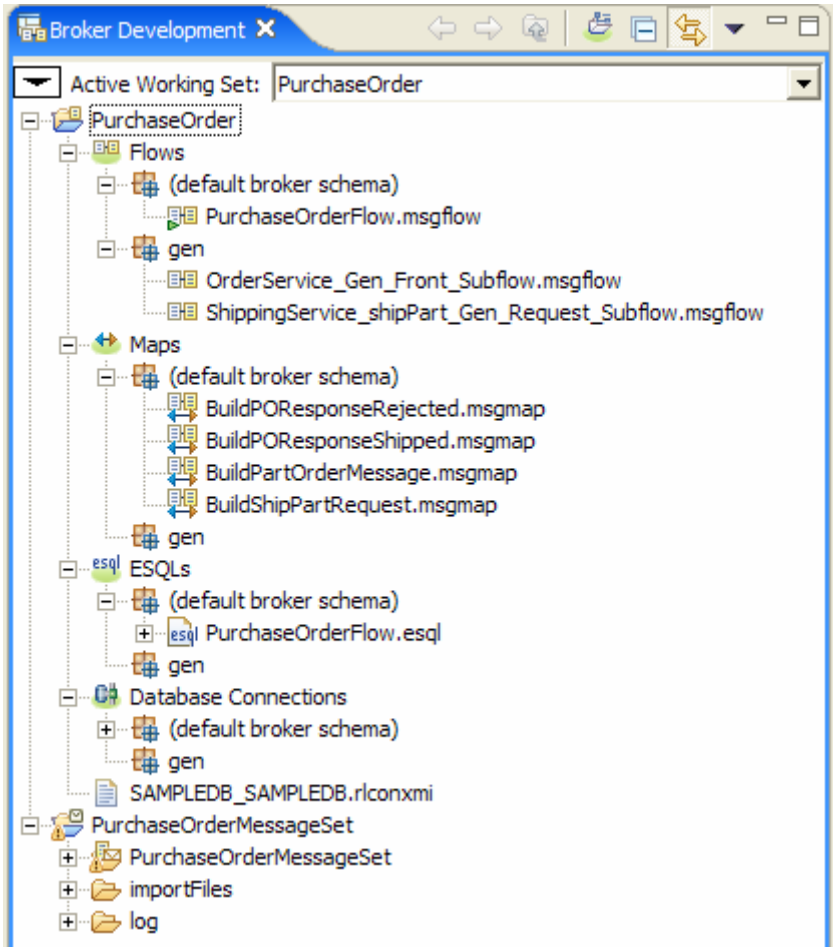


Scenario: Step 11d – Wire Map Node

- Wire as shown below: from `shipPartResponse` terminal to map, and from map's out terminal to generated reply `SOAPEnveloper` node



AUTHORING STEPS ARE DONE!

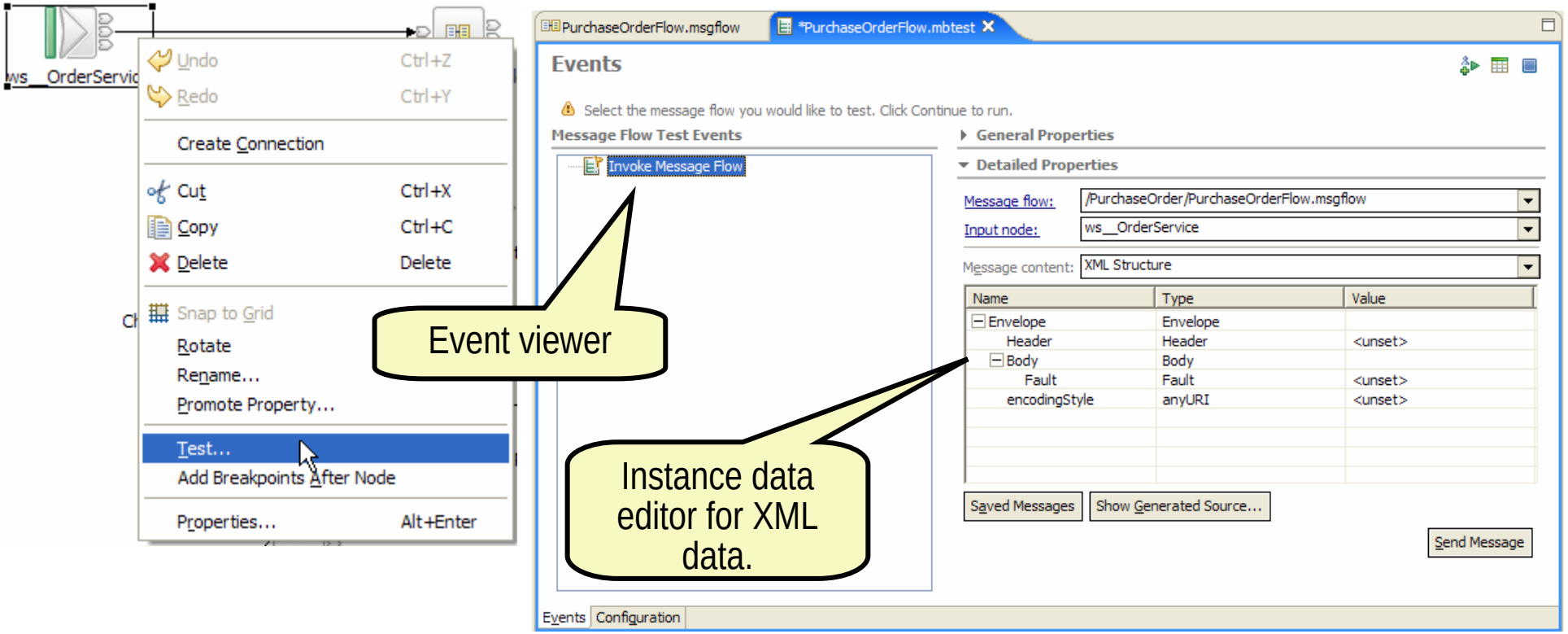


Time to Test!

- In previous releases, there was no built-in way to test a flow that starts with HTTPInput.
- New in 6.0.2 is support for a new “Test client” for testing flows that are started via HTTPInput or MQInput nodes.

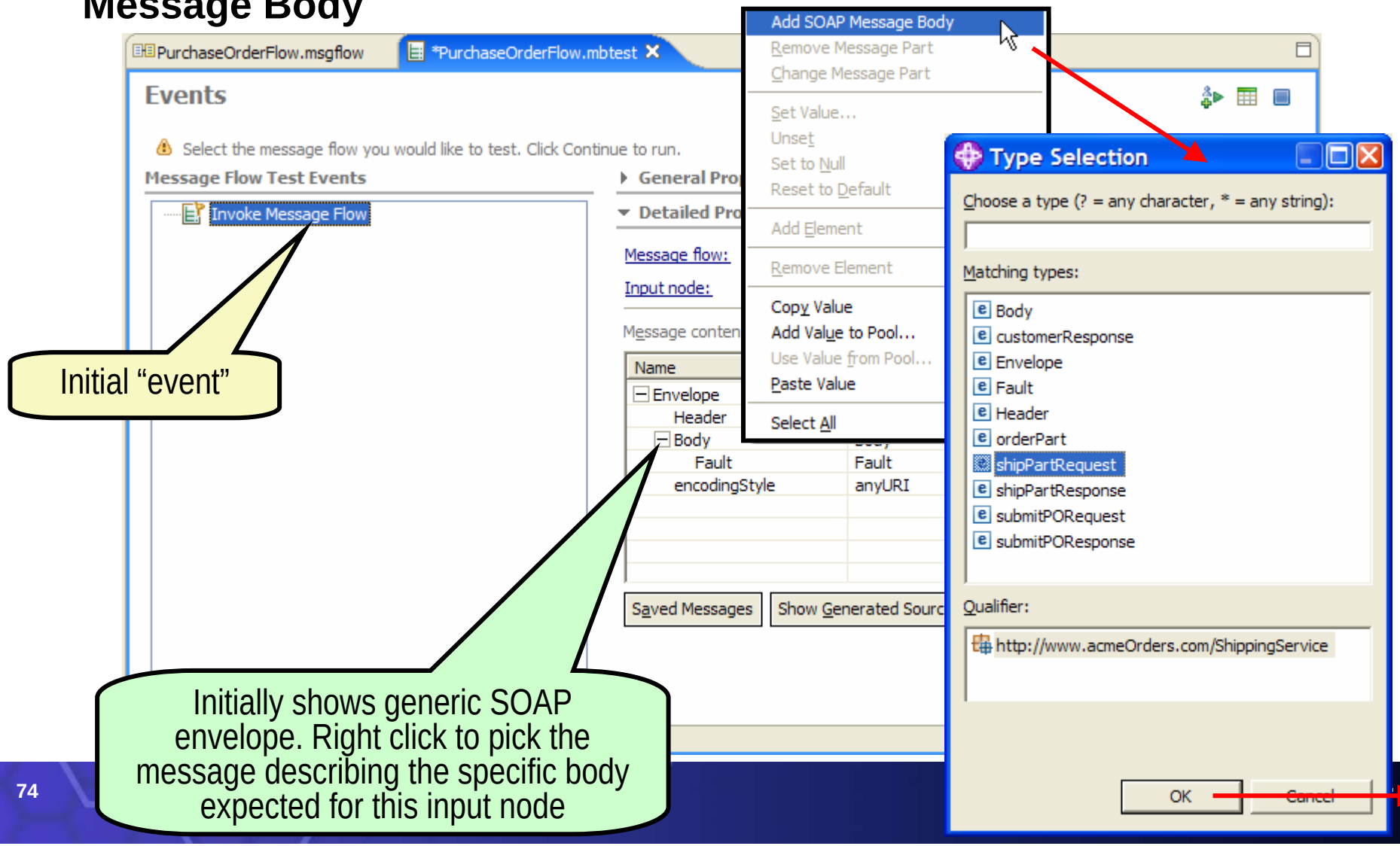
Scenario: Step 12a – Invoke Test Client

- Assuming you have created all necessary configurations (config mgr, queue mgr listener, broker, queues, etc), you can test your flow by right clicking on the HTTPInput node:



