



IBM Software Group

WebSphere® Message Broker Version 6

Configuration – Version control



@business on demand.

© 2007 IBM Corporation
Updated March 15, 2007

This presentation discusses the new capabilities for versions of artifacts in WebSphere Message Broker Version 6.

Version and keywords

- Provides details of the message flow or message set after deployment
- Can be viewed in the workbench
- Properties include
 - ▶ Deployment dates and times
 - ▶ Modification dates and times
 - ▶ User defined version and keyword information

When developing a message flow or message set, you can define the version and other key information that you want to be associated with it. After the message flow or message set has been deployed, you can view the properties in the workbench. These properties include the deployment and modification dates and times and any additional version or keyword information that you have set.

You can define information to give details of the message flow or message set that has been deployed; therefore, you can check that it is the message flow or message set that you expect.

Message flow version and keywords

- Version
 - ▶ Set in message flow Version property
 - ▶ Define default message flow version in Default version tag
- Keywords
 - ▶ Extracted from the compiled message flow (.cmf file)
 - ▶ Add keywords in only these places:
 - Label property of a Passthrough node
 - ESQL comments or string literals
 - Long Description property of the flow

3

Configuration – Version control

© 2007 IBM Corporation

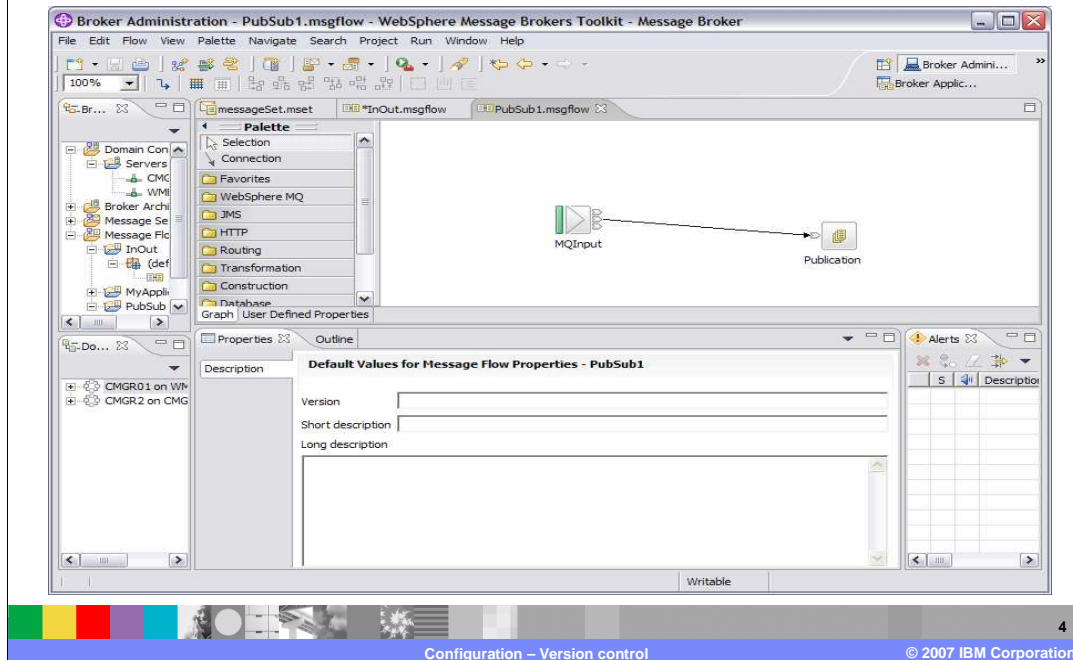
In message flows you can:

- set the version in the message flow in the Version property
- or define a default message flow version in the Default version tag of the Message Flow preferences. All new message flows that are created after this value has been set have this default applied to the Version property at the message flow level.

