

# Unit 19. Introduction to WebSphere MQ publish/subscribe

## What this unit is about

This unit covers the publish and subscribe capability of WebSphere MQ V7. It explains the role of the publishing application, the role of the subscribing application, the meaning of "topics" and different ways of defining publications using the MQSC command interface and WebSphere MQ Explorer.

## What you should be able to do

After completing this unit, you should be able to:

- Explain how WebSphere MQ supports a publish/subscribe environment
- Describe how to create the objects required in an WebSphere MQ publish/subscribe environment
- Explain the merits of publish/subscribe versus point-to-point messaging
- Design a publish/subscribe system incorporating multiple queue managers
- Use utilities to manage and monitor topics, publications, and subscriptions

## How you will check your progress

- Class discussion

## References

System Administration Guide

Publish/subscribe User's Guide

## Unit objectives

After completing this unit, you should be able to:

- Explain how WebSphere MQ supports a publish/subscribe environment
- Describe how to create the objects required in an MQ publish/subscribe environment
- Explain the merits of publish/subscribe versus point-to-point messaging
- Design a publish/subscribe system incorporating multiple queue managers
- Use utilities to manage and monitor topics, publications, and subscriptions

© Copyright IBM Corporation 2008

Figure 19-1. Unit objectives

WM203 / VM2032.0

### Notes:

## What is publish/subscribe?

Publish/subscribe is a term used to define an application model in which the provider of some information is decoupled from the consumers of that information.

- Providers of information need have no knowledge of consumers
- Consumers of information need have no knowledge of providers
- New providers/consumers can be added without disruption
- Providers of information are called **publishers**
- Consumers of information are called **subscribers**

© Copyright IBM Corporation 2008

Figure 19-2. What is publish/subscribe?

WM203 / VM2032.0

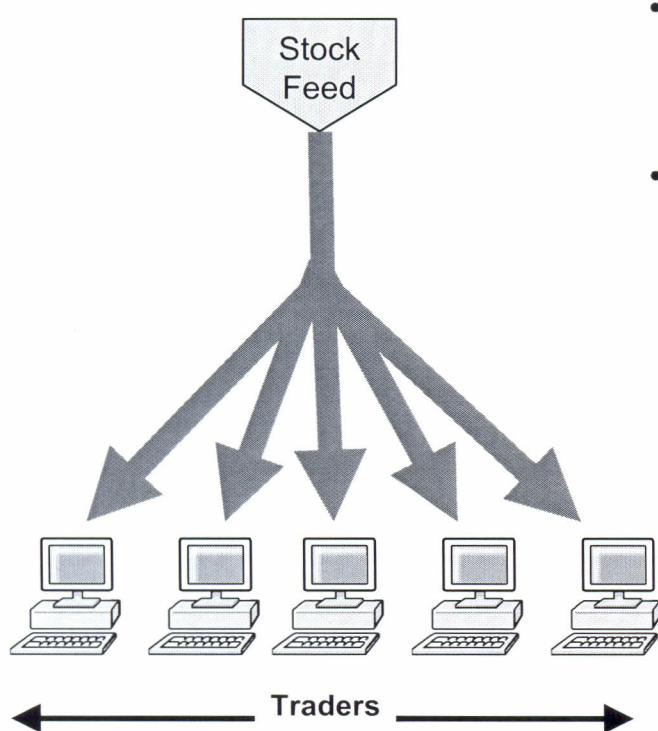
### Notes:

Publish/subscribe is the mechanism by which subscribers can receive information, in the form of messages, from publishers.

Publisher applications of data messages do not need to know the identity or location of the subscriber applications which receives the messages. Similarly, the subscribing applications do not need to know the identity or location of the publishing application providing their information. In this sense, the providers and consumers are said to be loosely coupled.

- Publishers supply information about a subject, without needing to know anything about the applications that are interested in that information. Publishers generate this information in the form of messages, called *publications* that they want to publish and define the topic of these messages.
- Subscribers create *subscriptions* that describe the topic that the subscriber is interested in. Thus, the subscription determines which publications are forwarded to the subscriber. Subscribers can make multiple subscriptions and can receive information from many different publishers.

