



IBM Software Group

# WebSphere® Message Broker Version 6

## *Deployable XSLT style sheets*



@business on demand.

© 2006 IBM Corporation  
Updated 15 December 2006

This session looks at the new support for Deployable XML Transformation Style Sheets introduced in Message Broker Version 6.

## Agenda

- Overview
- Examples
- Migration
- Summary



This session gives a brief overview of the XSLT function with specific examples; and it provides guidance migrating XML Transformation from earlier versions of Message Broker.

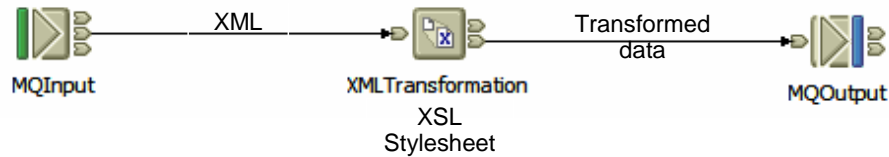
## Section

# *Overview*

This section gives a brief overview of the XSLT function.

## XSLT recap

- Transform XML data using an XSL stylesheet



.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<message>Transformed using
XSL!</message>
```

+

.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" encoding="UTF-8"/>
<!-- match the contents of the message tag and copy the text out -
->
<xsl:template match="/">
<xsl:value-of select="message"/>
</xsl:template>
</xsl:stylesheet>
```

=

```
Transformed using XSL!
```

Support for XSLT transformation was available in Version 5 of the product, and enables data transformation to be accomplished using a standard approach and tools. The XMLTransformation node is included in a message flow in the normal way, and the transformation is applied to the data in the message tree. This is usually XML data, although XSLT can be applied to data in other formats.

## Changes for Version 6

- **Deployable style sheets**
  - ▶ XSLT style sheets now included in a bar file and deployed directly to a broker.
- **Simplifies user's deployment process**
  - ▶ No need to hand-deploy style sheets
- **Simplifies maintenance**
  - ▶ Transactional deploy, update, deletion
  - ▶ Provides possible association of style sheets with message flows through bar files



When using the XMLTransformation node in Message Broker Version 5, it is necessary to manually copy the style sheets to the target deployment broker. Then, when the message flow node invokes this function, it loads the XSLT file directly from the file system on the broker. The deployment process was therefore managed outside of the normal Broker process.

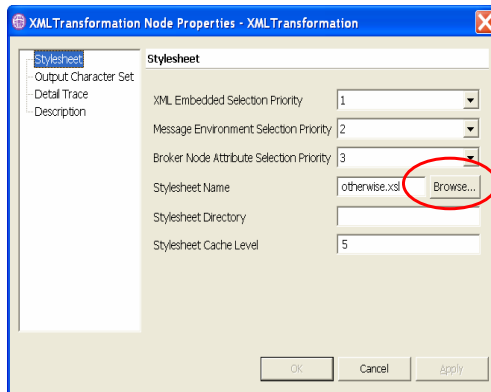
In Message Broker Version 6 the Style Sheet is now included in the deployment broker archive file, which is deployed using the normal broker deployment process. This brings benefits in the maintenance and deployment process. For example, if you want to modify your style sheet, all you need to do is to make the modification in your Broker workspace, and then redeploy it through the bar-file mechanism.

## Section

# *Examples*

This section gives specific examples.

