



IBM Software Group

WebSphere® Message Broker Overview

Application Development



@business on demand.

© 2006 IBM Corporation
Updated October 12, 2006

This presentation provides an introduction to WebSphere Message Broker Application Development

Agenda

- Application Development
- Summary and References

This presentation covers the WebSphere Message Broker Application Development tools and processes.

Section

Application development



This section will discuss the basics of WebSphere Message Broker application development.

Application development tools

- Toolkit (Workbench)
 - ▶ Eclipse based platform
 - ▶ Application Developer Perspective
- Application developer building blocks
 - ▶ Message flows
 - ▶ Message processing nodes
 - ▶ Message model

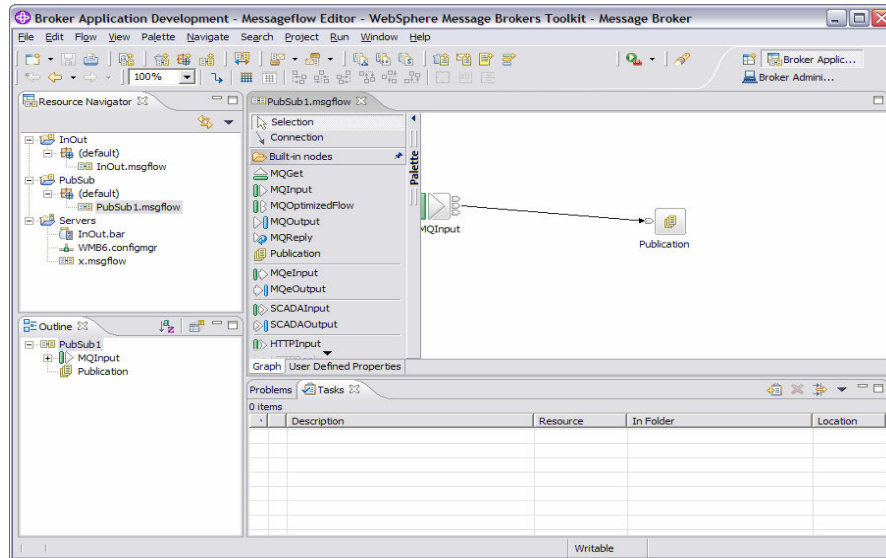


The Toolkit, or Workbench, is an Eclipse based platform. The Application Developer Perspective provides a development workspace.

The building blocks for a broker application developer are:

- Message flows
- Message processing nodes
- Message model

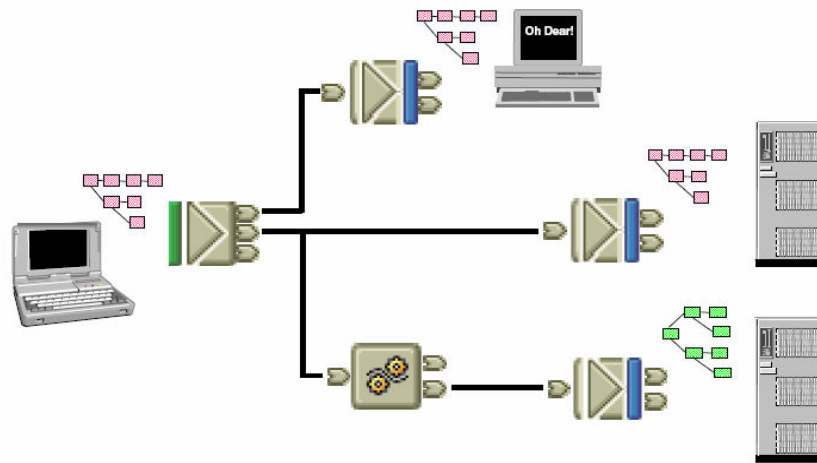
Broker Application Development perspective



This is a screen print of the broker application development perspective in the toolkit showing a publication/subscription message flow.

Message flows

The Message Flow is the first important object to understand. It represents connections between the applications you wish to integrate. A Message flow is a reusable, scalable, integration transaction.