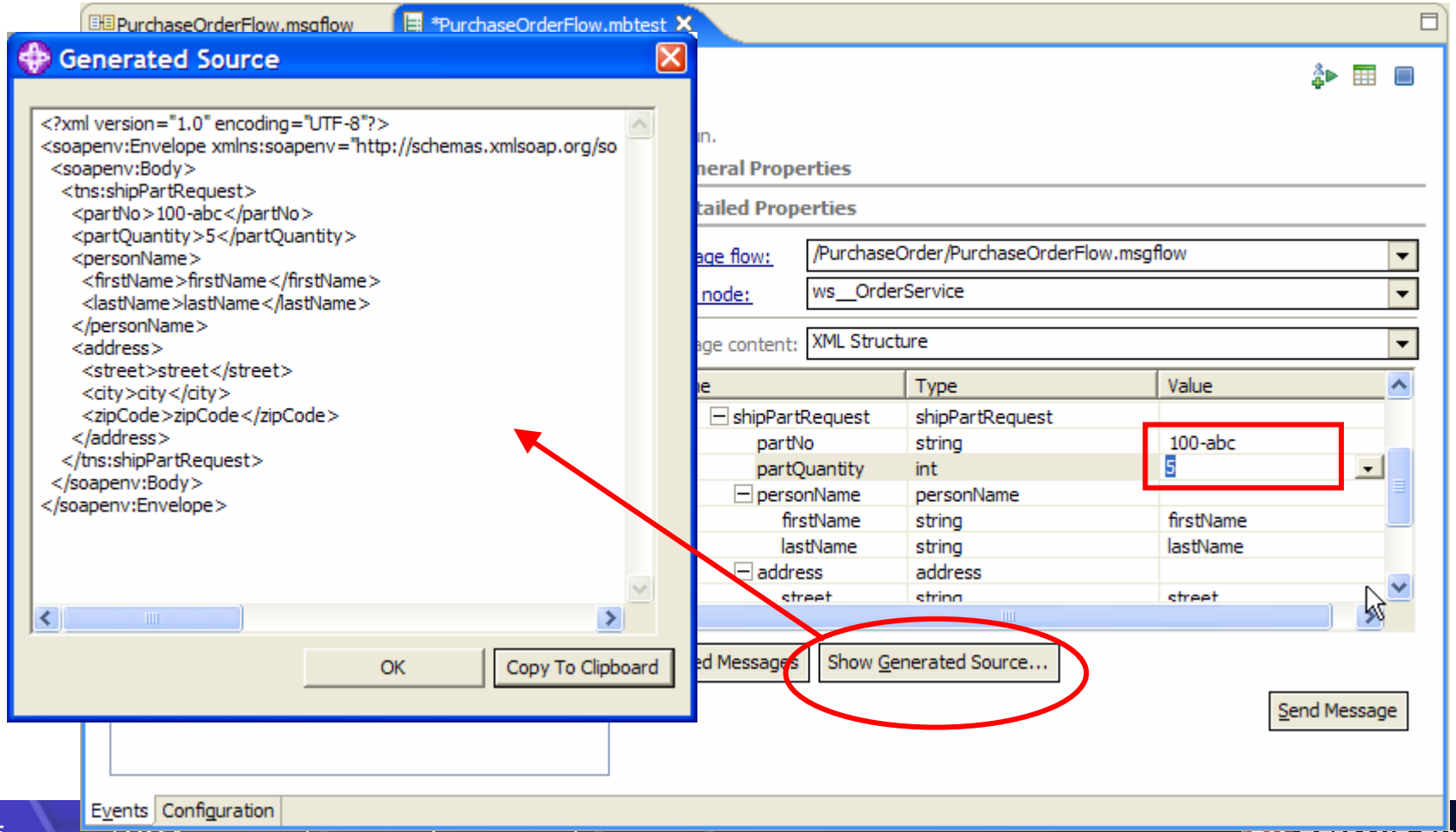
Scenario: Step 12b – Choose Message Body

- Right click on “Body” in message content, and select “**Add SOAP Message Body**”



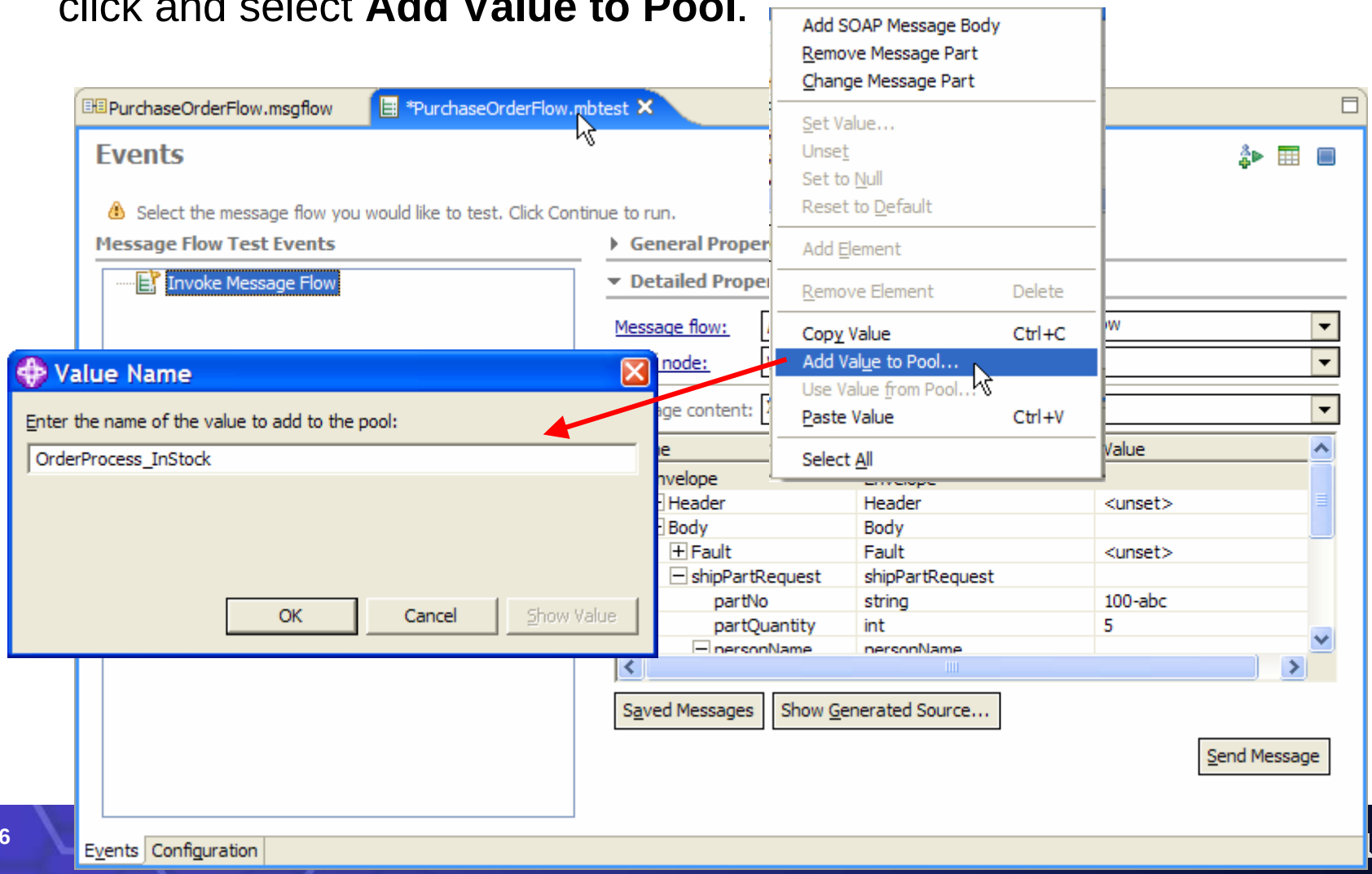
Scenario: Step 12c – Set Input Data

- Set `partNo` to 100-abc and `partQuantity` to 5. Press **Send Message**. Then press **Show Generated Source**…



