



IBM Software Group

WebSphere® MQ V7.0

Publish/subscribe administration - Part 2



@business on demand.

© 2008 IBM Corporation
Updated July 31, 2008

This is the second of two modules discussing administration of publish/subscribe in WebSphere MQ V7.

Agenda

- Introduction
- Topic tree administration
- Subscription administration
- Additional administrative topics
- Summary



Subscription administration and monitoring is covered in this module. Publish/subscribe introduction and topic tree administration are discussed in Publish/subscribe administration part 1.

Section

Subscription administration



This section covers subscription administration and monitoring.

Topic attributes for subscriptions

- DURSUB
 - ▶ Specifies whether applications are permitted to make durable subscriptions on this topic
- SUB
 - ▶ Controls whether applications are to be permitted to subscribe to this topic
- SUBSCOPE
 - ▶ Subscribes to publications in this queue manager
 - ▶ Subscribes to a network of connected queue managers.
- PROXYSUB
 - ▶ Controls whether a proxy subscription can be sent for this topic to directly connected queue managers, even if no local subscriptions exist
- WILDCARD
 - ▶ Controls the behavior of wildcard subscriptions with respect to this topic



The TOPIC attribute DURSUB determines whether the creation of durable subscriptions is allowed at this point in the topic tree. If set to NO, and if the MQSUB call uses MQSO_DURABLE, MQSUB will fail with MQRC_DURABILITY_NOT_ALLOWED.

The attribute SUB determines whether subscribing is allowed at this point in the topic tree at all. If set to DISABLED, an MQSUB call fails with MQRC_SUB_INHIBITED.

SUBSCOPE determines whether this queue manager subscribes to publications in this queue manager or in a network of connected queue managers.

PROXYSUB controls whether a proxy subscription can be sent for this topic to directly connected queue managers, even if no local subscriptions exist.

WILDCARD is a special attribute to block the propagation of subscriptions to very generic wildcard subscriptions, such as MQSUB('#'), where you don't really want portions of your topic tree exposed to such subscribers. It doesn't have an ASPARENT value as it only applies at that specific point in the topic tree.

DISPLAY CONN

```
DIS CONN(832AC44720013b05) TYPE(ALL)
AMQ8276: Display Connection details.
CONN(832AC44720016401)
EXTCONN(414D51435445535431202020202020)
TYPE(CONN)
APPLTAG(D:\q.exe)                APPLTYPE(USER)
USERID(hughson)

OBJNAME(Q1)                        OBJTYPE(Queue)
OPENOPTS(MQOO INPUT SHARED,MQOO FAIL IF QUIESCING)

OBJNAME( )                         OBJTYPE(Topic)
DEST(Q1)                           DESTQGR(TEST1)
SUBNAME(Fruit Prices)
SUBID(414D51205445535431202020202020832AC44720016403)
TOPICSTR(Price/Fruit/+)
```



Display connection also provides the subscription ID for a topic string on this connection.

DISPLAY SBSTATUS

```
DIS SBSTATUS SUBID(414D51205445535431202020202020832AC44720013D07
AMQ8099: webSphere MQ subscription status inquired.
SUB(Fruit Prices)
SUBID(414D51205445535431202020202020832AC44720013D07)
SUBUSER(hughson) RESMDATE(2008 02 26)
RESMTIME(19:10:07) LMSGDATE(2008 02 26)
LMSGTIME(19:10:28)
ACTCONN(414D51435445535431202020202020832AC44720013D05)
DURABLE(NO) NUMMSG(5)
SUBTYPE(API)
```



The subscription ID retrieved from a display connection or tpstatus for subscribers can be used as input to the display for subscription status. Output is shown here.

Topic status for subscribers in MQ Explorer

The screenshot displays the IBM WebSphere MQ Explorer interface. The main window shows a tree view on the left with 'Queue Managers' expanded to 'TEST1'. The 'Topics' pane on the right shows a filter set to 'Default for Topics'. A 'TEST1 - Status' window is open, showing the status for the 'Price/Fruit/Apples' topic. Below this, a 'Price/Fruit/Apples - Status' window is open, displaying a table of subscribers for the topic 'Price/Fruit/Apples'.

