



IBM Software Group

WebSphere® Message Broker Version 6.1

Administration



@business on demand.

© 2008 IBM Corporation
Updated January 21, 2008

This presentation will give an overall view of the key new Administration functions in WebSphere Message Broker Version 6.1

Agenda

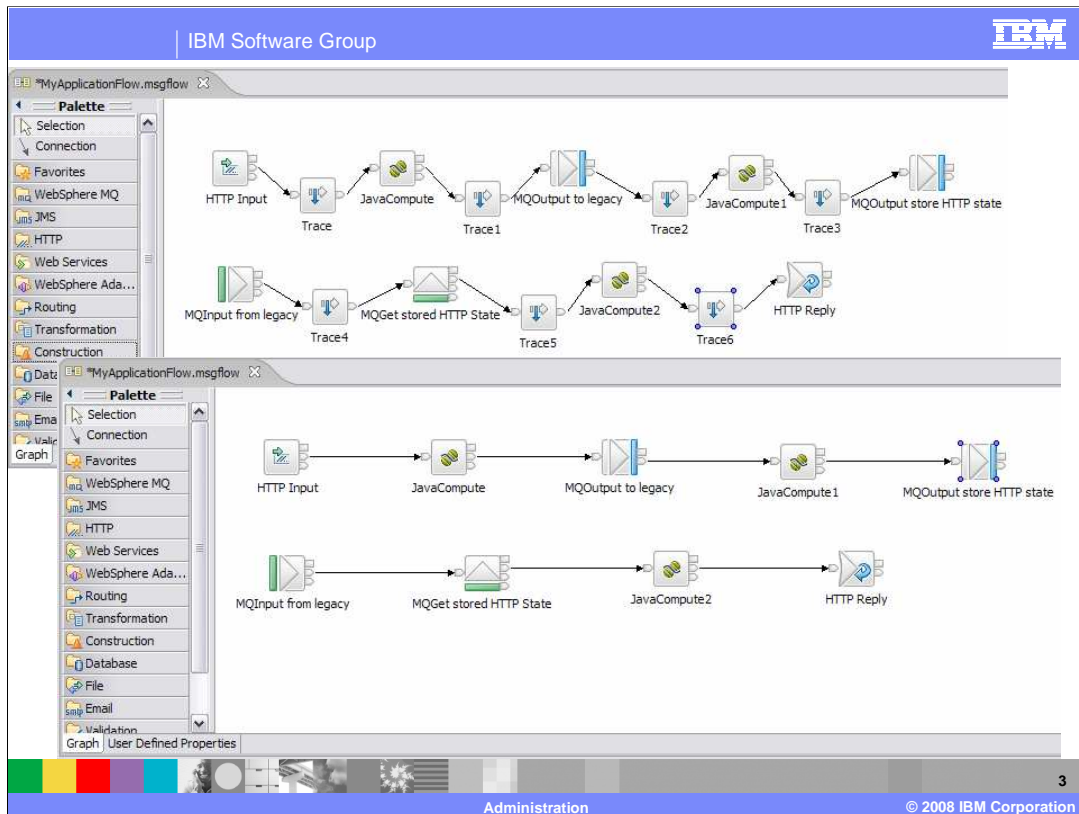
- Disabling trace nodes
- Additional instances
- JDBC™ XA support
- Improved BAR file processing
- Configuration manager proxy
- Runtime version handling
- IS02

This presentation covers the improvements in WebSphere Message Broker Version 6.1 in the area of administration.

These improvements include the ability to disable and enable trace nodes, additional instances configuration parameters, and JDBC XA support.

Improvements have also been made to BAR file processing and the configuration manager proxy.

Finally, this presentation will address improvements in runtime version handling capabilities and the MQ Explorer Broker Administration SupportPac® IS02.



