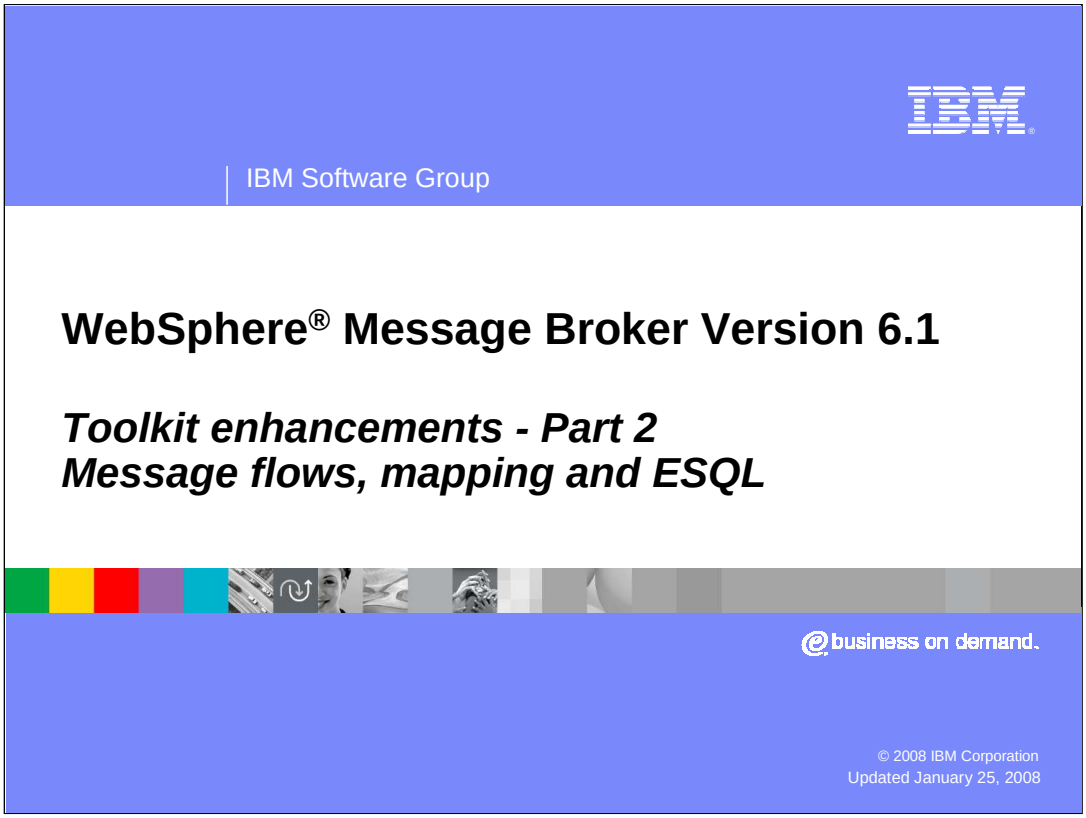
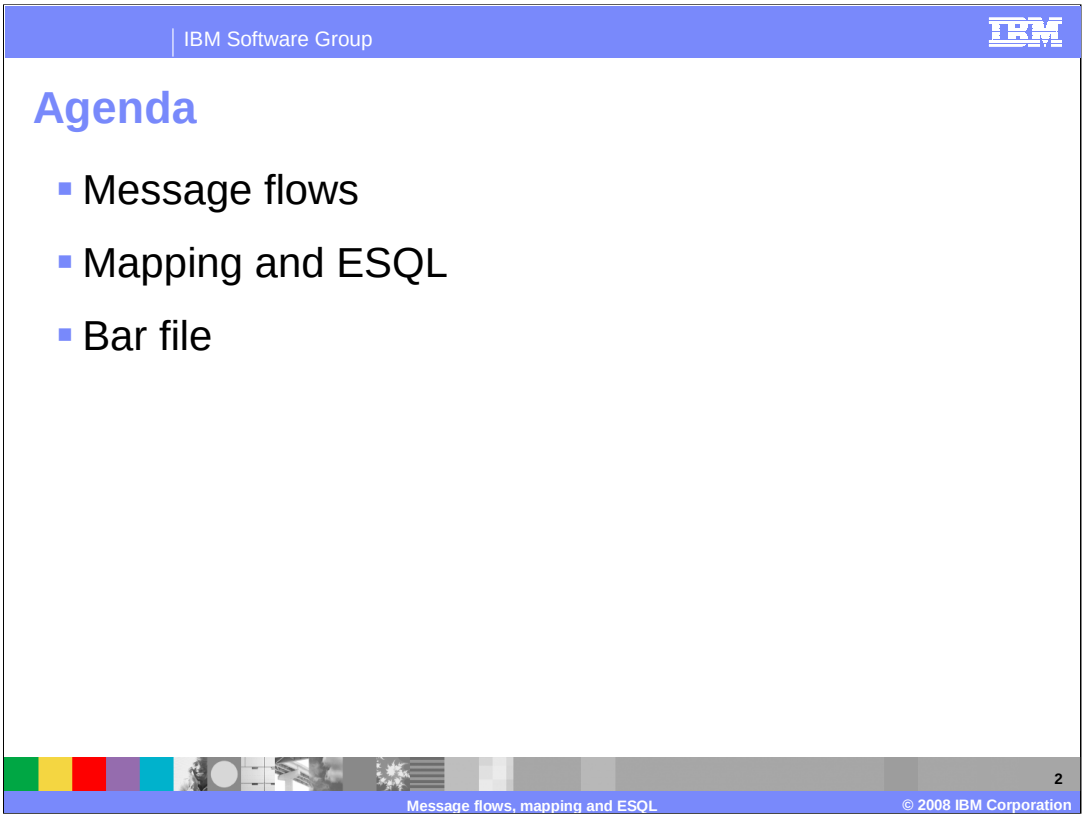
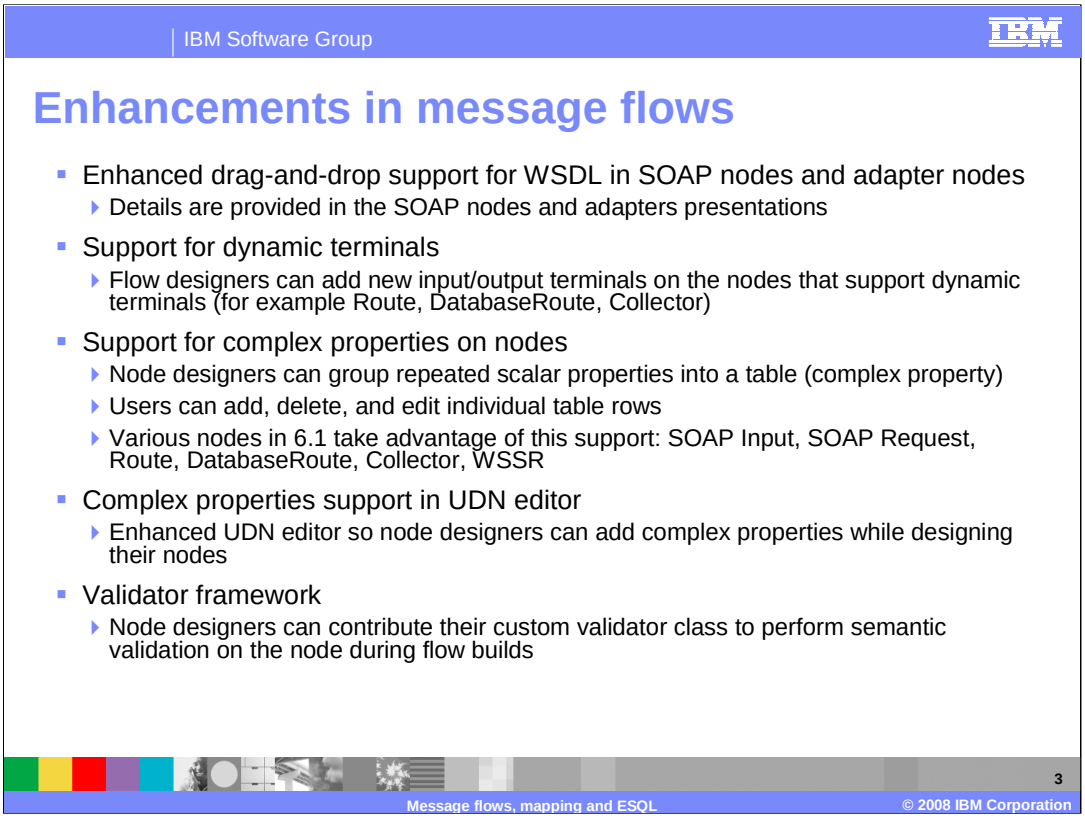


