

WebSphere MQ Publish/Subscribe Part II

Capitalware's MQ Technical Conference v2.0.1.3

Agenda

- **Part I**
 - ▶ Introduction
- **What is Pub/Sub?**
 - ▶ Pub/Sub in WebSphere MQ
 - ▶ Pub/Sub and the WMQ API
 - ▶ Topic Tree Design
- **Part II**
 - ▶ **WMQ Pub/Sub Administration**
 - ▶ **WMQ Pub/Sub Operations**
 - ▶ **Tricks of the Trade**
 - ▶ **Topologies**
 - ▶ **Using WMQ V7 Pub/Sub and WebSphere Application Server**
 - ▶ **What's New**

Capitalware's MQ Technical Conference v2.0.1.3

Administration

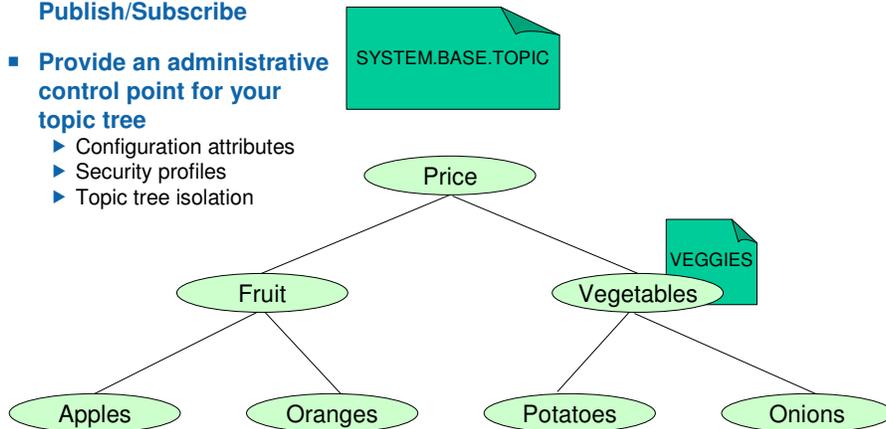
Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

Topic Objects and the Base Topic Object

- Not necessary for Publish/Subscribe
- Provide an administrative control point for your topic tree
 - ▶ Configuration attributes
 - ▶ Security profiles
 - ▶ Topic tree isolation



Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- By default there is always at least one Topic Object present – this is the base topic object, that has the name SYSTEM.BASE.TOPIC. This object contains all the default settings that control the behavior of topic objects that are lower in the hierarchy. If you want your whole topic tree to behave in the same way and have no need for any other topics, you can alter this object to have the behavior you require.
- Note that while you can delete this default object, but doing so will have no effect - the queue manager will act as if the SYSTEM.BASE.TOPIC was defined with the default attributes that come out of the box. If you delete this object and later decide that you need to change that behavior, you will first need to re-define this object again.
- While no other Topic Objects are necessary in order to make use of Publish/Subscribe in WebSphere MQ, there are a number of advantages that can be had by making use of them.
- The purpose of Topic Objects are to provide an administrative control point for your topic tree. Without them, all nodes in the topic tree are essentially the same, in terms of security, properties, etc.
- Topic Objects can be used to override the default configuration attributes for a particular node in the tree. This in turn can influence the behavior of publishers and/or subscribers accessing that point in the topic tree.
- Topic Objects also enable you to create security profiles for particular points in the tree. You can control who can Publish, who can Subscribe, etc to particular topics in the topic tree.
- A very powerful capability for which you can use Topic Objects is to provide what is called Topic Tree isolation. What this is and why you want to consider making use of it was discussed in the first of these two sessions.

Capitalware's MQ Technical Conference v2.0.1.3

Defining a topic object

SYSTEM.BASE.TOPIC

Price

Fruit

Apples

Oranges

FRUIT

```

DEFINE TOPIC
ALTER TOPIC
DELETE TOPIC
DISPLAY TOPIC
    
```

```

Command Prompt - runmqsc TEST1
Starting MQSC for queue manager TEST1.
DEFINE TOPIC(FRUIT)
TOPICSTR('Price/Fruit') DURSUB(NO)

DISPLAY TOPIC(FRUIT)
AMQ8633: Display topic details.
TOPIC(FRUIT)                                TYPE(LOCAL)
TOPICSTR('Price/Fruit')                      DESCR( )
CLUSTER( )                                    DURSUB(NO)
PUB(ASAPARENT)                               SUB(ASAPARENT)
DEFPERSIST(ASAPARENT)                        DEFPRTY(ASAPARENT)
DEFPRESP(ASAPARENT)                          ALTDTE(2008-02-26)
ALTIME(15.05.22)                             PMSGDLV(ASAPARENT)
NPMSGDLV(ASAPARENT)                          PUBSCOPE(ASAPARENT)
SUBSCOPE(ASAPARENT)                          PROXYSUB(FIRSTUSE)
WILDCARD(PASSTHRU)                            MDURMDL( )
MNDURMDL( )
    
```

Capitalware's MQ Technical Conference v2.0.1.3

Defining a topic object

IBM WebSphere MQ Explorer

WebSphere MQ Explorer - Content

Topics

Filter: Default for Topics

Topic name	Topic type	Topic string	Description	Publish	Subscribe	Durable subscriptions	Default pri
FRUIT	Local	Price/Fruit		As parent	As parent	Inhibited	As parent

Scheme: [] Default priority: []

Last upd: []

< Back Next > Finish Cancel

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- Let's say you need to disallow creation of durable subscriptions for one half of the topic tree. We can create one TOPIC object at the highest point where we need this behavior to start, and that behavior will be inherited by the nodes in the topic tree below that point without the need for any further TOPIC object definitions.
- As you might expect, this new object type has DEFINE, ALTER, DELETE and DISPLAY commands. One thing to note about ALTER is that the TOPICSTR parameter of a TOPIC object cannot be altered. Think of this attribute as the other name of the TOPIC object – you cannot alter the name of an object, you must delete and redefine an object to do that.
- Looking at the DISPLAY output from the object we just defined, we can see that many of the attributes that we didn't specify have the value ASPARENT (or for the character strings – have blanks, which means the same thing as the ASPARENT value). ASPARENT means that the value for this attribute is taken from the next TOPIC object found by walking up the topic tree. If the next TOPIC object found also says ASPARENT for the value that is being resolved we carry on up the tree – eventually we may get to the very top and thus use the values in the SYSTEM.BASE.TOPIC.

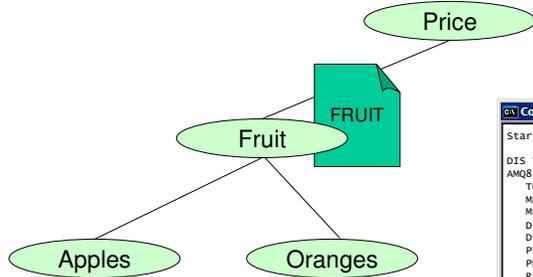
This Slide Intentionally Left Blank

Resolving ASPARENT

Topic status:

Topic string	Publish	Subscribe	Admin topic name	Durable subscriptions
/				
Price	Allowed	Allowed		Allowed
Fruit	Allowed	Allowed	FRUIT	Inhibited
Apples	Allowed	Allowed		Inhibited
Oranges	Allowed	Allowed		Inhibited

DISPLAY TPSTATUS



```

Command Prompt - runmqsc TEST1
Starting MQSC for queue manager TEST1.

DIS TPSTATUS('Price/Fruit')
AMQ8754: Display topic status details.
  TOPICSTR('Price/Fruit')
  MDURMDL(SYSTEM.DURABLE.MODEL.QUEUE)
  MNDURMDL(SYSTEM.NDURABLE.MODEL.QUEUE)
  DEFPRTY(0)
  DEFPRES(SYNC)
  PUB(ENABLED)
  PMSGDLV(ALLDUR)
  RETAINED(NO)
  SUBCOUNT(0)
  SUBSCOPE(ALL)
  ADMIN(FRUIT)
  DURSUB(NO)
  SUB(ENABLED)
  NPMSGDLV(ALLAVAIL)
  PUBCOUNT(0)
  PUBSCOPE(ALL)
  
```

Capitalware's MQ Technical Conference v2.0.1.3

IBM WebSphere MQ Explorer

WebSphere MQ Explorer - Content

Topics

Filter: Default for Topics

TEST1 - Status

Queue Manager: TEST1

Topic status:

Topic string	Publish	Subscribe	Admin topic name	Durable subscriptions	Default priority	Default persistence	Model durable c
/							
Price	Allowed	Allowed		Allowed	0	Not persistent	SYSTEM.DURAB
Fruit	Allowed	Allowed	FRUIT	Inhibited	0	Not persistent	SYSTEM.DURAB
Apples	Allowed	Allowed		Inhibited	0	Not persistent	SYSTEM.DURAB
Oranges	Allowed	Allowed		Inhibited	0	Not persistent	SYSTEM.DURAB

Scheme: Default for Topic Status - Distributed

Table last refreshed: 16:18:39 Selected item last updated: 16:18:39

Refresh Refresh All Close

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

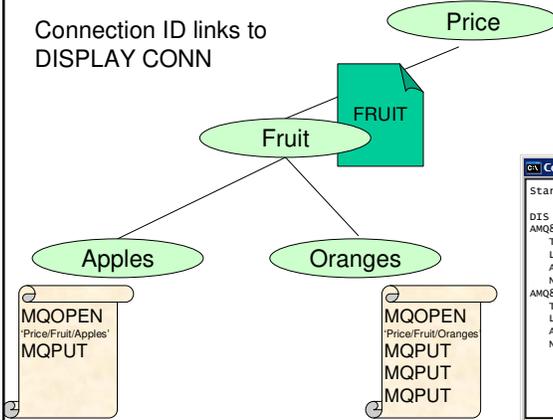
- In order to see what the real values being used for the attributes that have the value ASPARENT, you can use the DISPLAY TPSTATUS command.
- This command takes a topic string, not a topic object as its input. This means you can find the actual values that are going to be used at any point in the topic tree – not just at those points which have defined TOPIC objects.