Queue Manager: TEST1 Topic Name: Price/Fruit/Apples

Topic status - subscribers for the topic "Price/Fruit/Apples":

Filter: Standard for Topic Status - Subscriber

Topic string	Subscription ID	User	Message count	Durable	Type	Connection ID	Resume date	Resume time	Date of last r
Price/Fruit/Apples	414D5120544...	hughson	5	No	API	414D51435...	26-Feb-2008	19:39:46	26-Feb-2008

Scheme: Default for Topic Status - Subscriber - Distributed

Last updated: 19:40:14

Buttons: Refresh, Close, Refresh All, Close

MQ Explorer can be used to display the topic status for subscribers as shown here.

Subscriptions perspectives

```

DIS TPSTATUS( Price/Fruit/+ ) TYPE(SUB) ALL
AMQ8754: Display topic status details.
TOPICSTR(Price/Fruit/Oranges)
SUBID(414d51205445535431202020202020832ac44720013d07)
SUBUSER(hughson) DURABLE(NO)
ACTCONN(414d51435445535431202020202020832ac44720013d05)
NUMMSGS(2) SUBTYPE(API)

DIS SBSTATUS SUBID(414d51205445535431202020202020832ac44720
AMQ8099: webSphere MQ subscription status inquired.
SUB(Fruit Prices)
SUBID(414d51205445535431202020202020832ac44720013d07)
SUBUSER(hughson)
ACTCONN(414d51435445535431202020202020832ac44720013d05)
DURABLE(NO) NUMMSGS(5)
SUBTYPE(API)
  
```

Using DISPLAY TPSTATUS TYPE (SUB) you can see the details of the current publishers on this topic string. One of the attributes returned is the Active Connection ID (ACTCONN) which links to DISPLAY CONN which shows you the details about that specific application. The single subscription to 'Price/Fruit/+' shows subscribers on two topic strings. This is because this display is shown from the perspective of the topic string. The Subscription ID (SUBID) links to DISPLAY SBSTATUS where there is a single subscription with that ID; this display is from the perspective of the subscription.

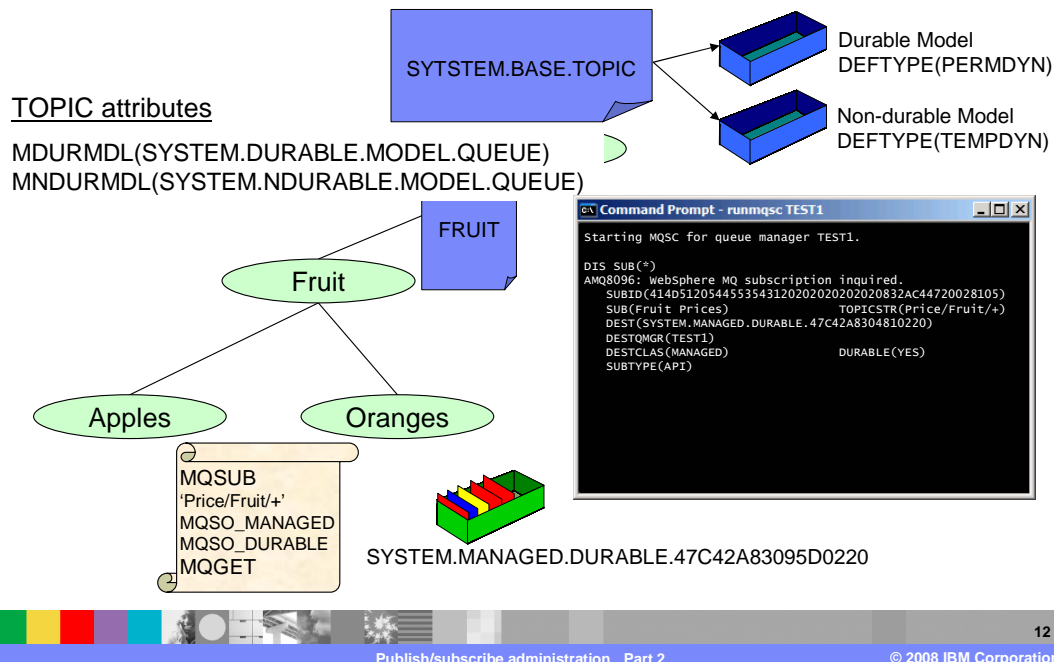
Section

Additional administration topics



This section describes additional administrative features in WebSphere MQ V7.