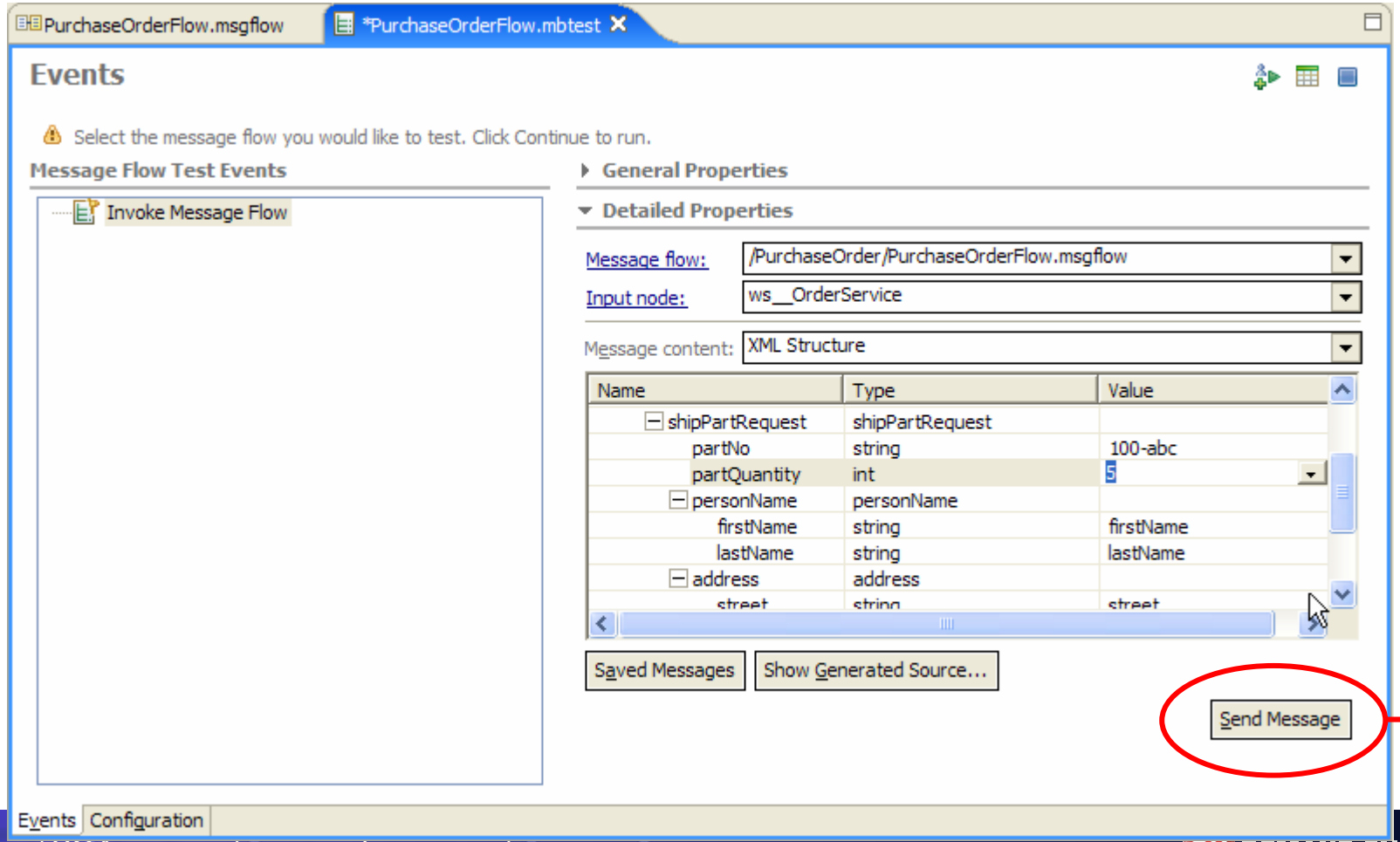
Scenario: Step 12d – Save Input Data

- Since we don't want to have to enter the data in each time, right click and select **Add Value to Pool**.



Scenario: Step 12e – Run Test

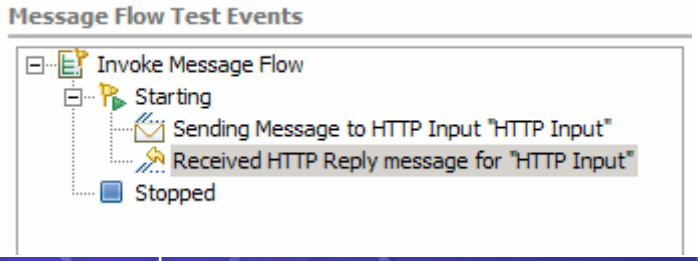
- Ok, time to launch! Press Send Message….

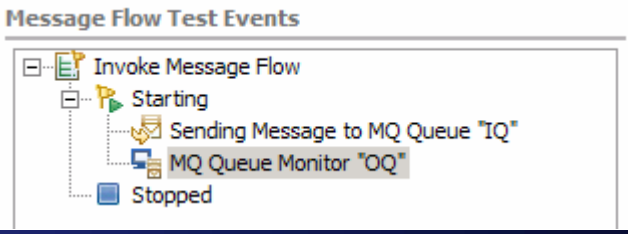


Scenario: Step 12e – Events

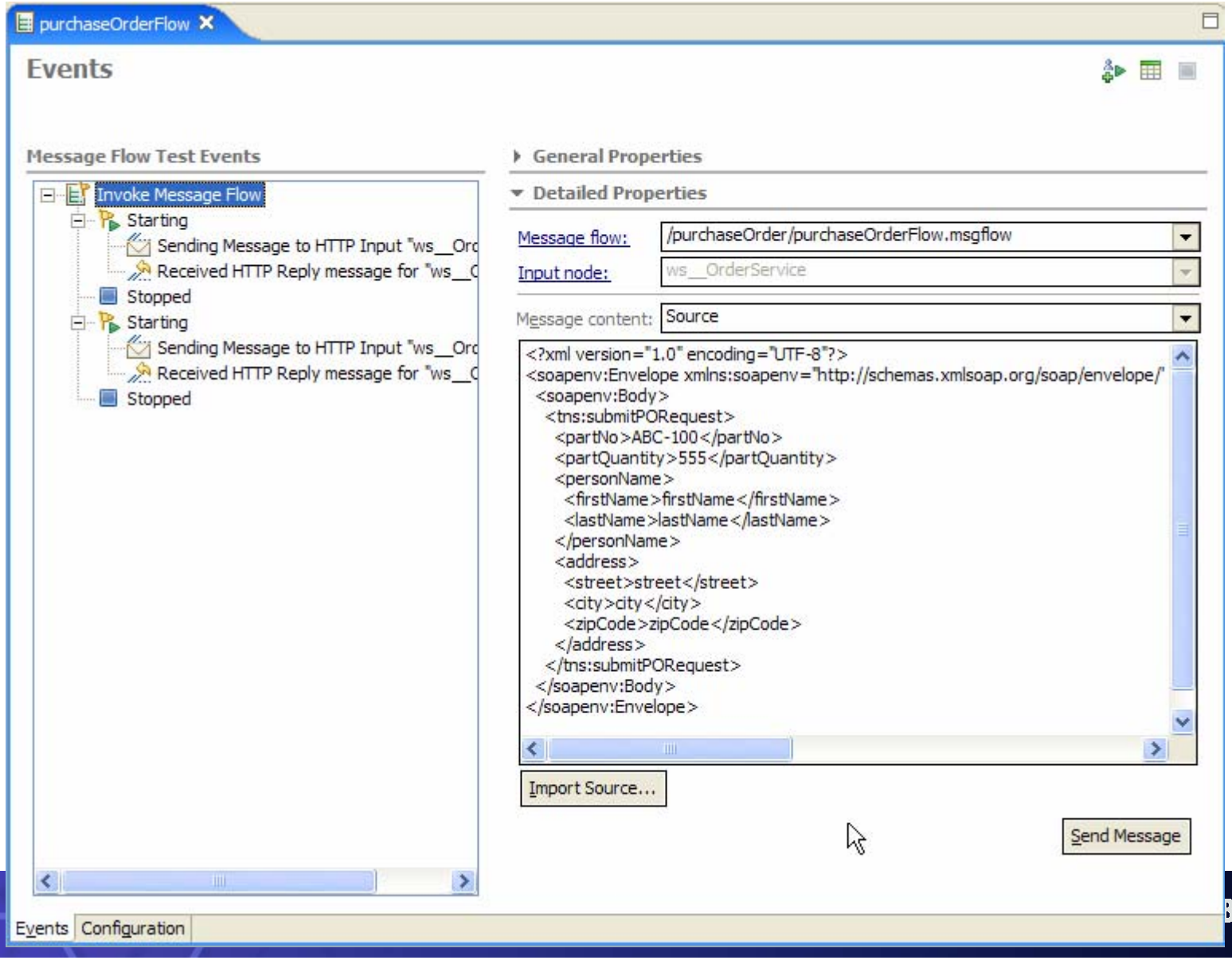
- The Message Flow Test Events tree will show events:

Event Type	What you can see
Starting	All MQ output queue monitors
Sending message to HTTP Input node	Request message in a readonly source editor
Sending message to MQ Input node queue	The queue, queue manager, host, port of the MQ queue that the message came from The message in a readonly source editor.
Receiving message from HTTP Reply node	Response message in a readonly source editor.
Receiving message at MQ Output node queue	The queue, queue manager, host, port of the mq queue that is being listened to. The reply message in a readonly source editor
Stopping	



