IBM Software Group

WebSphere MQ V7.0

Overview of Publish / Subscribe

An IBM Proof of Technology

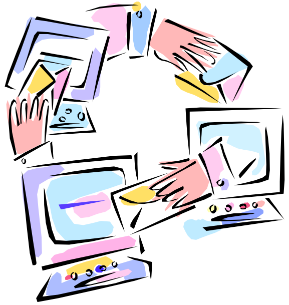
Unit Agenda

- Publish/Subscribe and WebSphere MQ
  - Advantages of Pub/Sub compared to point-to-point
  - Terminology - what are:
    - Topic objects, topic strings, topic space
    - Subscriptions, durable and non-durable
    - Destinations, managed and unmanaged
    - Publications, retained and otherwise
  - Topologies for pub/sub networks
  - Administration overview
- Lab 3 - Pub/Sub Administration using the WebSphere MQ Explorer

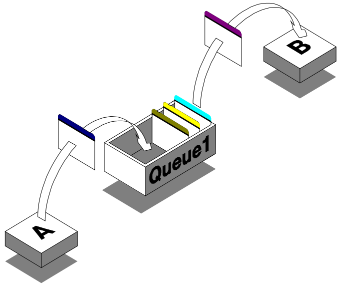
Point to Point Examples

- Post Card
  - Goes to just the person I send it to

- eMail
  - Might go to lots of people but I get to choose exactly who gets it



- Message Queuing
  - If I put a single message it will go to a single consumer

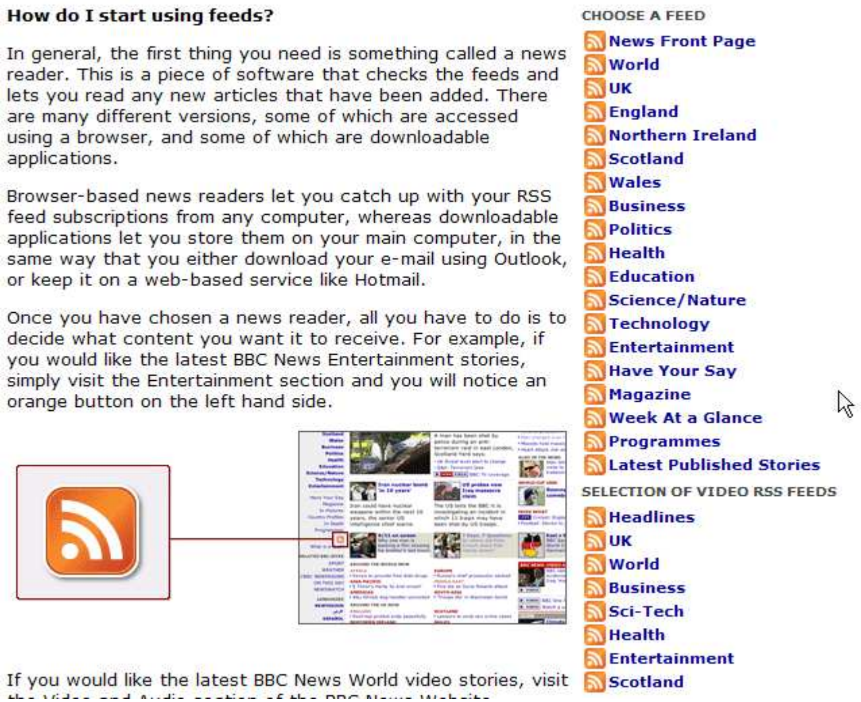


Publish / Subscribe Examples

- Magazine Publishing
  - In the US, over 10,000 titles published

- Airline Departure Boards
  - Boards might display (subscribe to)
    - All departures
    - Departures from this terminal
    - Departures by this airline

- RSS News Feeds



How do I start using feeds?

In general, the first thing you need is something called a news reader. This is a piece of software that checks the feeds and lets you read any new articles that have been added. There are many different versions, some of which are accessed using a browser, and some of which are downloadable applications.

Browser-based news readers let you catch up with your RSS feed subscriptions from any computer, whereas downloadable applications let you store them on your main computer, in the same way that you either download your e-mail using Outlook, or keep it on a web-based service like Hotmail.

Once you have chosen a news reader, all you have to do is to decide what content you want it to receive. For example, if you would like the latest BBC News Entertainment stories, simply visit the Entertainment section and you will notice an orange button on the left hand side.

If you would like the latest BBC News World video stories, visit

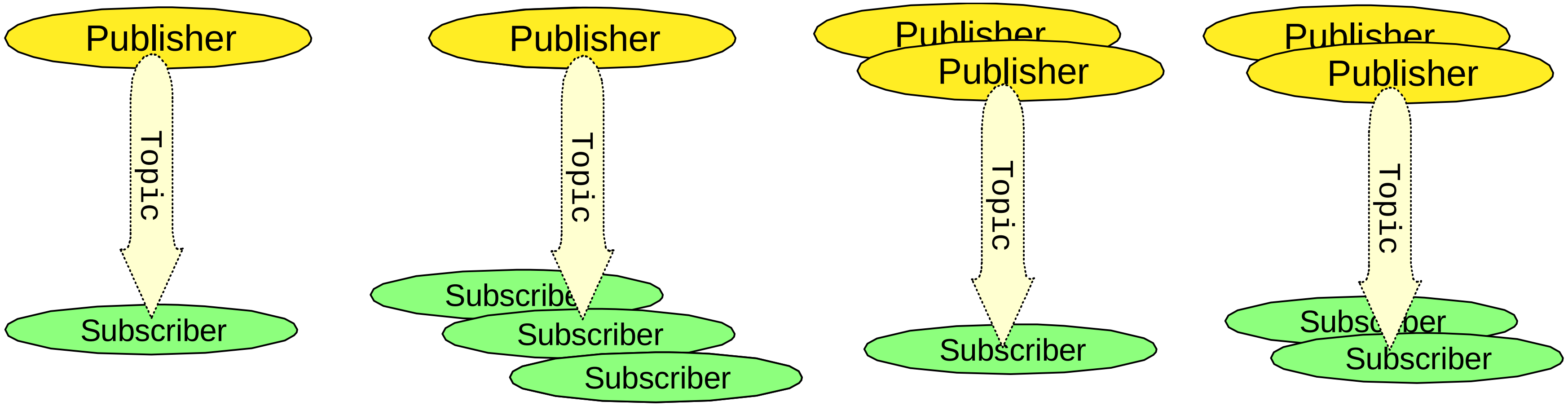
CHOOSE A FEED

- News Front Page
- World
- UK
- England
- Northern Ireland
- Scotland
- Wales
- Business
- Politics
- Health
- Education
- Science/Nature
- Technology
- Entertainment
- Have Your Say
- Magazine
- Week At a Glance
- Programmes
- Latest Published Stories

SELECTION OF VIDEO RSS FEEDS

- Headlines
- UK
- World
- Business
- Sci-Tech
- Health
- Entertainment
- Scotland