This presentation is the second of four presentations looking at the Toolkit in Message Broker Version 6.1. This presentation will look at topics concerned with the building and testing of Message Flows.



The main topics in the presentation are changes in the toolkit around message flows themselves, followed by changes made to the mapping node and ESQL.

Finally, it looks at improvements made in the deployment of artifacts to the Broker.



This slide summarizes the enhancements in Message Flows in version 6.1.

WSDL drag-and-drop support has been enhanced for the SOAP and Adapter nodes. This is covered in detail in Part 3 of the Toolkit presentations.

Message Broker version 6.1 has added support for Dynamic Node Terminals. Dynamic terminals are terminals that you can add to certain nodes after you have added them to a message flow in the Message Flow editor. For example, you can add dynamic output terminals to the Route and DatabaseRoute nodes, or you can add dynamic input terminals to the Collector node. You can also delete and rename dynamic terminals.

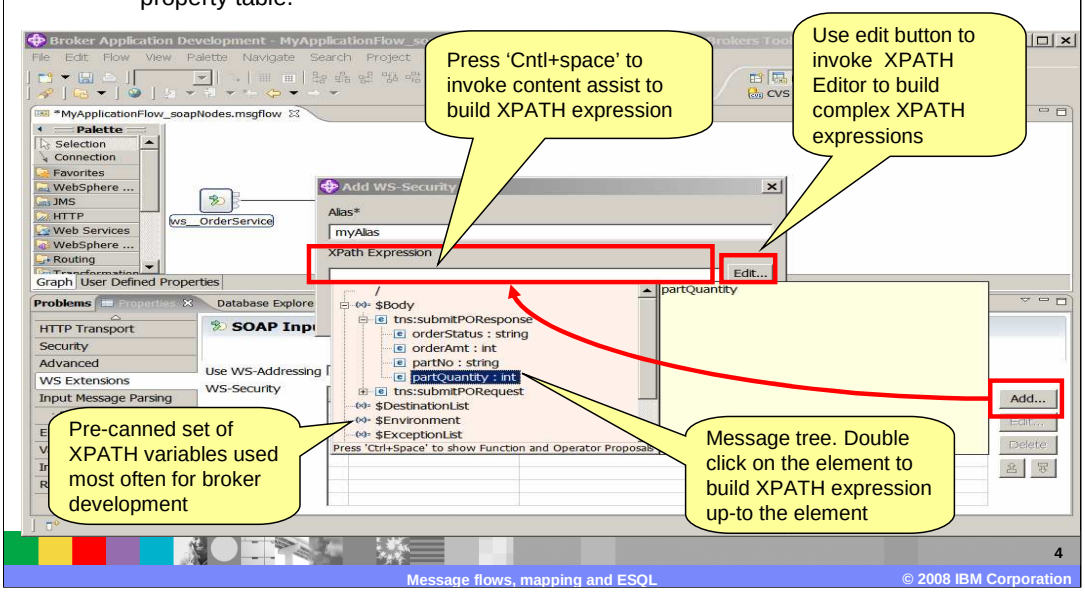
Another new feature is support for complex properties on nodes. A complex property is a property to which you can assign multiple values. Complex properties are displayed in a table in the Properties view, where you can add, edit, and delete values, and change the order of the values in the table. Node designers can group repeated scalar properties and users can edit and delete the individual rows. Various nodes in Version 6.1 take advantage of this new feature.

Complex properties are supported in the “User Defined Node” editor, allowing the node designer to add complex properties while designing their nodes.

Node designers can also add their custom validator class to perform semantic validation on the node, during flow builds.