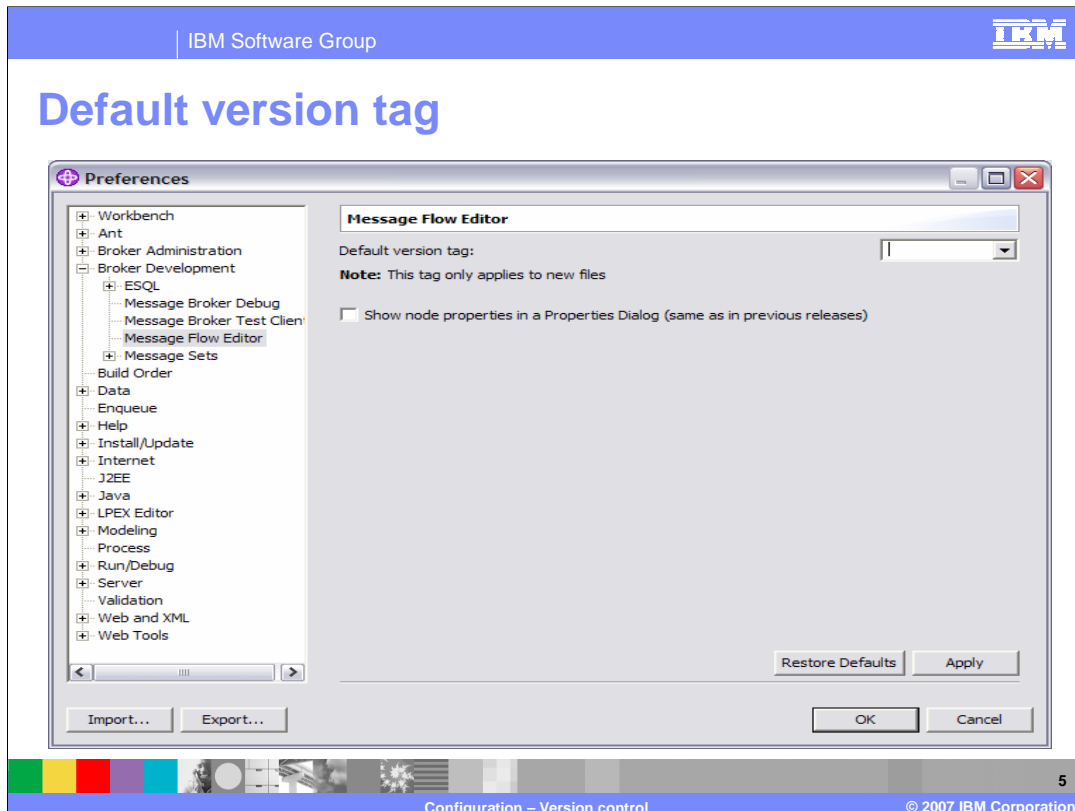
Keywords are extracted from the compiled message flow (the .cmf file) rather than the message flow source (the .msgflow file). Not all of the source properties are added to the compiled file. Therefore, add message flow keywords in only three places:

The label property of a Passthrough node, ESQL comments or string literals, and the Long Description property of the message flow.

Message flow Version property



Setting the value of the Version field for message flows can be done by setting the value in the Version property of the message flow. This is accessed using the properties of the message flow in the Toolkit, by right-clicking on the flow panel



This screen capture shows the Default version tag setting for the Message Flow Editor. You can set Message flow preferences from Window > Preferences and selecting Message Flow in the left pane. If you select Message Sets, you get a similar menu to set the Default version tag for message sets.

Message set version and keywords

- **Version**
 - ▶ Set in the message set Version property in the Documentation view of the message set
 - ▶ Define default message set version in Default version tag
- **Keywords**
 - ▶ Must be defined in the Documentation property of the message set
 - ▶ Dictionary file format is tightly controlled; be careful when using