Reusable, Scalable, Transaction

The message flow is the first important object to understand. It represents connections between the applications you want to integrate.

Message flows



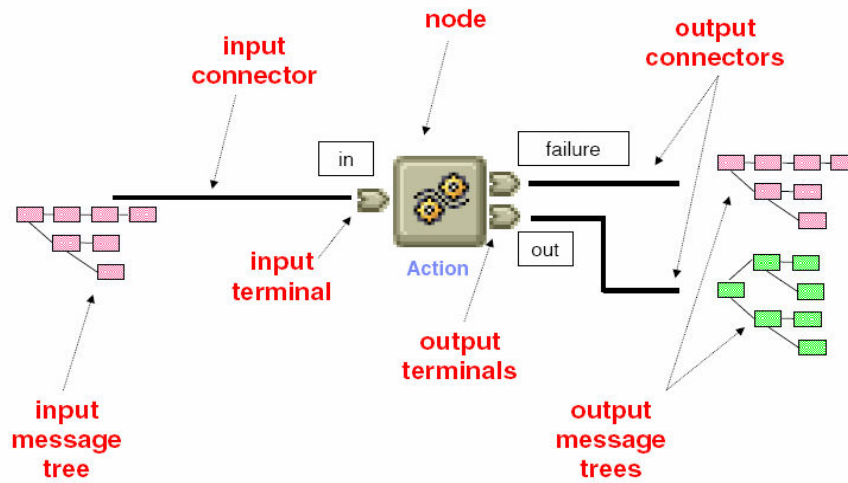
Message flows are general purpose, reusable integration applications. They provide the processing sequence required to connect applications together.

A message flow represents the connections between the applications you want to integrate. A message flow is a reusable, scalable, integration transaction.

A message flow may be transactional, multi-threaded, nested, or chained.

Message processing nodes

The Message processing node is the next object type to understand. There are many nodes, which both transform and route messages; they are the atoms of message integrations. They are combined to form message flows.



Message processing node is the next object type to understand. There are many nodes which both transform and route messages. These nodes are the atoms of message integration. They combine to form message flows.

Message processing nodes (cont.)

- Provide the individual processing elements that make up a message flow
 - ▶ Nodes represent functional routines encapsulating integration logic
 - ▶ Terminals represent the various outcomes possible from node processing
 - ▶ Connectors join the various nodes through their terminals
- Define a single logical operation on a message

- Input and output nodes are the endpoints of processing



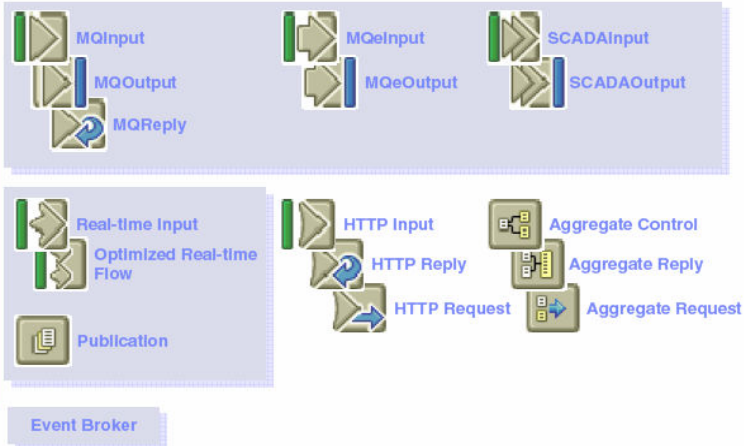
Message processing nodes are the individual elements that comprise a message flow. The nodes represent functional routines containing the logic. The terminals on a node represent the various outcomes of node processing, providing the capability to route the message accordingly. The connectors join the nodes through their terminals, creating the flow of the message.

A node is the single logical entity doing manipulation on a message. Multiple logical operations are represented by multiple nodes connected within the flow.

The input node is the start of the process; the output node is the end of the process.