Loose coupling with Pub/Sub



WebSphere MQ Implementation of Publish/Subscribe

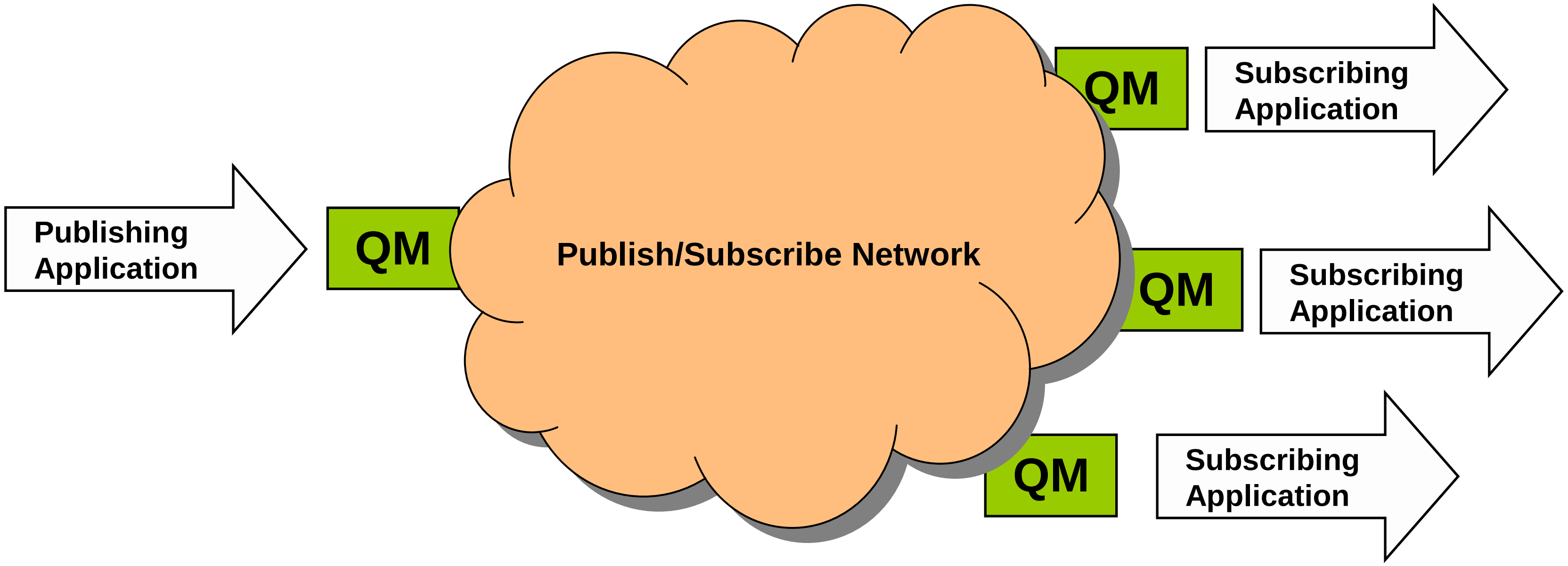
- Central concept is the *TOPIC STRING*

- Messages are “published” to a Topic string
  - Each act of publishing is to a single Topic string

- Subscribers are delivered messages that were published to the Topic string(s) they registered an interest in (subscribed to)
  - Subscribers may register interest in multiple topics
  - Wildcards can be used to filter topic names
  - Selectors can be used to filter message properties

- Topic strings can be created and secured administratively, or dynamically by simply being referenced by publishers and subscribers

Publish/Subscribe applications

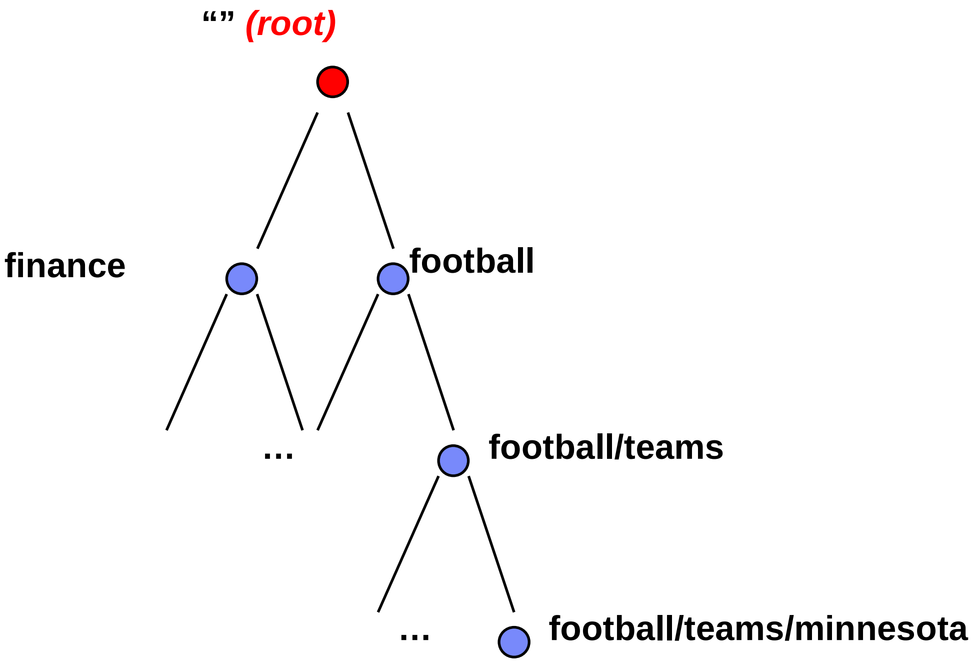


- Applications connect to local queue managers
- They can be different queue managers connected by a network.
- Can use JMS as well as any MQ API
- They are “connected” through a *topic string*

Concepts – Topic String

- A Topic *string* is an unlimited length string
- A topic string “connects” publishers and subscribers
- Has a “structure” and a semantics in it
  - The structure is supplied by the “/” slash character
  - Similar to directory structure on UNIX® or Windows®
    - e.g. ***”/news/sports/football/teams/minnesota”***
- The semantics are implied by the use of <u>wildcards</u> in subscriptions
  - “#” and ‘+’ are used as wildcard characters in subscriptions
    - e.g. ***”/news/+/football/#”***

Concepts – Topic Tree



- A topic tree is an internal representation of the topic hierarchy
- It has a root node at the very top
- It is implied from the complete set of topic strings in use - defined, published to, subscribed to.
- There is ***not necessarily*** a one-to-one mapping between topic *objects* and *nodes* in the tree