## The classic example



- A "feed" provides a continuous flow of information which is pushed to interested parties
- Traders consume this information and use it as a basis for the buying and selling stock

© Copyright IBM Corporation 2008

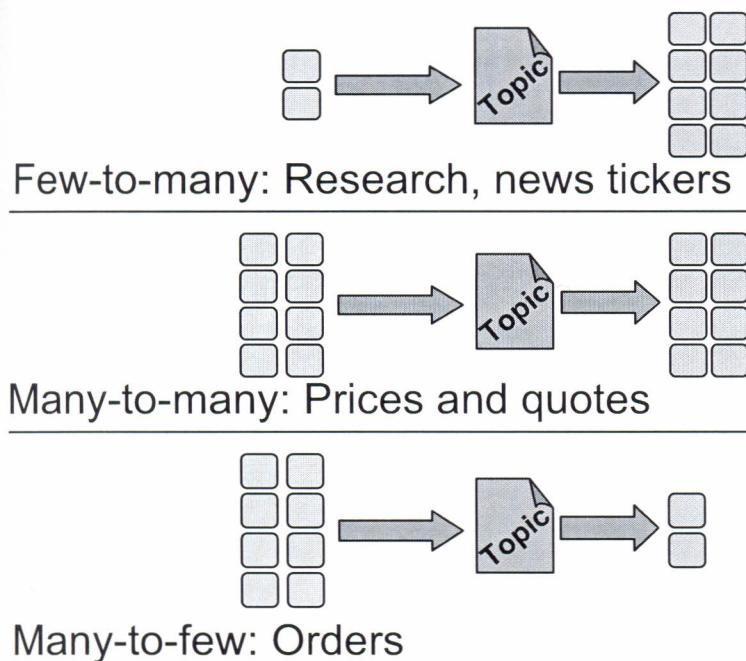
Figure 19-3. The classic example

WM203 / VM2032.0

### Notes:

Perhaps the most-commonly quoted example of a publish/subscribe system is one which provides stock-market information. Here a "feed" provides (publishes) a continuous flow of information containing the latest stock prices. The latest stock prices are required by traders who need this information in order to conduct trades. Traders register their interest in (subscribe to) particular stock prices and receive updates as prices change. Traders can be added and removed without disruption to the providers of the information who have no knowledge of who is receiving their information.

## Loose-coupling with publish/subscribe



© Copyright IBM Corporation 2008

Figure 19-4. Loose-coupling with publish/subscribe

WM203 / VM2032.0

### Notes:

In the WebSphere MQ publish/subscribe model the only thing which connects publishing and subscribing applications is the topic or subject which the publisher associates with his information. Publishers and subscribers need only agree on the topic to become connected to one another. Each different piece of information has its own topic associated with it. Subscribers nominate which types of information they want to receive by subscribing to specific topics.

Published information is sent in a WebSphere MQ message, and the subject of the information is identified by its *topic*. The publisher specifies the topic when it publishes the information, and the subscriber specifies the topics about which it wants to receive publications. The subscriber is sent information about only those topics it subscribes to.

Publishers of information are unaware of subscribers to the extent that they may publish information even if there are no subscribing applications requiring it. Publishing and subscribing are dynamic processes. New subscribers and new publishers can be added to the system without disruption.

## Publications and subscriptions

- Subscribers make **subscriptions** with the queue manager to register their interest in information relating to specific topics.
  - They use the MQSUB verb
  - Or define using DEFINE SUB command
- Publishers provide information about specific topics by sending **publications** to the queue manager
  - They use the MQPUT verb
- The queue manager forwards each publication it receives to all subscribers with a subscription which matches the associated topic