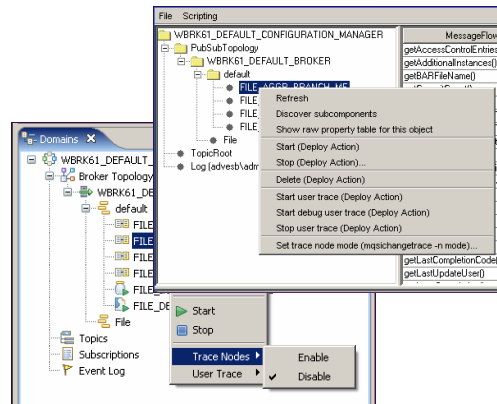
When developing message flows, using the trace node to generate trace records is a very useful debugging tool. These can incorporate text, message content, and date and time information, and helps you to monitor the behavior of the message flow. You can write the records to your user-trace file, another file, or the local error log.

The operation of the trace node is independent of the setting of user tracing for the message flow that contains it. In particular, records that are written by the trace node to the user trace log are written even if user trace is not currently active for the message flow.

Up until now, to stop the writing of trace records, you had to remove the trace nodes from the message flow and redeploy the flow to the broker. This has been changed in Message Broker Version 6.1.

Disabling trace nodes

- Turn off all trace nodes without having to remove them from your flow
- Administrative GUI, command line, and CMP control available
- For example:
 - ▶ `mqsichangetrace -n [on | off]`
 - ▶ `mqsireporttrace -n`



```
> mqsireporttrace broker -n -e default -f *
BIP8869I: The execution group trace node switch setting is 'on'.
When set to 'off', the execution group switch setting overrides the trace
node switch settings of the message flows within; otherwise, the effective
settings are determined by the individual message flows' trace node switch
setting. Use the 'f' flag on this command to discover individual message
flow trace node switch settings, if required.
BIP8869I: The trace node switch setting for message flow 'f1' is 'off'.
BIP8869I: The trace node switch setting for message flow 'f2' is 'on'.
```

In Version 6.1 of the Message Broker you can now disable trace nodes, without having to remove them from your message flow.

There are several ways of disabling trace nodes in a message flow.

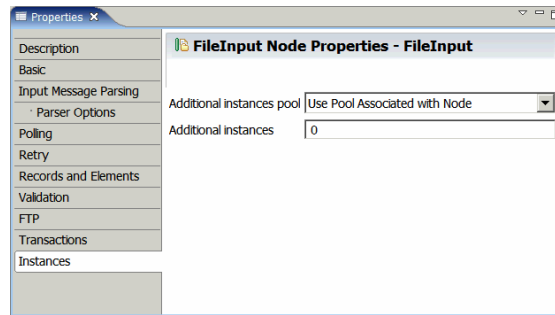
First, this can be performed from the Message Broker Toolkit in the Broker Administration perspective. Right click on the execution group or individual message flow where you want to disable trace nodes and select Trace Nodes, Disable (or Enable).

You can also turn off trace nodes using the command line. The “mqs-change-trace” command has flags that allow trace node options to be switched at execution group and individual flow level. The “mqs-report-trace” command allows you to see the current trace node options for a execution group or message flows within the execution group.

Finally, the configuration manager proxy API exerciser also allow trace node settings to be modified.

Additional instances

- Two new properties on selected input nodes
 - ▶ SOAPInput, FileInput, MQInput and all adapters input nodes



- Define an additional pool of threads that are associated with a specific input node

There are two new properties on the SOAP Input node, File Input node, MQInput node and the input nodes for all adapters. These properties allow more control over thread pools that are associated with specific input nodes.

The Additional Instances Pool property specifies whether additional instance threads for an input node are allocated from a thread pool for the whole message flow, or from a thread pool for use by that node only. The value of the Additional Instances property that is specified for the node controls the number of instances in the pool. If your message flow contains multiple input nodes, you can use the Additional Instances Pool and Additional Instances properties to ensure that each of the input nodes is allocated the required number of additional instances. Setting these properties allows you to fine tune your message flow operation. For example, if you know that one input node will receive twice as much input as another node, you can give it twice the number of additional threads.