## Details: Selecting style sheets to be deployed

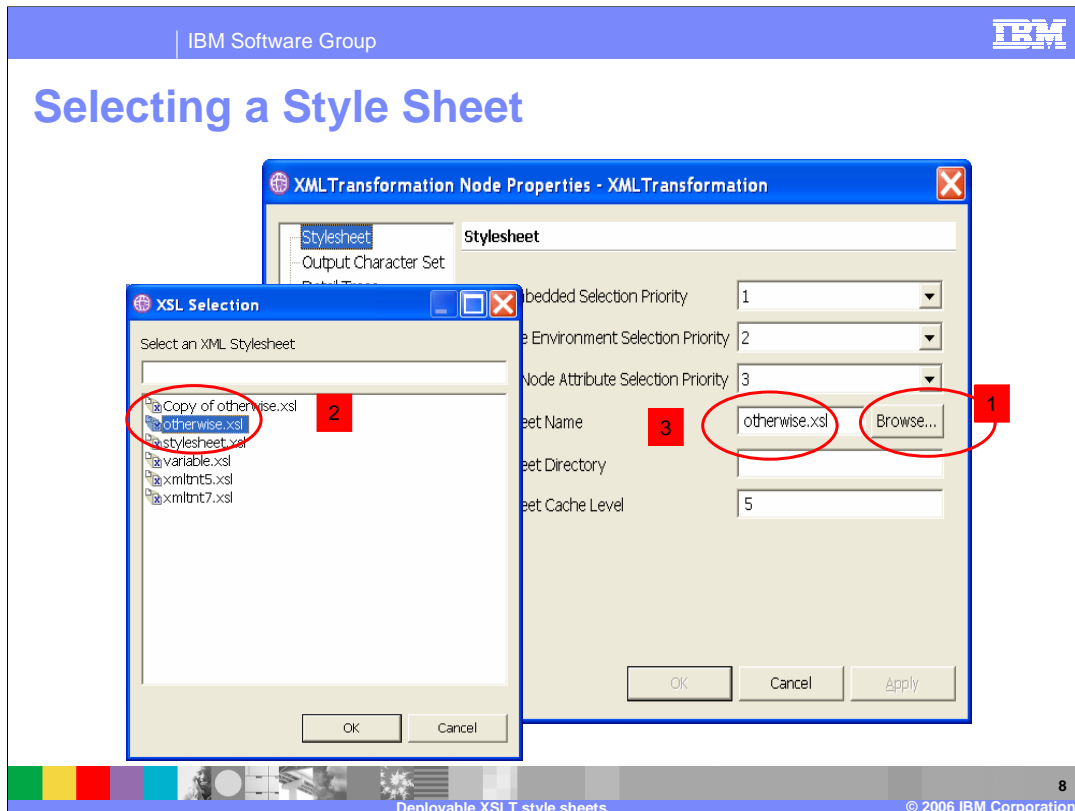


- Style sheets to be deployed must be imported into a project first.
- Style sheet selected by “Browse...” will be automatically pulled into a bar with the XMLTransformation node.

This screen capture shows the properties of the XMLTransformation node.

Prior to referencing a style sheet in this node, the style sheet must first be imported into the user’s Broker workspace. Note that the Message Broker Toolkit itself does not provide facilities for the creation of style sheets, so these must be created elsewhere.

Once the style sheet has been imported into the user’s workspace, open the Properties of the XMLTransformation node. From this, you can click on the Browse button in the properties window, to select a style sheet from your workspace. This is shown circled on this slide.

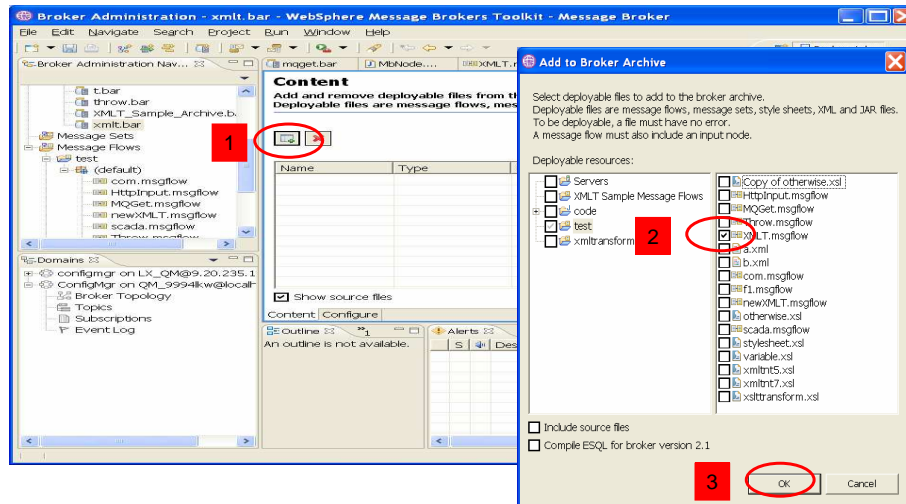


When you click on the browse button, you will see a new window identifying all the style sheets that exist in your current project. These style sheets should have an extension name of “xsl” or “xslt”, which are the two supported extensions. From this list choose the required style sheet and click ok. The stylesheet field will then contain the name of the selected style sheet.