Concepts – Topic Object

- A Topic *Object* is a WebSphere MQ object that can be administered

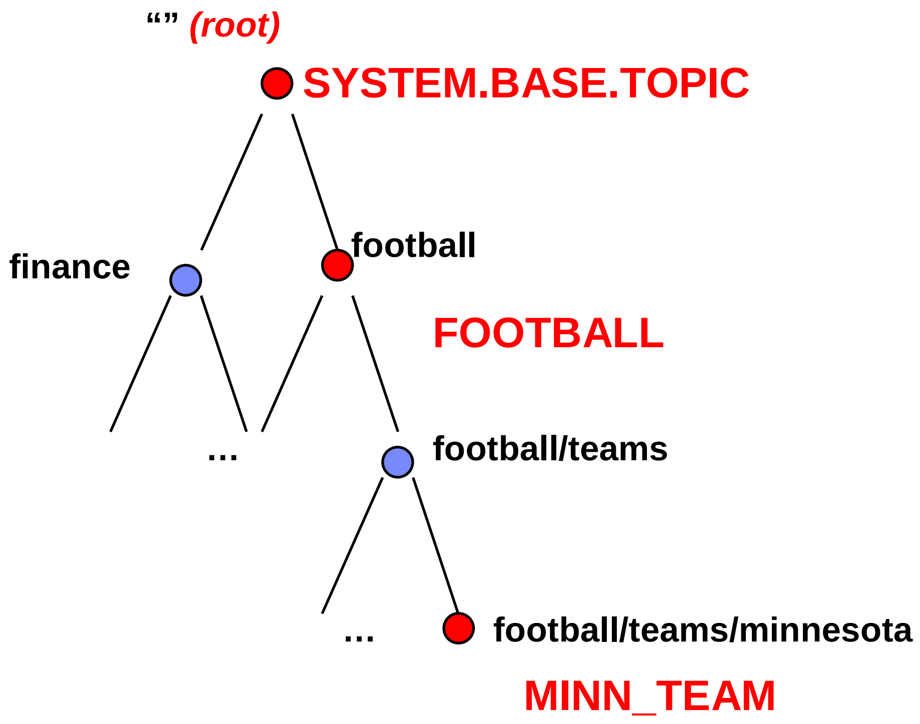
- A Topic Object has a name following the same rules as other WebSphere MQ objects (Queues etc)

- The Topic *String* is one of the properties of a Topic Object
  - Topic Objects can have other properties set

- Security can be applied to Topic Objects

- Application programs using the MQI can use the topic object as a full or partial alternative to topic string when publishing or subscribing
  - JMS programs would use the Destination object, which maps to a Topic or Queue object in WebSphere MQ

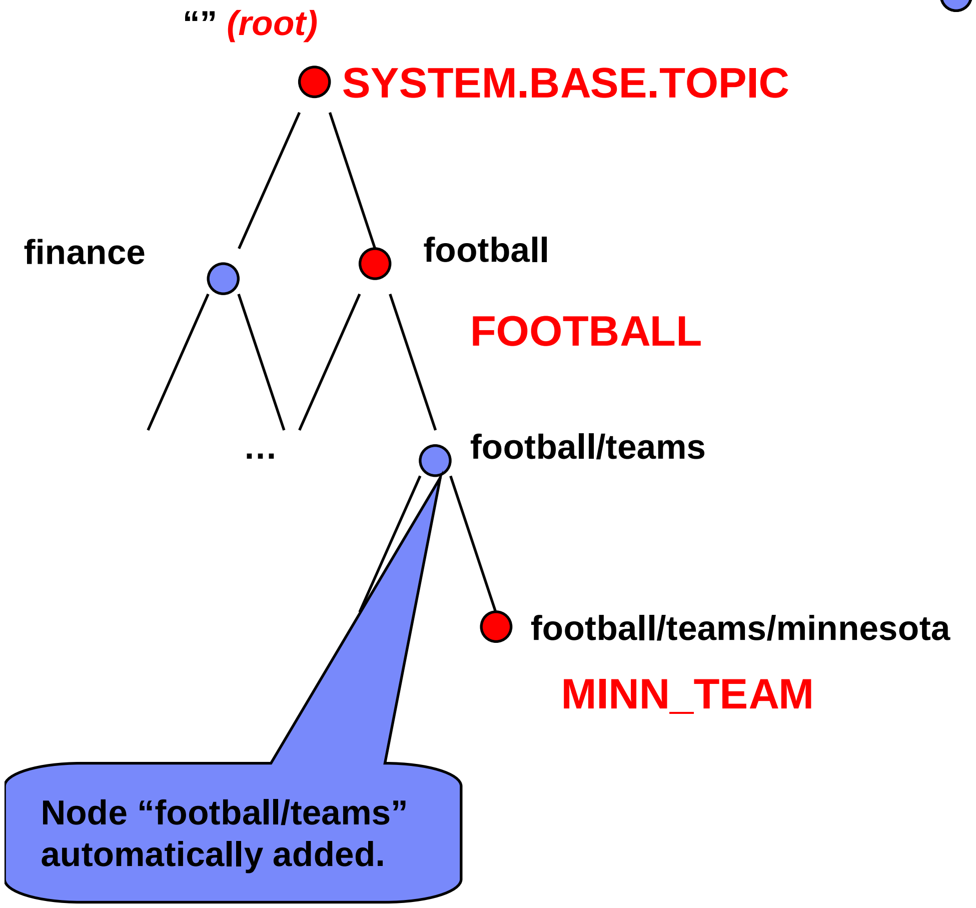
Concepts – Topic Tree



- Nodes which have topic objects defined for them can be thought of as *administered nodes*
- They are permanent parts of the hierarchy
- They can have properties specified for them
- They have topic object names, which need not reflect the topic strings

DEFINE TOPIC(FOOTBALL) TOPICSTR(‘football’)