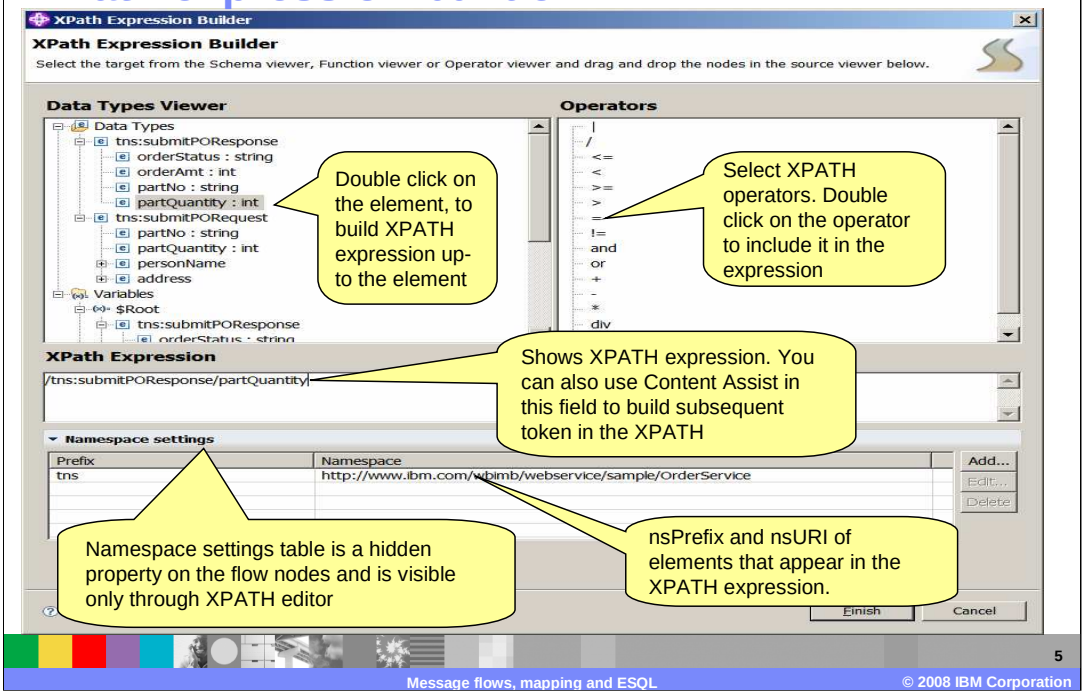
XPATH expressions on node properties

- Integrated XPath support for scalar and complex properties
  - Ability to invoke XPath builder for the scalar property or a column in the complex property table.



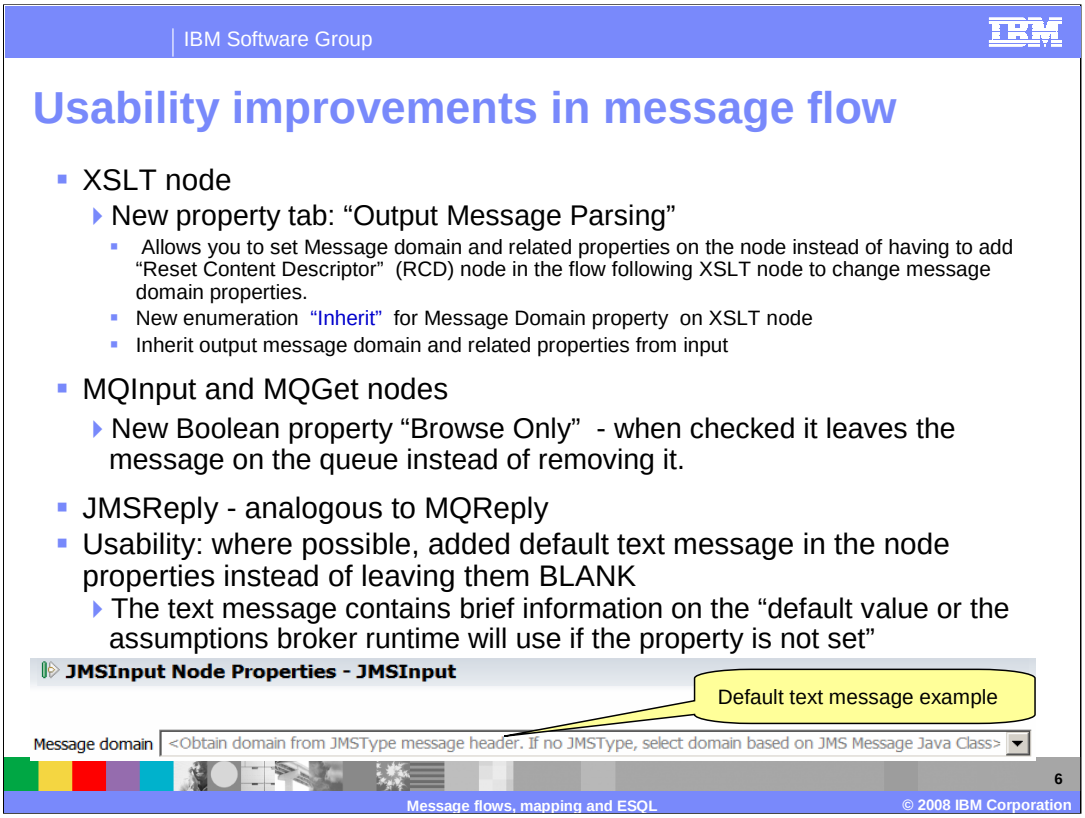
Also included is a comprehensive XPath support of node properties. Integrated XPath support for scalar and complex properties gives you the ability to invoke the XPath builder using Content Assist for the scalar property or the column in the complex property. There is a set of pre-loaded set of XPath variables that are commonly used in Broker Development, such as $Body, $Environment and so on. Within the message tree, double clicking on the element allows you to build an XPath expression up to the element.

XPath expression builder



Clicking on the Edit button will invoke the XPath editor. As mentioned on the previous slide, double clicking on an element in the tree will build an XPath expression up to that element. You can also use Content Assist to do the same thing in the XPath expression pane, which also shows the XPath expression that is being built. Also included are a list of XPath operators, double clicking on an operator will add it into the expression.

The namespace settings table is a hidden property on flow nodes, but is visible in the XPath editor. You can see the namespace prefix and namespace URI of the elements that appear in the XPath expression.



This slide discusses several usability improvements in message flow development.

Firstly, XSL transform node has changes. This node was formerly known as the XML transformation node. A new Output Message Parsing tab allows you to set message properties on the node, rather than having to add a reset content descriptor node in the flow after the XSLT node. An additional message domain property option is “Inherit.” This means that the output message will inherit the message domain and related properties from the input.

MQInput and MQGet nodes also now have a property which allows you to browse the message, rather than removing the message from the queue. This is covered in more detail in the session covering “Other New Nodes”.