To maintain compatibility with the equivalent function in Message Broker Version 5, you are still able to type in the name of the style sheet manually.



## Adding to a bar file (1)

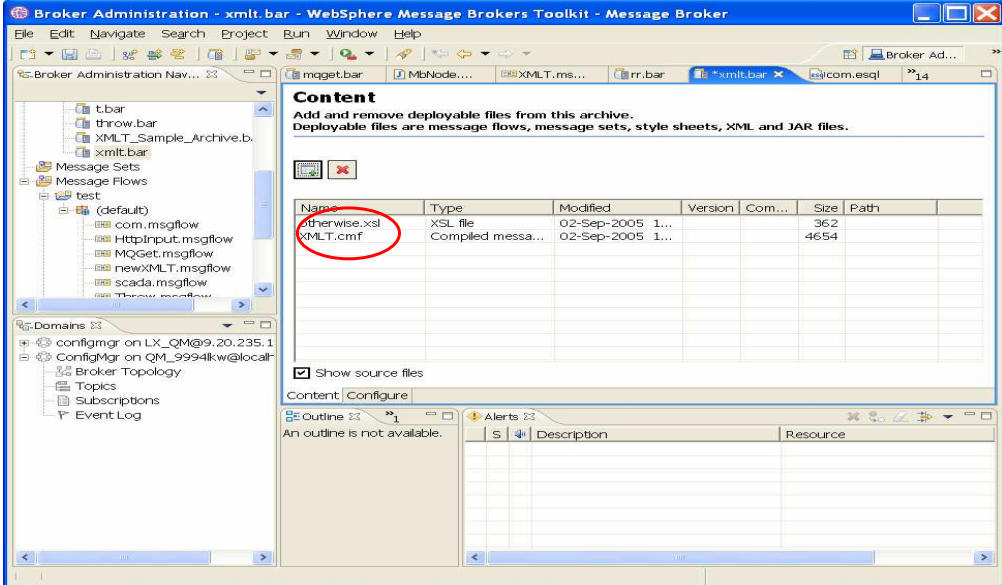


The process of adding these components to a bar file follows the normal approach as for other components.

If you identified the style sheet in the XMLTransformation node by using the Browse button, as shown on the previous slide, you don't have to identify your style sheet here. When populating the bar file, identify the message flow, as shown above, and click ok. The style sheet will be automatically added into your bar file.

Additional style sheets can also be added explicitly to the bar file, even if the style sheets will not be used by this particular node.

## Adding to a bar file (2)



Content

Add and remove deployable files from this archive.  
Deployable files are message flows, message sets, style sheets, XML and JAR files.

Name	Type	Modified	Version	Com...	Size	Path
otherwise.xml	XSL file	02-Sep-2005 1...			362	
XMLT.cmf	Compiled messa...	02-Sep-2005 1...			4654	