Concepts – Topic Tree

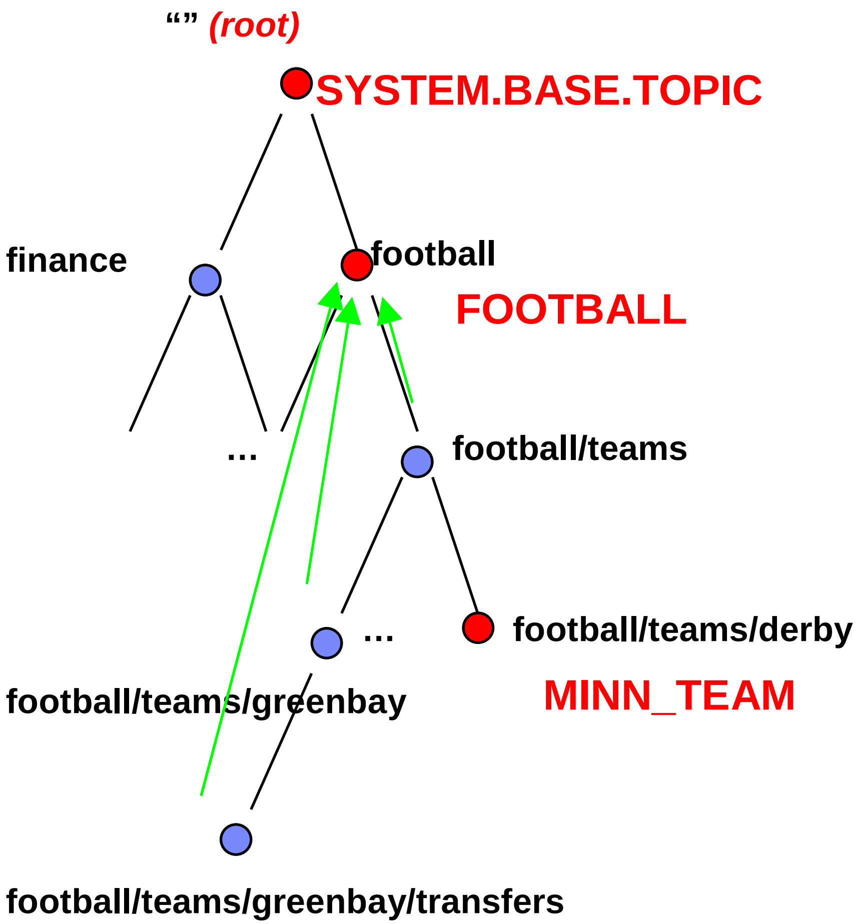


● **Admin node**

● **Non-admin node**

- The Topic Tree can be dynamically extended by a publisher and/or subscriber
  - Any missing parts of the tree hierarchy will be "filled in" when administered nodes are defined
- Dynamically added nodes in the Topic Tree inherit all properties from their parent, i.e. from an administered node above them

Concepts – Topic Tree



- 🔴 **Admin node**
- 🔵 **Non-admin node**

- Publishing or subscribing to a topic string that is not yet in the tree causes the tree to be extended
- They are temporary parts of the hierarchy and will be removed when no remaining publishers or subscribers exist
- These nodes inherit all properties from their parent, and so from an administered node above them

MQSUB(‘/football/teams/greenbay/transfers’)

Concept - Subscriptions

- A subscription is to a *topic string*
  - Wildcards can be used to set them up, but in essence they are to a topic string
  - Corresponds to a node in the topic tree
- Subscriptions can be *durable*, or not
  - Non-durable subscriptions mean messages are delivered to you only while you are connected
  - Messages for durable subscriptions are delivered even when disconnected, and can be retrieved when you next connect
- Subscription destinations can be *managed*, or not
  - With a managed destination a subscription queue is automatically created when you create a subscription, and is tidied up automatically in accordance with the durability of the subscription
  - With an unmanaged destination, you specify your own queue at the time the subscription is opened
- Subscriptions can be registered programmatically or administratively

Concept - Retained Publications

- Most recent message published on a given topic
  - Each node on the topic tree can have at most **one** Retained Publication
- Whether a publication is “retained” is determined at publish time by the application program
- A subscriber can request, at anytime, the current retained publication for a given topic
- Example *StockPrice.CurrentPrice* is a retained publication
  - I can be notified whenever it changes – ticker tape
  - Or I can request it at any time

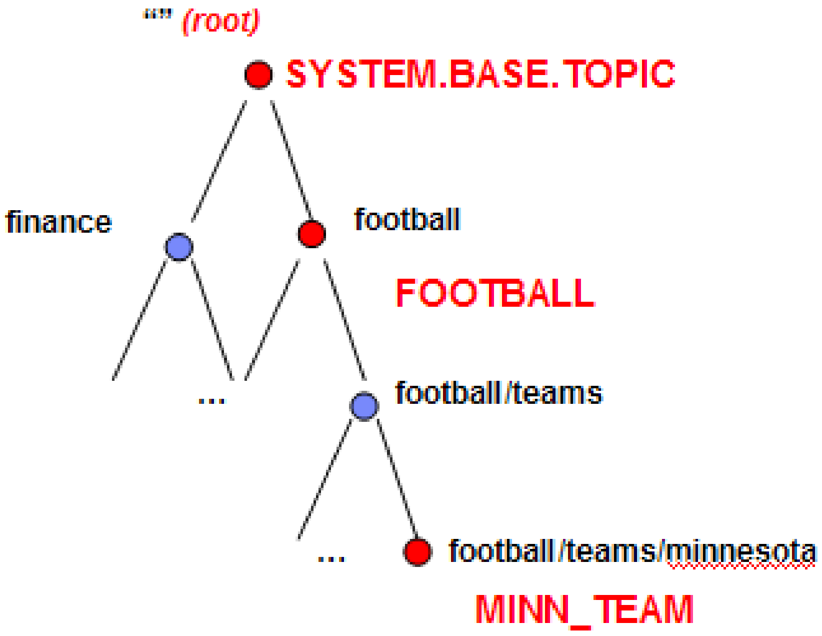
Administered Subscription

- Has a name conforming to usual MQ rules
  - Sample attributes
    - **TOPICSTR** – Topic string, basically unlimited length

      TOPICSTR('/football/teams/#')
    - **DEST** – Name of object to which messages for this subscription will be forwarded

      DEST(MY.QUEUE)
    - **DESTQMGR** – Remote queue manager to which messages are forwarded

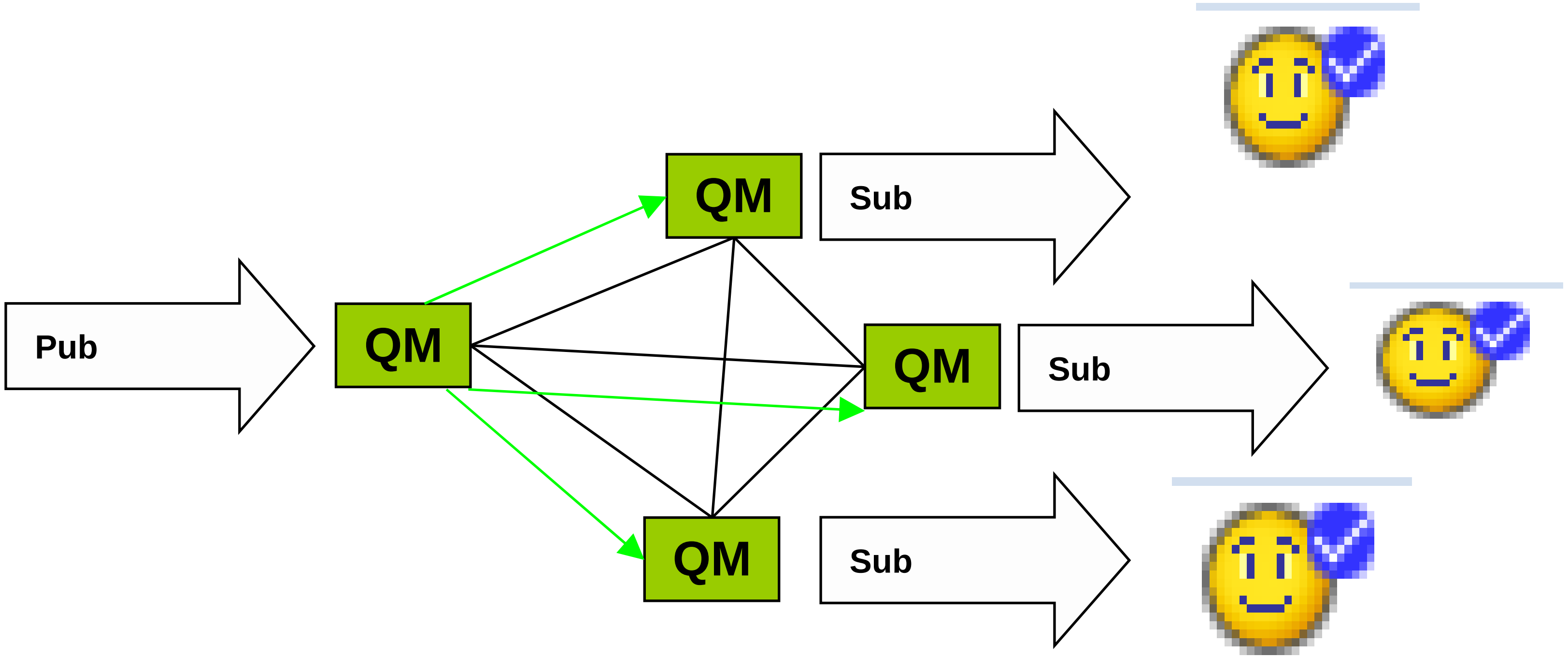
      DESTQMGR(REMOTE.QMGR)