© Copyright IBM Corporation 2008

Figure 19-5. Publications and subscriptions

WM203 / VM2032.0

### Notes:

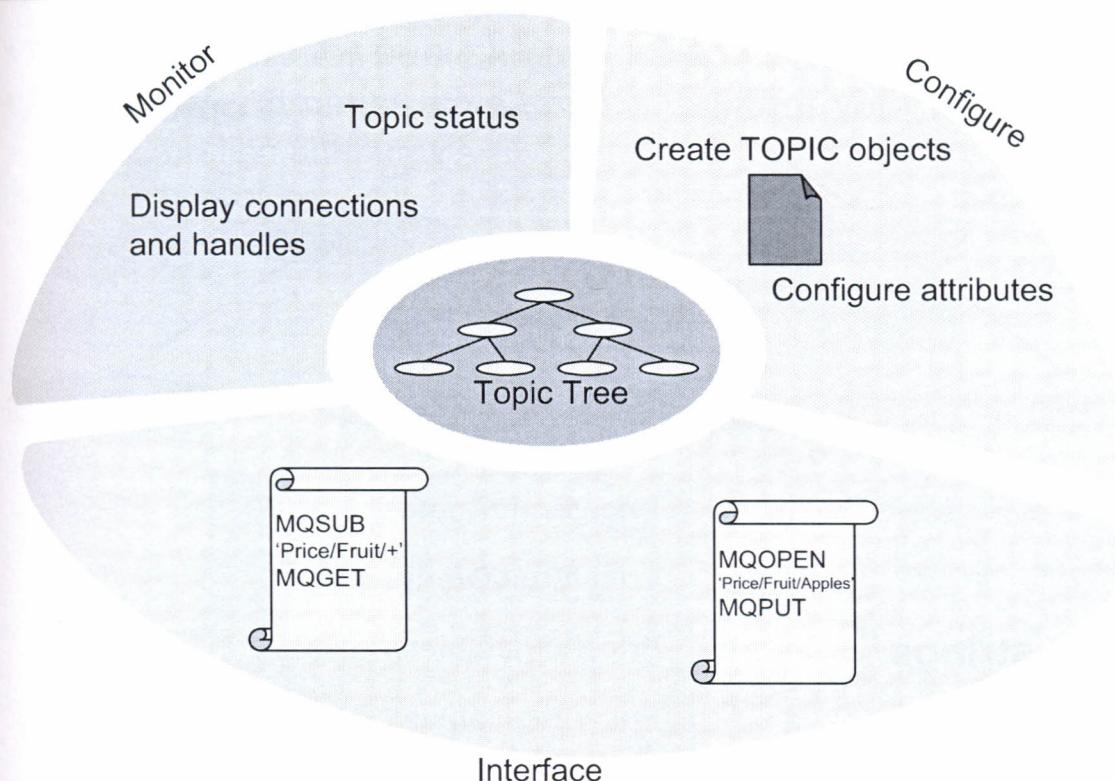
Just to recap, applications which provide information are called publishers. Applications which consume information are called subscribers.

A subscriber specifies the topic it is interested in receiving information about by specifying it on the MQSUB verb. A subscriber may make multiple subscriptions to the queue manager.

A publisher publishes its information by putting a message to a topic.

It is the job of the queue manager, or queue manager network if multiple queue managers have been connected together, to ensure that all subscribing applications with matching subscriptions to the topic being published receive the publisher's message, known as a publication.

## Publish/subscribe in IBM WebSphere MQ



© Copyright IBM Corporation 2008

Figure 19-6. Publish/subscribe in WebSphere MQ

WM203 / VM2032.0

### Notes:

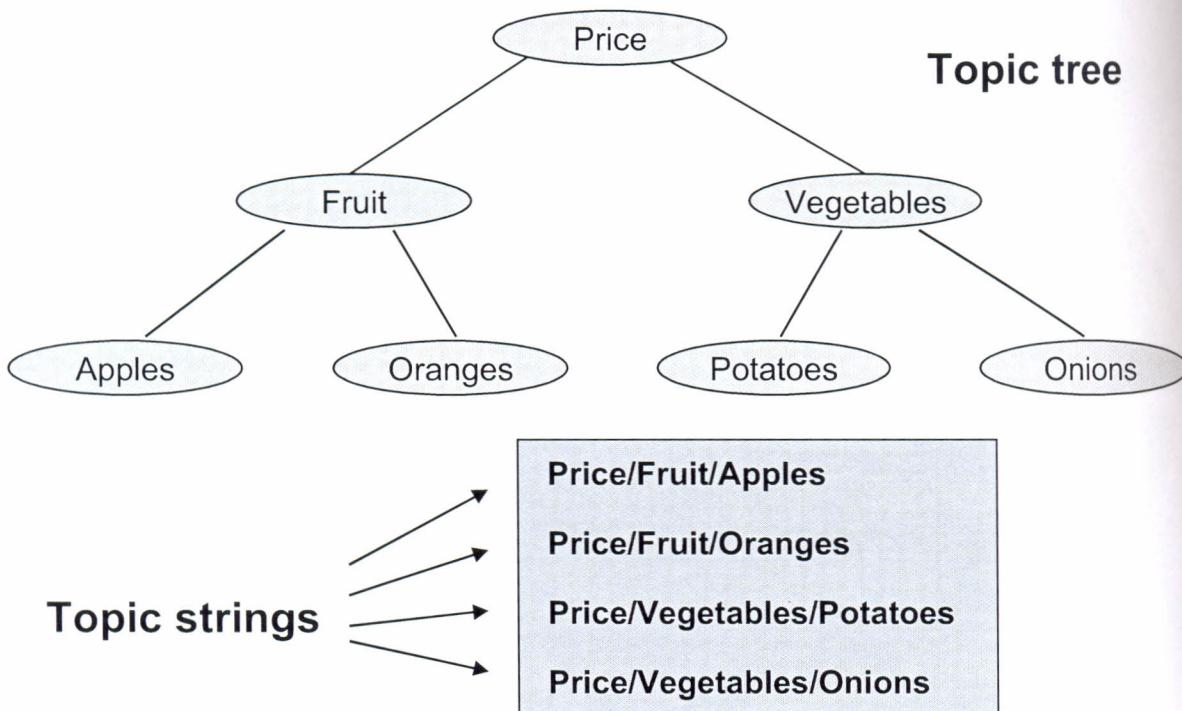
The queue manager holds a view of all the topic strings you are using in a hierarchical construct known as the topic tree (see next page). This topic tree is the central control point for all publish/subscribe. As a user you interact with the topic tree in several different ways.