Show source files

Content | Configure

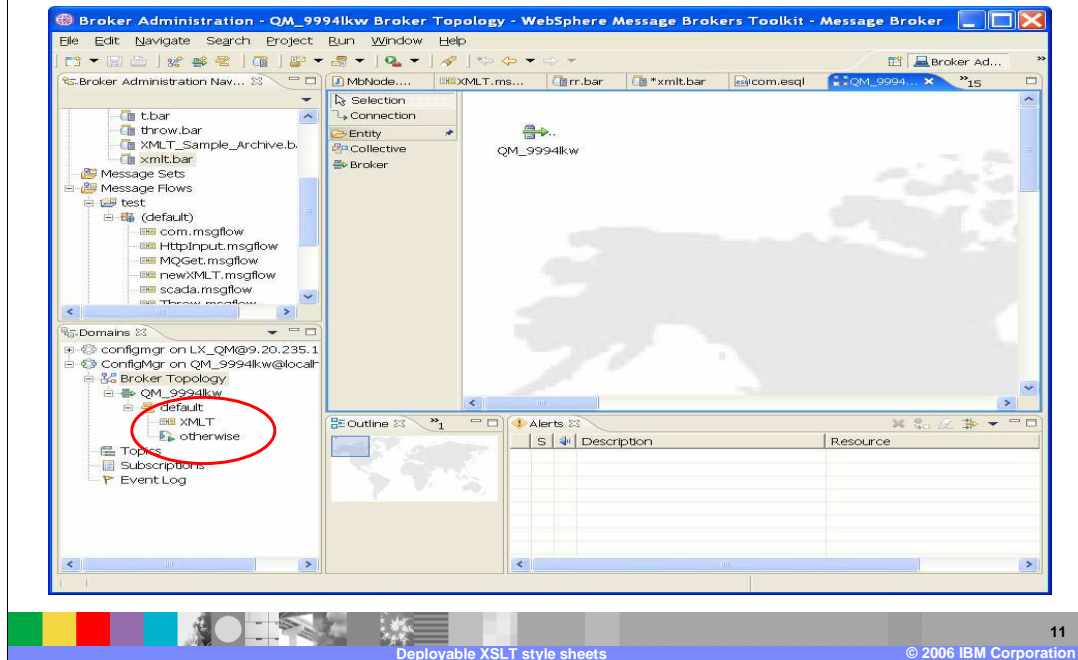
Outline: 1  
An outline is not available.

Alerts: S | Description | Resource

10  
Deployable XSLT style sheets © 2006 IBM Corporation

This screen capture shows the result of adding the style sheet transformation. The compiled message flow has the suffix “.cmf”, and in this case, it refers to a style sheet called “otherwise.xml”.

## After deployment



After deployment of the bar file, in the toolkit Broker Administration perspective, the deployed message flow and the deployed style sheet are displayed. Note the different icon for deployed style sheets.

## Section

# *Migration*

This section discusses migration.

## Migrating to V6 – Terminology Reference, 1

- Principal style sheet: The main style sheet used by an XMLTransformation node.
- Parent style sheet: A style sheet that references a style sheet/XML file.
- Child style sheet/XML file: A style sheet/XML file referenced by a style sheet.
- Style sheet/XML file reference: A naming of a style sheet/XML file to be used. It can be defined on an XMLTransformation node (using the node “Stylesheet Name” property), in a flow local environment, inside a style sheet, or in an input XML document.
- Embedded reference: An identification of a principal style sheet embedded in an input XML document,
  - `<?xml version="1.0">`  
`<?xml-stylesheet type="text/xml" href="c:\a.xml"?>`
- Embedded style sheet: A principal style sheet embedded in an input XML document
  - `<?xml-stylesheet type="text/xml" href="#style1"?>`  
`<xsl:stylesheet id="style1" version="1.0" ...>...`

This slide explains the various terms used in working with style sheets.

## Migrating to V6 – Terminology Reference, 2

- Absolute file reference: A reference expressed in an absolute form
  - “C:\a\b\c.xml”
  - “file:///C:\a\b\c.xml”.
- Relative file reference: A reference expressed in a relative form
  - “./c.xml”.
  - A child style sheet/XML file is always considered an integral part of its parent style sheet. Therefore, for a relatively referenced style sheet/xml file, the relativity is always interpreted as relative to the storage location of its parent style sheet storage location, local location or not.
- Hand-carried style sheet/XML file: A style sheet/XML file manually stored into a broker's file system
- Ascendant style sheets: All style sheets on the reference chain starting from the principal style sheet to the style sheet/XML file in question.
- Location dependent descendant style sheets/XML files: All relatively referenced child style sheets/XML files of the style sheet in question plus the location dependent descendants of these child style sheets.

This slide continues to explain the various terms used in working with style sheets.

