

# WebSphere Message Broker Version 7

## Toolkit enhancements (Part 2) Impact analysis and mapping



This is the second of two sessions describing the enhancements that have been made in the application development toolkit in WebSphere® Message Broker version 7.

## Toolkit enhancements – part 2

- Part 1: (review the first Toolkit session)
  - Application development perspective
  - New samples gallery
  - Simplified compile / test / debug
- Part 2: (this session)
  - Impact analysis
  - Mapping enhancements

The first session looked at the new application development toolkit, and the simplifications that have been made with the application development perspective, the samples gallery and the test and debug tools.

This session will describe the new impact analysis tool, and enhancements in the mapping node

## Impact analysis

- New tool to identify which other message broker artifacts are impacted if a particular artifact is changed or moved
- Example: If the name of *message* “xyz” is changed, which other artifacts referencing this message are impacted. The list can include:
  - .mxsd files (for example, complex types and groups having element reference to global element on which the message is defined)
  - Deployable WSDL files
  - message maps
  - message flows (nodes within flow referencing the message),
  - eSQL paths in ESQL modules and functions

Impact analysis is a new function that enables you to understand the impact of changing artifacts in the broker toolkit.

For example, if you change the name of a message, then this can affect a range of other artifacts that form the complete application. This can be any of those shown on this slide, including ESQL statements contained within a compute node, which reference the particular message.

## Impact analysis - operations on Message Broker artifacts

The following operations are supported:

- Rename
  - Files
    - mxsd
    - Deployable WSDL
    - Map files
    - Message flows files (including subflows)
  - Artifacts defined in mxsd file
    - Global artifacts (messages, elements, types, groups, attributes)
    - Local artifacts (elements, attributes)
  - Artifacts defined in ESQL file
    - ESQL modules
    - Schema scope ESQL routines
    - ESQL constants
- Move (within same project)
  - Map files
  - Message flows files
  - ESQL modules
  - Schema scope ESQL routines

The impact analysis functions are available on these primary artifacts. This means that you can perform this function on all the artifacts shown on this slide. Two operations are supported, namely Rename and Move.

For example, you might want to assess the impact of moving an ESQL file from one schema to another, but within the same project.

## Impact analysis – reported dependent artifacts

- Message set artifacts
  - MXSD files and the artifacts defined in the file
  - Global artifacts (messages, elements, types, groups, attributes)
  - Local artifacts (elements, attributes)
  - Deployable WSDL
- Maps and submaps
- Message flows
  - Main flows and sub flows
    - Message node properties referencing Message Set artifacts
    - Java™ classes
    - XPath
- ESQL
  - ESQL files and the artifacts defined in the file
  - ESQL modules, schema scope routines
- ESQL paths

When you invoke the impact analysis function, the tool will check the artifacts shown on this slide. If any of these artifacts contains a reference to the artifacts that is going to be changed or moved, then this is reported by the tool.

## Invoking impact analysis (1)

Pop-up menu **Impact Analysis**

- Global artifacts (from navigator)
- Local artifacts (from within the editor)

Rename Artifact dialog box:

New name: submitPORequest\_1

Update references

Analyze Impact Cancel

6 Toolkit enhancements (Part 2) - Impact analysis and mapping © 2010 IBM Corporation

The impact analysis tool can be invoked in two ways.

The first way to invoke the tool is to highlight the affected artifact in the project navigator, and right-click the item to show the pop-up menu. Select Impact Analysis from this menu, and then select the proposed action. For example, if you select Rename, then the tool will show a window similar to the screen capture shown here. Specify the new name, and click “Analyze Impact”.

## Invoking impact analysis (2)

The screenshot displays the IBM WebSphere Message Broker Application Development Toolkit interface. The main window shows a project tree on the left with 'CustomerInfo.msxd' selected. The central pane displays the 'orderStatus (xsd:string)' element details, including a table of its structure and a context menu. The 'Impact Analysis' option is highlighted in the context menu. The bottom left pane shows the 'Outline' view with the 'orderStatus' element selected.