To request additional instance allocation from the message flow thread pool, set Additional Instances Pool to Use pool associated with message flow. To request additional instance allocation from the node's own thread pool, set Additional Instances Pool to Use pool associated with node.

The Additional Instances property specifies the number of additional threads that the broker can use to service the message flow. These additional threads are created only if there are sufficient input messages. You can have up to 256 threads. The default value is 0. Additional threads can increase the throughput of a message flow but you should consider the potential impact on message order.

JDBC XA support

- Message Broker V6.1 adds support for type 4 JDBC drivers
- Enables brokers to include database interactions as part of a globally coordinated Message Broker transaction

```
public java.sql.connection  
    getJDBCType4Connection(String dataSourceAlias,  
                           int TransactionLevel)  
        throws MbException
```

- Also callable from ESQL
- Distributed platforms only

JDBC is an API developed by Sun Microsystems that provides a standard way to access data using Java. Using JDBC, an application can access a variety of databases and run on any platform with a Java Virtual Machine. The JDBC API defines a set of Java interfaces that encapsulate major database functionality, such as running queries, processing results, and determining configuration information. Because JDBC applications are written in Java, applications should work on any platform.

Today, there are four types of JDBC drivers in use:

Type 1 is a JDBC-ODBC bridge.

Type 2 is a partial Java driver.

Type 3 is a pure Java driver for database middleware.

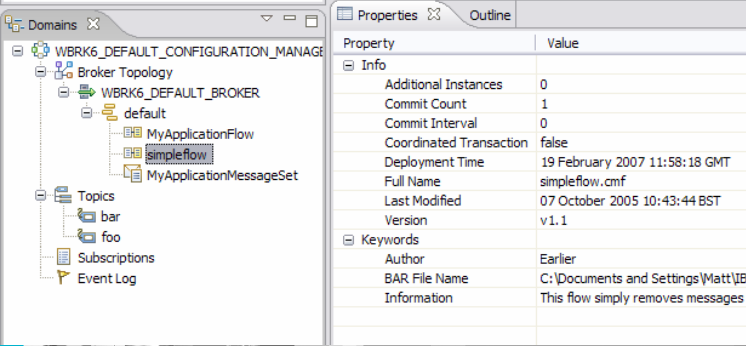
Type 4 is a pure Java driver for direct-to-database.

For most applications, the best choice is a pure Java driver, either Type 3 or Type 4. Type 4 drivers are the most common and are designed for a specific database product.

The scope of this feature is to provide support for connecting to Type 4 drivers through the Message Broker Mb-Node Java plug-in interface. You can enlist JDBC Type 4 database interactions as part of a globally coordinated Message Broker transaction on the distributed platforms.

Runtime version handling

- The configuration manager is able to store version handling information about each file being deployed inside the broker archive
- This information is remembered by the configuration manager and returned to configuration manager proxy applications, including the Message Broker Toolkit



The screenshot displays the IBM Message Broker Toolkit interface. On the left, a tree view shows the configuration structure under 'Domains'. The selected item is 'simpleflow', which is a message flow. The right pane shows the 'Properties' view for this message flow, listing various attributes and their values.

Property	Value
Info	
Additional Instances	0
Commit Count	1
Commit Interval	0
Coordinated Transaction	false
Deployment Time	19 February 2007 11:58:18 GMT
Full Name	simpleflow.cmf
Last Modified	07 October 2005 10:43:44 BST
Version	v1.1
Keywords	
Author	Earlier
BAR File Name	C:\Documents and Settings\Matt\B
Information	This flow simply removes messages

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

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

Runtime version handling – Embedding keywords

- XML-based files
 - ▶ **\$MQSI keyword = value**
MQSI\$ eyecatchers
embedded in the file
- Zip-based files
 - ▶ **\$MQSI keyword = value**
MQSI\$ eyecatchers
embedded in the zip entry
(META-INF\keywords.txt)
- Not all editors contain toolkit support for version handling