This Slide Intentionally Left Blank

Administration for Publishers

Topic string	Date of last publication	Time of last publication	Publish count	Connection ID
Price/Fruit/Oranges	26-Feb-2008	16:50:44	3	414D51435...

Connection ID links to
DISPLAY CONN



TOPIC attributes

- DEFPRTY
- DEFPSIST
- DEFPRESP
- PUB
- PUBSCOPE
- PMSGDLV
- NPMSGDLV

```

Command Prompt - runmqsc TEST1
Starting MQSC for queue manager TEST1.

DIS TPSTATUS('Price/Fruit/*') TYPE(PUB) all
AMQ8754: Display topic status details.
TOPICSTR(Price/Fruit/Oranges)      LPUBDATE(2008-02-26)
LPUBTIME(16:50:44)
ACTCONN(414D514354455354312020202020202020832AC44720005E02)
NUMPUBS(3)
AMQ8754: Display topic status details.
TOPICSTR(Price/Fruit/Apples)      LPUBDATE(2008-02-26)
LPUBTIME(16:50:37)
ACTCONN(414D5143544553543120202020202020832AC44720007601)
NUMPUBS(1)
  
```

IBM WebSphere MQ Explorer

TEST1 - Status

Queue Manager: TEST1

Topic status:

Topic string	Publish	Subsc
/		
Price	Allowed	Allowe
Fruit	Allowed	Allowe
Apples	Allowed	Allowe
Oranges		

Price/Fruit/Oranges - Status

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

Topic status - publishers for the topic 'Price/Fruit/Oranges':

Filter: Default for Topic Status - Publisher

Topic string	Date of last publication	Time of last publication	Publish count	Connection ID
Price/Fruit/Oranges	26-Feb-2008	16:50:44	3	414D514354455...

Scheme: Default for Topic Status - Publisher - Distributed

Last updated: 18:25:28

Refresh Close

N
O
T
E
S

- There are a few attributes on the topic object that are relevant to publishers. We will look at those here along with the topic status display that shows information about publishers.
- There are various options on MQPUT that can be left to resolve from the object that was opened. Priority, Persistence and Asynchronous Put Response (new in V7). Using MQPRI_PRIORITY_AS_TOPIC_DEF, MQPER_PERSISTENCE_AS_TOPIC_DEF and MQPMO_RESPONSE_AS_TOPIC_DEF (all of which constants have the same numeric value as their equivalent AS_Q_DEF constants) means that that the actual value is resolved from the topic object.
- The TOPIC attribute PUB determines whether publishing is allowed at this point in the topic tree. If set to DISABLED, an MQPUT call will fail with MQRC_PUT_INHIBITED. PUBSCOPE will be discussed later when we cover "Distributed Publish/Subscribe". PMSGDLV and NPMSGDLV will be covered later when we talk about configuring behavior for publication failures.
- As already discussed, with the resolution of ASPARENT values, the object that finally resolved the value may be further up the topic tree than the point at which you are publishing.
- Using DISPLAY TPSTATUS TYPE(PUB) 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.

This Slide Intentionally Left Blank

Monitoring your Application Subscriptions

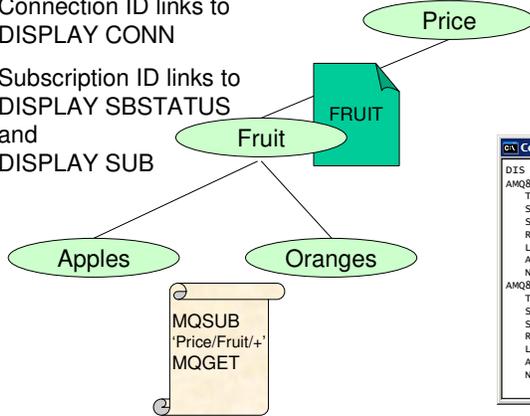
Topic string	Subscription ID	User	Message count	Durable	Type	Connection ID	Resume date	Resume time
Price/Fruit/Oran...	414D5120544...	hughson	0	No	API	414D51435...	26-Feb-2008	19:54:33

TOPIC attributes

- DURSUB
- SUB
- SUBSCOPE
- PROXYSUB
- WILDCARD

Connection ID links to
DISPLAY CONN

Subscription ID links to
DISPLAY SBSTATUS
and
DISPLAY SUB



```

Command Prompt - runmqsc TEST1
DIS TPSTATUS('Price/Fruit/+') TYPE(SUB) ALL
AMQ8754: Display topic status details.
TOPICSTR(Price/Fruit/Oranges)
SUBID(414D51205445535431202020202020832AC44720013D07)
SUBUSER(hughson) RESMDATE(2008-02-26)
RESMTIME(18:53:35) LMSGDATE(2008-02-26)
LMSGTIME(18:53:41) DURABLE(NO)
ACTCONN(414D51435445535431202020202020832AC44720013D05)
NUMMSG(2) SUBTYPE(API)
AMQ8754: Display topic status details.
TOPICSTR(Price/Fruit/Apples)
SUBID(414D51205445535431202020202020832AC44720013D07)
SUBUSER(hughson) RESMDATE(2008-02-26)
RESMTIME(18:53:35) LMSGDATE(2008-02-26)
LMSGTIME(18:53:41) DURABLE(NO)
ACTCONN(414D51435445535431202020202020832AC44720013D05)
NUMMSG(2) SUBTYPE(API)
    
```

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

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

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

Filter: Standard for Topic Status - Subscriber

Scheme: Default for Topic Status - Subscriber - Distributed

Last updated: 19:40:14

N
O
T
E
S

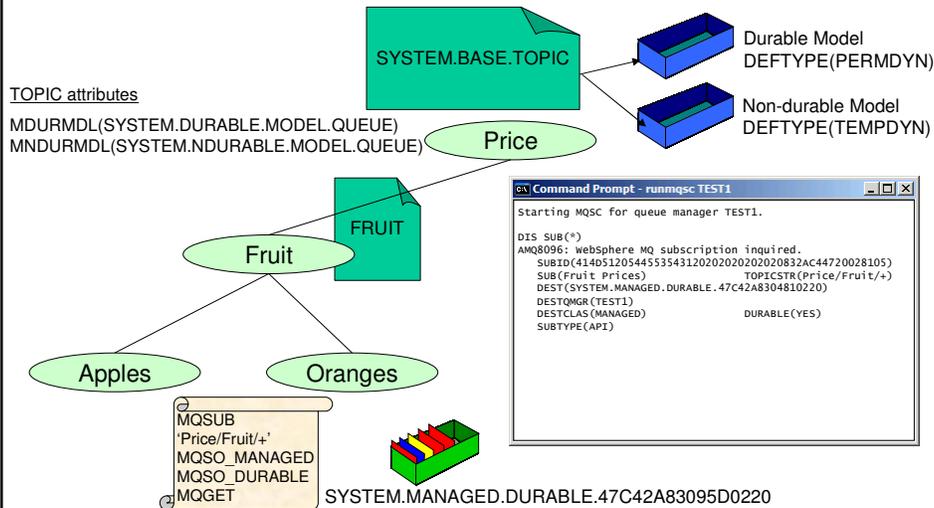
- There are a few attributes on the topic object that are relevant to subscribers. We will look at those here along with the topic status display that shows information about subscribers.
- The TOPIC attribute DURSUB determines whether the creation of durable subscriptions is allowed at this point in the topic tree. If set to NO, and MQSUB using MQSO_DURABLE 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 will fail with MQRC_SUB_INHIBITED. SUBSCOPE and PROXYSUB will be discussed later when we cover "Distributed Publish/Subscribe". WILDCARD is a special attribute to block the propagation of subscriptions to very generic wildcarded 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.
- As already discussed, with the resolution of ASPARENT values, the object that finally resolved the value may be further up the topic tree than the point at which you are publishing.

Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

Configuring managed destinations



Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- If your applications are using managed destinations for delivery of their publications, the queue that publications reside upon is not something the application has to worry about, but an administrator may wish to configure things about this queue. Managed destinations are dynamic queues and are 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) as SYSTEM.DURABLE.MODEL.QUEUE and SYSTEM.NDURABLE.MODEL.QUEUE. These names are provided in the MDURMDL and MNDURMDL keywords on the TOPIC object definition.
- 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. If you want to over-ride these models at different points in the tree, there is one thing to remember. The model for the durable subscriber (MDURMDL) 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. We will look at the changes in DISPLAY CONN in detail a little later.

Capitalware's MQ Technical Conference v2.0.1.3

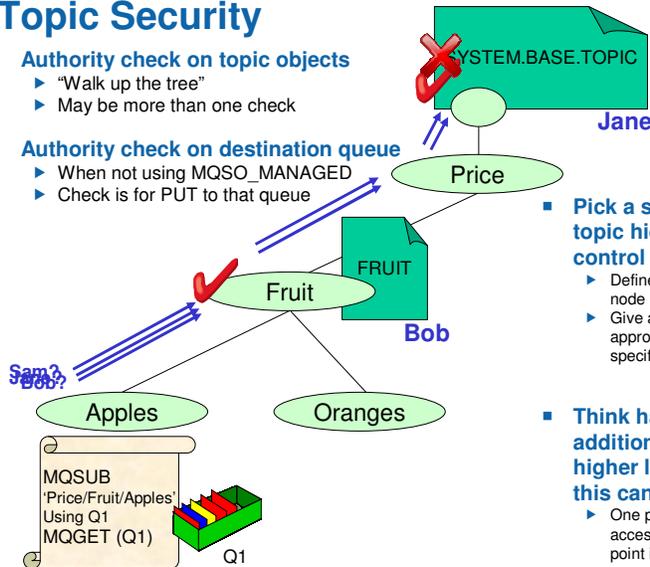
Topic Security

- **Authority check on topic objects**

- ▶ “Walk up the tree”
- ▶ May be more than one check

- **Authority check on destination queue**

- ▶ When not using MQSO_MANAGED
- ▶ Check is for PUT to that queue



- **Pick a suitable layer of the topic hierarchy and set access control at this point.**

- ▶ Define topic objects for each topic node at that layer.
- ▶ Give all publishers and subscribers the appropriate level of access for each specific branch of the topic tree

- **Think hard before adding additional access control at higher levels in the tree as this can cause confusion.**

- ▶ One potential is to grant a wider access to ‘super users’ from a higher point in the tree.