Configuring managed destinations



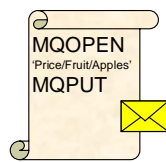
Applications using managed destinations for delivery of their publications do not need to deal with the queue where those publications reside. However, an administrator can configure things about this queue. Managed destinations are dynamic queues created based on the model queue defined at the specific point in the topic tree. The SYSTEM.BASE.TOPIC defines the two model queues, one for durable subscriptions and one for non-durable subscriptions. These topic attributes are shown here.

If you don't define any TOPIC objects with model queues in these attributes then all TOPIC objects will inherit these attributes from the SYSTEM.BASE.TOPIC. To over-ride these models at different points in the tree, the model for the durable subscriber must be permanent dynamic. The model for the non-durable subscriber should be temporary dynamic.

The dynamic queues created for subscribers using MQSO_MANAGED or DESTCLAS(MANAGED) will have a stem of SYSTEM.MANAGED.DURABLE or SYSTEM.MANAGED.NDURABLE depending on the durability of the subscription using it. You can see the queue name being used in DISPLAY SUB and DISPLAY CONN.

Configuring behavior on failure

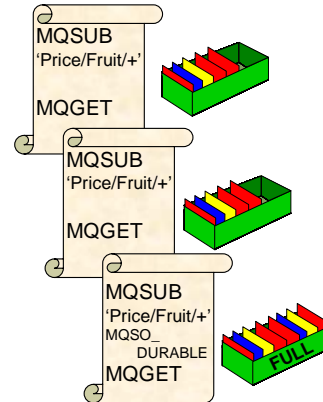
MSGDLV	Behaviour
ALL	If one failure – no-one gets it
ALLDUR	Any failure to a durable subscriber – no-one gets it
ALLAVAIL	Failures don't affect the delivery to subscribers who can accept it



MQRC_PUBLICATION_FAILURE

TOPIC attribute defaults

PMSGDLV (ALLDUR)
NPMSGDLV (ALLAVAIL)



If a publication satisfies several of subscribers and the queue manager is unable to deliver the publication to one of the subscribers, for example if its destination queue is full, what happens? Three choices can be configured on the topic in question using the persistent and non-persistent message delivery attributes shown here.

You can configure your system so that all subscribers must be delivered the publication or none of them get it. This is the value ALL. If one of the subscribers cannot accept it, the MQPUT fails with MQRC_PUBLICATION_FAILURE.

You can configure your system so that all durable subscribers must be delivered the publication or none of them get it. This is the value ALLDUR. If one of the durable subscribers cannot accept it as in this diagram, the MQPUT fails, again with MQRC_PUBLICATION_FAILURE.

You can configure your system so that no failure of one specific subscriber has any affect on the delivery of the publication to other subscribers. This is the value ALLAVAIL.

Creating administrative subscriptions

Subscription name	Topic string	Type
APPLES.TO.Q1	Price/Fruit/Apples	Admin
Fruit Prices	Price/Fruit/+	API

```

Command Prompt - runmqsc TEST1
Starting MQSC for queue manager TEST1.
DEFINE SUB(APPLES.TO.Q1)
TOPICSTR( Price/Fruit/Apples )
DEST(Q1)
  
```

DEFINE SUB
 ALTER SUB
 DELETE SUB
 DISPLAY SUB

You do not have to create subscriptions by coding applications to use MQSUB. You can create them administratively. You can use an application that does an MQGET from a specific queue and enable it to consume publications by creating an administrative subscription that sends publications to the queue the application reads.

There are DEFINE, ALTER, DELETE and DISPLAY commands for subscriptions. DELETE SUB can be useful in tidying up durable subscriptions that applications have made and forgotten about. This can happen if the application did not call MQCLOSE when finished.

There is a SUBTYPE field associated with a subscription. If the subscription was created from an application issuing MQSUB the SUBTYPE is API. If it was created through an administrative command the SUBTYPE is ADMIN.