Distributed Pub/Sub Topologies

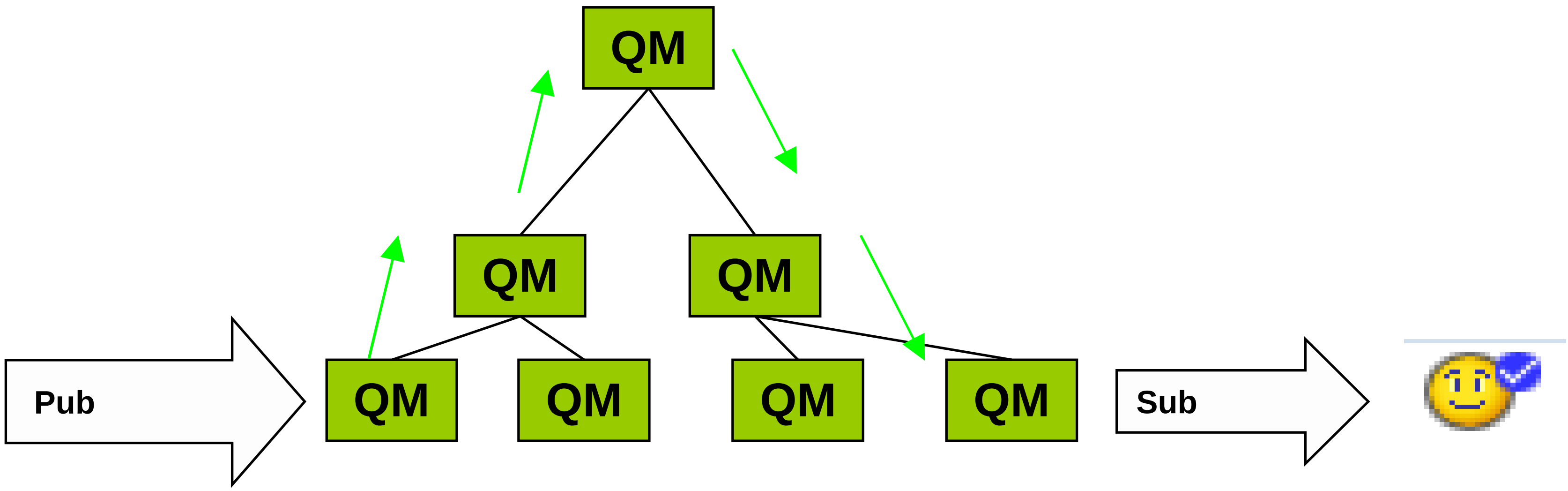
- Two types of distributed Pub/Sub topologies are supported
  - Publish/Subscribe *Clusters*
    - Queue managers are connected using a “network” or “mesh” connectivity model
    - MQ Clustering technology is used to provide connectivity between participating queue managers
    - Publications from any queue manager in the cluster may be subscribed to by any other
    - Enables multiple paths for publications, can eliminate single points of failure
  - Publish/Subscribe *Hierarchies*
    - Queue managers are arranged in a Parent/Child Hierarchy
    - Each Queue manager may have only one Parent
    - Queue managers may exchange Pub/Sub information directly between only their parent or children
    - This approach may require multiple hops and/or create single points of failure
- A Publish/Subscribe network can be created that combines both of these topologies