Capitalware's MQ Technical Conference v2.0.1.3

N O T E S

- When MQOPENING a topic (MQOT_TOPIC) for MQOO_OUTPUT – that is, in order to publish, or when making an MQSUB call to subscribe to a topic, a security check is done to see if your user ID has authority to use that topic.
- In our example we have called MQSUB at the point in the topic tree, “Price/Fruit/Apples”. There is no topic object at this point in the topic tree, so to find the profile we need to check authorities against we walk up the topic tree to find a node which does have a topic object. The next point is “Price/Fruit”. This does have a topic object, FRUIT, so we will check that this user ID has subscribe authority on the profile for the FRUIT topic. If that user ID does have authority, our search stops there. If it does not, we carry on searching up the topic tree and will check the SYSTEM.BASE.TOPIC to see if this user ID has subscribe authority there.
- Be aware that an additional authorisation check is done for an MQSUB call when the application wishes to use a specific destination queue (i.e. is not using the MQSO_MANAGED option). In this case we also check that this user ID has authority to PUT to that destination queue.
- Strategies for how best to handle security:
 - Pick a suitable layer of the topic hierarchy and set access control at this point. Define topic objects for each topic node at that layer. Give all publishers and subscribers the appropriate level of access for each specific branch of the topic tree.
 - Think very carefully before adding additional access control at higher levels in the tree as this can cause confusion.
 - One potential is to grant a wider access to ‘super users’ from a higher point in the tree.

Capitalware's MQ Technical Conference v2.0.1.3

Displaying a subscription

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

- MQSD.SubName**: Points to the 'Subscription name' field.
- MQSD.ObjectName** and **MQSD.ObjectString**: Point to the 'Topic name' and 'Topic string' fields.
- MQSO_WILDCARD_CHAR** and **MQSO_WILDCARD_TOPIC**: Point to the 'Wildcard usage' dropdown.
- MQSO_MANAGED or not**: Points to the 'Scope' dropdown.
- MQSUB hObj parameter**: Points to the 'Destination queue manager' field.
- MQSD.SubCorrelId**: Points to the 'Correl ID' field.
- MQSO_DURABLE or not**: Points to the 'Durable' checkbox.
- MQSD.SubUserData**: Points to the 'User data' field.
- MQSD.SelectionString**: Points to the 'Selector' field.

Version: v2.0.1.3

N
O
T
E
S

- We have already seen DISPLAY SBSTATUS which shows the run-time status of a subscription. DISPLAY SUB is slightly different – it shows the more static attributes of the subscription, whether it was created using a DEFINE SUB command or an MQSUB call in an application.
- On these next slides we can see the mapping between the MQSD (Subscription Descriptor) and the DISPLAY SUB output.

Displaying a subscription

The screenshot shows the 'Fruit Prices - Properties' dialog box with the 'Extended' tab selected. The following fields and their corresponding callouts are visible:

- Variable user ID:** Fixed (Callout: MQSO_FIXED_USERID, MQSO_ANY_USERID)
- User:** hughson (Callout: MQSD.PubAppIdentityData)
- Application identity:** (Callout: MQSD.PubAccountingToken)
- Accounting token:** 00000 16 01 05 15 00 00 00 CA--F1 62 65 65 29 DC B5 A1 |
00010 53 04 6E F4 01 00 00 00--00 00 00 00 00 00 0B | z.n.....
- 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)

v2.0.1.3

```

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  PubAppIdentityData;
    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
};
    
```

DISPLAY CONN additions

The screenshot displays two command prompts and two tables from the MQ console. The left command prompt shows the output of the 'DISPLAY CONN' command for a queue manager, listing connection details such as CONN, EXTCNN, and OBJNAME. The right command prompt shows the output of the 'DISPLAY CONN' command for a specific connection, listing details like OBJNAME, DEST, and SUBID. Below the command prompts are two tables showing queue manager objects accessed by an application.

Table 1: 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

Table 2: 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	414051205445535431202020202020832AC44720038206	SYSTEM.MANAGED.DURABLE.47C42A8305820320

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- 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 you will see 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.
- This slide shows some examples of these displays.

Capitalware's MQ Technical Conference v2.0.1.3

Questions?



Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

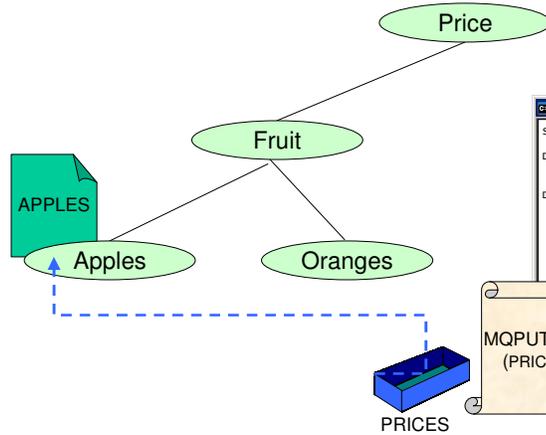
Tricks of the Trade

Capitalware's MQ Technical Conference v2.0.1.3

Alias Queues

Base object:
Base type:

Publish Your MQ Event Messages!



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

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- In the same way that you can make a point-to-point consumer into a consumer of publications simply by means of an administrative command, you can also make a point-to-point producer of messages into a publisher of messages, again by means of 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 (as we have just seen) and requesting that publication are sent to the original getting application's queue joins the two original application up again, but now via publish/subscribe.
- One thing to note, this will only work if the point-to-point producer and point-to-point consumer were not previously using exactly the same physical queue. If they were you might first want to convert the putter to use an alias queue targeting the getters queue, and then from there convert to publish/subscribe.
- Now that you are using publish/subscribe, other interested parties can also subscribe to this topic without conflict on the getting queue or complicated logic in the putting application.

This Slide Intentionally Left Blank

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)
    
```

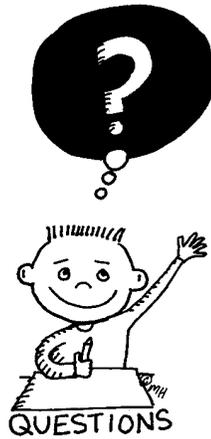
DEFINE SUB
 ALTER SUB
 DELETE SUB
 DISPLAY SUB

Capitalware's MQ Technical Conference v2.0.1.3

NOTES

- You don't have to create subscriptions by coding applications to use MQSUB, you can create them administratively. This means that you can take an application that is coded simply to MQGET from a specific queue and have it consume publications by creating an administrative subscription that sends publications to the queue it is getting from.
- There are DEFINE, ALTER, DELETE and DISPLAY commands for SUB. DELETE SUB may be useful in tidying up durable subscriptions that applications have made and forgotten about – i.e. not called MQCLOSE for them when they were finished with them.
- One thing to note about subscriptions is the SUBTYPE field. If the subscription was created from an application issued MQSUB it will be SUBTYPE(API) but if it were created through an administrative command it will be SUBTYPE(ADMIN).
- We will look at DISPLAY SUB a little later.

Questions?



Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

Publish/Subscribe Topologies

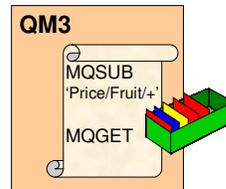
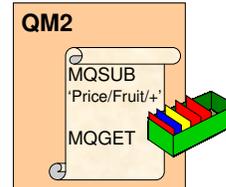
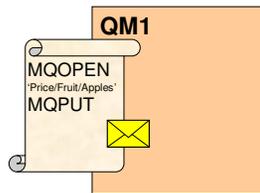
Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

Publish/Subscribe Topologies

- Local Queuing -> Distributed Queuing
- Publish/Subscribe -> Distributed Publish/Subscribe
- Application API calls remain the same
- Administration changes have the effect



Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

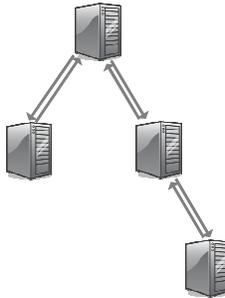
- In the previous presentation we concentrated on the publish/subscribe API and Administration within a single queue manager. Of course, just as queuing can be on a single queue manager or can be moving messages between multiple queue managers – known as distributed queuing – so can publish/subscribe. We call this distributed publish/subscribe.
- This is the concept (and the features to implement it) that an application may be publishing to a topic on QM1 and other applications may be subscribing on that topic on others queue managers, here QM2 and QM3, and the publication message needs to flow to those other queue managers to satisfy those subscribers.
- The application code stays the same, you still call MQSUB or MQOPEN and MQPUT, the difference, as with distributed queuing is in the administrative set-up of your queue managers.
- We are going to look at the different ways you can set up your queue managers to publish messages to another queue manager.

Capitalware's MQ Technical Conference v2.0.1.3

Multi-Queue Manager Pub/Sub Approaches

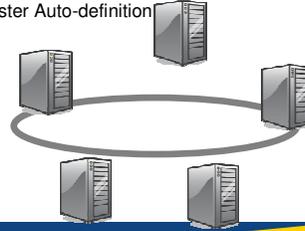
Pub/Sub Hierarchies

- Similar to WebSphere MQ V6 hierarchies
- Very scalable
- Lack of availability if intermediate queue manager or connections to it fail
 - ▶ Highly available hub and spokes
- Inflexible
 - ▶ Channels must exist
 - ▶ Transmit queues of same name as remote queue manager



Pub/Sub Clusters

- Similar to WebSphere Message Broker Collectives
- Scalability limited to cluster sizes that can cope with the all-to-all cluster connectivity
- Availability – Direct links between queue managers
- Flexible, low cost administration
 - ▶ Cluster Auto-definition



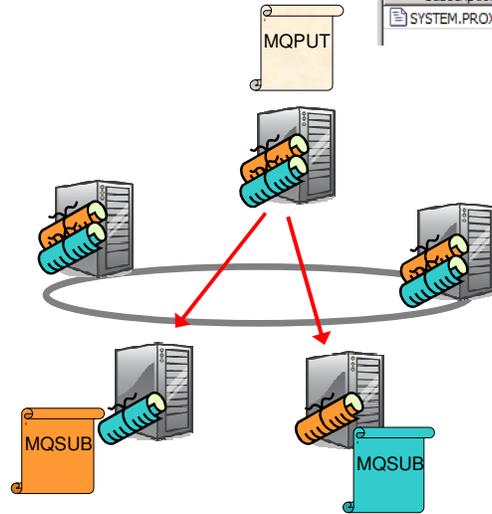
Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- Pub/sub clusters and hierarchies provide different benefits. You should choose the appropriate topology that gives you the benefits you need.
- Pub/sub clusters are very similar to the Message Broker collectives topology in that you have any-to-any connectivity, a fully-connected set. In fact the queue managers are connected together with all-to-all connectivity which means that all the queue managers in the cluster will have channels running to all the other queue managers in the cluster. This may limit the scalability of this solution, with the number of channels being the gating factor. It is however a very available solution since to one queue manager is a bottle-neck or single point of failure for the cluster. It also provides the same benefits of clustering that you get for queued clusters, that of reduces administration definitions.
- Hierarchies are very similar to, and interoperable with, WebSphere MQ V6 Queued Pub/Sub hierarchies. They have interconnection only between parent and child queue managers and so the loss of one queue manager, or connectivity to it, can cause a break in the link for other queue managers. Less availability therefore, but more scalable as less channels are needed for each queue manager. It is less flexible and definitions (channels and transmit queues) need to be set up manually.

Capitalware's MQ Technical Conference v2.0.1.3

Pub/Sub Clusters



Subscription name	Topic string	Type
SYSTEM.PROXY.QM2.DEMO	Price/Fruit/Apples	Proxy

TOPIC attributes

CLUSTER
SUBSCOPE
PUBSCOPE
PROXYSUB