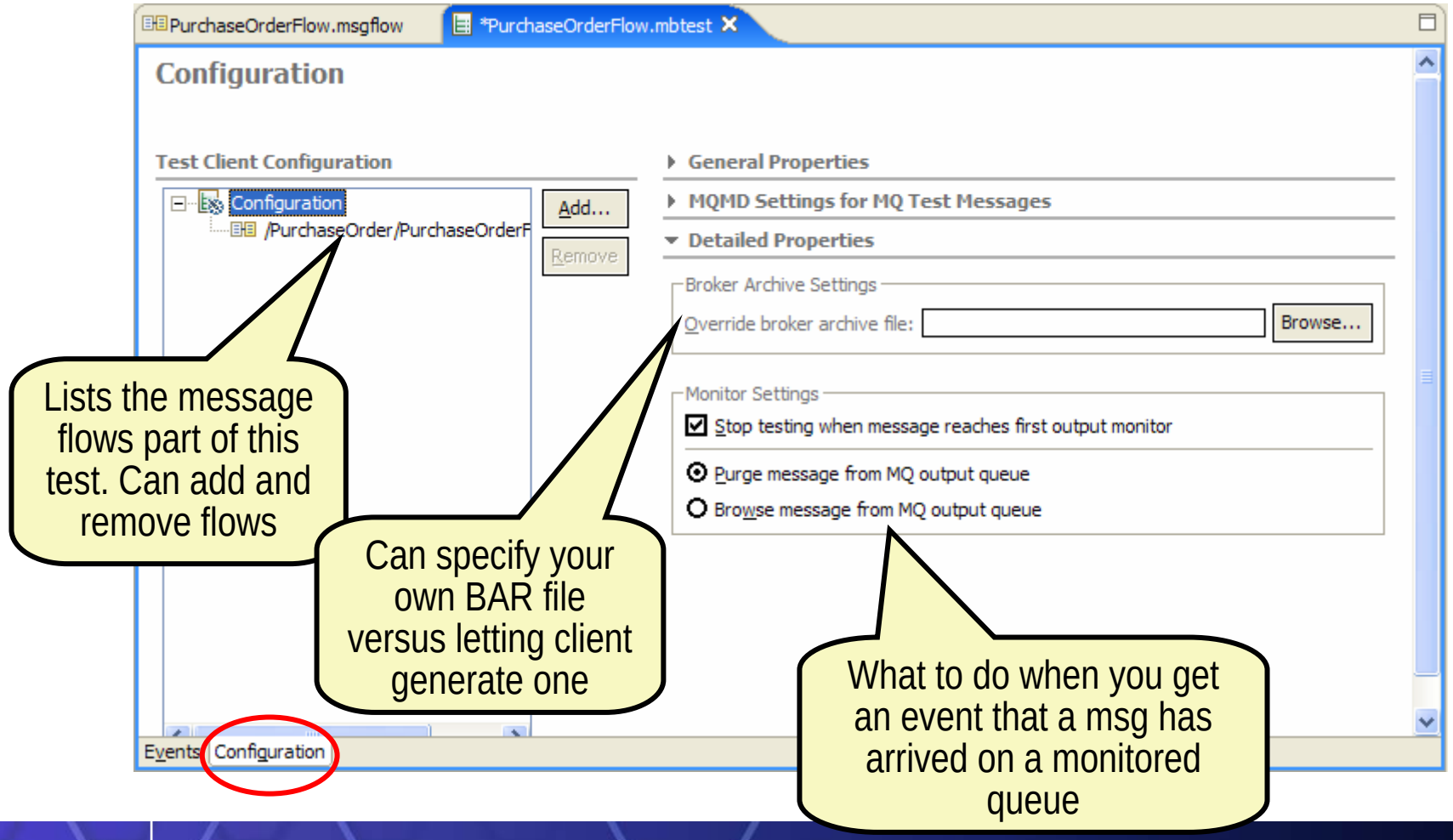


Test Client after two runs



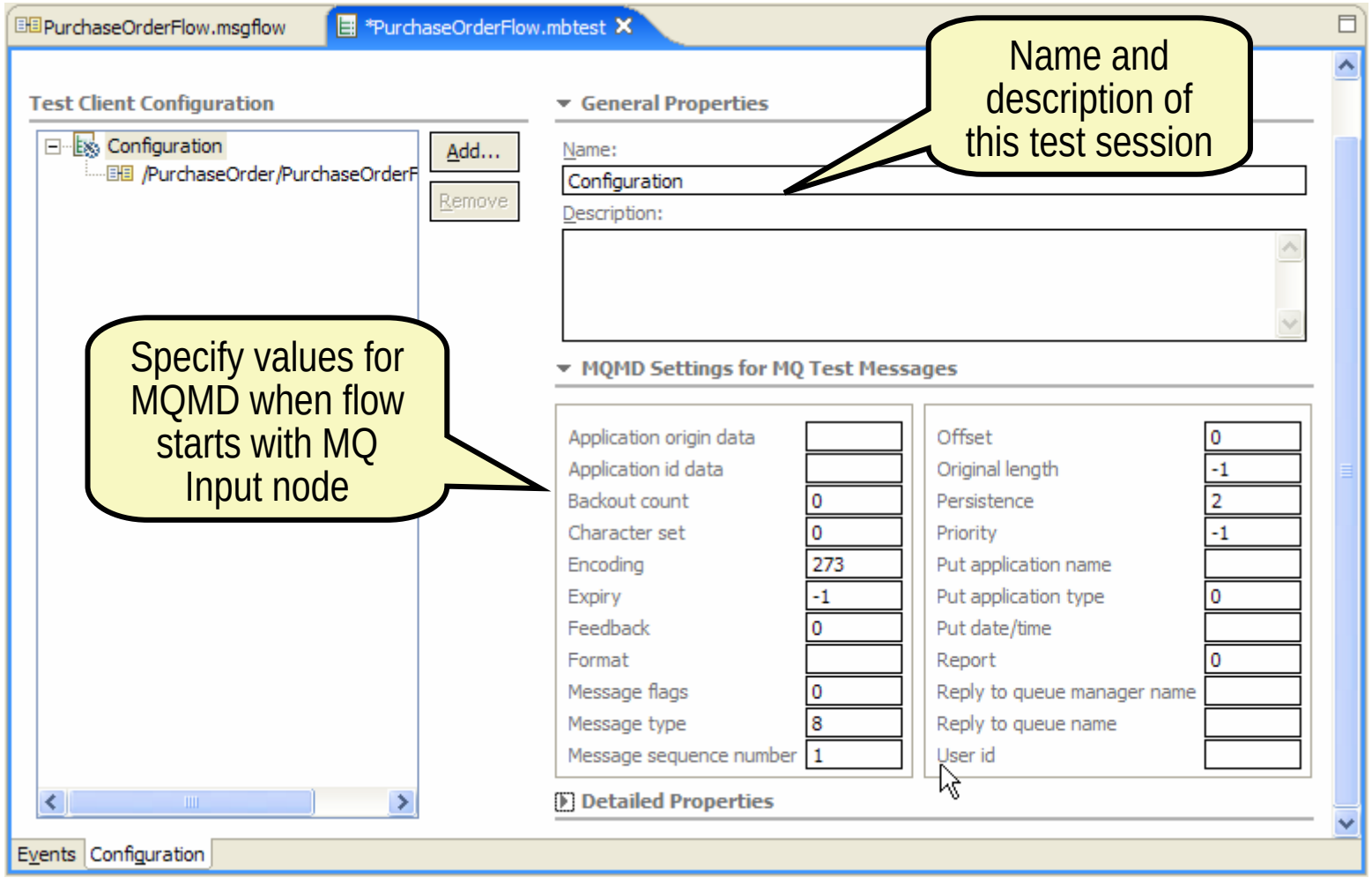
More about Test Client

- The Configuration page is for configuring your test session:



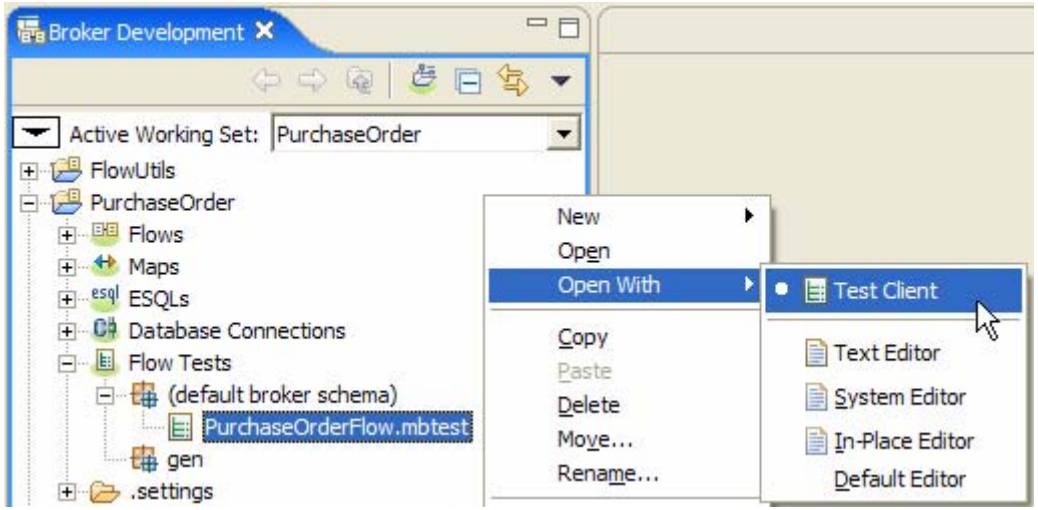
More about Test Client

- The Configuration page is for configuring your test session:



More about Test Client

- You can save your test session configuration. This allows you to reload it from the navigator:



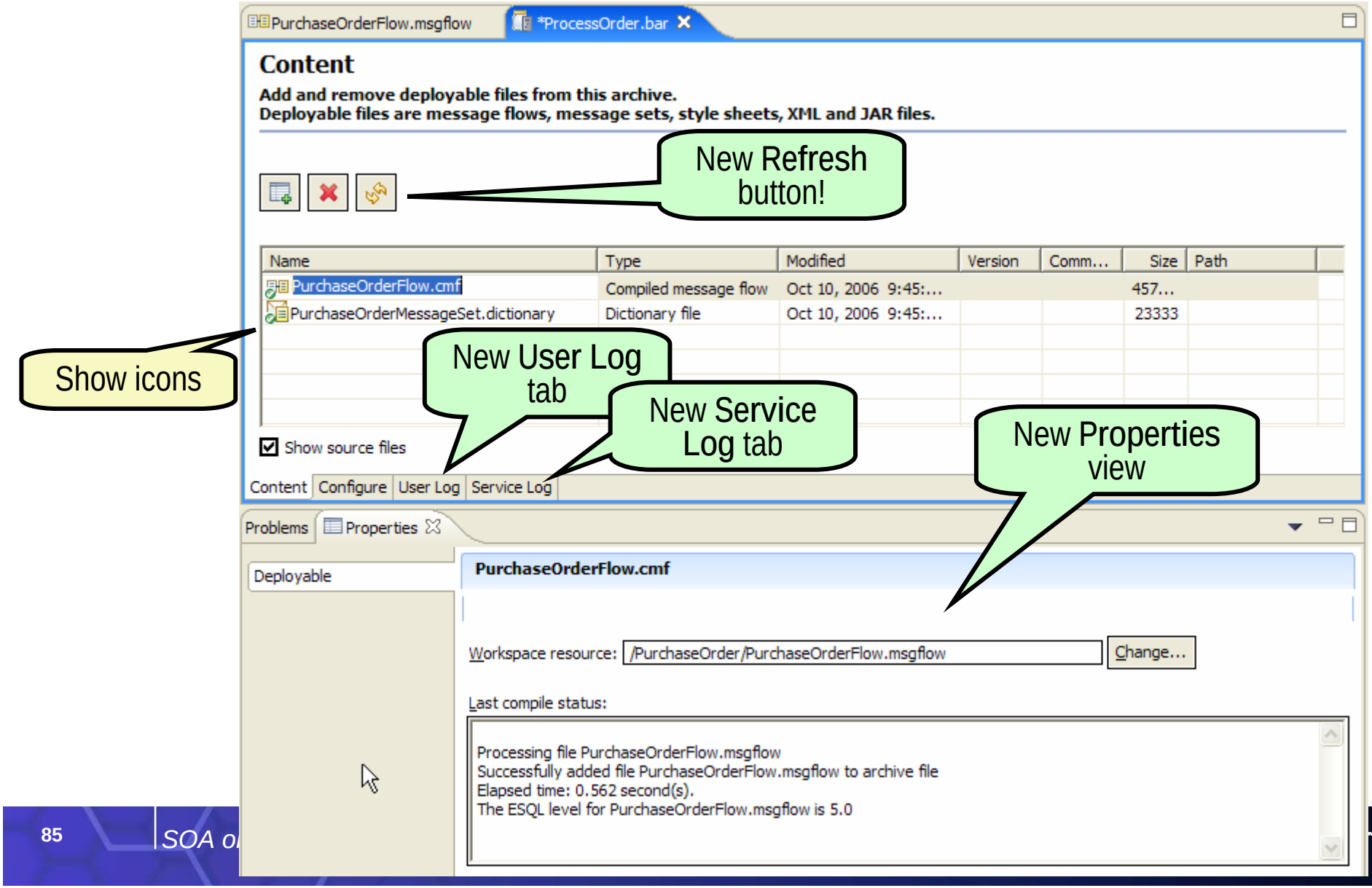
Looking Ahead for the Test Client

- This is only the first release for this exciting new technology. It is technology we are sharing with WebSphere Integration Developer (WID), where it has proved very popular.
- Over time we have many ideas for evolving it beyond this first release, including:
  - Replacing Enqueue/Dequeue support for MQ.
  - Offer support for other protocols beyond just MQ and HTTP. Eg: JMS
  - Support events for entering and exiting each node in the flow
  - Support emulating nodes, so you don't have to actually invoke payroll. Instead, you just supply a result value as thought it had been invoked
  - Support testcase creation, management and execution