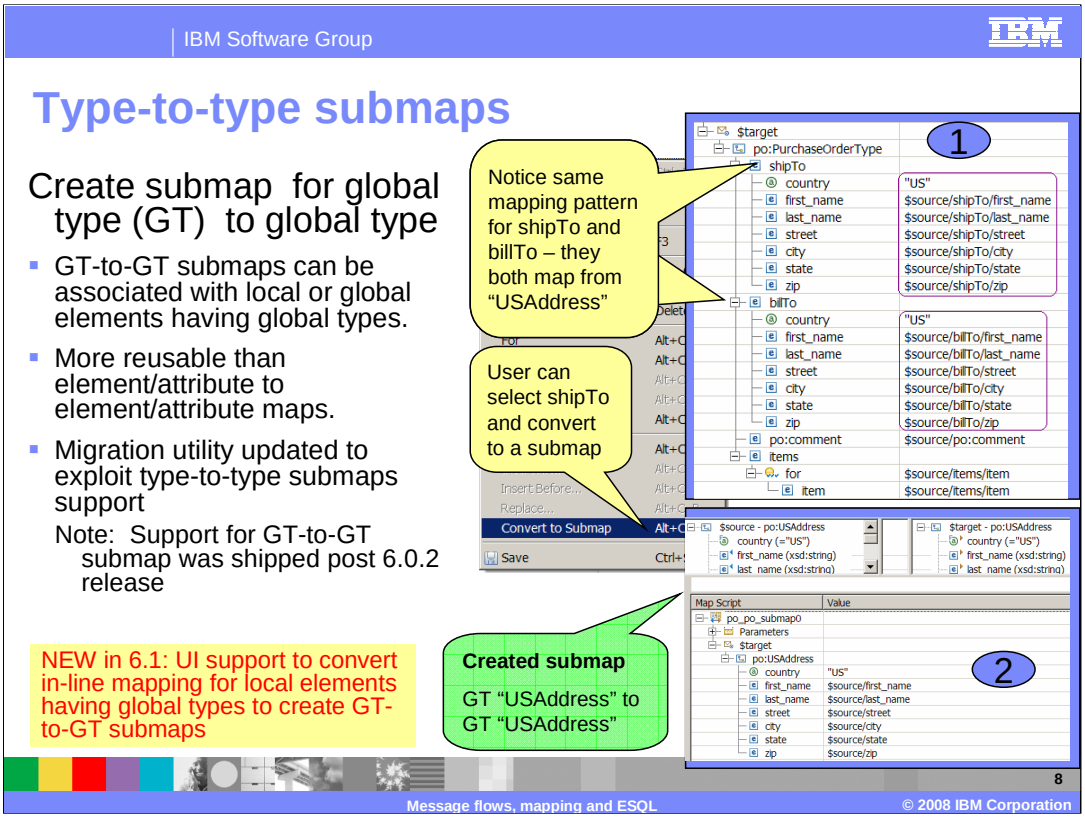
The JMSReply node is similar to the MQReply node. The JMSReply node sends JMS messages to the reply destination that is supplied in the “JMS-Reply-To” header field of the JMS message tree.

Finally, where possible, in node properties, a default text message has been added which contains brief information regarding default values and assumptions that will be used by the broker runtime if this property is not set.

Section

Mapping and ESQL

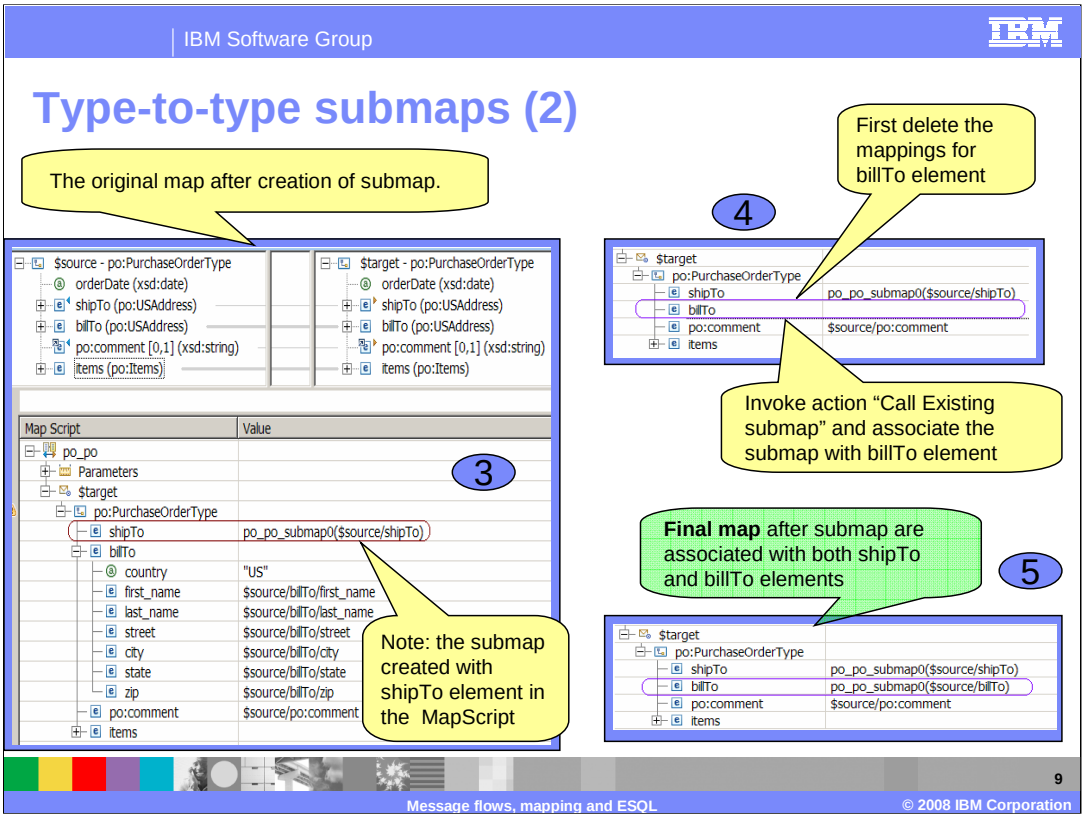
This presentation now looks at the new functions and improvements provided in the Mapping Node, and in the area of ESQL deployment.



Many enhancements have been made to mapping in Message Broker Version 6.1. The first of these is Type-to-Type submaps.