```

Command Prompt - runmqsc TEST1
Starting MQSC for queue manager TEST1.
DEFINE TOPIC(APPLES)
  TOPICSTR('Price/Fruit/Apples')
  CLUSTER(DEMO)

DISPLAY SUB(*) SUBTYPE(PROXY) ALL
1 : DISPLAY SUB(*) SUBTYPE(PROXY) ALL
AMQ8096: websphere MQ subscription inquired.
SUBID(41405120514D3120202020202020204F5786482000F02)
SUB(SYSTEM.PROXY.QM2.DEMO.Price/Fruit/Apples)
TOPICSTR(Price/Fruit/Apples) TOPICOBJ( )
DEST(SYSTEM.INTER.QMGR.PUBS) DESTOMGR(QM2)
DESTCLAS(PROVIDED) DURABLE(YES)
SUBSCOPE(ALL) SUBTYPE(PROXY)
    
```

Capitalware's MQ Technical Conference v2.0.1.3

N O T E S

- A pub/sub cluster is a cluster of queue managers, with the usual CLUSRCVR and CLUSSDR definitions, but that also contains a TOPIC object that has been defined in the cluster.
- With a cluster you have "any-to-any" connectivity. There are direct links between all queue managers in the cluster. This provides good availability for the delivery of messages, if one route is unavailable, there may well be another route to deliver the messages to the target subscription.
- With a TOPIC object defined in the cluster, an application connected to one queue manager in the cluster can subscribe to that topic or any node in the topic tree below that topic and receive publications on that topic from other queue managers in the cluster.
- This is achieved by the creation of proxy subscriptions on the queue managers in the cluster, so that when a publication to the topic in question happens on their queue manager, they know to forward it to the appropriate other members of the cluster.
- You can view these proxy subscriptions through the same commands we saw earlier. By default proxy subscriptions are not shown to you because the default value for SUBTYPE is USER. If you use SUBTYPE(ALL) or SUBTYPE(PROXY) you will see these subscriptions.
- There are a few attributes that are specifically related to Distributed Publish/Subscribe. PUBSCOPE and SUBSCOPE determine whether this queue manager propagates publications to queue managers in the topology (pub/sub cluster or hierarchy) or restricts the scope to just its local queue manager. You can do the equivalent job programmatically using MQPMO_SCOPE_QMGR / MQSO_SCOPE_QMGR.
- PROXYSUB is an attribute that controls when proxy subscriptions are made. By default it has value FIRSTUSE and thus proxy subscriptions are only created when a user subscription is made to the topic. Alternatively you can have the value FORCE which means proxy subscriptions are made even when no local user subscriptions exist.

Capitalware's MQ Technical Conference v2.0.1.3

Questions?



Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

WebSphere Application Server Considerations

Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

WebSphere Application Server

Publish/Subscribe Settings for the WebSphere MQ Provider

- **WebSphere Application Server supports different JMS Provider types**
 - ▶ Default Provider (SIB)
 - ▶ Foreign Providers (@&#\$\$%!)
 - WebSphere MQ
 - Native awareness of specific MQ features
- **Configuring WMQ Publish/Subscribe**
 - ▶ Connection Factories
 - ▶ Topics
 - ▶ Differences between:
 - WebSphere MQ V6 “queued” pub/sub
 - WebSphere Message Broker V6 “queued” pub/sub
 - WebSphere MQ V7.x “native” pub/sub

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- WebSphere Application Server supports different JMS Provider types:
 1. There is a Default Provider that is part of the application server itself, built on System Integration Bus (SIBus) technology
 2. There is support for what are called Foreign Providers. In principle, any JEE-compliant provider can be plugged into WebSphere Application Server, although support for these is very limited; from the standpoint of administration, the application server sees these essentially as a “black box”
 3. Finally, there is support for the Websphere MQ Provider.
- It is possible for WebSphere MQ to be defined as a “black box” foreign provider, but the native awareness of specific MQ features that WebSphere Application Server possesses enables customers to administratively utilize many of the proprietary features of WebSphere MQ, particularly features of the publish/subscribe components, while still enabling applications to be written in a manner that is portable to other application servers.
- WebSphere Application Server supports Message Driven Beans (MDBs) via ListenerPorts and ActivationSpecs (the latter utilizing the WebSphere MQ JCA 1.5 Resource Adapter). Either can be used with the WebSphere MQ publish/subscribe components, although be aware that ListenerPorts are deprecated.
- The remainder of this presentation will focus on configuring the WebSphere MQ JMS features that are related to Publish/Subscribe, including Connection Factories and Topics.
- Also covered will be differences between WebSphere MQ V6 “queued” Publish/Subscribe, WebSphere Message Broker V6 “queued” Publish/Subscribe, and WebSphere MQ V7.x “native” Publish/Subscribe. While these V6 products are no longer supported, I want to mention them as part of this presentation because V6-specific properties still appear on the WebSphere Application Server console, and you want to be aware of which of these properties still have meaning when using WebSphere MQ V7, and which are now obsolete.

Capitalware's MQ Technical Conference v2.0.1.3

Connection Factory Properties (General)

[JMS providers](#) > [WebSphere MQ messaging provider](#) > [Connection factories](#) > [MyCF](#)

A unified JMS connection factory can be used to create JMS connections to both queue and topic

Configuration

General Properties

Administration

Scope
[Node=thinkNode01]

Provider
[WebSphere MQ messaging provider]

* Name
[MyCF]

* JNDI name
[myJNDI]

Description
[]

Connection

Queue manager
[MyQmgr]

Transport
[bindings, then client]

* Hostname
[localhost]

Port
[1414]

Server connection channel
[SYSTEM.DEF.SVRCONN]

- The **General** tab covers standard ConnectionFactory details for both Point-to-Point and Publish/Subscribe
- The portion shown here has no Pub/Sub-specific properties

Capitalware's MQ Technical Conference v2.0.1.3

N

- The General tab covers standard ConnectionFactory details for both Point-to-Point and Publish/Subscribe
- The portion shown here has no Pub/Sub-specific properties

O

T

E

S

Capitalware's MQ Technical Conference v2.0.1.3

Connection Factory Properties (General) (continued)

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- This slide, and those that follow, require some explanation. Specifically, the slides that follow will highlight various items of interest on the WebSphere Application Server configuration console, with the "text bubbles" showing the WebSphere MQ JMS Connection Factory and Topic properties that correspond to the item indicated.
- The "text bubbles" will be one of three different colors:
 - **GREEN** indicates features that are applicable to WebSphere MQ V7 Publish/Subscribe. In some cases they may also have been applicable to WebSphere MQ V6, and/or WebSphere Message Broker V6 Publish/Subscribe, although the meaning of some may be different with V7 than with these earlier products. These cases will be pointed out.
 - **YELLOW** indicates features that are applicable to WebSphere MQ V6 Publish/Subscribe and/or WebSphere Message Broker V6 Publish/Subscribe. If specified for ConnectionFactories or Topic objects that are used with WebSphere MQ V7 Publish/Subscribe, they will be ignored.
 - **RED** indicates features that are applicable, not to Publish/Subscribe, but to point-to-point producers and consumers. Because ConnectionFactory definitions can be used to describe either TopicConnections or QueueConnections, there can be some confusion regarding which properties are applicable to Publish/Subscribe. One goal of this presentation is to make it clear which ConnectionFactory properties are related to Publish/Subscribe, and which are not.
- Starting at the top: ClientID (CLIENTID MQ property) can be set to whatever you like. It is used to uniquely identify a durable subscription consumer, to enable a connection to a durable subscription to be resumed.
- Cloned Durable Subscriptions (CLONESUP) are a very useful feature that permits two or more instances of the same durable topic subscriber to run simultaneously, providing horizontal scalability to high-volume subscribers. To utilize this, each instance must a) specify the same ClientID, and b) run in a separate Java virtual machine (JVM). Be aware that running two or more instances of the same durable topic subscriber simultaneously contravenes the JMQ 1.1 specification, although this may not be an issue for you, it does not require the JMS code to be written in a non-portable manner.
- Provider Version (PROVIDERVERSION) is very important, as the choice you make here will dictate whether the consumer will run in "native" (V7) mode or in "migration" (V6) mode. This in turn may influence how you choose to specify the PSMODE queue manager property, and can restrict access to new WMQ V7 features, such as Conversation Sharing, Async Put, Read-ahead, etc.

Capitalware's MQ Technical Conference v2.0.1.3

Connection Factory Properties (Advanced)

JMS providers > WebSphere MQ messaging provider > Connection factories > MyCF > Advanced properties

These properties are used to configure advanced functionality when using a JMS connection factory to create connections.

Configuration

General Properties

Message compression

Compress message headers

Compression algorithm for message payloads
NONE

Temporary destinations

WebSphere MQ model queue name
SYSTEM.DEFAULT.MODEL.QUEUE

Temporary queue prefix

Temporary topic prefix

Connection consumer

Retain messages, even if no matching consumer is available

Polling interval
5000 milliseconds

Rescan interval
5000 milliseconds

Maximum batch size
10

TEMPMODEL

TEMPQPREFIX

TEMPTOPICPREFIX

MSGRETENTION

POLLINGINT

RESCANINT

MSGBATCHSZ

V7

V6

N/A

Capitalware's MQ Technical Conference v2.0.1.3

- N**
- Remember that in this and subsequent slides, **Red** balloons refer to properties that are related to the point-to-point messaging domain, and **NOT** related to publish/subscribe.
- O**
- Under "Temporary destinations", "WebSphere MQ model queue name" (TEMPMODEL) and "Temporary queue prefix" (TEMPQPREFIX) only apply when the destination is a queue, and so are not applicable to the publish/subscribe messaging domain. "Temporary topic prefix" (TEMPTOPICPREFIX) is new with WMQ V7, and does apply to the publish/subscribe domain. An application can create a TemporaryTopic object by using the createTemporaryTopic() method of a Session object. The temporary topic remains until the connection ends or the application explicitly deletes the temporary topic by using the TemporaryTopic.delete() method. When an application creates a temporary topic, the WMQ JMS classes will create a topic with the form "TEMP/<TEMPTOPICPREFIX>/<unique_id>"; if TEMPTOPICPREFIX is not specified, the generated topic string will be of the form "TEMP/<unique_id>".
- T**
- Under "Connection consumer", "Rescan interval (RESCANINT) only applies when the destination is a queue, and so is not applicable to the publish/subscribe messaging domain. "Polling interval" (POLLINGINT) only applies to WMQ V6. "Maximum batch size" (MSGBATCHSZ) also only applies to WMQ V6, but be aware that a new V7 property, READAHEADALLOWED, can be used to perform a similar function, and will be discussed on a later slide.
- E**
- S**

Capitalware's MQ Technical Conference v2.0.1.3

Connection Factory Properties (Broker)

JMS providers > WebSphere MQ messaging provider > Connection factories > MyCF > Broker properties

These properties are used to configure publish-subscribe functionality when using the JMS resource to create

Configuration

General Properties

Queues

- Broker control queue (SYSTEM.BROKER.CONTROL.QUEUE) → BROKERCONQ
- Broker publication queue (SYSTEM.BROKER.DEFAULT.STREAM) → BROKERPUBQ*
- Broker subscriber queue (SYSTEM.JMS.ND.SUBSCRIBER.QUEUE) → BROKERSUBQ
- Broker connection consumer subscription queue (SYSTEM.JMS.ND.CC.SUBSCRIBER.QUEUE) → BROKERCCSUBQ

Capabilities

Version

- Version 1 broker → BROKERVER
- Version 2 broker

Specify where message selection occurs

→ MSGSELECTION

Legend: V7 (Green), V6 (Yellow), N/A (Red)

Capitalware's MQ Technical Conference v2.0.1.3

NOTES

- On the "Broker properties" page, under "Queues", are several queues that apply to the queued publish/subscribe interface, and so only apply to WMQ V6 or WMB V6 (with one exception). "Broker control queue" (BROKERCONQ) is the name of the queue where API control messages like "RegSub", "RegPub", etc are sent. "Broker publication queue" (BROKERPUBQ) is the stream queue in WMQ V6 that is targeted by this ConnectionFactory. "Broker subscriber queue" (BROKERSUBQ) and "Broker connection consumer subscription queue" (BROKERCCSUBQ) identify the name of the subscriber queue(s) to be used.
- Under "Capabilities" we find several settings that some customers have found confusing. In the "Version" box we see the option to choose between a "Version 1 broker" and a "Version 2 broker" (BROKERVER), as well as "Specify where message selection occurs" (MSGSELECTION). These are all applicable to the queued pub/sub interfaces of WMQ V6 and WMB V6, and so do not apply to WMQ V7 unless the queued pub/sub interface is being used by this ConnectionFactory. It is probably easier to think of the "Version 1" and "Version 2" brokers as "RFH1" (WMQ) and "RFH2" (WMB) brokers; the "versions" really apply to this, and not to any specific version numbers of the products. The last item, "Specify where message selection occurs" (MSGSELECTION), only applies to the queued pub/sub interface, and indicates where message selection it to take place; "Client" says that message selection is to be done by the client, and only applies when the version corresponds to an "RFH1" (WMQ) broker, while "Broker" says that message selection is to be done by the broker, and only applies when the version corresponds to an "RFH2" (WMB) broker.
- Notice that one of the queues listed (Broker publication queue) is indicated as also applying to WMQ V7 pub/sub. The corresponding property (BROKERPUBQ) is tagged with an "*" because this property serves a different purpose in WMQ V7 than it does in WMQ V6, as the next slide will discuss in more detail.

Capitalware's MQ Technical Conference v2.0.1.3

JMS and Topic Tree Isolation

General Properties

Queues

Broker control queue

Broker publication queue
FRUIT.ANCHOR

Broker subscriber queue

Broker connection consistency subscription queue

Capabilities

Version

Version 1 broker

Version 2 broker

Specify where message selection is made

In broker

BROKERPUBQ