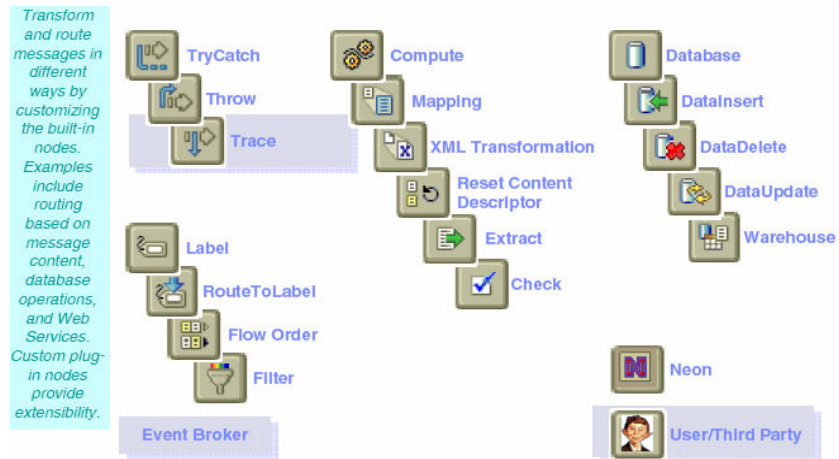
Built-in nodes

The broker comes with a built-in set of nodes that form an integration "Starter Pack." You can see that many of the most common transport protocols are supported, enabling you to get 'out of the box' quickly.



There are built-in nodes provided with the broker. Several transport protocols are supported, as shown in this slide.

Built-in nodes (cont.)



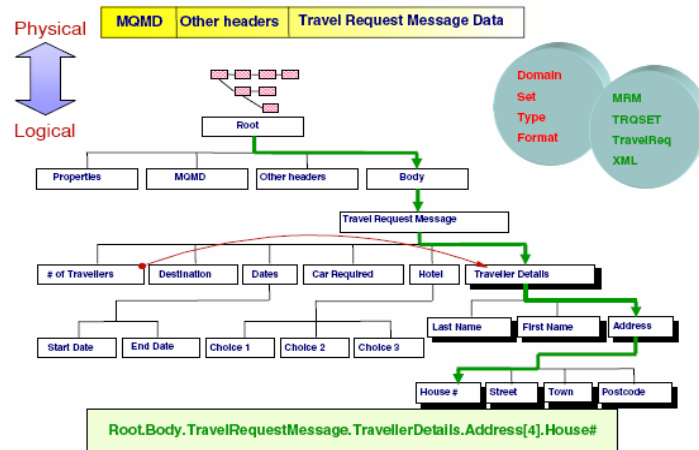
This slide shows transformation and routing nodes provided. These nodes provide capabilities such as:

- Routing based on message content
- Data base operations
- Web services

Custom plug-in nodes provide you with capabilities to extend the broker capabilities.

Logical message

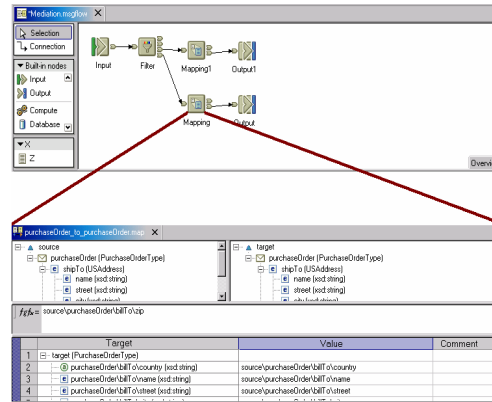
Outside the broker a message exists as a physical "bitstream". When a message is manipulated inside the broker, the logical form is used. The broker uses the domain, set, type and format to read and write the message between these forms.



As a message comes to the broker it exists as a 'bitstream'. Within the broker, a message is mapped to a logical form. The broker uses the domain, set, type and format to read and write the message between these forms.

The role of mapping in brokers

- Message flows are a graphical language for sequencing and decision making in brokers.
- Maps are transformation steps contained by a message flow.
- Maps concentrate on structural transformations of the message, not on sequencing steps



The message flows are a graphical paradigm for sequencing steps and decisions on complete messages.