Structure	Type	Min Occurs
CustomerInfo.msxd		
Messages		
customerResponse	CustomerResponseMsgType	1
orderStatus	xsd:string	1
partNo	xsd:string	1
partQuantity	xsd:decimal	1
orderAmt	xsd:decimal	1
customerName	{Local complex.Type}	1
customerAddress	{Local complex.Type}	1
customerEmailAddress	xsd:string	1
msg_CustomerResponseMsgType	CustomerResponseMsgType	1

7 Toolkit enhancements (Part 2) - Impact analysis and mapping © 2010 IBM Corporation

The second way to invoke the tool is to open the required artifact, and select the Impact Analysis tool from the item details.

In the example shown here, the item `orderStatus` has been selected, and Impact Analysis selected from the pop-up menu. Alternatively, the same action can be selected from the elements shown in the display shown in the Outline view, in the bottom left pane.

## Impact analysis report

**Primary artifact**

**Dependent artifacts**

Copy the analysis report to clip board and paste to spreadsheet or document for printing or distribution to developers

Click here to view a list of artifact types that might contain impacted references

Copy Analysis To Clipboard Close

Resource	Project	Path	Description
OrderService_InlineSchema1.mxsd	SOAPMessageSet	/SOAPMessageSet/ SOAPMessageSet/com/acmeorders/www/orderservice	Renaming XSD element from 'submitPC
BuildPOResponseRejected_submap0.msgmap	SOAP	/SOAP	Element '{http://www.acme
BuildPOResponseShipped_submap0.msgmap	SOAP	/SOAP	Element '{http://www.acme
OrderService_InlineSchema1.mxsd	SOAPMessageSet	/SOAPMessageSet/com/acmeorders/www/orderservice	Change message definition i
OrderService.wsdl	SOAPMessageSet	/SOAPMessageSet/com/acmeorders/www/orderservice	Change XSD reference from

This screen capture shows the output of a rename of message definition.

The proposed operation is shown on the top line of this window. In this example, it is a rename operation.

The primary artifact is shown in the next section. This shows the current name of the primary artifact, where it is located, and the outcome of the proposed rename action. This is out of view on this screen capture.

The lower part of the display shows the secondary artifacts, which are those that contain a direct reference to the primary artifact. In this example, there are four items, including two mapping nodes, an “mxsd” file, and a “wsdl” file.

If you want to copy this report into a spreadsheet or document, you can copy directly to the clipboard.



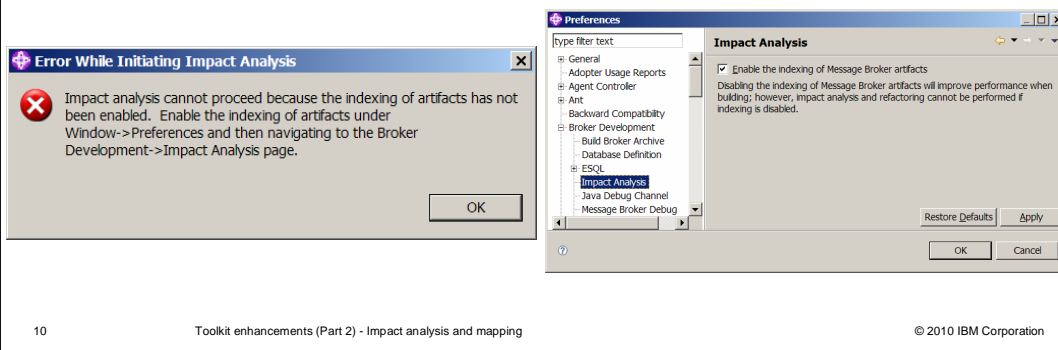
## Impact analysis report imported into a spreadsheet

Operation	Resource	Project	Path	Description	Details
Rename 'submitPO Request' to 'submitPO Request_1'	OrderService_Inlin eSchema1.mxsd	PurchaseOrderMessageSet	/PurchaseOrderMessageSet/ com/ibm/www/wbimb/webser vice/sample/orderservice	Renaming XSD element from 'submitPO Request' to 'submitPO Request_1'	Renaming XSD element from 'submitPO Request' to 'submitPO Request_1'
<b>Resources and artifacts that contain impacted references:</b>					
Operation	Resource	Project	Path	Description	Details
Rename 'submitPO Request' to 'submitPO Request_1'	OrderService_Inlin eSchema1.mxsd	PurchaseOrderMessageSet	/PurchaseOrderMessageSet/ PurchaseOrderMessageSet/ com/ibm/www/wbimb/webser vice/sample/orderservice	Change message definition name from 'submitPO Request' to 'submitPO Request_1'	Change message definition name from 'submitPO Request' to 'submitPO Request_1'
Rename 'submitPO Request' to 'submitPO Request_1'	OrderService.wsdl	PurchaseOrderMessageSet	/PurchaseOrderMessageSet/ PurchaseOrderMessageSet/ com/ibm/www/wbimb/webser vice/sample/orderservice	Change XSD reference from '(http://www.ibm.com/wbimb/web service/sample/OrderService)submitPO Request' to '(http://www.ibm.com/wbimb/web service/sample/OrderService)submitPO Request_1'	Change the referenced element from '(http://www.ibm.com/wbimb/w ebservice/sample/OrderService)submitPO Request' to '(http://www.ibm.com/wbimb/w ebservice/sample/OrderService)submitPO Request_1'