	File format
Compiled Message Flows (.cmf)	XML
Message Dictionaries (.dictionary)	XML
XSL Stylesheets (.xsl, .xslt, .xml)	XML
Java archives (.jar)	Zip
WebSphere TX maps (.mar)	Zip
XSD archives (.xsdzip)	Zip
Inbound adapter configuration (.inadapter)	Zip
Outbound adapter configuration (.outadapter)	Zip

8

Administration

© 2008 IBM Corporation

Several objects in WebSphere Message Broker can have additional information added to the object. This information can display information about an object after the object has been deployed.

You can define custom keywords, and their values that the configuration manager will interpret as additional information to be displayed, in the properties view.

Any keywords that you define must follow certain rules to ensure that the information can be parsed. In XML based files, such as compiled message flows, for example, these keyword definitions are embedded in the file. For zip based files, such as jar files within the bar file, the keyword definitions are embedded in the “keywords.txt” file of the jar file, under the “META-INF” directory.

mqsireadbar

- Displays the contents of a .bar file, including runtime version handling information

```
BIP1051I: Reading Bar file using tooling mqsireadbar...
C:\Documents and Settings\Matt\My Documents\BAR Files\test.bar:
simpleflow.cmf (14/06/07 17:28):
  Author = Matt
  Build = S000-L070614
  VERSION = v1.1
  Information = This flow simply removes messages from INPUT.QUEUE
Deployment descriptor:
simpleflow#additionalInstances
simpleflow#commitCount
simpleflow#commitInterval
simpleflow#coordinatedTransaction
simpleflow#MQInput.topicProperty
simpleflow#MQInput.validateMaster
simpleflow#MQInput.queueName = INPUT.QUEUE
simpleflow#MQInput.serializationToken

BIP8071I: Successful command completion.
```

- Improvements in V6.1
 - ▶ Displays properties in the deployment descriptor
 - ▶ Now available in both toolkit and runtime installations



The “mqsiread-bar” command lists to the console the keywords defined for each deployable file within a deployable broker archive file.

It also displays any deployment descriptor and the list of parameters that have been overridden.

In Version 6.1 of the Message Broker this is a command that is common to both runtime and toolkit environments. If this command is run on a workstation that has both the runtime and toolkit components installed, the version of the command that is used is determined by the relative locations of the toolkit and runtime directories.

mqsiapplybaroverride

- Replace an entire deployment descriptor with another (specifying either the new broker.xml file or a BAR file that contains it)
- Manually override properties specified in a text file or on the command line

```
BIP1137I: Applying overrides using tooling mqsiapplybaroverride...
BIP1139I: Replacing deployment descriptor with C:\Documents and Settings\Matt\My
Documents\BAR Files\broker.xml...
BIP1140I: Overriding property FLOW#NODE.PROPERTY with 'NEW_PROPERTY'...
BIP1140I: Overriding property FLOW2#NODE2.PROPERTY2 with 'NEW_PROPERTY2'...
BIP1143I: Saving Bar file C:\Documents and Settings\Matt\My Documents\BAR
Files\test_output.bar...

BIP8071I: Successful command completion.
```