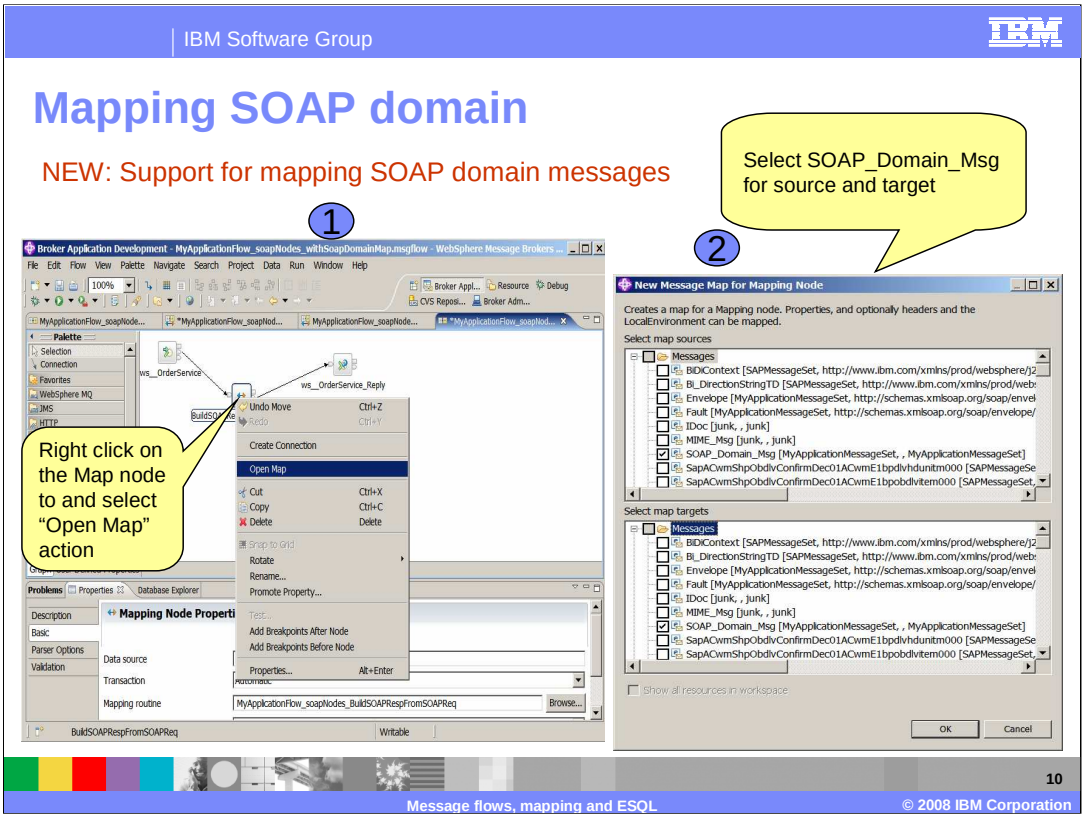
A Global Type to Global Type submap can be associated with local or global elements that have global types. It allows for more reuse than an element/attribute to element attribute map. The example here shows that both the shipTo and BillTo addresses are mapped from the same address. In this case, you can select the shipTo and convert to a submap.

The migration utility has also been updated to exploit the Type to Type submap support.



Following on from the previous slide you will see the original map after the submap has been created. In the shipTo, you will notice the submap is created with the shipTo element in the MapScript.

If you now move to the billTo type, first delete the mapping for this element. You then repeat the “Call an Existing Submap” and associate the submap with the billTo element. You can now see that in the final map, the shipTo and billTo elements are associated with the submap.



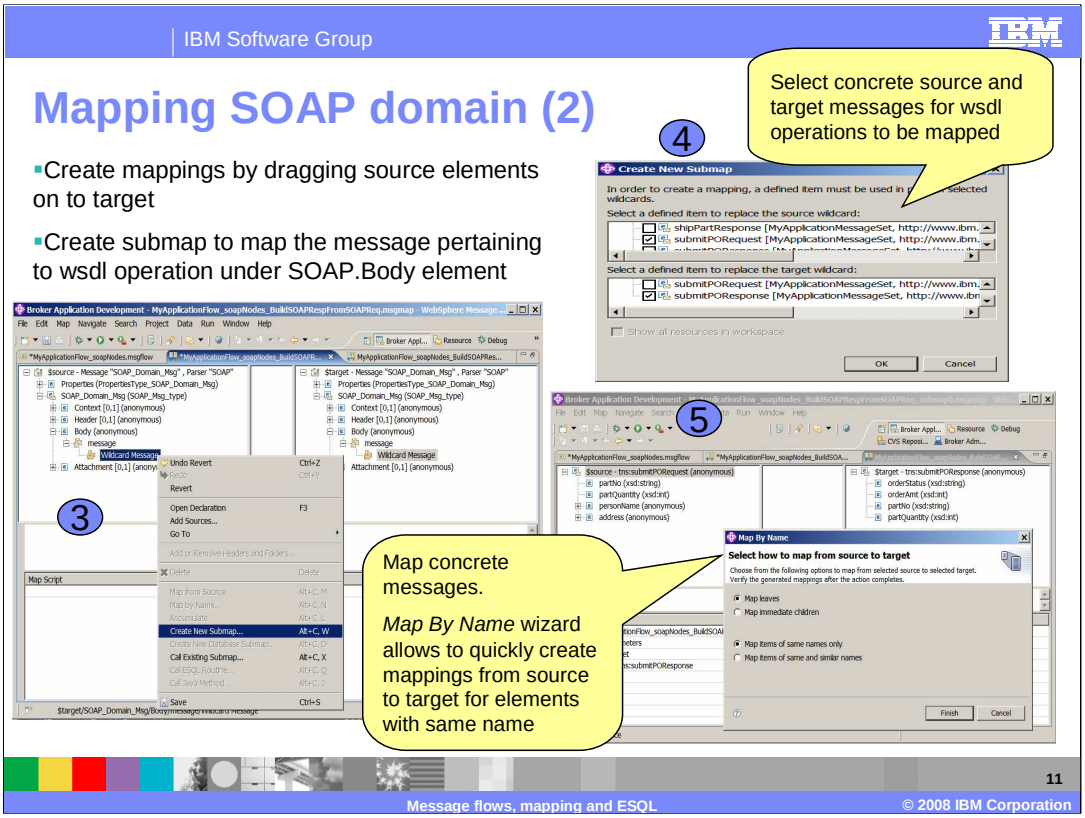
A very important addition to the mapping support is the mapping SOAP domain messages.

Where both the envelope and the messages that can appear within that envelope have to be modeled, you can use the Message Mapping editor to select from available messages at points in the model. This is done by using the Submap function.

To create the primary map, right-click the Mapping node, and select "Open Map", shown as task 1.

To map messages in the SOAP domain you open the map as normal, select the SOAP messages for source and target. In this case, select the SOAP_Domain_Msg, which comes from the SOAP parser domain. This is shown as task 2.