A map is a single transformation step in brokers. You can transform a message programmatically or graphically:

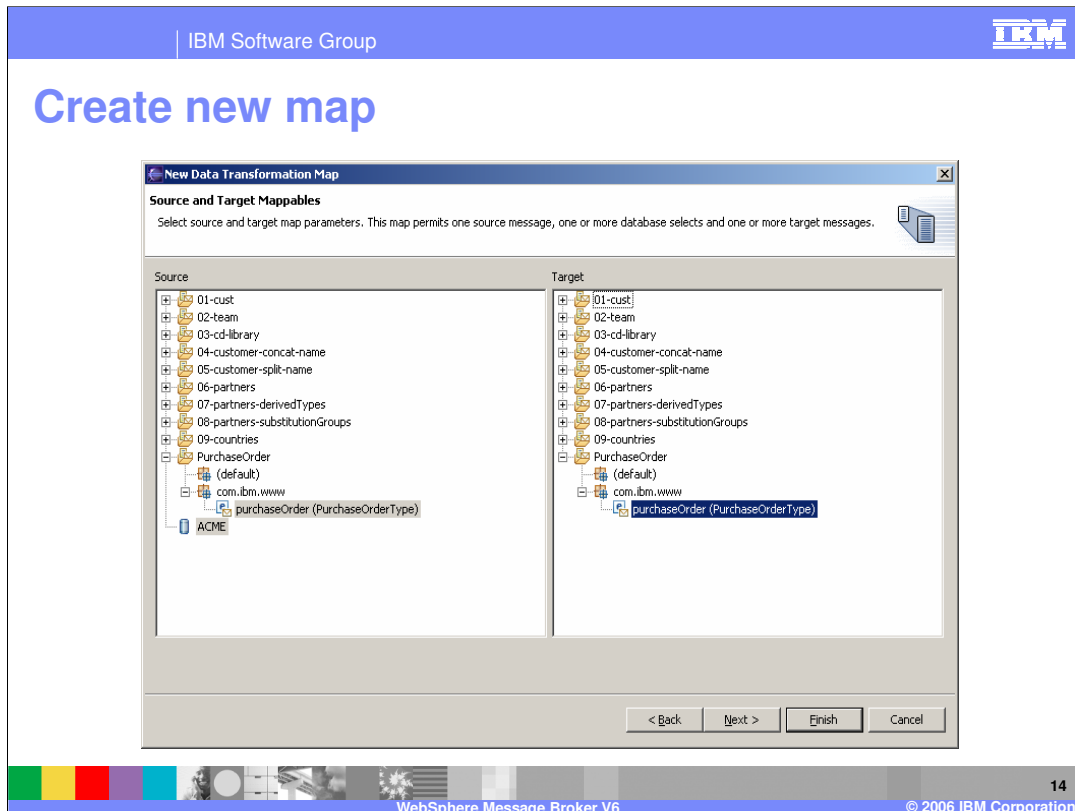
- Programmatically using ESQL in Compute node, Java™ in a Java compute node or XSLT in an XML Transformation node, or
- Graphically using the Mapping node.

A natural starting point for creating a map is to create a mapping node in a message flow.

Within the Mapping node properties, you can give a name for the map to use. You can also reuse an existing map.

You configure a message mapping using the Message Mapping editor, which you use to set values for:

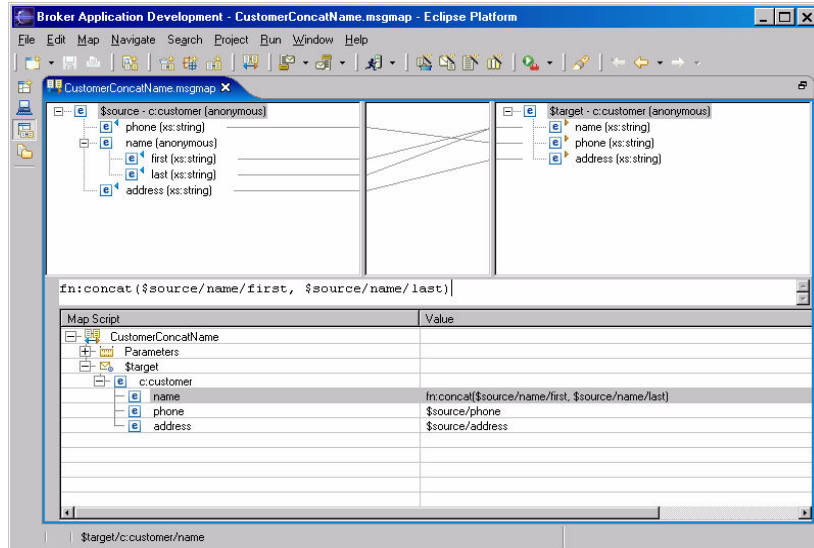
- the message destination
- message headers
- message content



When you create a new mapping, the New>Map wizard guides you through the process of configuring the map. This slide shows just one page in the wizard, where a database, an input message and an output message are selected.

The last page of the wizard provides a summary of the actions that will be taken, so that you can make sure everything is exactly like how you want it.

Submap scenarios



15

WebSphere Message Broker V6

© 2006 IBM Corporation

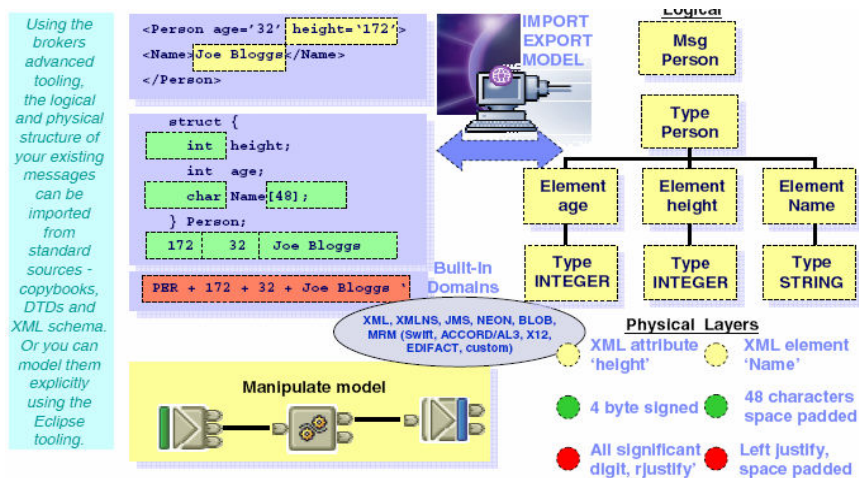
Creating a map is a matter of associating source and target elements through drag and drop.

When you perform a drag and drop, little “mapped” triangles appear in the source and target.

New in version 6, lines are drawn connecting the sources and targets as well. This is a visual guide to where the mappings are, and when the source and target trees collapse the lines are replaced with a ‘compound’ line. This helps you navigate around the source and target trees, to locate which fields are mapped and which are not.

When the source or target is collapsed, a ‘compound’ mapping line indicates that there are mapped items inside. This is a powerful navigation feature that helps you find a mapping even if the mappings are fairly sparse. A single mapping line then indicates where the source contributes to the target. This navigation tool makes it efficient and simple to understand which elements are mapped.

Message model: Import export



16

WebSphere Message Broker V6

© 2006 IBM Corporation

The broker provides tools to import messages from standard sources:

- COBOL copybooks
- C header files
- XML DTD
- WSDL
- XML schema

You may also model a message explicitly using the Eclipse tools within the broker.