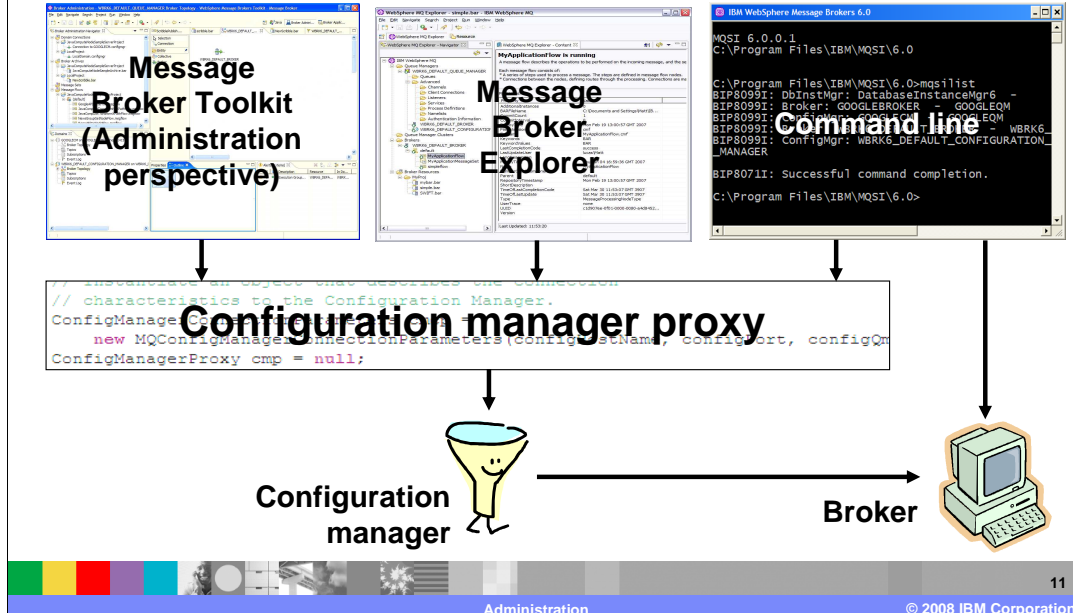
- Overwrite existing BAR file or write out to a new file
- Available in both toolkit and runtime installs



The “mqsi-apply-bar-override” is another command that is common to both the toolkit and runtime environment.

With the mqsi-apply-bar-override command, you can replace configurable values in the broker archive deployment descriptor with new values that you specify in a properties file.

Administration architecture



The configuration manager is the interface between the workbench and an executing set of brokers. It provides brokers with their initial configuration, and updates them with any subsequent changes. It maintains the broker domain configuration. This slide is a visual representation of the high-level administration architecture between the toolkit and command environments, the configuration Manager and the brokers, using the configuration manager proxy.

Configuration manager proxy

- Comprehensive Java API for administering broker domains:

```
import com.ibm.broker.config.proxy.*;

public class CreateBrokerReference {
    public static void main(String[] args) {
        ConfigManagerProxy cmp =
            ConfigManagerProxy.getInstance(...);
        TopologyProxy topology = cmp.getTopology();
        topology.createBroker("MYBROKER", "QMGR");
    }
}
```

- Used for administering brokers, execution groups, message flows, dictionaries (and other deployed message flow dependencies), subscriptions, the publish/subscribe topology, collectives, the event log, topics hierarchy and the configuration manager
- The Message Broker Toolkit administration perspective, IS02 and various command line utilities are visualizations of this API.
- Powerful samples available (for example, configuration manager proxy exerciser)

The configuration manager proxy is an application programming interface that your applications can use to control broker domains through a remote interface to the configuration manager. This proxy is sometimes known as the CMP.

Your applications have complete access to the configuration manager functions and resources through the set of Java classes that constitute the CMP. The CMP is a set of Java classes that sit logically between the user application and the configuration manager, inside the Java Virtual Machine of the user application.

As shown on the previous slide, the Broker Administration Perspective in the Message Broker Toolkit is a visualization of this API. The Message Broker Explorer in SupportPac IS02 also uses the same API. Many of the command line utilities that interact with the configuration manager also use this API.