This function is intended for complex scenarios, where you need a detailed breakdown of the SOAP messages.



You then create mappings by dragging source elements to target elements.

The created map will contain a wildcard message. Right-click this, and select “Create New Submap”, shown as task 3 on the slide. This will open the wizard to specify the submap. In this wizard, select the Body of the SOAP message that you really want to map. In this example, the source is “submit-PO-Request”, and the target is “submit-PO-Response”. Clicking OK move to task 5, where you can choose whether to map exact names only, or to allow the mapper to detect similar names.

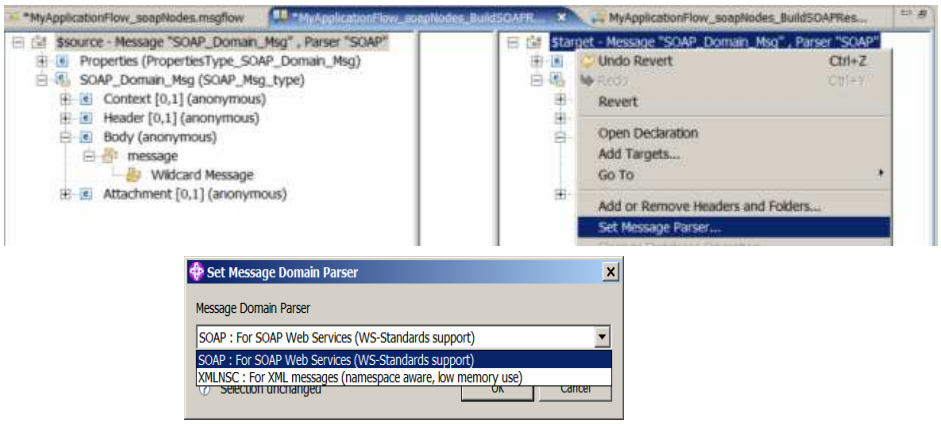
In the main map, expand the levels (both source and target) of Envelope and Body until you find the wildcard message, and select this on both the source and target sides.

Select the concrete source and target messages for WSDL operation to be mapped. The Map by Name Wizard allows you to quickly create mappings from source and target for elements with the same name.

If you do not need to have direct access to the SOAP message components, you should use the WSDL drag-and-drop mapping, which is covered in Part 3 of the Toolkit presentations.

Mapping editor enhancements

- Display domain parser for message assemblies on source and target
- On the target – “Set Message Parser” action to change the domain parser to one of the supported domains on the target message set

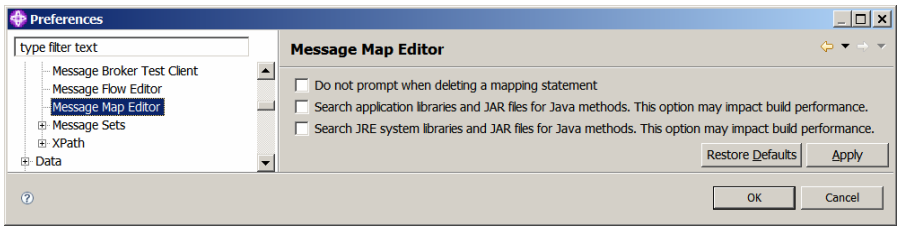


One of the additional Mapping Editor Enhancements is to display the domain name for message assembly on both the source and the target. On the target message, you can select the Set Message Parser action to change to one of the domains which is supported on the target message set.

It is possible to have more than one message domain supported for a message set. When you create and open a map for the first time, it will use the default domain that is specified on the message set definition. For example, you may need the map node to change the domain from the SOAP domain to the XMLNSC domain. This facility is only available for domains which provide the whole message.

Call Java™ methods directly from map

- Call public static Java methods directly from map instead of wrapping them under the ESQL functions
  - Right click action “Call Java Method” on source and target pane
    - Lists set of public static Java methods (from workspace projects having Java nature) with number of parameters matching the source and target selection of mapping.
    - Scope of search can be set through preference page
      - By default, search for all methods in .java source files in the Java build path of each Java project

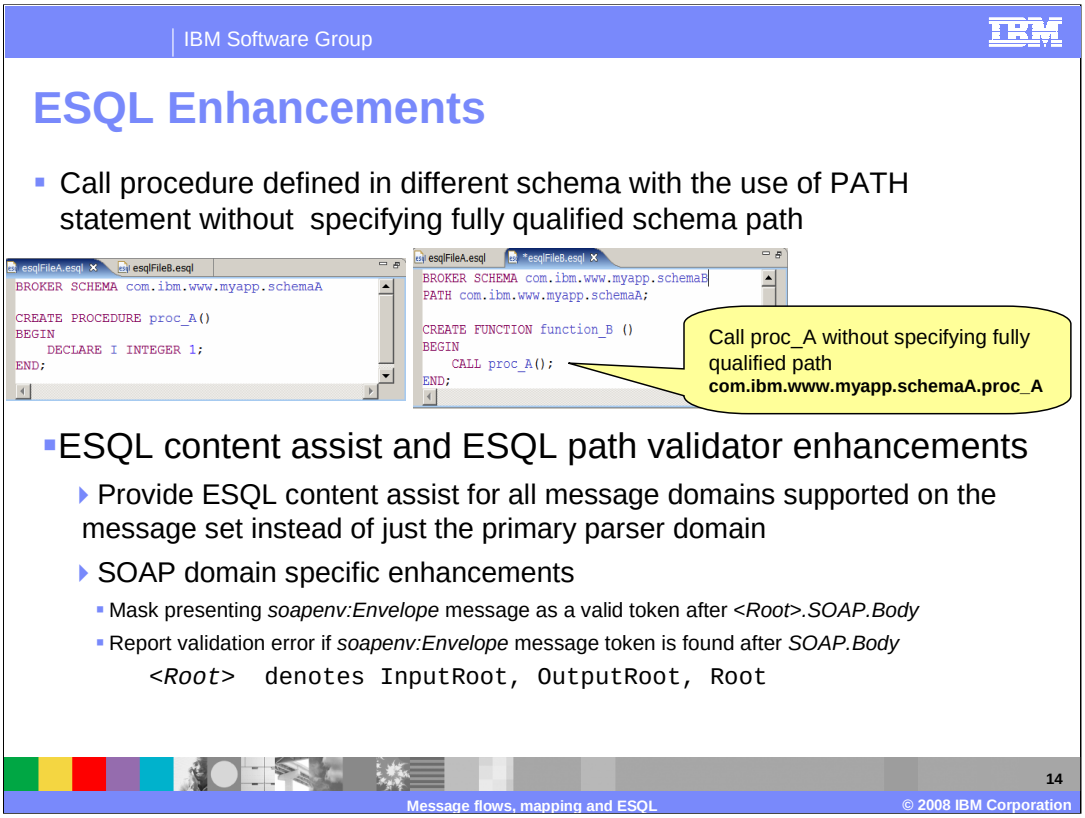


- Content assist for Java methods under new category “java:” in the edit pane
  - List all methods matching the search criteria *but this list is NOT restricted to methods with parameters matching the source and target of mapping.* You can type ahead to narrow the list as required

Version 6.1 allows you to call some Java methods directly from the map node, rather than having to wrap them under ESQL functions.