This screen capture shows how the impact analysis report will look when it has been imported into a spreadsheet.

## Impact analysis - enable/disable preference (1)

- Impact analysis requires creation of an index for all artifacts in workspace
- If you have a large number of projects in workspace or have large mxsd, map, ESQL or flow files, it can:
  - Affect build performance (because indexing is done during the build phase)
  - Increase memory foot print and disk space
- You can enable / disable Impact analysis support
  - Select Window, Preferences, Broker Development, Impact Analysis



The impact analysis function requires the creation and maintenance of an index for all artifacts contained in the toolkit workspace. The index is disabled initially. If you invoke the impact analysis function when the index is disabled, you will see an error window similar to the screen capture shown here on the left. To enable impact analysis, use the preferences option under the Window option, and set the check box as shown.

If there are a large number of workspace items, toolkit performance can be affected. To improve performance, you can temporarily disable the indexing tool.

## ***Mapping node enhancements***

The next topic that will be discussed is the enhancements made to the mapping node in version 7.

## Relational database stored procedures

- Map RDB stored procedure (written in any of the supported languages on the database)
- Provide input for IN parameter of stored procedure
- Map stored procedure output to target message
  - OUT parameters
  - INOUT parameters
  - Return value
  - Results sets
- Supported databases: DB2® and Oracle

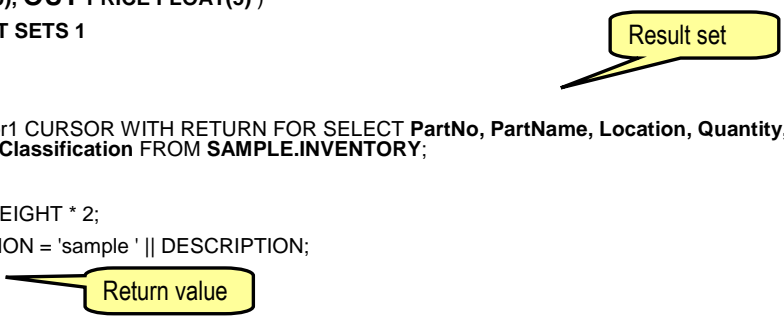
The primary enhancement for mapping nodes in Message Broker version 7 is the support for database stored procedures.

Support for stored procedures is provided in the same way as a table in a relational database. The fields in the input message are mapped to the input parameters of the stored procedure. The output of the stored procedure, including the return value and result set, are mapped to the target message. This is done using the standard tools of the mapping node.

Message Broker version 7 supports DB2 and Oracle stored procedures. However, it is important to note that stored procedures contain several different styles of deployment and invocation. These differences are described later in this session.

## Example : DB2 stored procedure definition

```
CREATE PROCEDURE SAMPLE.SQLINVENTORY ( IN WEIGHT BIGINT, INOUT DESCRIPTION
CHARACTER(50), OUT PRICE FLOAT(5) )
DYNAMIC RESULT SETS 1
LANGUAGE SQL
BEGIN
  DECLARE cursor1 CURSOR WITH RETURN FOR SELECT PartNo, PartName, Location, Quantity,
LastTransDate, Classification FROM SAMPLE.INVENTORY;
  OPEN cursor1;
  SET PRICE = WEIGHT * 2;
  SET DESCRIPTION = 'sample ' || DESCRIPTION;
  RETURN 2;
END
```