Configuration manager proxy API exerciser

ConfigManagerProxy Method	Result
getAccessControlEntries()	(USER) matt F
getConfigManagerFullVersion()	1 06/02/15 11:16:04
getConfigManagerOSArch()	x86
getConfigManagerOSName()	Windows XP
getConfigManagerOSVersion()	5.1
getConfigManagerProxyVersion()	6
getConfigManagerVersion()	6
getConfigurationObjectType()	ConfigManagerProxy
getConfigurationObjectTypeOfParent()	null
getLastBIPMessages()	
getLastCompletionCode()	success
getLastUpdateUser()	dibble\matt

```

(18/08/06 13:25:47) Successfully registered for domain object <TopicRoot>.
(18/08/06 13:25:47) Successfully registered for domain object <Log (dibble\matt)>.
(18/08/06 13:25:47) Successfully registered for domain object <GOOGLECM>.
(18/08/06 13:25:47) The Configuration Manager's object hierarchy is available for use.
(18/08/06 13:25:47) Click on an object in the hierarchy to select it and display its properties.
(18/08/06 13:25:47) Right-click on a selected object to manipulate it.
(18/08/06 13:25:47) <---- cmp.exerciser.ClassTesterForConfigManagerProxy.testConnect
  
```

Connected to Configuration Manager 'GOOGLECM'.

Through using the configuration manager proxy samples, you can deploy a BAR file, manage a broker domain, or use the configuration manager proxy API exerciser to perform various tasks. The CMP samples introduce you, at a basic level, to the features available with the configuration manager proxy.

Configuration manager proxy changes in V6.1

- New classes and methods to read and manipulate BAR files:
 - ▶ BARFile
 - ▶ Deployment descriptor
- Set and get many additional broker properties
 - ▶ “mqsichangeproperties” and “mqsireportproperties” capability
 - ▶ For example, HTTP listener port, debug port additional instances, policy sets
- Ability to synchronize broker and configuration manager

14

Administration

© 2008 IBM Corporation

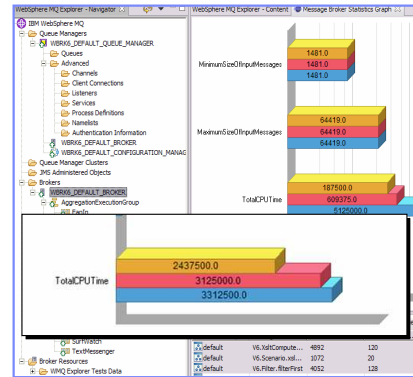
In Version 6.1 new classes and methods have been added to the configuration manager proxy to extend the API. These include those to manipulate BAR files and the BAR file Deployment Descriptor.

Additional broker properties, such as HTTP listener port, policy sets and others, can use the “mqsi-change-properties” command to set additional broker properties. The “mqsi-report-properties” can be used to report those properties.

Version 6.1 also adds the ability to synchronize the broker and configuration manager.

MB explorer Eclipse administration

- Alternative MQ Explorer based administration
 - ▶ Simplifies administration of MQ, MB networks in single Explorer console
 - ▶ Uses MQ Eclipse 'extension points' to provide seamless experience
- Comprehensive administration facilities
 - ▶ All features in broker administration
 - ▶ Also includes new features such as multiple execution group deployment
- Performance Monitor
 - ▶ Easily view processor, IO and other metrics in Eclipse
- IS02 Category 3 SupportPac
 - ▶ Fully supported in production



The Message Broker Explorer allows for an alternative to the Message Broker Toolkit, for administration. This is an MQ Explorer-based administration tool which allows for comprehensive administration of MQ and Message Broker networks from a single console. Most Broker Administration features can be performed from this console, including new features such as deployment to multiple Execution Groups.

The Performance Monitor allows you to easily view metrics such as processor usage and I/O activity in an Eclipse-based environment.

This Message Broker Explorer is delivered as a Category 3 Support Pack, number IS02.

Summary

- Disabling trace nodes
- Additional instances
- JDBC XA support
- Improved BAR file processing
- Configuration manager proxy
- Runtime version handling
- IS02

This presentation has covered the new features and improvements in Message Broker Version 6.1 for administration. These topics are summarized on this slide.

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_Admin.ppt

This module is also available in PDF format at: ..\\WMB61_IEA_Admin.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:

IBM SupportPac WebSphere

Java, JDBC, 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 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 2008. 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.