Only public static Java methods can be called. To call a Java method, select both the source and the target, select the Call Java Method action. A list of public static Java methods from workspace projects that have a Java nature will be returned. The scope of this search can be set using the preferences page.

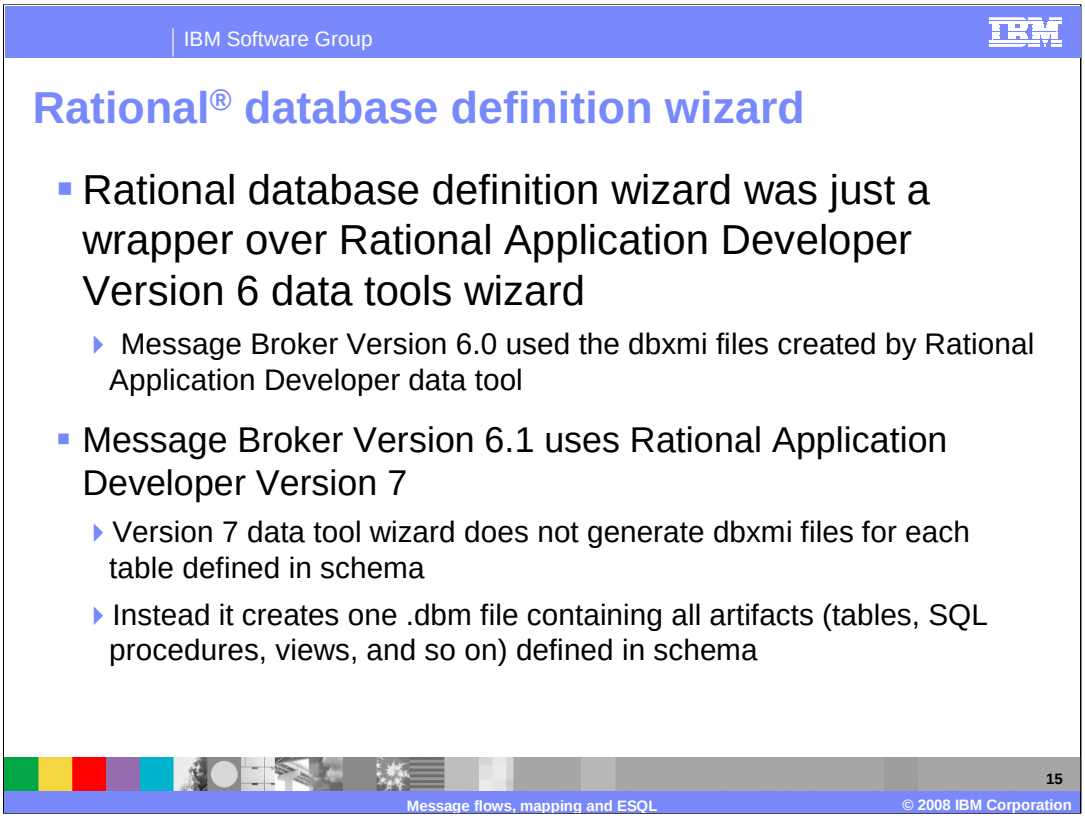
There is Content Assist for Java methods under a new category java, in the edit pane. It lists all methods matching the source criteria. However, this list is not restricted to methods with parameters matching the source and target of the mapping.



Enhancements have also be made to ESQL.

A PATH statement can be used to call procedures that are defined in different schema, without having to specify a fully qualified schema path. The PATH clause specifies a list of additional schemas to be searched when matching function and procedure calls to their implementations. The schema in which the call lies is implicitly included in the PATH. The PATH is used to resolve unqualified function and procedure names.

The ESQL Content Assist and ESQL Path Validator have also been enhanced. The Content Assist is now available for all message domains that are supported on the message set, rather than just the primary parser. Specific enhancements for the SOAP domain have also been added to the ESQL Path Validator.

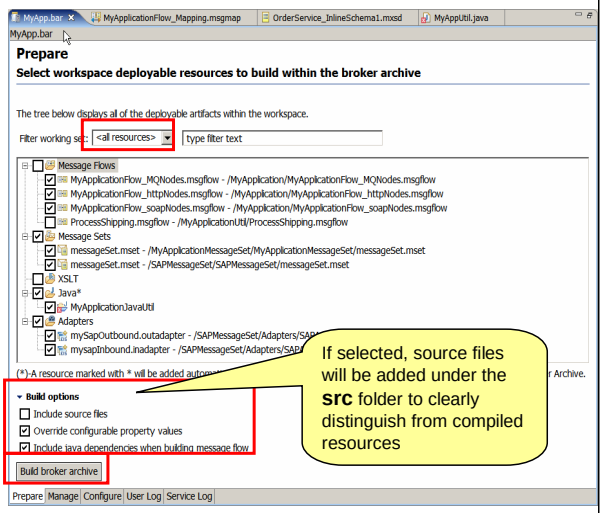


The Relational Database Definition wizard in Version 6.1 of the Message Broker Toolkit is just a wrapper over the Rational Application Developer Version 7 Data Tools Wizard. In Versions 6.0 and 6.0.2 of the Message Broker, the dbxmi files that were created from the Rational Application Developer data tool were used.

The presentation now moves on to look at improvements made to the processing of the Message Broker bar file mechanism.

Message Broker archive wizard

- Improved Broker archive wizard consistent with other wizards
- Bar files can be stored in any project (Server project removed from 6.1)
- Prepare tab replaces the 6.0.2 “Add to Broker Archive Dialog”
  - Lists categorized workspace resources under check-box tree.
  - Allows you to pick resources to be included in the build.
  - Filter resources using working sets and filter text
  - Indication of build error by tagging appropriate resource
  - Properties view shows the specific error details



The management of the Message Broker Archive File, known as a “bar” file, has been improved in version 6.1.