Example of a Publish/Subscribe Cluster

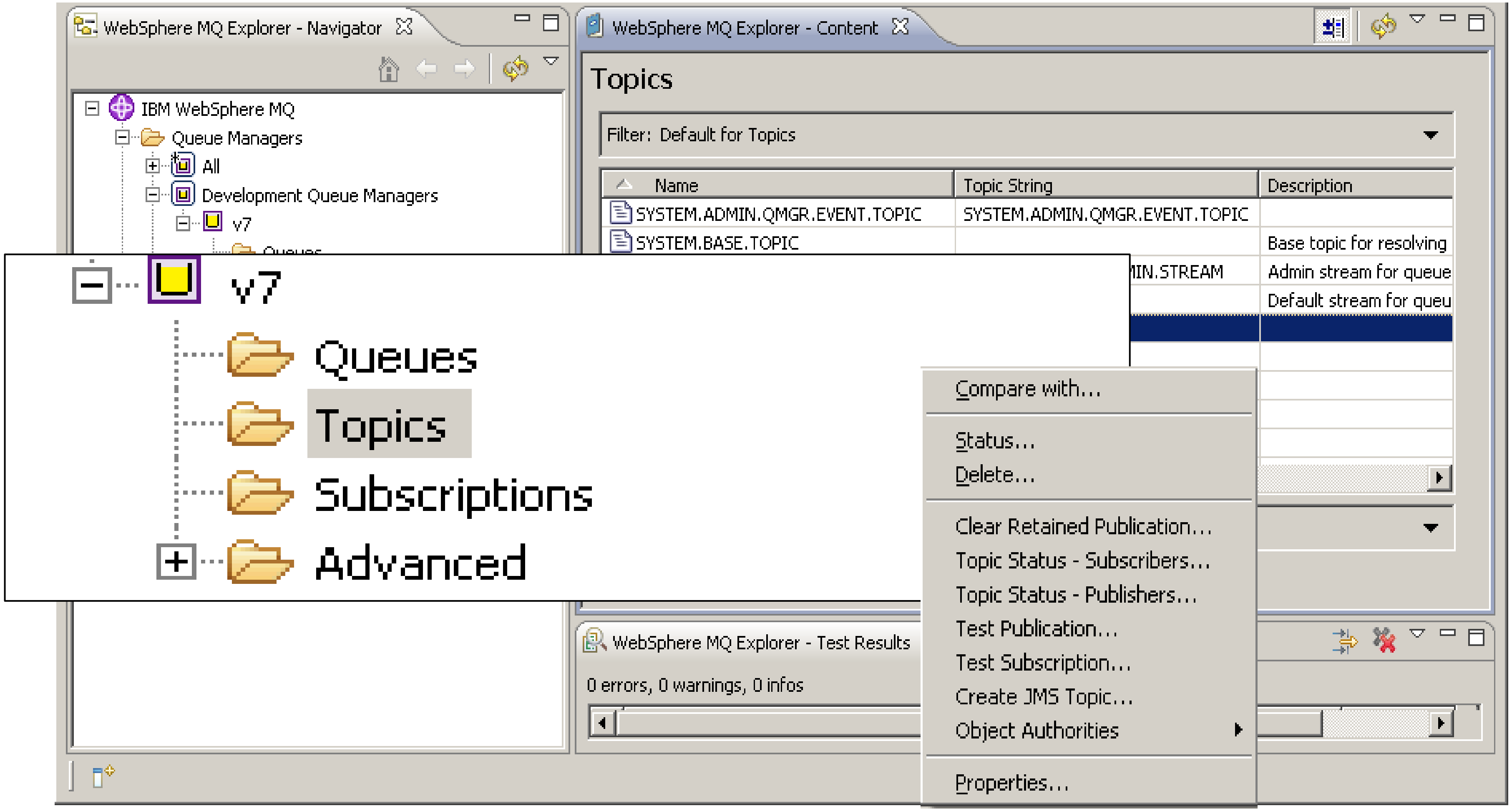


- **Consistent definitions in cluster**
- **Multiple routes across cluster**

Example of a Publish/Subscribe Hierarchy



Publish/Subscribe in the WebSphere MQ Explorer


WebSphere MQ Explorer - Navigator
IBM WebSphere MQ
Queue Managers
All
Development Queue Managers
v7
Queues
Topics
Subscriptions
Advanced
WebSphere MQ Explorer - Content
WebSphere MQ Explorer - Test Results
Compare with...
Status...
Delete...
Clear Retained Publication...
Topic Status - Subscribers...
Topic Status - Publishers...
Test Publication...
Test Subscription...
Create JMS Topic...
Object Authorities
Properties...

Topics

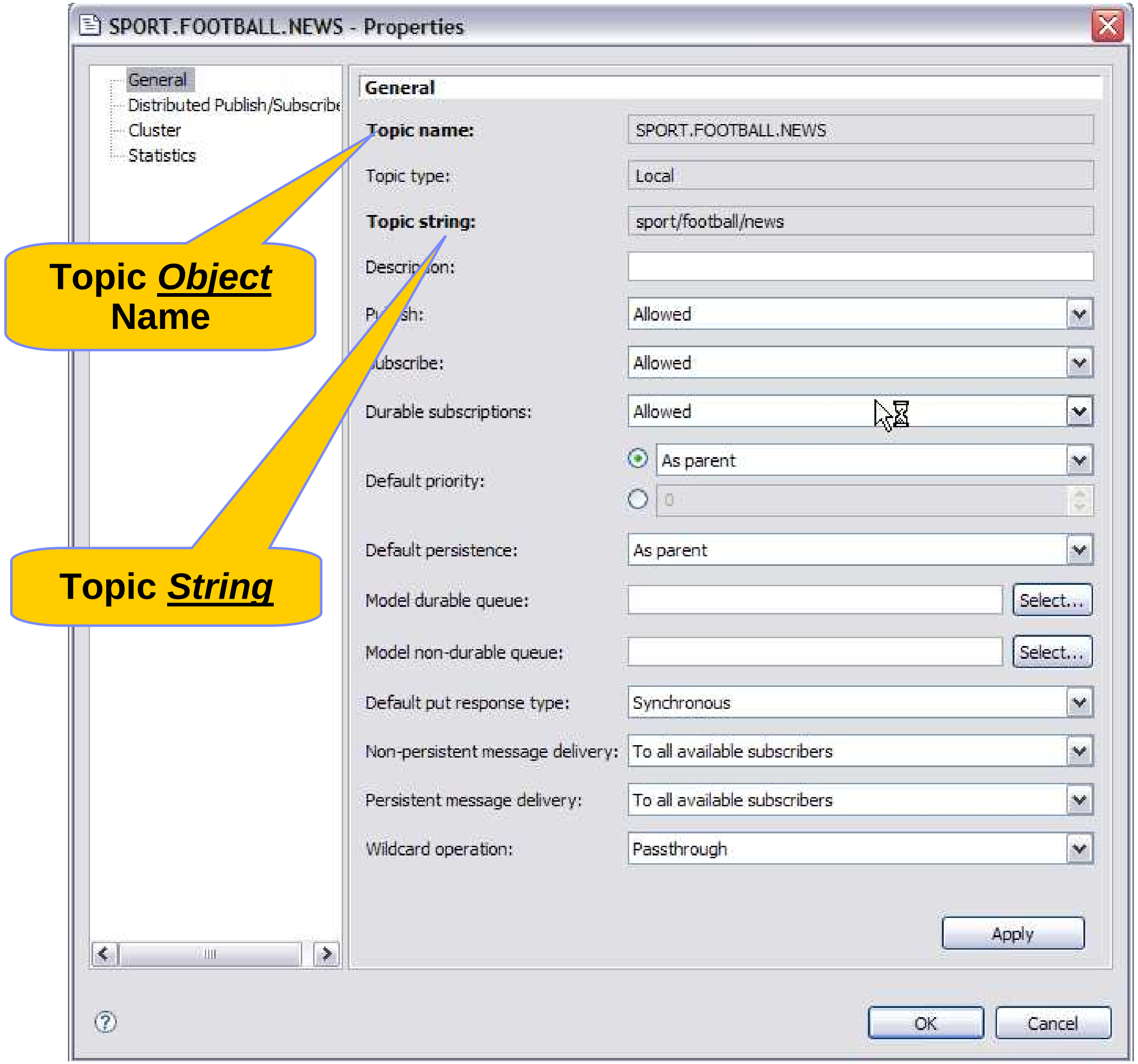
Filter: Default for Topics

Name	Topic String	Description
SYSTEM.ADMIN.QMGR.EVENT.TOPIC	SYSTEM.ADMIN.QMGR.EVENT.TOPIC	
SYSTEM.BASE.TOPIC		Base topic for resolving
	IN.STREAM	Admin stream for queue
		Default stream for queu

0 errors, 0 warnings, 0 infos

Topic Objects

- Topic Attributes are associated with a Topic ***Object***
- The Topic Object is a WebSphere MQ object that can be administered
- The Topic ***String*** is one of the properties of a Topic Object
- The topic string “connects” publishers and subscribers
- A Topic string is an unlimited length string
  - Topic Objects can have other properties set



Topic ***Object*** Name

Topic ***String***

SPORT.FOOTBALL.NEWS - Properties

General

Topic name: SPORT.FOOTBALL.NEWS

Topic type: Local

Topic string: sport/football/news

Description:

Publish: Allowed

Subscribe: Allowed

Durable subscriptions: Allowed

Default priority: ● As parent ○ 0

Default persistence: As parent

Model durable queue: Select...