IBM Software Group IBM

Alias queues

Base object:
 Base type:

```

graph TD
    Price([Price]) --- Fruit([Fruit])
    Fruit --- Apples([Apples])
    Fruit --- Oranges([Oranges])
    APPLES[APPLES] --> Apples
    PRICES[PRICES] -.-> Apples
    MQPUT[MQPUT (PRICES)]
  
```

Command Prompt - runmqsc TEST1
 Starting MQSC for queue manager TEST1.
 DEFINE TOPIC(APPLES)
 TOPICSTR(Price/Fruit/Apples)
 DEFINE QALIAS(PRICES)
 TARGETTYPE(TOPIC)
 TARGET(APPLES)

15

Publish/subscribe administration Part 2 © 2008 IBM Corporation

Not only can you make a point-to-point consumer into a consumer of publications by means of an administrative command, you can also make a point-to-point producer of messages into a publisher of messages by an administrative command.

Changing the queue that the putting application uses into an alias queue which points to a topic, turns that application into a publishing application.

Creating an administrative subscription and requesting that publications be sent to the original getting application's queue joins the two original applications up again using publish/subscribe.

However, this will work only if the point-to-point producer and point-to-point consumer are not using the same physical queue. If this is the case, you can first convert the putter to use an alias queue targeting the getters queue. Then convert to publish/subscribe.

Having converted the applications to publish/subscribe, other interested parties can subscribe to this topic without conflict on the getting queue or complicated logic in the putting application.

Displaying a subscription

The screenshot shows the 'Fruit Prices - Properties' dialog box with the following fields and callouts:

- Subscription name:** Fruit Prices (Callout: MQSD.SubName)
- Topic:**
 - Topic name: (Callout: MQSD.ObjectName)
 - Topic string: |Price/Fruit/+ (Callout: MQSD.ObjectString)
- Wildcard usage:** Topic level wildcard (Callout: MQSO_WILDCARD_CHAR, MQSO_WILDCARD_TOPIC)
- Scope:** All (Callout: MQSO MANAGED or not)
- Destination:**
 - Destination class: Provided (Callout: MQSO MANAGED or not)
 - Destination queue manager: TEST1 (Callout: MQSUB hObj_parameter)
 - Destination name: Q1 (Callout: MQSD.SubCorrelId)
- Correl ID:**

00000	41	4D	51	20	54	45	53	54--31	20	20	20	20	20	20
00010	83	2A	C4	47	20	02	15	03--						
- Durable:** Yes (Callout: MQSO DURABLE or not)
- Type:** API (Callout: MQSO DURABLE or not)
- Properties:** None (Callout: MQSD.SubUserData)
- User data:** (Callout: MQSD.SubUserData)
- Selector:** (Callout: MQSD.SelectionString)

DISPLAY SBSTATUS shows the run-time status of a subscription. DISPLAY SUB shows the more static attributes of the subscription as shown here. Type shows whether the subscription was created using a DEFINE SUB command or an MQSUB call in an application. In this instance the type is API meaning it was created by an MQSUB call in an application.

Displaying a subscription

The screenshot shows the 'Fruit Prices - Properties' dialog box with the 'Extended' tab selected. The following settings are visible and annotated:

- Variable user ID:** Fixed (Callout: MQSO_FIXED_USERID, MQSO_ANY_USERID)
- User:** hughson (Callout: MQSD.PubAppIdentityData)
- Application identity:** (Empty)
- Accounting token:** 00000 16 01 05 15 00 00 00 CA--F1 62 65 65 29 DC B5 A1 |1
00010 53 04 6E F4 01 00 00 00--00 00 00 00 00 00 0B | Z.n..... (Callout: MQSD.PubAccountingToken)
- Publish priority:** As published (Callout: MQSD.PubPriority)
- Subscription ID:** 00000 41 4D 51 20 54 45 53 54--31 20 20 20 20 20 20 | AMQ TEST1
00010 83 2A C4 47 20 02 15 03-- | .*.G ...
- Expiry:** Unlimited (Callout: MQSD.Expiry)
- Request only:** All (Callout: MQSO PUBLICATIONS ON REQUEST)
- Subscription level:** 1 (Callout: MQSD.SubLevel)

Buttons at the bottom include 'Apply', 'OK', and 'Cancel'.