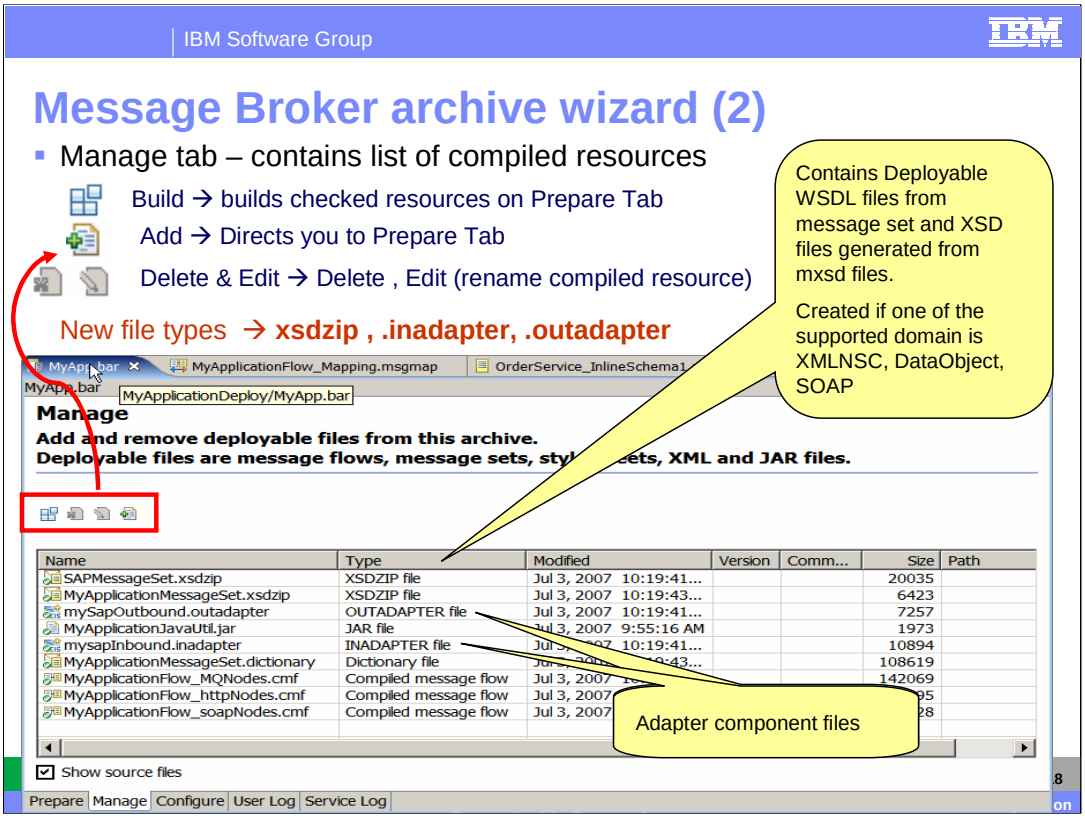
First, when you create a new bar file, the first task that you will do is to populate the bar file with artifacts from the workspace. This is done with the “prepare” tab, and the toolkit will be positioned automatically on this tab. You should pick the resources that you need, and click on “Build Broker Archive” to create the bar file.

The available resources in the workspace are shown, and are shown in various categories, for example, message sets, XSLT transformation, and adapters.

By default, the Prepare tab will show all resources available in the workspace. It is possible to limit the display to those resources that are just contained within a particular working set. If you have defined a working set containing just the resources needed for a particular application, you can select this working set on the bar file builder. The bar file builder will then show just the relevant resources.

When source files are included in the bar file, these resources are now stored in a separate folder, named “src”. This enables you to clearly distinguish between source components, and the component that is used by the broker runtime.

In earlier versions, the bar file was stored in the Server project in the workspace. In version 6.1, the bar file can now be stored in any project. This means that you can store the bar file in the same project as the application resources needed for the particular application. For example, it can be stored in the project which contains the primary message flow.



On the “Manage” tab of the bar file builder, you can see what resources are contained by the bar file, and you can perform actions on these resources.

You can rebuild the bar file, add new resources to the bar file, or delete resources from the bar file. You can also Edit the resource; this means that you can change the name of the bar file resource.

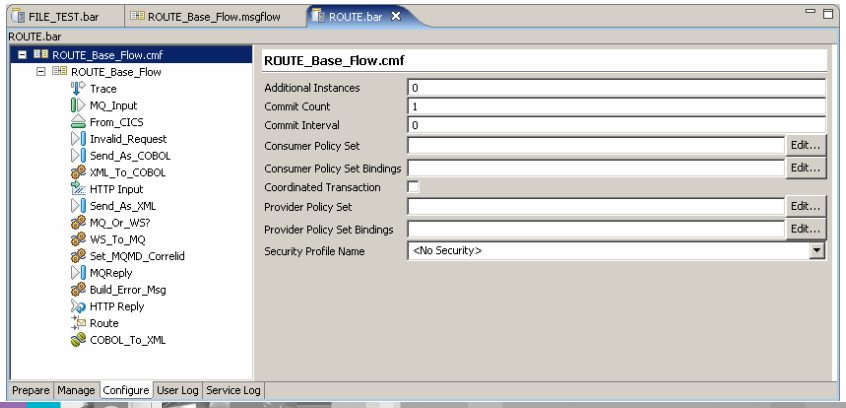
If you click “Add”, the wizard will take you immediately to the Prepare tab wizard, as shown on the previous slide.

Version 6.1 has added some additional file types to the possible resources. “xsd-zip” is used to contain the message set definitions, if the target domain is either XML-NSC, DataObject or SOAP. The DataObject domain is used for the WebSphere Adapters.

If you want to refresh the bar file, you must invoke the Build wizard to rebuild the bar file.

Message Broker archive wizard (3)

- Configure tab
  - New : Policy set properties and security profile
- *User Log* and *Service Log* largely unchanged



On the Configure tab, you specify values for promoted properties. In addition to those available in earlier releases, you now also specify the policy set properties for security, and properties needed for security profiles.

Summary

- Message flows
- Mapping and ESQL
- Bar file

In summary, this presentation has discussed new function and improvements made for message flows, and the mapping node and ESQL. Improvements to the bar file mechanism were also discussed.

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_WMB61_IEA_Toolkit2_Flows_ESQL.ppt

This module is also available in PDF format at: ../WMB61_IEA_Toolkit2_Flows_ESQL.pdf

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:

Rational WebSphere

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

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 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 (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.

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.