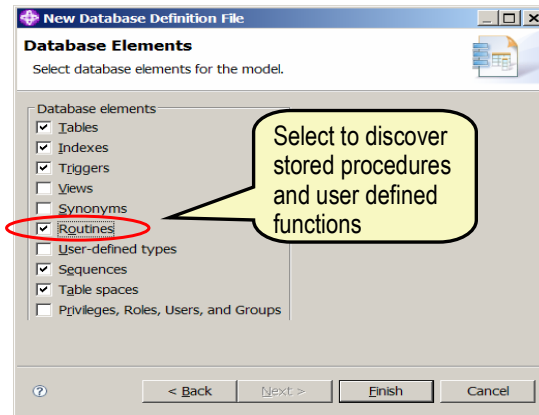


The diagram includes two callout boxes. One is a yellow speech bubble pointing to the 'DYNAMIC RESULT SETS 1' line, containing the text 'Result set'. The other is a yellow speech bubble pointing to the 'RETURN 2;' line, containing the text 'Return value'.

This slide shows an example of a DB2 stored procedure. It shows the IN parameter, the INOUT parameter, and the OUT parameter. Its purpose is to multiply the IN parameter by 2, and pass this back as the OUT parameter. The INOUT parameter is updated by concatenating it with the word “sample “, and passed back to the calling application.

## Database definition wizard

- Discover definition of stored procedure from database using database definition wizard
- On database elements page, select “**routines**” check box to discover both stored procedures and user-defined functions



14

Toolkit enhancements (Part 2) - Impact analysis and mapping

© 2010 IBM Corporation

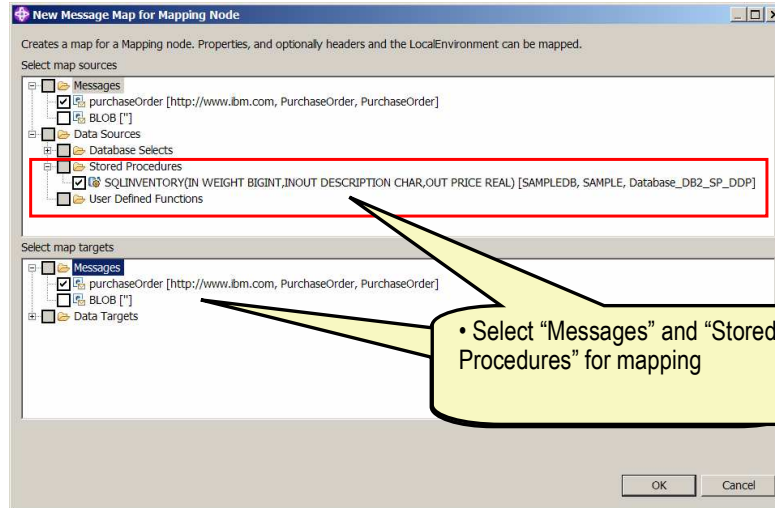
Before you can do any form of database manipulation, including any work with stored procedures, you must obtain a definition of the required database. This includes table definitions, column definition and information about the specific stored procedures that are to be used.

This is done by using the database definition wizard in the Message Broker toolkit. As shown on this screen capture, you will see that this wizard has now been enhanced to include a check box to retrieve information about stored procedures.

Using this check box will tell the wizard to retrieve all information about stored procedures and user-defined functions. The wizard will place this information in the generated .dbm file in the toolkit workspace.

## Message map wizard

- Lists all **stored procedures** and **user defined functions** from **.dbm** files in the working set.



15

Toolkit enhancements (Part 2) - Impact analysis and mapping

© 2010 IBM Corporation

Once the database definition has been created in the workspace, this can now be used in the mapping node wizard. The mapping node wizard is invoked by either double-clicking the mapping node in a flow, or by using “File, New, Mapping Node” in the toolkit.

This has also been enhanced in version 7 to use stored procedures. The example shown here shows the stored procedures category, when you expand the Data Sources in the input messages part of the wizard. The wizard will show all stored procedures that are available as .dbm files within your toolkit workspace.

Select the required messages for input and output and the required stored procedure from the Data Sources, and click OK. This will take you to the main mapping window.