```
strncpy (ObjDesc.ObjectName,
        "FRUIT.ANCHOR",
        MQ_TOPIC_NAME_LENGTH);
ObjDesc.ObjectString.VSPtr = "Apples";
ObjDesc.ObjectString.VSLength = MQVS_NULL_TERMINATED;
```

V7

V6

N/A

Capitalware's MQ Technical Conference v2.0.1.3

NOTES

- Think back to the earlier slide that discussed Topic Tree Isolation; the idea behind it is that a developer (or a business unit) may want their application(s) to be insulated (or "isolated") from subsequent changes to the topic tree structure that would affect the topic string structure they chose to use. Topic Tree Isolation enables this to be accomplished by using a Topic Object definition as an index into the topic tree, isolating the portion of the topic tree from that point down from changes elsewhere in the topic tree structure.
- To utilize this feature, both a topic object name and a topic string must be specified. But the JMS specification does not know anything about topic objects; they are a WMQ-specific feature.
- When you think about it, we already had a way to isolate topic structures from one another; that was through the use of separate streams. With WMQ V6 pub/sub, separate streams could be used to isolate topic name spaces from one another.
- WMQ V7 does not support the concept of streams, but it does allow for topic tree isolation through the use of topic objects (which, incidentally, is the approach used by the V7 queued pub/sub interface to simulate the behavior of streams for backwards compatibility). And since "Broker publication queue" (BROKERPUBQ) was the way in WMQ V6 to reference a specific stream queue, it follows that using this same property in WMQ V7 to refer to a topic object would enable a JMS publisher or subscriber to accomplish something similar – reference a portion of the topic tree in isolation from other portions of the tree. And that's what this slide is showing; "Broker publication queue" is referencing a topic object (FRUIT.ANCHOR, which acts as an index into the topic tree for the topic string that is specified in the corresponding JMS (not WMQ) Topic object definition.
- There are two important things to note here. First, older versions of the WMQ V7 Infocenter did not reflect this use of the BROKERPUBQ, and says if it is used it will be ignored when connecting to a V7 queue manager in V7 mode; this has been corrected in current versions of the Infocenter. Second, there was a defect that causes this approach to topic tree isolation not to work when originally released (in fixpack 7.0.1.3). APAR IZ88852 (in fixpack 7.0.1.6) is needed to make this behave properly.

Capitalware's MQ Technical Conference v2.0.1.3

Connection Factory Properties (Broker) (continued)

Subscription store: (SUBSTORE)

Durable subscription state refresh interval: milliseconds (STATREFRESHINT)

Subscription cleanup level: (CLEANUP)

Subscription cleanup interval: milliseconds (CLEANUPINT)

Subscription wildcard format: (WILDCARDFORMAT)

V7
V6
N/A

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- "Subscription store" (SUBSTORE), "Durable subscription state refresh interval" (STATREFRESHINT), "Subscription cleanup level" (CLEANUP) and "Subscription cleanup interval" (CLEANUPINT) all refer to the method for deregistering ("cleaning up") non-durable subscriptions in WMQ and WMB V6 when the subscriber lost its connection. These properties have no effect in WMQ V7, which can now immediately detect when the subscriber loses its connection.
- "Subscription wildcard format" (WILDCARDFORMAT) is used to specify the wildcard format the subscriber will be using; either character wildcards only (for consistency with applications that previously used WMQ V6 pub/sub) or topic level wildcards (which are used by WMB V6 and WMQ V7).

Capitalware's MQ Technical Conference v2.0.1.3

Topic Properties

Topics > WebSphere MQ messaging provider > myTopic
Topic destinations provided for publish and subscribe messaging by the WebSphere administrative objects to manage topic destinations for the WebSphere MQ mess

Configuration

General Properties

Administration

Scope
Node=thinkNode01

Provider
WebSphere MQ messaging provider

+ Name
myTopic

+ JNDI name
myJNDI3

Description

WebSphere MQ topic

+ Topic name
Apples

Broker durable subscription queue
As connection

Broker durable subscriber connection consumer queue
As connection

Broker publication queue
As connection

Broker publication queue manager
As connection

TOPIC

V7
V6
N/A

N
O
T
E
S

- For the JMS Topic object definition, the only property we'll focus on here is the "Topic name" field. Here you specify the topic string (NOT the topic object name), and the topic string entered here will either be:
 - The fully-qualified topic string
 - The topic string relative to the topic string that is defined by the topic object that was referenced in the ConnectionFactory (using the BROKERPUBQ property).

Topic Properties - Advanced

[Topics](#) > [WebSphere MQ messaging provider](#) > [myTopic](#) > [Advanced properties](#)
View and set the advanced properties of a WebSphere MQ destination

Configuration

General Properties

Delivery

Persistence
As set by application

Priority
As set by application

Specified priority
0

Expiry

As set by application

Specified expiry
0

Message format

Coded character set identifier
1208

Use native encodings

Integer encoding
Normal

Decimal encoding
Normal

Floating point encoding
IEEENormal

Append RFH version 2 headers to messages sent to this destination

Capitalware's MQ Technical Conference v2.0.1.3

- N**
- The Configuration tab covers message delivery and format options. These are analogous to what you would see in a point-to-point environment.

O

T

E

S

Capitalware's MQ Technical Conference v2.0.1.3

Topic Properties (Advanced) - Continued

Optimizations

Asynchronously send messages to the queue manager
 → PUTASYNCALLOWED

Read ahead, and cache, non-persistent messages for consumers
 → READAHEADALLOWED

Read ahead consumer close method
 → READAHEADCLOSEPOLICY

V7
 V6
 N/A

Capitalware's MQ Technical Conference v2.0.1.3

N
 O
 T
 E
 S

- "Optimizations" are all new with WMQ V7, and apply only when the message producer or consumer is client-connected.
- "Asynchronously send messages to the queue manager" (PUTASYNCALLOWED). Indicates whether or not message producers are allowed to use asynchronous puts to publish messages to the topic.
- "Read ahead, and cache, non-persistent messages for consumers" (READAHEADALLOWED). Indicates whether or not message consumers are allowed to use read ahead to get nonpersistent messages from this destination into an internal buffer before receiving them
- "Read ahead consumer close method" (READAHEADCLOSEPOLICY). If read ahead is allowed, this option is used to indicate what is to happen to messages in the internal read ahead buffer awaiting delivery to an asynchronous message listener when the message consumer is closed. "Wait for all cached messages to be delivered" says that all messages in the internal read ahead buffer are to be delivered to the asynchronous message listener before returning. Otherwise, only the current message listener invocation completes before returning, potentially leaving messages in the internal read ahead buffer, which are then discarded.

Capitalware's MQ Technical Conference v2.0.1.3

Questions?



Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

Migration Considerations

Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

Publish/Subscribe Queue Manager Properties

- **A Number of new Queue Manager Properties were introduced with WebSphere MQ V7.0**
 - ▶ The following are not new, but were formerly specified in the *Broker* stanza of the *qm.ini* file