6

Configuration – Version control

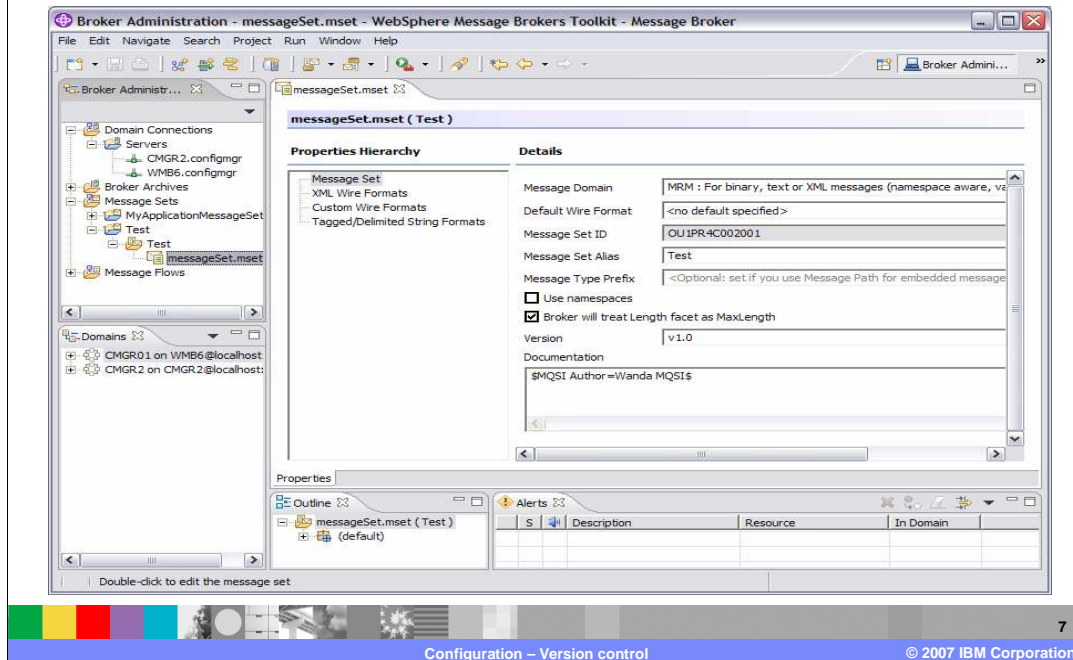
© 2007 IBM Corporation

Message Dictionaries, or Message Sets, can also be versioned. You can set the version of the message set in the Version property. This is in the Documentation view of the message set.

As with message flows, you can define a default message set version in the Default version tag of the Message Set Preferences. All new message sets that are created after this has been set have this default applied to the Version property at the message set level.

Keywords for message sets must be defined in the Documentation property of the message set. These follow certain rules to ensure that the information can be parsed.

Message set Version property



Opening a message set by double clicking and selecting "message set" allows you to set the Version property. There is a space for entering keyword information in the Documentation property as shown here.

Configuration Manager deployment

At deployment time, the Configuration Manager gathers various pieces of version information about each file being deployed inside the broker archive

- ▶ Message flows (.cmf)
- ▶ Dictionaries (.dictionary) (Message sets)
- ▶ XSL stylesheets (.xsl)
- ▶ XML (.xml)
- ▶ JavaCompute node logic (.jar)

This information is remembered by the Configuration Manager and returned to the Message Broker Toolkit and Configuration Manager Proxy applications



When a broker application artifact is deployed to the runtime environment, using the normal bar file mechanism, the Configuration Manager gathers certain pieces of information about these artifacts. The artifacts include all components that might be required for a message broker application, including the new JavaCompute node.

The information includes date and time information and version information. This information is held by the Configuration Manager, and is available to the Toolkit Administration perspective, and to applications which access the Configuration Manager through the Proxy API.

Version information gathered

The time the object was deployed

The time the object was last saved in the broker archive

The name of the broker archive file that deployed the object

A user-specified 'version' keyword

Other user-defined keywords and their values:

- ▶ "Author" -> "Matt"
- ▶ "Buddy Checker" -> "Dave"
- ▶ "Release name" -> "S000"

This slide shows the information that is collected by the Configuration Manager when the application is deployed to the broker.

Some of the information is fixed and cannot be changed by the application developer, such as the date and time of deployment, and the name of the broker archive file that was used to perform the deployment.

It is also possible to specify version information and user-defined keywords which can be used to contain any information useful to maintain a record of the deployed artifact. The example on this slide shows that the "Author" of the item was "Matt", that it was checked by "Dave", and the "Release name" is "S000".