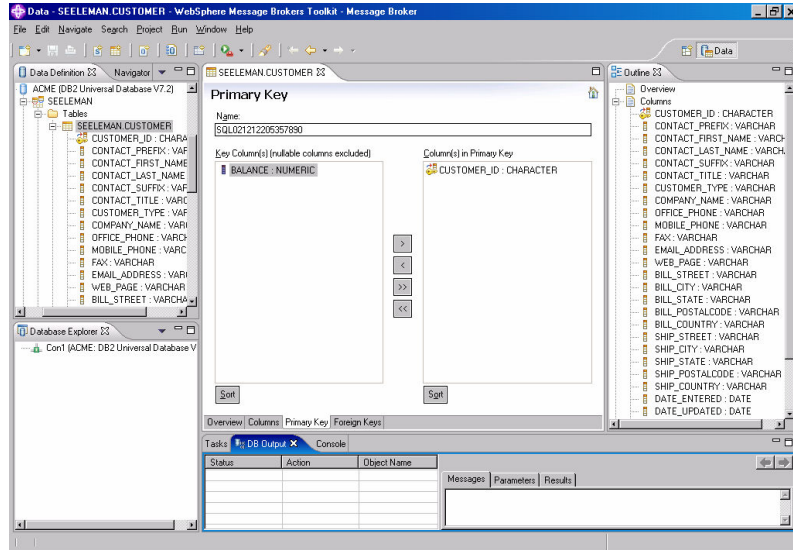
New tools in V6

- The data tools provide:
 - ▶ a mechanism to import database definitions used in editing maps
 - ▶ ESQL content assist
 - ▶ validation
- The XML tools include:
 - ▶ XML document editor
 - ▶ XSL style sheet editor
 - ▶ content assist.



In Version 6 of the message broker, there are new tools to facilitate mapping. The data tools provide a method to import database definitions that are used in editing maps. The XML tools provide an XML document editor and an XSL style sheet editor.

Data tools



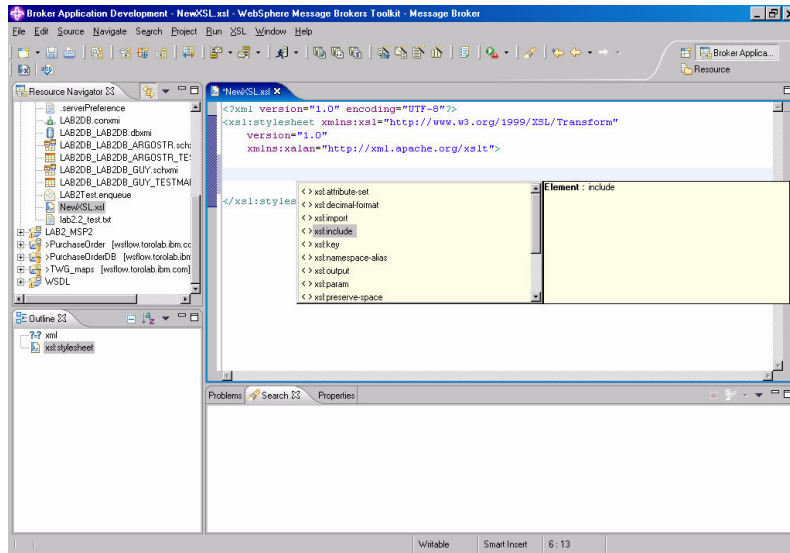
18

WebSphere Message Broker V6

© 2006 IBM Corporation

The data tools provide the mechanism to import database definitions that are used in editing maps and ESQL content assist and validation.

XML tools



19

WebSphere Message Broker V6

© 2006 IBM Corporation

The XML tools include an XML document editor and an XSL style sheet editor with content assist.

Section

Summary and references

The last portion of the presentation contains a summary and references.

Summary

- WebSphere Message Broker Toolkit application developer perspective
- Building blocks
 - ▶ Message flows
 - ▶ Message processing nodes
 - ▶ Message model

The Toolkit (Workbench) is an Eclipse based platform which provides the application developer with a workspace in the Application Developer Perspective.

The building blocks for a broker application developer are Message flows, Message processing nodes, and the Message model.

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:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)/business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

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 and/or changes in the product(s) and/or program(s) 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 (e.g., 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 2005,2006. 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.