 - **PSNPMMSG** (*DiscardNonPersistentInputMsg*) - Controls how the queued publish/subscribe interface should handle failed attempts to process a non-persistent input message
 - **PSNPRES** (*DLQNonPersistentResponse* and *DiscardNonPersistentResponse*) - Controls how the queued publish/subscribe interface should handle failed attempts to send a response message being generated as a result of a non-persistent input message that could not be processed
 - **PSRTYCNT** (*MaxMsgRetryCount*) - Controls the number of times the queued publish/subscribe interface will reattempt to process failures encountered when attempting to process a command message put under syncpoint.
 - **PSSYNCP** (*SyncPointIfPersistent*) - Controls whether the queued publish/subscribe interface processes command messages (e.g. RegPub, RegSub) under syncpoint
 - ▶ We won't spend time on these – but want you to be aware of this change
- **We will spend time discussing these two very important properties:**
 - ▶ **PSMODE** - Controls whether the Publish/Subscribe engine for a queue manager is enabled or disabled, and if the former, which modes of operation are supported.
 - ▶ **PSCLUS** – (Added in V7.1) - Controls whether the queue manager will participate in publish/subscribe activity across any clusters in which it is a member
- **Let's explore these last two options in more detail**

Capitalware's MQ Technical Conference v2.0.1.3

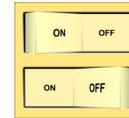
The PSMODE Queue Manager Property

- **New Queue Manager Property with WebSphere MQ V7.0**
 - ▶ "Publish/Subscribe Mode"
 - ▶ Three Options
 - COMPAT, ENABLED, DISABLED
 - Some have said the definition of these lacks clarity
- **What do these options actually mean?**
 - ▶ **PSMODE(COMPAT)**
 - The publish/subscribe engine is started
 - The queued publish/subscribe interface is NOT started
 - This setting intended for compatibility with WebSphere Message Broker V6
 - ▶ **PSMODE(ENABLED)**
 - The publish/subscribe engine is started
 - The queued publish/subscribe interface is also started
 - This is the queue manager's initial default value
 - ▶ **PSMODE(DISABLED)**
 - Neither the publish/subscribe engine or the queued publish/subscribe interface are started



Capitalware's MQ Technical Conference v2.0.1.3

The PSCLUS Queue Manager Property



- **New Queue Manager Property with WebSphere MQ V7.1**
- **Publish Subscribe clusters allow a topic space to span multiple queue managers**
- **Doing so in large or stretched deployments may result in unacceptable overhead**
- **For existing large point-to-point clusters the recommendation has been to avoid suddenly introducing clustered Topic objects**
 - ▶ But prior to V7.1 there was no way to enforce this – often leading to problems
- **So in V7.1 a new queue manager attribute controls whether or not the QM will participate in pub/sub clustering**
 - ▶ PSCLUS (ENABLED/DISABLED)
- **Disables the definition of cluster topic objects, and the sending/receiving of proxy subscriptions.**
- **Ideally set on every queue manager if no pub sub to be used**
 - ▶ However, configuring at least full repositories gives majority of protection
 - Disables the 'everybody learns about everybody' aspect of a Pub/Sub cluster.

Capitalware's MQ Technical Conference v2.0.1.3

Questions?



Capitalware's MQ Technical Conference v2.0.1.3

Multicast

Capitalware's MQ Technical Conference v2.0.1.3

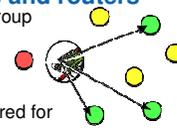
This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

Multicast – Technical Overview

IP Multicast is a low level form of Pub/Sub implemented in NICs and routers

- ▶ Receivers register their interest in receiving data on a pre-defined set of multicast group addresses)
 - For IPv4 this is 224.0.0.0 to 239.255.255.255 (RFC3171)
- ▶ Sender send datagrams to the multicast address
- ▶ Network cards/ routers make copies of data and send to receivers who have registered for an address



Multicast

- Multicast can be more efficient than traditional unicast pub/sub, scaling to a high number of subscribers



Unicast

- But Multicast is more complex to set up than unicast
 - ▶ Routers need to be configured to pass multicast traffic
 - ▶ ISPs starting to enable routers – allowing multicast over the internet

- Basic multicast is unreliable – no retries, persistence etc. - messages can be lost
- Reliable multicast uses sequence numbers and (negative) acknowledgements (ACKS/NAKs), allowing receivers to request messages to be replayed
 - ▶ Publisher & subscriber speed mismatches can lead to broadcast storms & flooding
 - NAK suppression and aggregation is used to circumvent
- Reliable protocols not interoperable across vendors - 'PGM' provides common denominator and is supported by routers

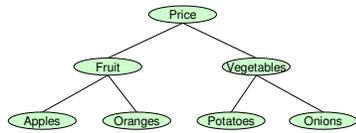
Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

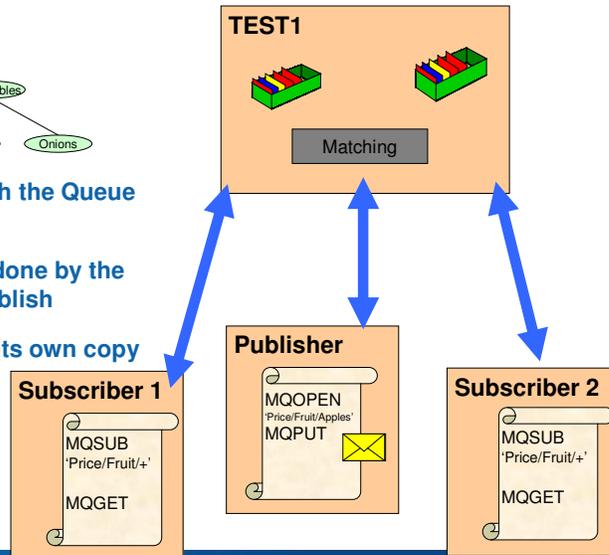
- IP Multicast is a form of pub /sub implemented at a low level in network cards and routers
 - Receivers register their interest in receiving data on a particular multicast address (pre-defined set of IP addresses)
 - For IPv4: 224.0.0.0 to 239.255.255.255 (RFC3171)
 - Sender send datagrams to the multicast address
 - Network cards/ routers make copies of data and send to all registered receivers
- Multicast can be more efficient than traditional unicast pub/sub and scales to a high number of subscribers because the source to sends packet only once
 - If it needs to be delivered to a large number of receivers the nodes in the network take care of replicating the packet to reach multiple receivers only when necessary.
- Routers need to be configured to pass multicast traffic, which can make multicast more complex to set up than unicast
 - ISPs slowly starting to enable their networks – allowing broadcasters like the BBC to stream media using multicast over the internet
- The base multicast transport is unreliable – no retries, persistence etc. This means data can be lost. But reliable protocols exist
 - Protocols like PGM and RMM use sequence numbers and (negative) acknowledgements (ACKS/NAKs) to allow receivers to request messages to be replayed if they are missed
 - Reliable protocols not interoperable – most multicast vendors have invented their own
 - Publisher and subscriber speed mismatches can lead to broadcast storms and flooding. NAK suppression and aggregation is used to circumvent this problem.

Capitalware's MQ Technical Conference v2.0.1.3

Standard MQ Publish/Subscribe



- Messages flow through the Queue Manager
- Subscriber matching done by the Queue Manager on publish
- Each subscriber sent its own copy of the message



Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

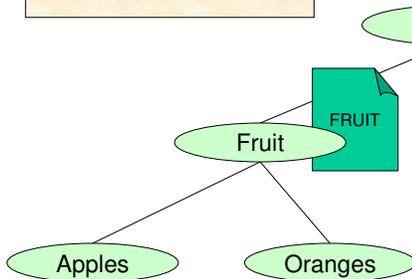
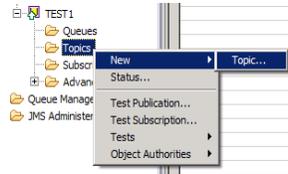
- WebSphere MQ Pub/Sub as we've discussed it up to this point is centered around a hierarchical Topic Tree.
- Publishers advertise data they want to share by assigning it a name that maps to a particular point in the topic tree.
- Subscribers register interest in data published to particular points in the topic tree.
- A message is published; it flows through the Queue Manager, who performs the matching of subscribers to publications at the time of publication
- Each matching subscriber is then sent its own copy of the message
- In this model, the matching engine is in the queue manager. At publish time this is where the subscribers are identified and copies of the publication messages are sent to each subscriber's queue
- In WebSphere MQ V7.1, Publish/Subscribe is extended to include the Multicast transport

Capitalware's MQ Technical Conference v2.0.1.3

Making it flow via multicast – modify your topic tree

DEFINE TOPIC
ALTER TOPIC
DELETE TOPIC
DISPLAY TOPIC

SYSTEM.BASE.TOPIC