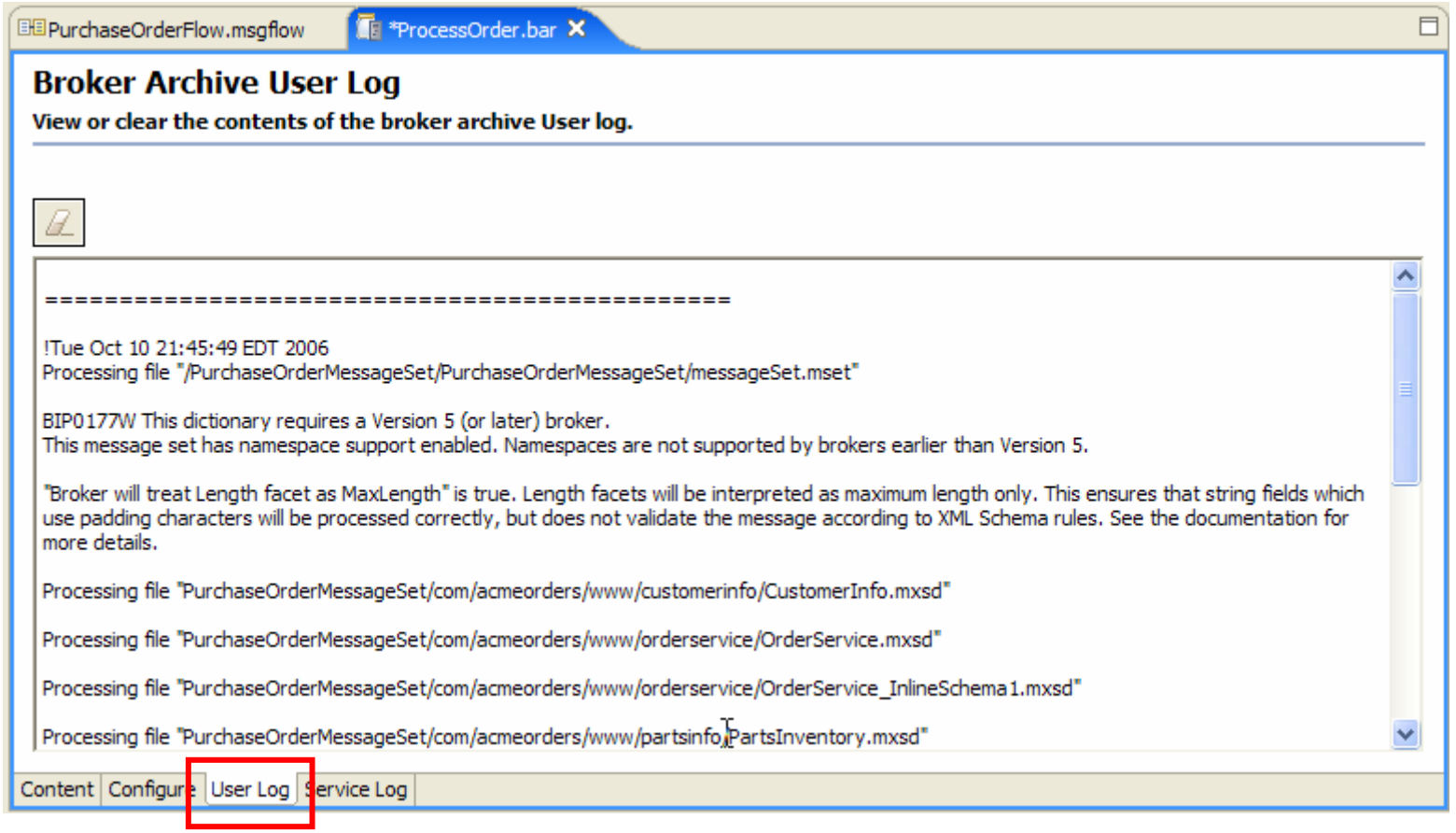
Other Enhancements

- There are plenty of other enhancements that we couldn't show in the scenario…

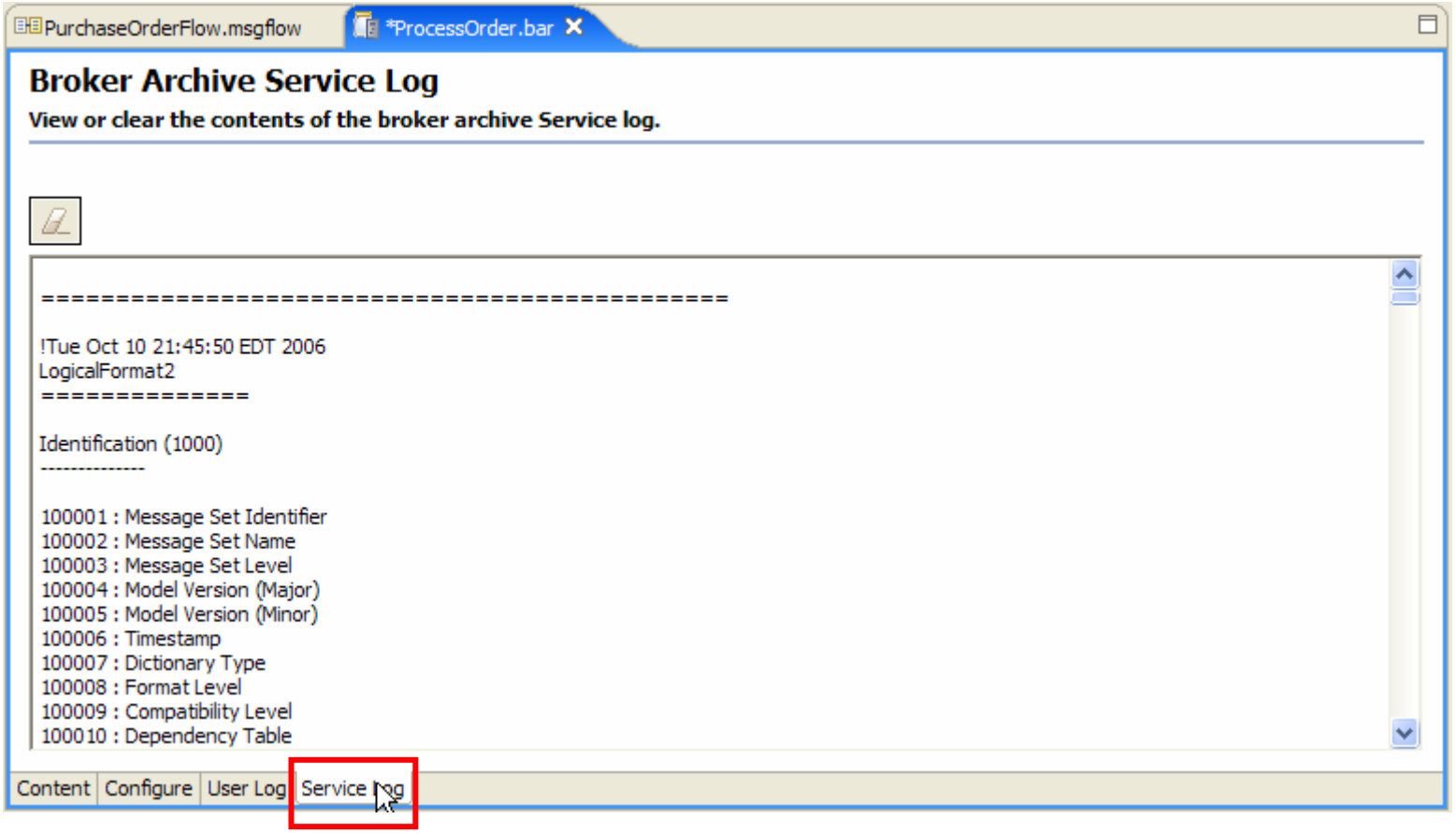
BAR Editor


PurchaseOrderFlow.msgflow
*ProcessOrder.bar
Content
Add and remove deployable files from this archive.
Deployable files are message flows, message sets, style sheets, XML and JAR files.
New Refresh button!
Name
Type
Modified
Version
Comm...
Size
Path
PurchaseOrderFlow.cmf
Compiled message flow
Oct 10, 2006 9:45:...
457...
PurchaseOrderMessageSet.dictionary
Dictionary file
Oct 10, 2006 9:45:...
23333
Show icons
New User Log tab
New Service Log tab
New Properties view
Show source files
Content
Configure
User Log
Service Log
Problems
Properties
Deployable
PurchaseOrderFlow.cmf
Workspace resource:
/PurchaseOrder/PurchaseOrderFlow.msgflow
Change...
Last compile status:
Processing file PurchaseOrderFlow.msgflow
Successfully added file PurchaseOrderFlow.msgflow to archive file
Elapsed time: 0.562 second(s).
The ESQL level for PurchaseOrderFlow.msgflow is 5.0