## Message map editor – add results set

- Stored procedures have prefix **\$db:proc**
- User defined functions have prefix **\$db:func**, behaves same as \$db:proc

db2SP.msgflow db2SP\_Mapping.msgmap

\$source - Message "po:purchaseOrder"

- Properties (PropertiesType\_po:purchaseOrder)
- po:purchaseOrder (po:PurchaseOrderType)
- \$db:proc - SAMPLEDB.SAMPLE.SQLINVENTORY**
  - WEIGHT (IN BIGINT)
  - DESCRIPTION (INOUT CHAR)
  - PRICE (OUT REAL)
  - ResultSet1 [0,unbounded]

Open Declaration F3

Show Substitutions...

Add Sources...

Go To

Add or Remove Headers and Folders...

**Toggle Add/Remove Stored Procedure Return Value**

Add or Remove Result Set Columns

Value

\$source/Properties

• Stored procedure *in, out, inout* parameters are listed here for mapping

• Do not have definition of *Return value and set of columns returned in the result set*

• Add *return value* from stored procedure

16 Toolkit enhancements (Part 2) - Impact analysis and mapping © 2010 IBM Corporation

The stored procedure that was selected in the wizard will now appear as an entry on the left side of the mapping window, along with the source message. This will include information about the IN, INOUT and OUT parameters. The result set is added as a repeating element. The stored procedure is indicated with the prefix \$db:proc.

A mechanism is now required to be able to map input and output data to these stored procedure fields. This is done by right-clicking on the \$db:proc item, which will open the dialogue menu as shown in this example. Select the action named "Toggle Add/Remove Stored Procedure Value." An element called ReturnValue of type INTEGER is added to the Source pane.



## Message map editor – add return value

- Stored procedure has prefix **\$db:proc**
- User defined function has prefix **\$db:func**, behaves same as \$db:proc

• Stored procedure in, out, inout, parameters are listed here for mapping

• Do not have definition of Return value and set of columns returned in the result set

• To add return value from stored procedure, right click **\$db:proc**

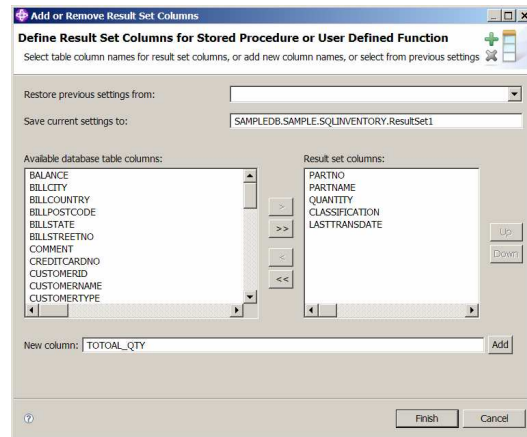
• To specify set of columns in the result set, right click **Result Set**

17 Toolkit enhancements (Part 2) - Impact analysis and mapping © 2010 IBM Corporation

You should also define the columns for the Result Set; to do this, right-click the Result Set, and select “Add or Remove Result Set Columns” in the pop-up menu. This will open a new window, where you will define the columns for the Results Set.

## Add or remove result set columns wizard

- All database table columns from the database definition file are listed
- You can add unlisted column like TOTAL\_QTY (created as a result of SUM(QUANTITY) AS TOTAL\_QTY) in the *New column* field
- Selected result set columns is saved in current workspace for reuse



When you select this option, the window shown in this screen capture is opened. All the columns that are available in the .dbm file are listed here. These were discovered by the database definition wizard earlier. Using your knowledge of the stored procedure, you can now select the particular columns that you require as part of the result set. You can also add additional fields, based on the original fields shown by the stored procedure. For example, shown in this screen capture, TOTAL\_QTY is a new value based on the value of QUANTITY.

## Map stored procedure parameters, result set

The screenshot displays the mapping editor interface with three main panes:

- Top Left Pane (\$source):** Shows the source message structure. A red box highlights the `ResultSet1` field, and a green box highlights the `Returnvalue` field.
- Top Right Pane (\$target):** Shows the target message structure with fields like `partNum`, `productName`, `quantity`, `USPrice`, and `shipDate`.
- Bottom Pane (\$db:proc):** Shows the stored procedure call `$db:proc - SAMPLEDB.SAMPLE.SQLINVENTORY` with its parameters: `WEIGHT (IN BIGINT)`, `DESCRIPTION (INOUT CHAR)`, `PRICE (OUT REAL)`, and `po:comment`. A red box highlights the `po:comment` parameter.