```

Command Prompt - runmqsc TEST1
Starting MQSC for queue manager TEST1.
DEFINE TOPIC(FRUIT)
  TOPICSTR('Price/Fruit') MCAST(ENABLED) COMMINFO(MC)

DISPLAY TOPIC(FRUIT)
AMQ8633: Display topic details.
TOPIC(FRUIT)                                TYPE(LOCAL)
TOPICSTR('Price/Fruit')                     DESCR( )
CLUSTER( )                                  DURSUB(ASAPARENT)
PUB(ASAPARENT)                              SUB(ASAPARENT)
DEFPST(ASAPARENT)                          DEFPRTY(ASAPARENT)
DEFPRESP(ASAPARENT)                        ALTDTE(2008-02-26)
ALTTIME(15.05.23)                           PMSGDLV(ASAPARENT)
NPMSGDLV(ASAPARENT)                         PUBSCOPE(ASAPARENT)
SUBSCOPE(ASAPARENT)                         PROXYSUB(FIRSTUSE)
WILDCARD(PASSTHRU)                          MDURMDL( )
MNDURMDL( )                                  MCAST(ENABLED)
COMMINFO(MC)
    
```

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- For WebSphere MQ publications to flow over a multicast network, you use Topic Object definitions to indicate those points in the topic tree that are to be made available to multicast subscribers. This is done by specifying MCAST(ENABLED) on the Topic Object definition.
- The only other property on the Topic Object definition that is related to multicast is the COMMINFO attribute, which in the example here is specified as "COMMINFO(MC)". This identifies the name of a COMMINFO object that contains the comms-specific details.

Capitalware's MQ Technical Conference v2.0.1.3

Making it flow via multicast - COMMINFO

- New MQ object
- Defines the behavior of the multicast traffic
 - ▶ Group address
 - ▶ Reliability setting
 - ▶ Data conversion
 - ▶ Communication Monitoring
 - ▶ Queue Manager Bridging
 - ▶ Property propagation