This screen capture shows the mapping between the MQSD (Subscription Descriptor) and the DISPLAY SUB output.

MQSD and options - Reference

```

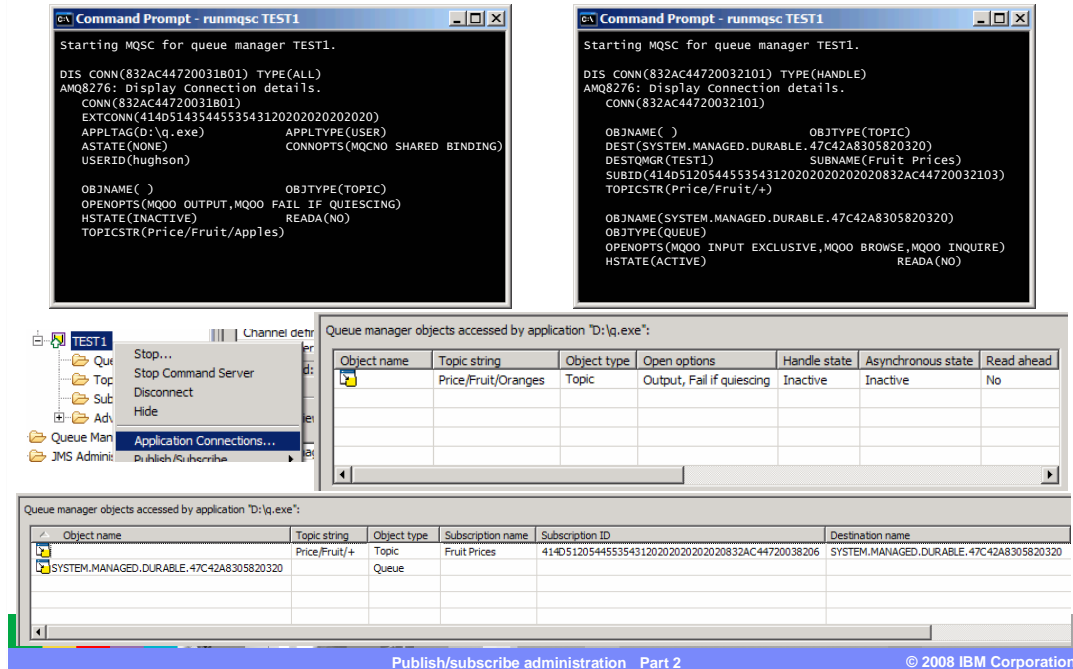
struct tagMQSD
{
    MQCHAR4   StrucId;
    MQLONG   Version;
    MQLONG   Options;
    MQCHAR48  ObjectName;
    MQCHAR12  AlternateUserId;
    MQBYTE40  AlternateSecurityId;
    MQLONG   SubExpiry;
    MQCHARV   ObjectString;
    MQCHARV   SubName;
    MQCHARV   SubUserData;
    MQBYTE24  SubCorrelId;
    MQLONG   PubPriority;
    MQBYTE32  PubAccountingToken;
    MQCHAR32  PubApplIdentityData;
    MQCHARV   SelectionString;
    MQLONG   SubLevel;
    MQCHARV   ResObjectString;

    MQSO_NON_DURABLE
    MQSO_READ_AHEAD_AS_Q_DEF
    MQSO_ALTER
    MQSO_CREATE
    MQSO_RESUME
    MQSO_DURABLE
    MQSO_GROUP_SUB
    MQSO_MANAGED
    MQSO_SET_IDENTITY_CONTEXT
    MQSO_FIXED_USERID
    MQSO_ANY_USERID
    MQSO_NOT_OWN_PUBS
    MQSO_PUBLICATIONS_ON_REQUEST
    MQSO_NEW_PUBLICATIONS_ONLY
    MQSO_FAIL_IF QUIESCING
    MQSO_ALTERNATE_USER_AUTHORITY
    MQSO_WILDCARD_CHAR
    MQSO_WILDCARD_TOPIC
    MQSO_SET_CORREL_ID
    MQSO_SCOPE_QMGR
    MQSO_NO_READ_AHEAD
    MQSO_READ_AHEAD
};

```

On the left are the MQ Subscription Descriptor fields. On the right are listed the subscribe options.