Model non-durable queue: Select...

Default put response type: Synchronous

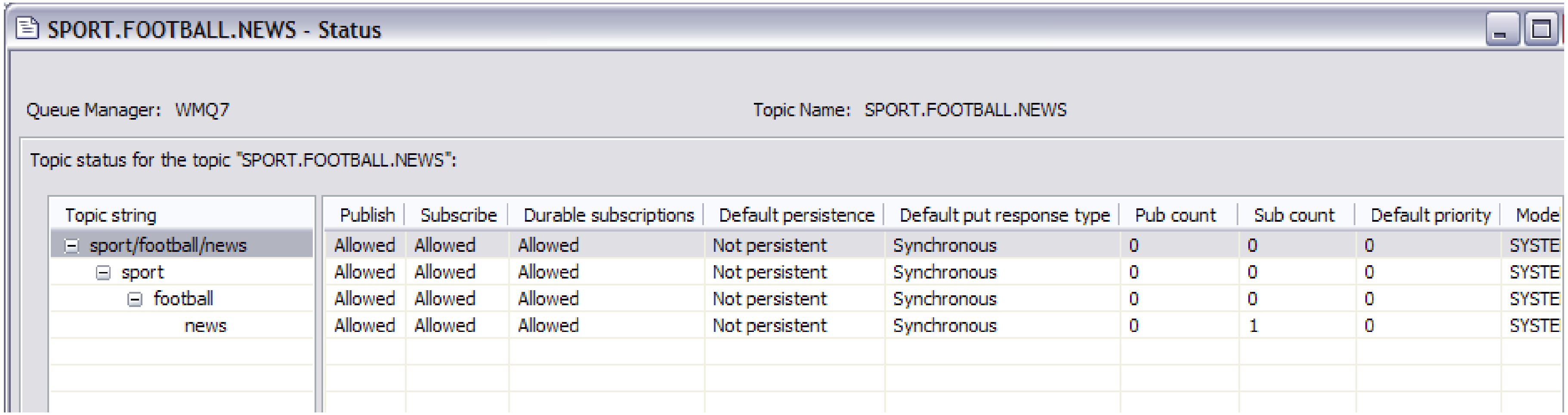
Non-persistent message delivery: To all available subscribers

Persistent message delivery: To all available subscribers

Wildcard operation: Passthrough

Topic Status – current usage of topics

- DISPLAY TPSTATUS(*topic-string*)
  - Topic attributes (TYPE(TOPIC))
    - Number of publishers and subscribers
    - Whether a retained publication exists
  - Subscriber info (TYPE(SUB)):
    - Last resume date of subscription and last message delivery time
    - Number of messages sent to this subscriber
  - Publisher info (TYPE(PUB)):
    - Last publish time
    - Number of messages published


SPORT.FOOTBALL.NEWS - Status

Queue Manager: WMQ7

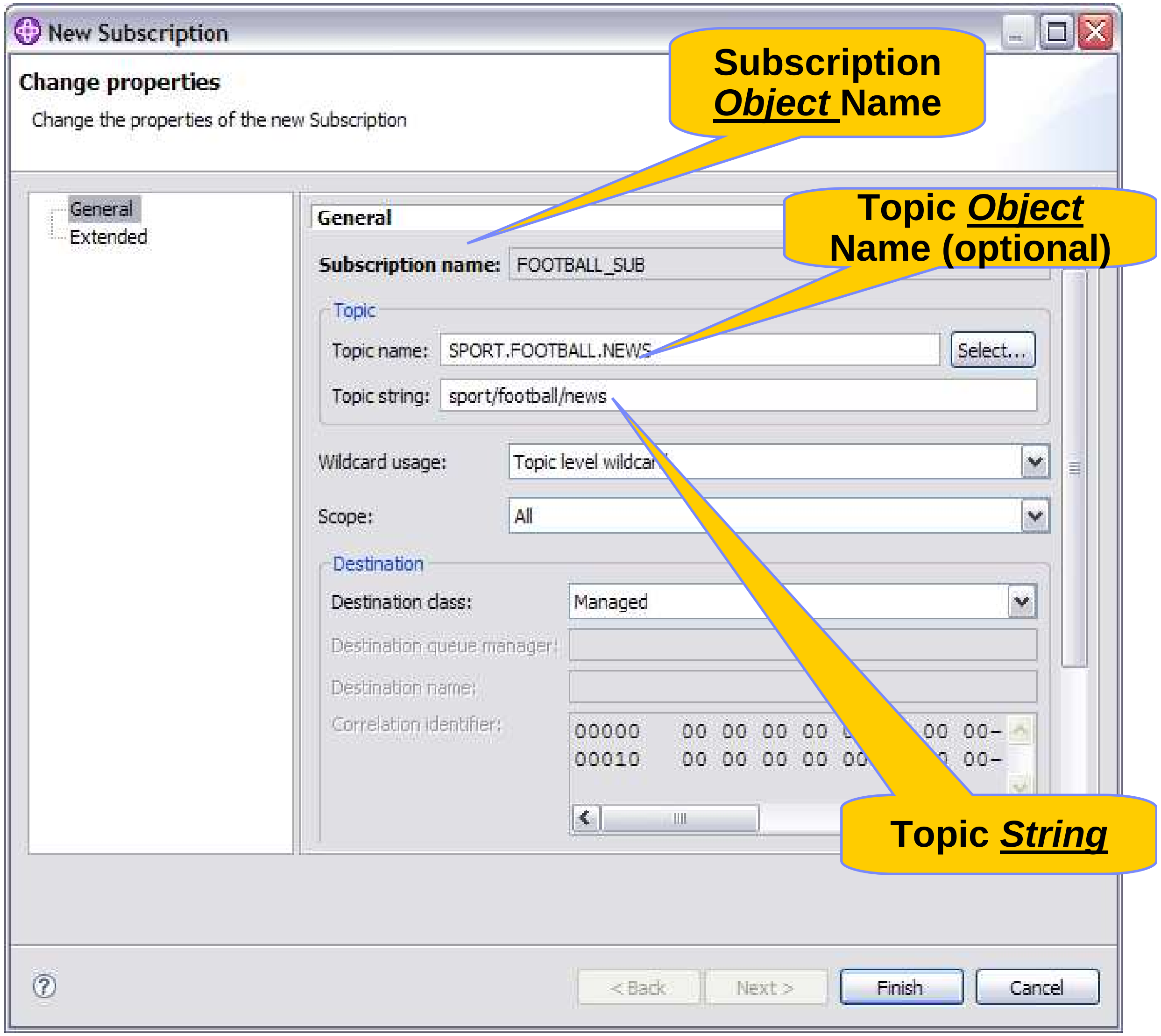
Topic Name: SPORT.FOOTBALL.NEWS

Topic status for the topic "SPORT.FOOTBALL.NEWS":