## Migrating to and using V6 – Tasks 1

- Absolute references to files will continue to work and no migration is necessary.
- You need to migrate your style sheets if you have relatively referenced hand-carried principal style sheets. This includes hand-carried location dependent descendant style sheets/XML files of an embedded style sheet.



In V6, absolute references to files will continue to work. Migrate your style sheets if they are relatively referenced hand-carried principal style sheets, including hand-carried location dependent descendant style sheets or XML files of an embedded style sheet.

## Migrating to and using V6 – Tasks 2

- Introduce new hand-carried style sheets/XML files into your broker file systems using relatively referenced principal style sheets and location dependent descendant style sheets/XML files that are stored in <broker work path>/XSL/external
  - for example, “a/b.xsl” stored in <broker work path>/XML/external/a/b.xsl
  - for example, “c/d.xml” referenced by “a/b.xsl” as “c/d.xml” stored in <broker work path>/XML/external/a/c/d.xml.
- Use soft links for relative files which cannot be relocated.



Use the <broker work path> to prefix hand-carried new style sheets and XML files into your broker. Construct soft links for relative files which cannot be relocated.



## Migration topics

- Running flows must be restarted in order to pick up first-time deployed style sheets
- A deployed style sheet must be referenced relatively; migration therefore might be necessary.



If you have a message flow running, and this references the style sheet using a hand-coded value (perhaps because it was created in the V5 Broker), and you then subsequently decide to use a deployed style sheet contained within the deployed bar file, you must stop and restart the flow. This will enable it to pick up the deployed style sheet from the bar file.

Full path referencing of style sheets can only be done using the manual hand-coding approach, as used in Message Broker Version 5. The new capabilities of Version 6 require the use of relative referencing of the style sheet. Thus, when migrating this type of referencing from Version 5, you may need to make some changes to enable the style sheet to be located correctly.

## Section

# *Summary and references*

The following slides contain a summary and references.

## Summary

- Style sheets are deployed using normal broker deployment process in WMB V6
- Screen captures using the XMLTransformation node show how to incorporate style sheets into your flow
- Migration considerations



This presentation discussed the new capabilities of style sheet processing in WMB V6.

In Message Broker Version 6 the Style Sheet is now included in the deployment broker archive file, which is deployed using the normal broker deployment process. Examples of using the toolkit to set up style sheets in the XMLTransformation node and deployment were given.

Migration considerations were also discussed.

# Samples

The screenshot displays the 'Samples Gallery' application window. On the left, a 'Contents' pane lists various sample categories: Showcase samples, Application samples (EJB, Web, Message Brokers), and Technology samples (Java, Web Site Designer, Web, XML, Message Brokers, Aggregation, Comma Separated Value (CSV) Message Set, EDIFACT Message Set, FIX Message Set, JavaCompute Node, JMS Nodes, Message Map, SWIFT Message Set, Timeout Processing, Web Service, X12 Message Set, and XMLT). The 'XMLT' sample is selected. The main content area shows the 'XMLT Sample' details, including a description, a list of demonstrated features, and links for 'Import and deploy' and 'Read about the sample'.

**XMLT Sample**

The XMLT sample is a message flow sample application that shows how a message flow can be used to transform an XML message to another form of XML message, according to the rules provided by an XSL (eXtensible Stylesheet Language) stylesheet.

It demonstrates:

- How the XMLTransformation node is used in message flow creation
- How XSL stylesheets can be deployed to the broker rather than existing only as files on the local operating system.

Click the following links to find out more about the sample and how to get the pre-built sample running using the wizards.

[Import and deploy](#): 5 minutes

[Read about the sample](#)

You can set up the sample in one of the following ways:

- [Import and deploy the sample](#)
- [Import the sample](#)

20  
Deployable XSLT style sheets © 2006 IBM Corporation

To see how this function is used in a real example, see the XMLTransformation node sample in the Samples Gallery.

## References

- WebSphere Message Broker library:

<http://www-306.ibm.com/software/integration/wbimessagebroker/library/>

- WebSphere Message Broker Information Center:

<http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp>

## 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:

### WebSphere

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

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR 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.

© Copyright International Business Machines Corporation 2006. All rights reserved.

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