Embedding keywords

The Configuration Manager scans through each file looking for a predefined eye catcher that describes each keyword and its value

```
$MQSI keyword = value MQSI$
```

Each time one of these strings appears in the file being deployed, it is associated with the Configuration Manager's record for that file

When browsing the list of deployed objects in the Broker Administration perspective, each keyword/value pair is displayed

At deployment time, the Configuration Manager scans all the included files, looking for a particular eye-catcher. This is indicated by the string "\$MQSI", and terminated with the string "MQSI\$". In between these eye-catchers is a keyword and an associated value for that keyword.

This information is stored in the Configuration Manager database, and is associated with the deployed artifact. These keyword-value pairs are displayed by the Broker Administration perspective.

Keyword restrictions

- Do not use the following characters within keywords because they cause unpredictable behavior:

`^$.|\<>?+*=&[]`

- These characters can be used in the values that are associated with keywords
 - For example:
 - `$MQSI RCSVER=$id$ MQSI$` is acceptable
 - `$MQSI $name=Fred MQSI$` is not acceptable
- 'VERSION' and 'BAR' keywords are reserved.
 - The toolkit automatically embeds the 'VERSION' keyword inside message flows and dictionaries if a version property is entered in the Broker Archive editor
 - The 'BAR' keyword is used by the Configuration Manager to report the name of the broker archive file that was used.

12

Configuration – Version control

© 2007 IBM Corporation

The special characters listed on the slide could cause unpredictable behavior and should not be used as part of a keyword; however, you can use these characters in the values that are associated with keywords.

Do not use 'VERSION' and 'BAR' as keywords. When you set the Version property in the toolkit message flow editor (Properties pane), a keyword called VERSION is added to the resulting cmf or dictionary file. The BAR keyword is associated with each object automatically when it is deployed and it contains the full path name of the broker archive file that deployed the object. The values of both keywords are defined programmatically in the class `com.ibm.broker.config.proxy.DeployedObject`.

Passthrough node

- Use the Passthrough node to add a label to your message flow or sub flow
- The Label property of Passthrough node provides the capability to specify the user information
- Passthrough node does not alter the message in any way

13

Configuration – Version control

© 2007 IBM Corporation

Use the Passthrough node to enable versions of a sub flow at run time. Use the Passthrough to add a label to your message flow or sub flow. By combining this label with keyword replacement from your version control system, you can identify which version of a sub flow is included in a deployed message flow. You can use this label for your own purposes. If you have included the correct version keywords in the label, you can see the value of the label:

- Stored in the broker archive (BAR) file, using the **mqsireadbar** command
- As last deployed to a particular broker, on the properties of a deployed message flow in the Message Brokers Toolkit
- In the run time, if you enable user trace for that message flow

The Passthrough node does not process the message in any way. The message that it propagates on its Out terminal is the same message that it received on its In terminal.

Adding keywords to ESQL files

- Add keywords in ESQL files in
 - ▶ Comment fields
\$MQSI compiled by = John MQSI\$
 - ▶ Static strings
Set target = '\$MQSI_target = production only MQSI\$'
 - ▶ Variable string
\$MQSI_VERSION=\$id\$MQSI\$

Keywords can be included in ESQL files in three ways:

As comment fields - Add the keyword as a comment in the ESQL file

As static strings - Include the keyword as part of a static string in the ESQL file

As variable string - Include the keyword value as a variable string in the ESQL file. In the example shown here, the message flow source is extracted from the file repository; the repository's plug-in has been configured to substitute the identifier *\$id\$* with the actual version number.

Keywords in sub flows

- You can embed keywords in each sub flow that you use in a message flow.
- When using sub flows, use a different version keyword in each sub flow.
- This allows you to see version information for each constituent part.

If the main message flow invokes several sub flows, each sub flow can have its own separate version keywords; hence each sub flow can be versioned independently. A different keyword must be used in each instance of a sub flow. This is because only the first recorded instance of each keyword within the message flow .cmf file is available to Configuration Manager Proxy applications and to the toolkit. The order that sub flows appear in the .cmf file is not guaranteed.