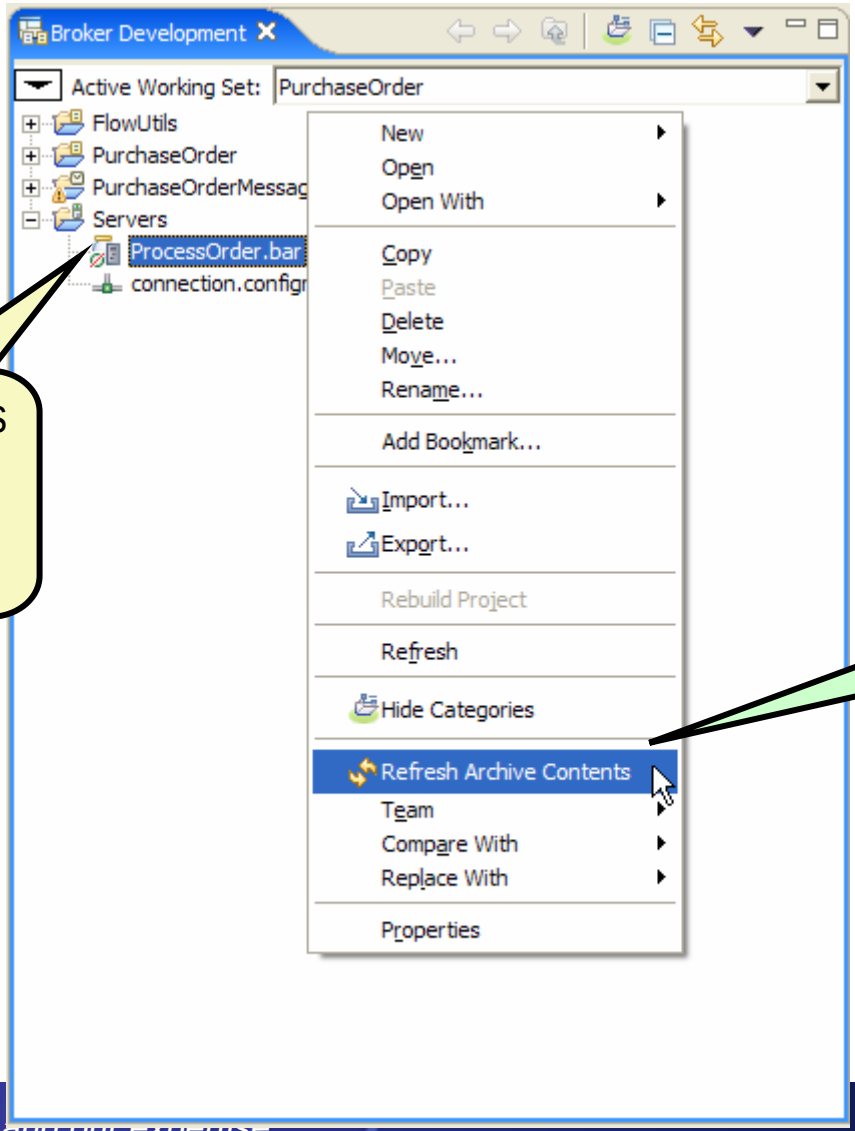
More BAR Editor



More BAR Editor



More BAR Editor - Refresh



Icon changes when BAR needs refreshing

New Refresh Archive Contents action

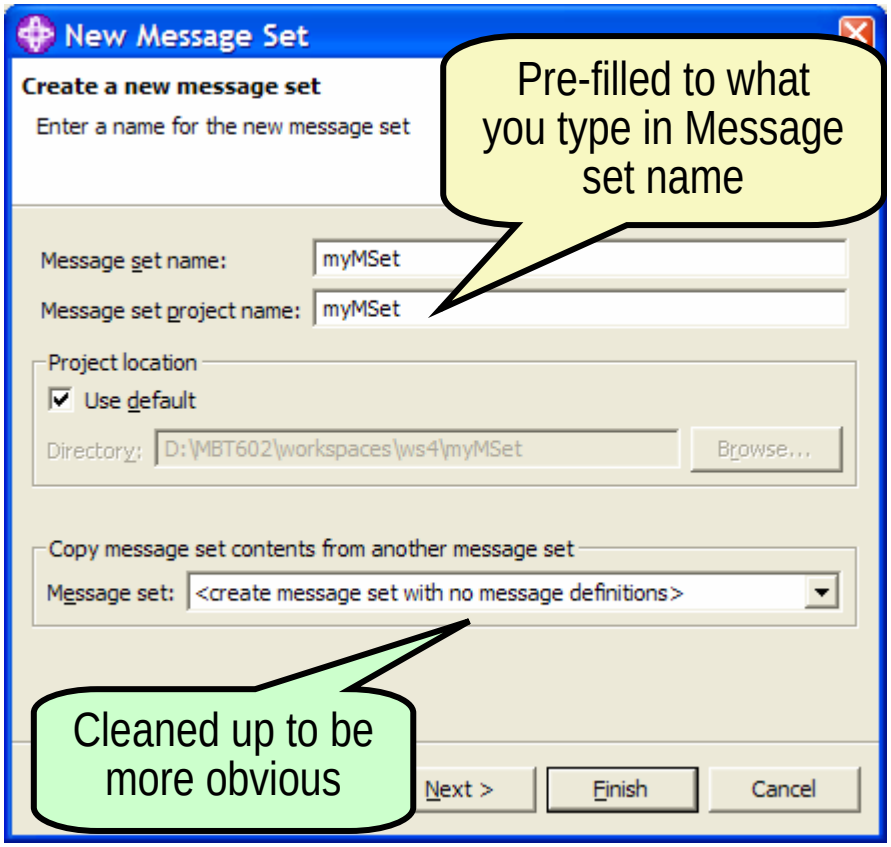
Command : mqsiapplybaroverride

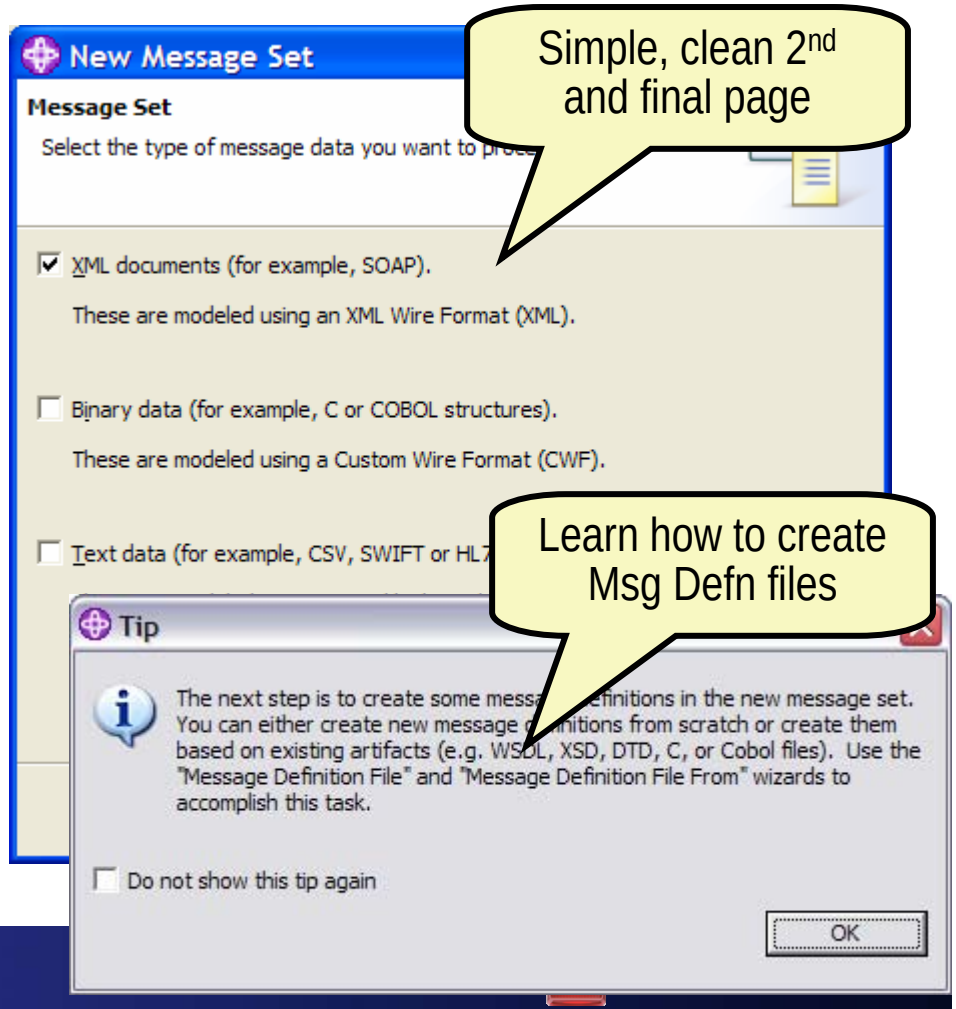
- There is also a new command for updating the override values in a BAR file:

  - >>-mqsiapplybaroverride-- -b --barFile -p --propertiesFile
  - Parameters
    - -b barFile
      - (Required) The path to the broker archive file to apply override values to.
      - ie relative from the executable or a full path to the broker archive file
    - -p propertiesFile
      - (Required) The path to the properties file containing the override values
      - i.e. relative from the executable or a full path to the properties file