Below the procedure call is a **Map Script** table:

Map Script	Value
<code>\$db:proc</code>	<code>\$db:proc.RESULTSET1</code>
<code>partNum</code>	<code>\$db:proc.RESULTSET1.PARTNO</code>
<code>productName</code>	<code>\$db:proc.RESULTSET1.PARTNAME</code>
<code>quantity</code>	<code>\$db:proc.RESULTSET1.QUANTITY</code>
<code>USPrice</code>	<code>\$db:proc.PRICE</code>
<code>po:comment</code>	<code>\$db:proc.RESULTSET1.CLASSIFICATION</code>
<code>shipDate</code>	<code>\$db:proc.RESULTSET1.LASTTRANSDATE</code>

Annotations in the image include:

- Return value:** A blue arrow pointing from the `Returnvalue` field in the source to the `ResultSet1` field in the target.
- Map Result Set:** A red arrow pointing from the `ResultSet1` field in the source to the `ResultSet1` field in the target.
- Stored Proc Parameters:** A green arrow pointing from the `Returnvalue` field in the source to the `po:comment` parameter in the procedure call.

A yellow callout box on the right contains the text: "•Map stored procedure parameters, result set, return value like any other Database select or message field".

When you have prepared your results set and returned values, you can proceed to the mapping editor. Since the return value and results set field were populated in the previous actions, you can now see these in the “source” definition in the top left pane. From now on, the mapping is performed in exactly the same way as a database select mapping. Fields can be mapped by dropping the source field onto the target fields, or by selecting the source field, and right-clicking the target field. This can be done with the stored procedure parameters, and the results set values.

## Support for special characters in database object names

- Most databases allow special characters like '#', '\$'
- Additional character enclosed in double quotation marks are also permitted
- Database objects affected
  - Database tables
  - Table columns
  - Database views
  - Stored procedure name and stored procedure parameters
- Not supported by mapping node, because of the use of the underlying XPath tools

The final enhancement provided in the mapping node in version 7 is the support for special characters in the database object names. In earlier versions of Message Broker, although it is possible to create databases and tables with special characters, this was not supported by Message Broker. This was because the mapping node uses the XPath tools to parse the database path, which do not support these special characters.

## Support for special characters in DB object names (2)

- Solution: reference database path having non-conforming XPATH object names through msgmap:db-path function
- Used when database path does not conform to the XML NCName format
- Two signatures:
  - msgmap:db-path(*databasePath*)
  - msgmap:db-path(*databasePath*, *delimiter*)
    - *databasePath* is a reference to either \$db:select, \$db:insert, \$db:update, \$db:delete, \$db:proc, or \$db:func
    - *delimiter* is a string literal used between the segments in the database path. *delimiter* has a default value of '.'
- When a source object is dragged onto a target object, mapping node automatically adds this function if source contains non-conforming database path

This has been addressed in Message Broker version 7 by introducing a new mapping function called db-path. This is used to resolve the real name of the database object.

If you want to specify your own delimiter between database objects segments, you can do this using the signature, as shown here.

If you already have a dbm file which represents a database which has these special characters, the mapping node will detect this, and create the appropriate db-path function.

## Support for special characters in database object names (3)

- Examples
- **msgmap:db-path("\$db:select.DB#1.SCH\$#1.CustInfo.Name")**  
In this case, the default value, '.', is used as the delimiter
- **msgmap:db-path("\$db:select/DB#1/SCH\$#1/Cust.Info/Name", "/")**  
In this case, '/' is used as the delimiter because the database table name includes the default value

This slide shows two examples of the db-path function for special characters. In both examples, the database object names contain the character "#", and so the "db-path" function has been introduced.

In the first example, the delimiter is a ".", so has not been over-ridden.

In the second example, the delimiter has been changed to a "/", because the table name, "Cust.Info", contains the default value.

---

## Summary

- Impact analysis
- Mapping node enhancements

In summary, this session has covered the impact analysis and mapping node enhancements in Message Broker version 7.



## 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\\_WMB7\\_ApplicationDevelopment\\_Toolkit\\_2.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WMB7_ApplicationDevelopment_Toolkit_2.ppt)

This module is also available in PDF format at: [./WMB7\\_ApplicationDevelopment\\_Toolkit\\_2.pdf](http://WMB7_ApplicationDevelopment_Toolkit_2.pdf)

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





## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DB2, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. in the United States, other countries, or both.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.