Topic string	Publish	Subscribe	Durable subscriptions	Default persistence	Default put response type	Pub count	Sub count	Default priority	Mode
⊟ sport/football/news	Allowed	Allowed	Allowed	Not persistent	Synchronous	0	0	0	SYSTE
⊟ sport	Allowed	Allowed	Allowed	Not persistent	Synchronous	0	0	0	SYSTE
⊟ football	Allowed	Allowed	Allowed	Not persistent	Synchronous	0	0	0	SYSTE
news	Allowed	Allowed	Allowed	Not persistent	Synchronous	0	1	0	SYSTE

Administered Subscriptions

- Subscriptions that are administratively created require a name
- A Topic ***Object*** can optionally be referenced
- The Topic ***String*** is required
- Wildcards may be used
- By definition, an administered subscriptions is ***durable***
- Administered subscriptions can use either managed or unmanaged destinations



New Subscription

Change properties

Change the properties of the new Subscription

General

Subscription name: FOOTBALL_SUB

Topic

Topic name: SPORT.FOOTBALL.NEWS

Topic string: sport/football/news

Wildcard usage: Topic level wildcar

Scope: All

Destination

Destination class: Managed

Destination queue manager:

Destination name:

Correlation identifier:

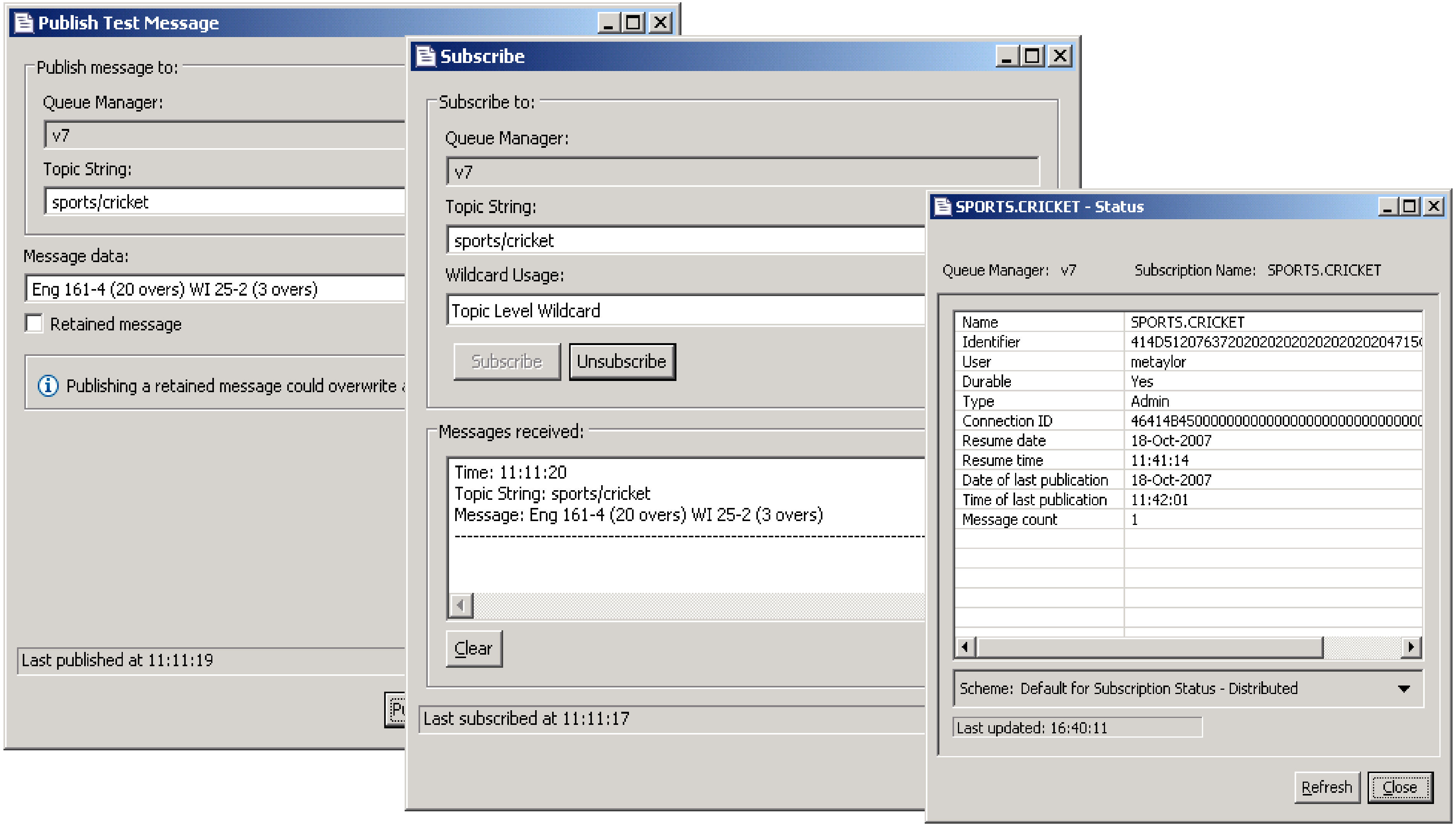
```
00000    00 00 00 00 0    00 00-
00010    00 00 00 00 00   0 00-
```

Subscription *Object* Name

Topic *Object* Name (optional)

Topic *String*

Testing Publish and Subscribe



Publish Test Message

Publish message to:

Queue Manager: v7

Topic String: sports/cricket

Message data: Eng 161-4 (20 overs) WI 25-2 (3 overs)

☐ Retained message

Publishing a retained message could overwrite a

Last published at 11:11:19

Subscribe

Subscribe to:

Queue Manager: v7

Topic String: sports/cricket

Wildcard Usage: Topic Level Wildcard

Messages received:

```
Time: 11:11:20
Topic String: sports/cricket
Message: Eng 161-4 (20 overs) WI 25-2 (3 overs)
--------------------------------------------------------------------------------
```

Last subscribed at 11:11:17

SPORTS.CRICKET - Status

Queue Manager: v7 Subscription Name: SPORTS.CRICKET

Name	SPORTS.CRICKET
Identifier	414D51207637202020202020202020202020204715
User	metaylor
Durable	Yes
Type	Admin
Connection ID	46414B450000000000000000000000000000000
Resume date	18-Oct-2007
Resume time	11:41:14
Date of last publication	18-Oct-2007
Time of last publication	11:42:01
Message count	1

Scheme: Default for Subscription Status - Distributed

Last updated: 16:40:11

Benefits of Publish/Subscribe with WebSphere MQ

- Applications are more loosely coupled than with point-to-point
  - No need to agree on queue names, number of receivers, etc
- Applications decide their message exchange patterns
  - One-to-one
  - One-to-many
  - Many-to-one
  - Many-to-many
- Messages are “published” to a Topic string
  - Which can be created dynamically
  - …or administratively for more control over security, etc
- Subscribers receive messages they registered an interest in
  - Subscribers may register interest in multiple topics
  - Wildcards can be used to filter topic names
  - Selectors can be used to filter message properties