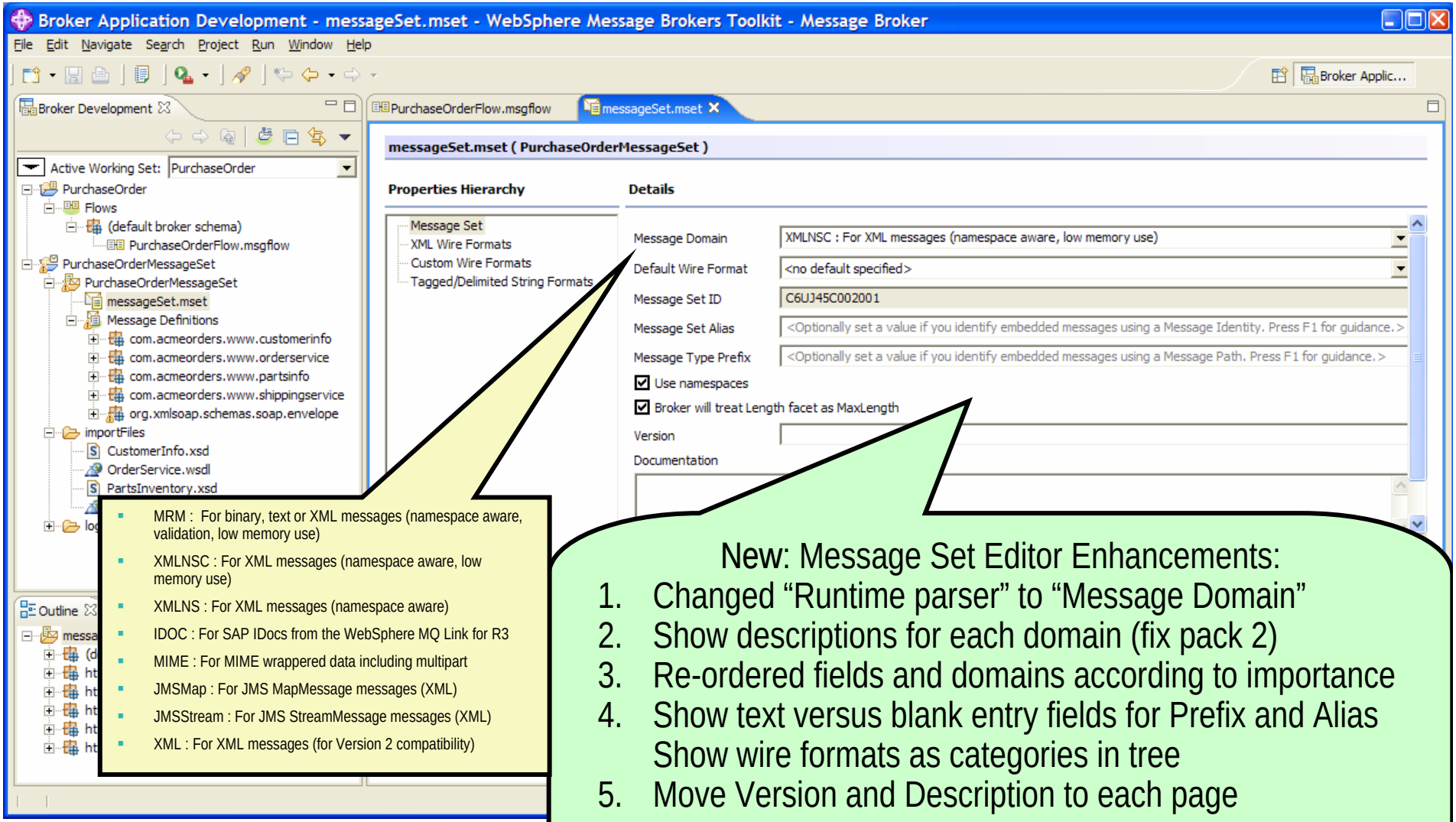
Simpler New Message Set wizard

- We have removed the New Message Set Project wizard, and instead enhanced the New Message Set wizard to allow creating the project at the same time:

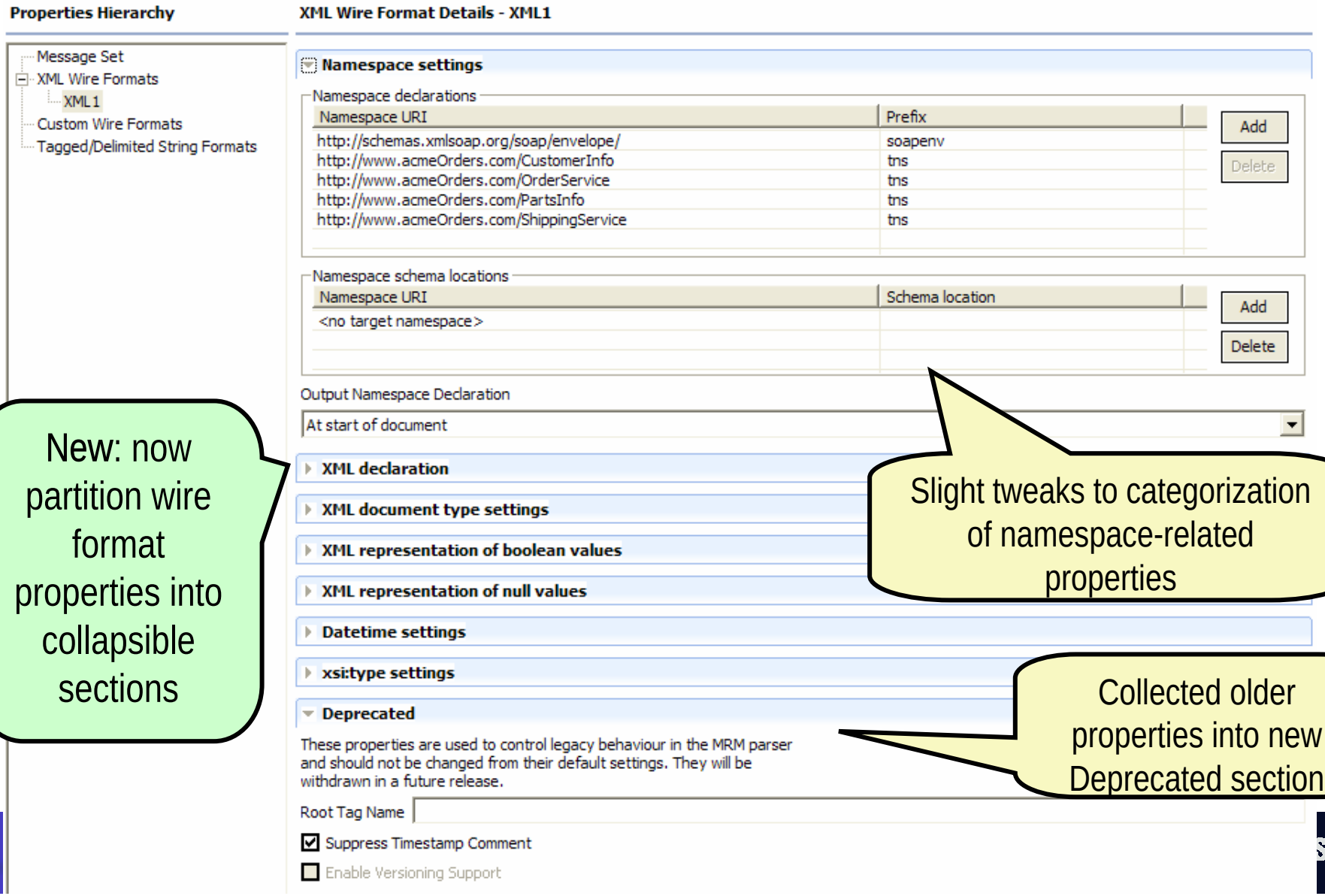




Message Set Editor

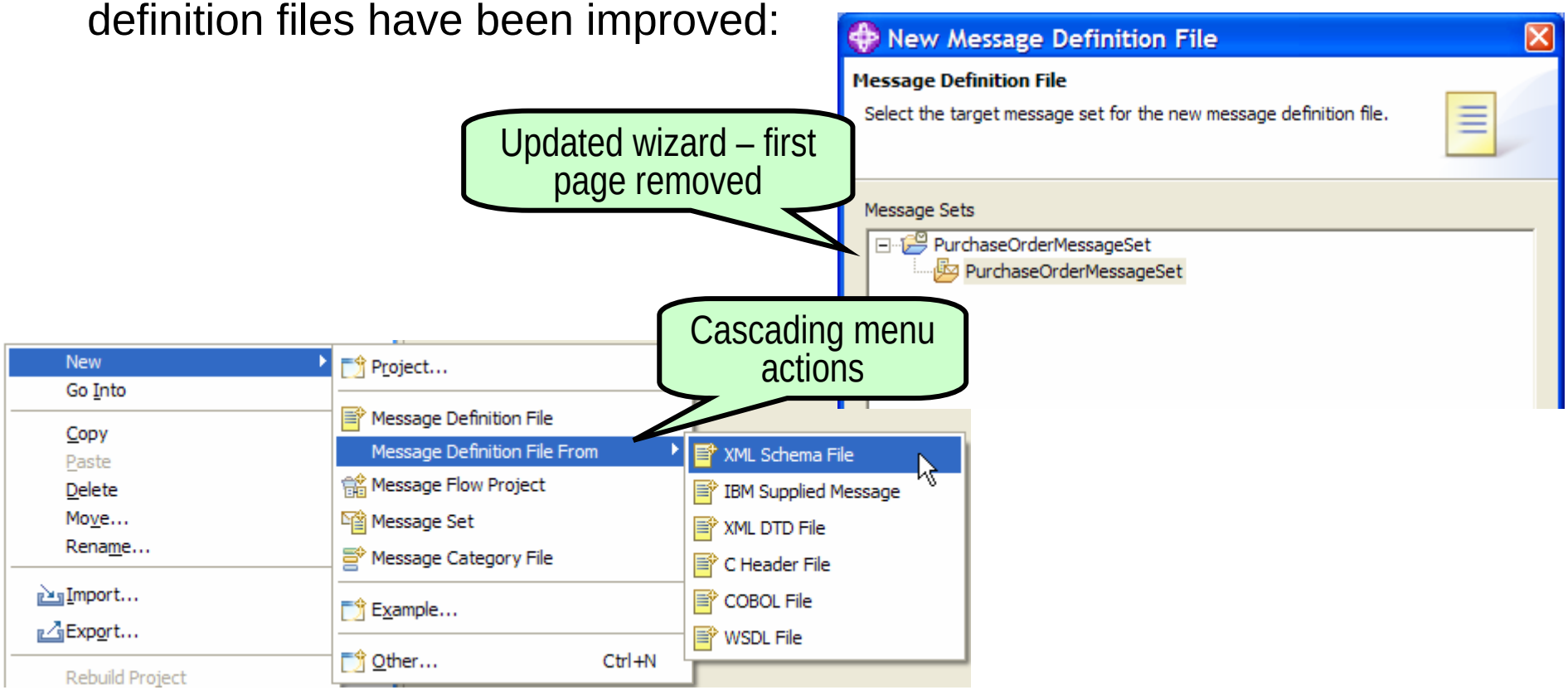


Message Set Editor for Wire Formats



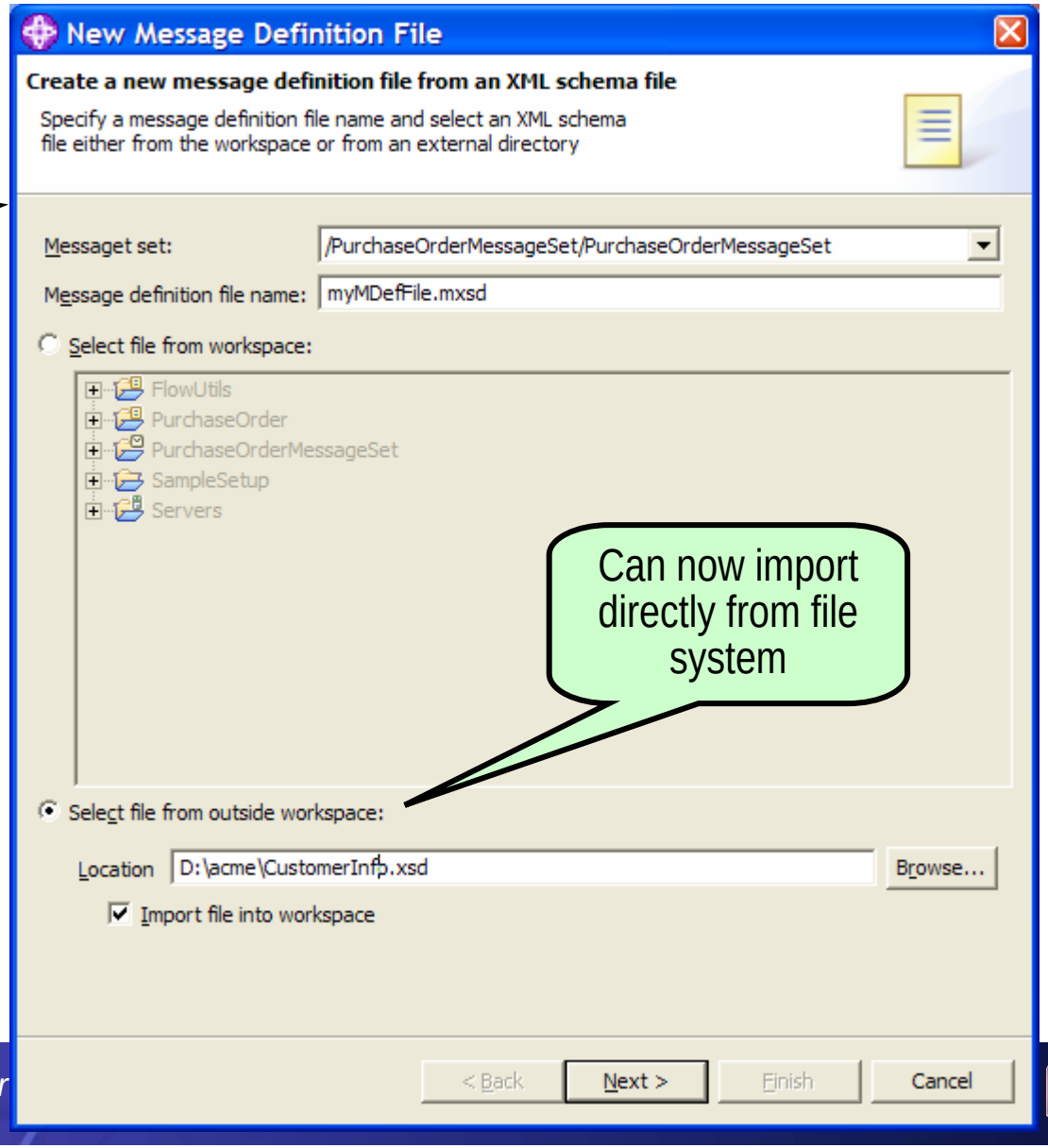
Simpler New Msg Defn File wizards

- The menu actions and most of the wizards for creating message definition files have been improved:

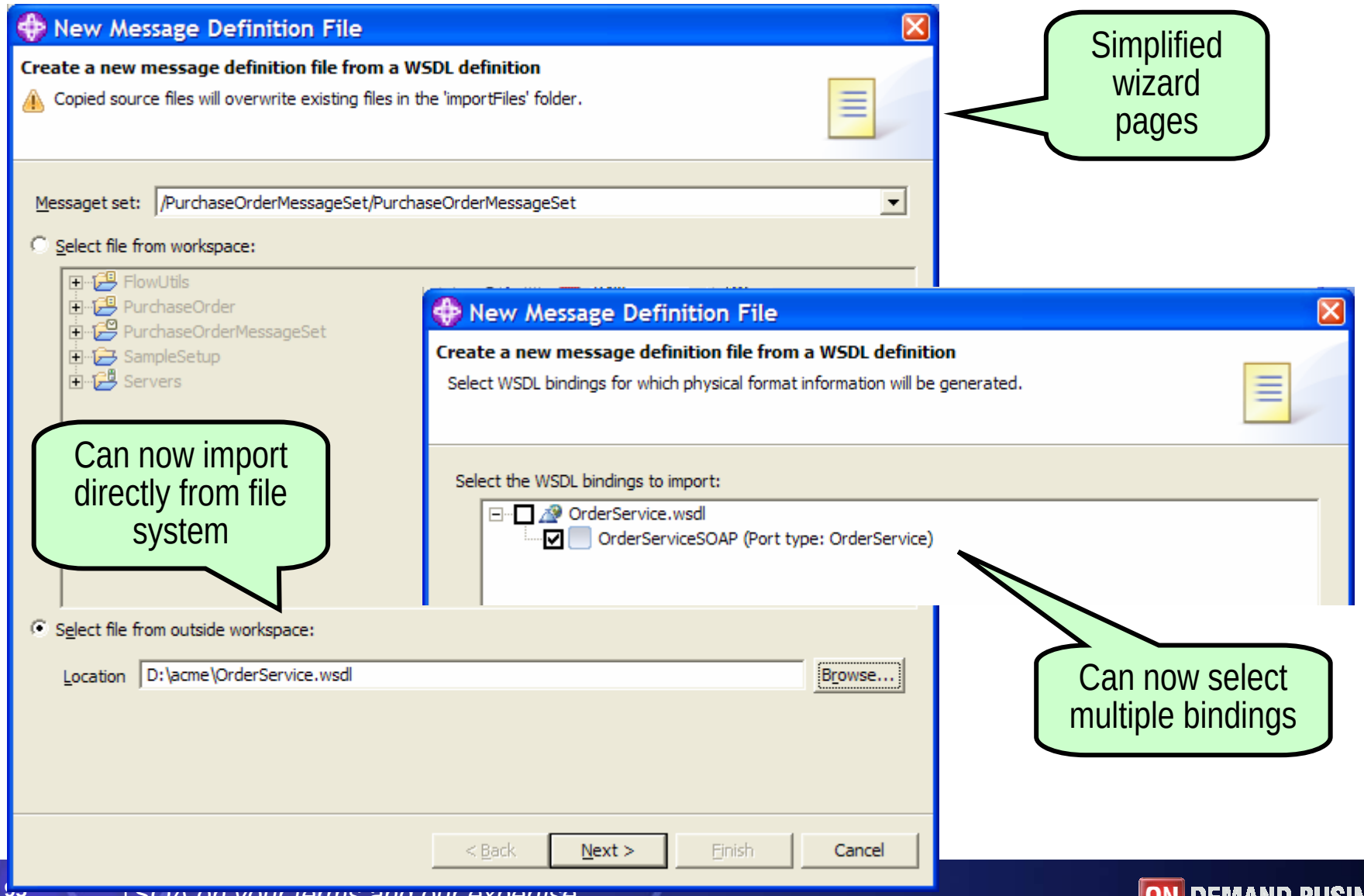


XSD, DTD, COBOL & C wizards

Simplified wizard pages



WSDL wizard


New Message Definition File
Create a new message definition file from a WSDL definition
Copied source files will overwrite existing files in the 'importFiles' folder.
Messaget set: /PurchaseOrderMessageSet/PurchaseOrderMessageSet
Select file from workspace:
FlowUtils
PurchaseOrder
PurchaseOrderMessageSet
SampleSetup
Servers
Select file from outside workspace:
Location D:\acme\OrderService.wsdl
Browse...
< Back
Next >
Finish
Cancel
New Message Definition File
Create a new message definition file from a WSDL definition
Select WSDL bindings for which physical format information will be generated.
Select the WSDL bindings to import:
OrderService.wsdl
OrderServiceSOAP (Port type: OrderService)
Simplified wizard pages
Can now import directly from file system
Can now select multiple bindings

Looking Ahead

- Other high priority consumability features we are hoping to do **:
  - New non-programming Filter nodes
    - Specifying XPath expressions into the message.
      First/all non-null expression fires associated terminal.
    - Specifying database query expressions using message content as input.
      First/all non-null database query fires associated terminal
  - New transport header nodes to create MQMD, HTTP, JMS headers
  - MQOutput nodes auto-creates an MQMD when none present
    - Saves having to use Compute node prior to output nodes
  - New JMSReply node
    - So JMSOutput isn't overloaded

***** These statements represent IBM's current intent for future product delivery. IBM directions and plans are subject to change without notice***

What are you waiting for!

- eGA was Dec 22nd 2006 for existing customers
- Go to WebSphere Message Broker Recommended Fixes site
  http://www-1.ibm.com/support/docview.wss?rs=849&uid=swg27006041
- SOAPExtract & SOAPEnvelope nodes available via SupportPac IA9O **
  http://www-1.ibm.com/support/docview.wss?rs=171&uid=swg24014070
- Installation using RPU

- eGA for Passport Advantage customers in Feb 2007
- pGA (CDs) in March 2007

**** Requires WMB V6.0 FixPack 3 runtime***