```
DEFINE COMMINFO  
ALTER COMMINFO  
DELETE COMMINFO  
DISPLAY COMMINFO
```

```
Command Prompt - runmqsc TEST1  
Starting MQSC for queue manager TEST1.  
DEFINE COMMINFO(MC) GRPADDR(239.0.0.0)  
DISPLAY COMMINFO(MC)  
AMQ8861: Display comminfo details.  
COMMINFO(MC) TYPE(MULTICAST)  
DESCR( ) ALTDAT(2010-06-04)  
ALTTIME(18.45.42) BRIDGE(ENABLED)  
CCSID(AS PUB) COMMEV(DISABLED)  
ENCODING(AS PUB) GRPADDR(239.0.0.0)  
MCHBINT(2000) MCPROP(ALL)  
MONINT(60) MSGHIST(100)  
NSUBHIST(NONE) PORT(1414)
```

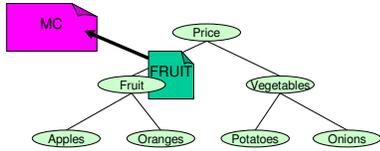
Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

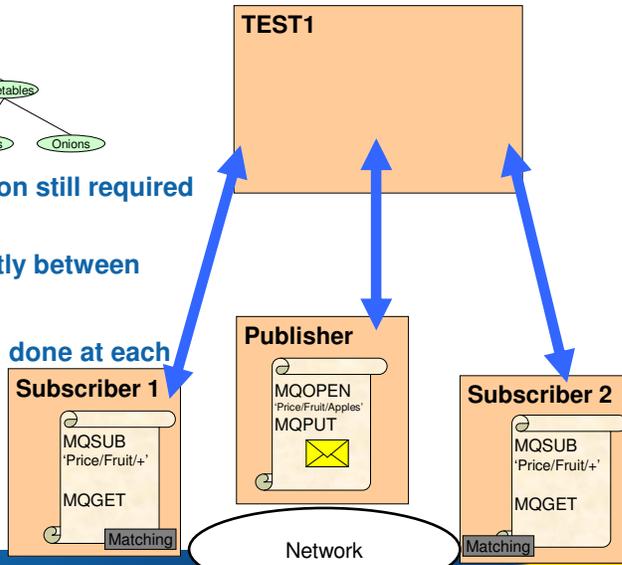
- Multicast support in WebSphere MQ V7.1 introduces a new MQ object type – the COMMINFO object. This object is used to define the behavior of the multicast traffic, and includes properties such as Group address, reliability and data conversion settings, and so on.
- The WebSphere MQ message descriptor (MQMD) is a fairly large structure (428 bytes) and often much of this structure is the same for every message. For small messages, the MQMD itself might represent most of the data traffic and can have a significant detrimental effect on the transmission rate. WebSphere MQ Multicast provides the ability to configure property propagation via the MCPROP attribute of the COMMINFO object definition. This allows you to specify which, if any, of the MQMD attributes are to be transmitted along with the message. By setting the value of this attribute to an appropriate level, you can control the size of the WebSphere MQ Multicast messages.
- As a side note, the underlying technology in WebSphere MQ V7.1 multicast support is the same as that in WebSphere MQ Low Latency Messaging (LLM). Note that this does NOT mean that the LLM product itself is embedded as part of WebSphere MQ – they are still separate products, although it is possible to bridge them together. A discussion of that is beyond the scope of this presentation.

Capitalware's MQ Technical Conference v2.0.1.3

MQ multicast Publish/Subscribe



- Normal MQ connection still required
 - ▶ Must be SHARECNV(>0)
- Messages flow directly between clients
- Subscriber matching done at each client.



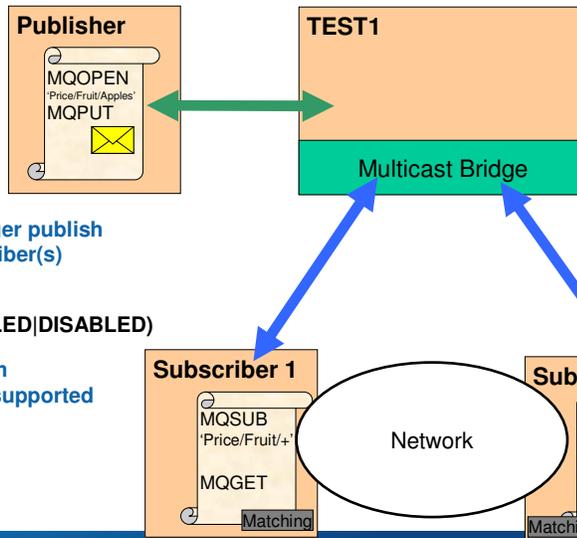
Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- To use MQ Multicast for Publish/Subscribe, a connection to the queue manager is still required – in order to discover the group address for publishing, for example. This uses a normal MQ SVRCONN connection, but note that SHARECNV(>1) must be specified.
- Multicast-related information is defined in a new MQ object called a COMMINFO object – which we discussed in a previous slide.
- Even though a connection to the queue manager is required, note that publication messages are not sent through the queue manager, but are instead published by the client directly to the group address and received directly by the subscribing clients from the group address.
- The matching engine is in the client. Matching must be done in the client to filter out topics on the group address that the application is not interested in because many topics can be on the same group address.

Capitalware's MQ Technical Conference v2.0.1.3

Bridging from Queue Manager to Multicast



- From Queue Manager publish to Multicast subscriber(s)
- COMMINFO object
 - ▶ BRIDGE(ENABLED|DISABLED)
- Only outbound from queue manager is supported

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

- Publications made by applications that are not using multicast can be bridged to multicast subscribers by a component in the queue manager.
- Bridging is enabled on the COMMINFO object using the BRIDGE field.
- When an application publishes on a topic whose COMMINFO object is configured as BRIDGE(ENABLED) then these messages are published out on the group address for that TOPIC/COMMINFO.
- Bridging is only available outbound from the queue manager.
- Bear in mind the multicast topic string length limitation of 255 characters when creating a topic to be bridged between MQ publishers and multicast subscribers.

Capitalware's MQ Technical Conference v2.0.1.3

N
O
T
E
S

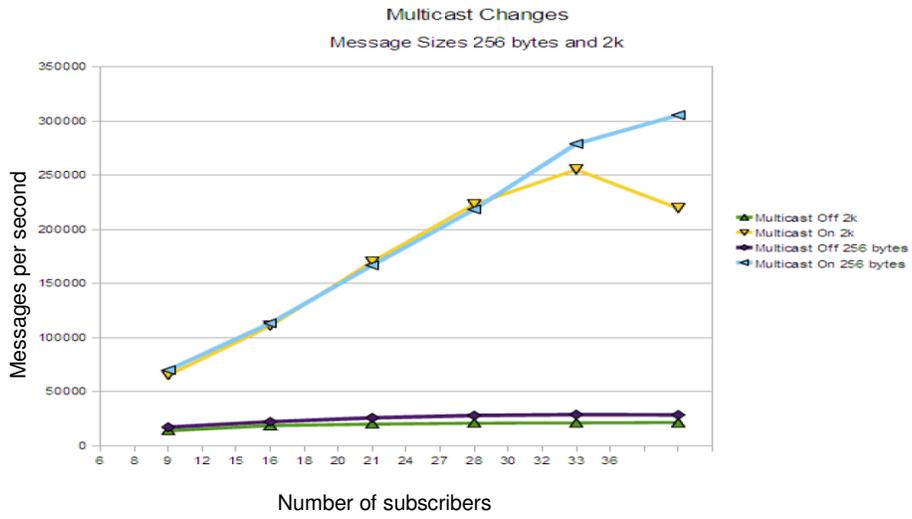
- WebSphere MQ V7.0 introduced the DISPLAY TPSTATUS command, enabling administrators to view the status of subscribers and publishers using a topic.
- Multicast publishers and subscribers also provide this status information, but in addition have one extra field, called MCASTREL, which is a pair of short and long term indicators showing the reliability (as a percentage) of the multicast transport. A value of 100% shows that all messages are being delivered successfully. A lower value indicates some issue with the transport that means some messages are not being delivered.
- To receive more detailed information about the problem you can turn on events which provide more detail. This is done using the COMMEV property on the COMMINFO object. This allows you to turn on events either all the time, or just when a problem is seen. COMMEV(EXCEPTION) will only write events when MCASTREL indicates reliability is 90% or less. Events will be written at an interval defined by the MONINT property on the COMMINFO object.
- The information that is used by the queue manager to display the reliability of the multicast transport and the other status fields, periodically is sent from the clients (as the messages being monitored do not go through the queue manager). Clearly if the queue manager is not available to the clients and they are in the process of reconnecting, this information may be out of date.

Capitalware's MQ Technical Conference v2.0.1.3

This Slide Intentionally Left Blank

Capitalware's MQ Technical Conference v2.0.1.3

Performance

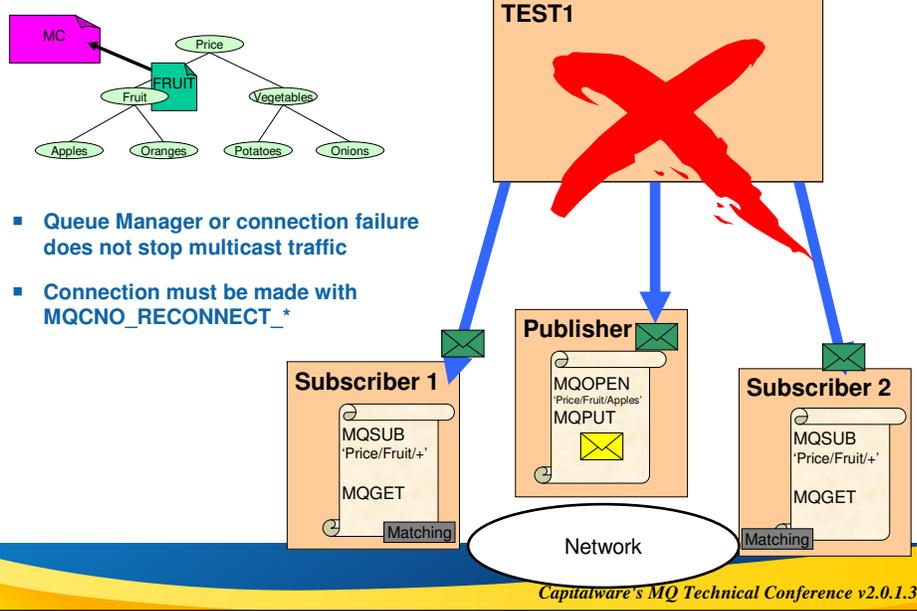


Capitalware's MQ Technical Conference v2.0.1.3

- N
O
T
E
S**
- Here is a brief look at the sort of difference using multicast can make when using it for a large fan-out to bigger numbers of subscribers.
 - In short, the more subscribers you have, the more messages can be sent per second as the publishing rate is independent of the number of subscribers.
 - More details can be found in the Performance Reports for WebSphere MQ V7.1 and V7.5.

Capitalware's MQ Technical Conference v2.0.1.3

MQ Multicast Publish/Subscribe High availability



N
O
T
E
S

- Applications that make use of the multicast transport can benefit from an availability standpoint – specifically because they will have the ability to keep on sending and receiving messages even if the queue manager they originally connected through is unavailable. This is possible because the messages are not going through the queue manager, but are flowing directly between publishers and subscribers.
- Note that this will only occur when using a reconnectable client - that is a client using one of the MQCNO_RECONNECT_* options. If the connection with the queue manager is lost this option will cause the client to try to reconnect to the queue manager in the background. During this reconnect process multicast traffic is able to continue.

Capitalware's MQ Technical Conference v2.0.1.3

Multicast and WebSphere MQ – What are the Benefits?

■ Reduced latency

- ▶ Much higher volumes than standard non-persistent messaging
- ▶ Messages do not pass through queues
- ▶ Peer to peer communication

■ Increased Scalability

- ▶ Additional subscribers cause no slow down
- ▶ Reduced network traffic

■ 'Fair delivery' of data

- ▶ Each subscriber 'sees' the data at the same time
- ▶ Fair delivery is critical to ensure that no recipient gains an advantage
- ▶ Multicast offers near simultaneous delivery

■ High availability

- ▶ Multicast uses the network so does not need a pub/sub engine to fan-out data
- ▶ Once a Topic is mapped to a group address there is no need for a Qmgr
 - Publishers and subscribers can operate in a "peer-to-peer" mode
 - Allows load to be reduced on Qmgr servers
 - Qmgr servers not a single point of failure

For detailed explanation
on how to set up and use
MQ V7.1 multicast



Capitalware's MQ Technical Conference v2.0.1.3

Questions?



Capitalware's MQ Technical Conference v2.0.1.3

Summary

Capitalware's MQ Technical Conference v2.0.1.3

Summary - WebSphere MQ Publish/Subscribe

- **Topic tree administration control**
 - ▶ TOPIC objects
 - Topic/Subtree default attributes
 - Security control point
- **Application Monitoring**
 - ▶ DISPLAY TPSTATUS
 - Publisher and subscriber status
 - ▶ DISPLAY SUBSTATUS
 - Additional subscriber status
 - ▶ DISPLAY CONN
 - Updated for pub/sub
- **Managed Destinations**
 - ▶ How to configure
- **No code change Publish/Subscribe**
 - ▶ QALIAS pointing at TOPIC
 - ▶ Admin SUB pointing to queue
- **Pub/Sub and WAS**
 - ▶ Connection Factory properties
 - ▶ Native vs Migration Mode considerations
- **Topologies**
 - ▶ Pub/Sub Clusters
 - ▶ Hierarchies
- **WebSphere MQ Multicast**
 - ▶ Publishers and Subscribers communicate directly
 - ▶ Interoperates with standard MQ Pub/Sub
 - ▶ Performance, HA advantages
 - ▶ COMMINFO object
 - Links to TOPIC object
 - Defines multicast behavior

Capitalware's MQ Technical Conference v2.0.1.3