DISPLAY CONN additions



The screenshot shows two command prompt windows and two GUI windows. The command prompts show the output of the DISPLAY CONN command, displaying connection details for a queue manager. The GUI windows show the queue manager objects accessed by an application, including a table of objects and a detailed view of a specific object.

Command Prompt - runmqsc TEST1 (Left):

```
Starting MQSC for queue manager TEST1.
DIS CONN(832AC44720031B01) TYPE(ALL)
AMQ8276: Display Connection details.
CONN(832AC44720031B01)
EXTCONN(414D514354453543120202020202020)
APPLTAG(D:\q.exe)      APPLTYPE(USER)
ASTATE(NONE)           CONNOPTS(MQCN0 SHARED BINDING)
USERID(hughson)

OBJNAME( )             OBJTYPE(TOPIC)
OPENOPTS(MQOO OUTPUT,MQOO FAIL IF QUIESCING)
HSTATE(INACTIVE)      READA(NO)
TOPICSTR(Price/Fruit/Apples)
```

Command Prompt - runmqsc TEST1 (Right):

```
Starting MQSC for queue manager TEST1.
DIS CONN(832AC44720032101) TYPE(HANDLE)
AMQ8276: Display Connection details.
CONN(832AC44720032101)

OBJNAME( )             OBJTYPE(TOPIC)
DEST(SYSTEM.MANAGED.DURABLE.47C42A8305820320)
DESTQMGR(TEST1)       SUBNAME(Fruit Prices)
SUBID(414D512054453543120202020202020832AC44720032103)
TOPICSTR(Price/Fruit/+)

OBJNAME(SYSTEM.MANAGED.DURABLE.47C42A8305820320)
OBJTYPE(QUEUE)
OPENOPTS(MQOO INPUT EXCLUSIVE,MQOO BROWSE,MQOO INQUIRE)
HSTATE(ACTIVE)        READA(NO)
```

Queue manager objects accessed by application 'D:\q.exe':

Object name	Topic string	Object type	Open options	Handle state	Asynchronous state	Read ahead
	Price/Fruit/Oranges	Topic	Output, Fail if quiescing	Inactive	Inactive	No

Queue manager objects accessed by application 'D:\q.exe':

Object name	Topic string	Object type	Subscription name	Subscription ID	Destination name
SYSTEM.MANAGED.DURABLE.47C42A8305820320	Price/Fruit/+	Queue	Fruit Prices	414D512054453543120202020202020832AC44720038206	SYSTEM.MANAGED.DURABLE.47C42A8305820320

© 2008 IBM Corporation

DISPLAY CONN provides information about the applications connected to the queue manager and the handles that they have open.

When an application opens a topic to publish messages to it, you will see this open object handle in DISPLAY CONN.

When an application subscribes to a topic to receive publications, it is returned a handle to the subscription which is in DISPLAY CONN. If the subscription was made using the option MQSO_MANAGED, the handle to the subscription destination queue that has been created by the queue manager for this subscribing application can also be seen in DISPLAY CONN.

Section

Summary

This section provides a summary of WebSphere MQ V7 publish/subscribe as presented in Publish/subscribe administration part 1 and part 2.

Summary

- Topic tree administration control
 - ▶ Topic object definition
 - ▶ Topic tree behavior
 - ▶ A point for security control
- No code change publish/subscribe
 - ▶ Queue aliases pointing to topics
 - ▶ Administrative subscription commands
- Administration and monitoring
 - ▶ Updates to Display CONN
 - ▶ New status displays
 - Topic status
 - Subscription status

Administration for publish/subscribe in WebSphere MQ V7 uses the standard MQ administration interfaces.

Topic object configuration and the behavior of the topic tree is controlled in the definition and setting of attributes on the topic objects. These objects are also the point for security control.

No code changes are required to use publish/subscribe; applications written to MQPUT to a queue or MQGET from a queue can be included in a publish/subscribe environment with only administrative changes.

There are updates to DISPLAY CONN and new status displays to allow monitoring of applications using the topic tree.

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_PubSub_Admin_part2.ppt

This module is also available in PDF format at: ../PubSub_Admin_part2.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 WebSphere

A current list of other IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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.