Adding keywords to XSL stylesheets

- XSL and XML files can be deployed in broker archives for use in the XMLTransformation node
- No version Property available since stylesheets are not a broker artifact
- Keywords can be added as an XML comment:

```
<?xml version="1.0" encoding="UTF-8">
<!-- $MQSI author = John MQSI$>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" encoding="UTF-8"/>
<xsl:template match="/">
<xsl:value-of select="message"/>
</xsl:template>
</xsl:stylesheet>
```

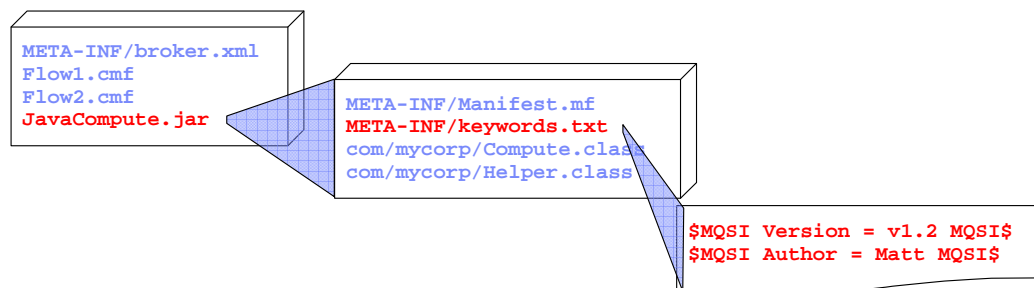
XSL stylesheets can be included in version control. However, no version property is available since these are not broker artifacts. However, keywords can be embedded at any place in an XSL stylesheet. The keyword can be added as an XML comment and must have the following format:

`$MQSI keyword = value MQSI$`

The example on this slide shows how to add the keyword of author with the value John to an XSL stylesheet. However, the Configuration Manager does not extract `version="1.0"` from this example because the value is not bounded by the `$MQSI` and `MQSI$` keywords.

Adding keywords to JAR files

- Associate keywords with JAR files:
 - ▶ Add a file called META-INF/keywords.txt to the root of the JAR file.
 - ▶ Add your keywords to the META-INF/keywords.txt file
- Example



17

Configuration – Version control

© 2007 IBM Corporation

If a bar file contains JAR files, you can associate keywords with the JAR files.

- Add a file called META-INF/keywords.txt to the root of the JAR file.
- Add your keywords to the META-INF/keywords.txt file, because this file is parsed for keywords when it is deployed.

For example, a deployed bar file contains *compute.jar*, and *compute.jar* contains the file META-INF/keywords.txt with the keyword information, between the \$MQSI eye catchers, as show on this slide.

This content means that the keywords “Version” and “Author” are associated with the deployed file *compute.jar* in the Configuration Manager Proxy and in the Message Brokers Toolkit.

Linking with version control systems

- Many version control systems (for example, CVS) use keyword systems that allow information to be expanded when files are checked out. For example:
 - ▶ CVS (RCS):
 - `$Revision: $` expands to `$Revision: 1.3 $`
 - ▶ SCCS:
 - `%I%` expands to `1.3`
- Use these tags inside `$MQSI..MQSI$` eye catchers to conveniently embed source code repository information:
 - ▶ `$MQSI Version = %I% MQSI$`
 - ▶ `$MQSI Revision = $Revision: $ MQSI$`
- When files containing these eye catchers are checked out, the version control system will resolve their values accordingly.

It may be necessary to link the version facility to an external code management repository such as CVS. Each version control system will have its own way of identifying code artifacts, and the syntax for creating version numbers will vary.

For example, the open-source CVS product uses the syntax “`$Revision: $`”. This must be spelled exactly as shown on this slide. Note that there is a single space between the colon and the final `$`.

When this string is processed by CVS, it is replaced by a generated version number, as shown on the slide.

These strings can be included on the Version property, where this is available through the broker Toolkit. Where this is not available, the string can be imbedded in the comments of the code source, as described on previous slides.

Summary

- Version control
 - ▶ Toolkit:
 - Version Property
 - Keywords
 - ▶ Linking to version-control systems

This session looked at the new WebSphere Message Broker Version 6 message flow and message set version and keyword capabilities. This can be accomplished:

- Directly through the Toolkit
- Linked to external code management systems such as CVS

Version control can be accomplished by:

- the Version property of broker artifacts
- creating your own keyword properties
- linking the values of these properties to external code management systems

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 NON-INFRINGEMENT. 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 2007. 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.