You can configure the behavior of the topic tree by defining topic objects and changing attributes on them.

You can programmatically interface with the topic tree as a subscriber using MQSUB and as a publisher using MQOPEN and MQPUT.

You can monitor the use of your topic tree by such applications using the Topic status command and the commands to display connections and their handles.

## Topic strings and topic tree



© Copyright IBM Corporation 2008

Figure 19-7. Topic strings and topic tree

WM203 / VM2032.0

### Notes:

Topic strings can be any characters you choose. You can, and should, add structure to your topic strings using the slash ( / ) character. This produces a topic tree with a hierarchical structure, as the example on this foil shows. Although this hierarchical topic tree was created by the use of the topic strings shown, generally picture it as a tree such as this.

## Topic objects

- Not necessary for publish/subscribe
- Provide an administrative control point for your topic tree
  - Configuration attributes
  - Security profiles
  - Topic tree isolation



© Copyright IBM Corporation 2008

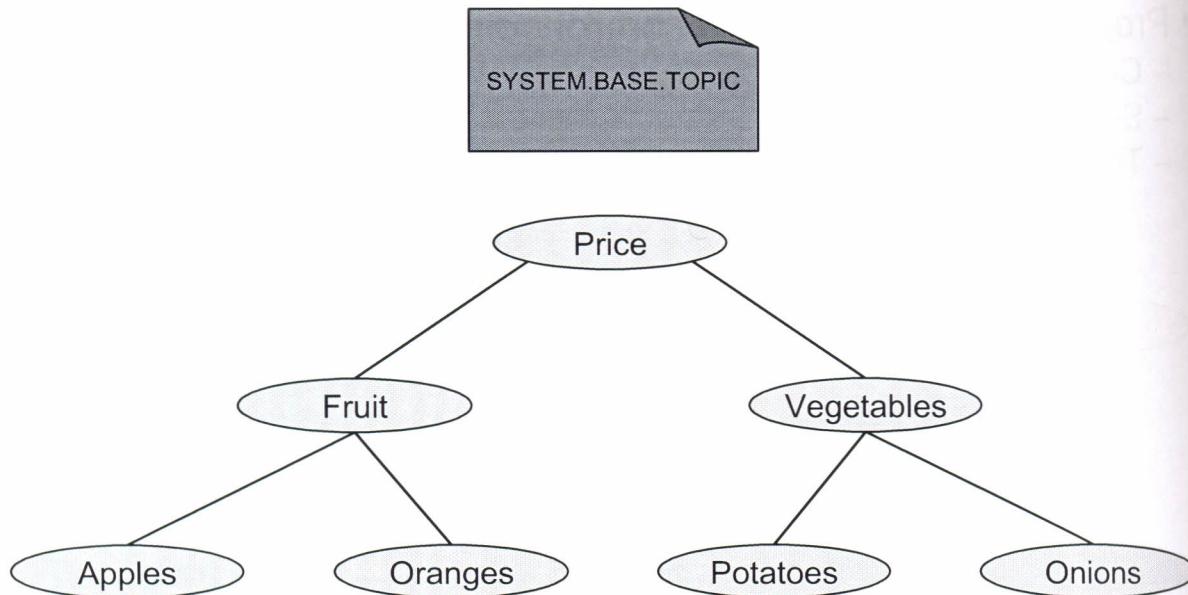
Figure 19-8. Topic objects

WM203 / VM2032.0

### Notes:

- Topic objects are a new construct in WebSphere MQ V7. They can be used to control the behavior of your topic tree.
- You do not need to define any topic objects in order to use publish/subscribe with WebSphere MQ V7, however you may want to define some if you need to configure the topic tree to use non-default attributes. For example, if you want to apply different security profiles to parts of your topic tree, or if you want to isolate your applications from administrative changes to the topic tree – rather like you do when you use remote queue and alias queue definitions.

## Base topic object



© Copyright IBM Corporation 2008

Figure 19-9. Base topic object

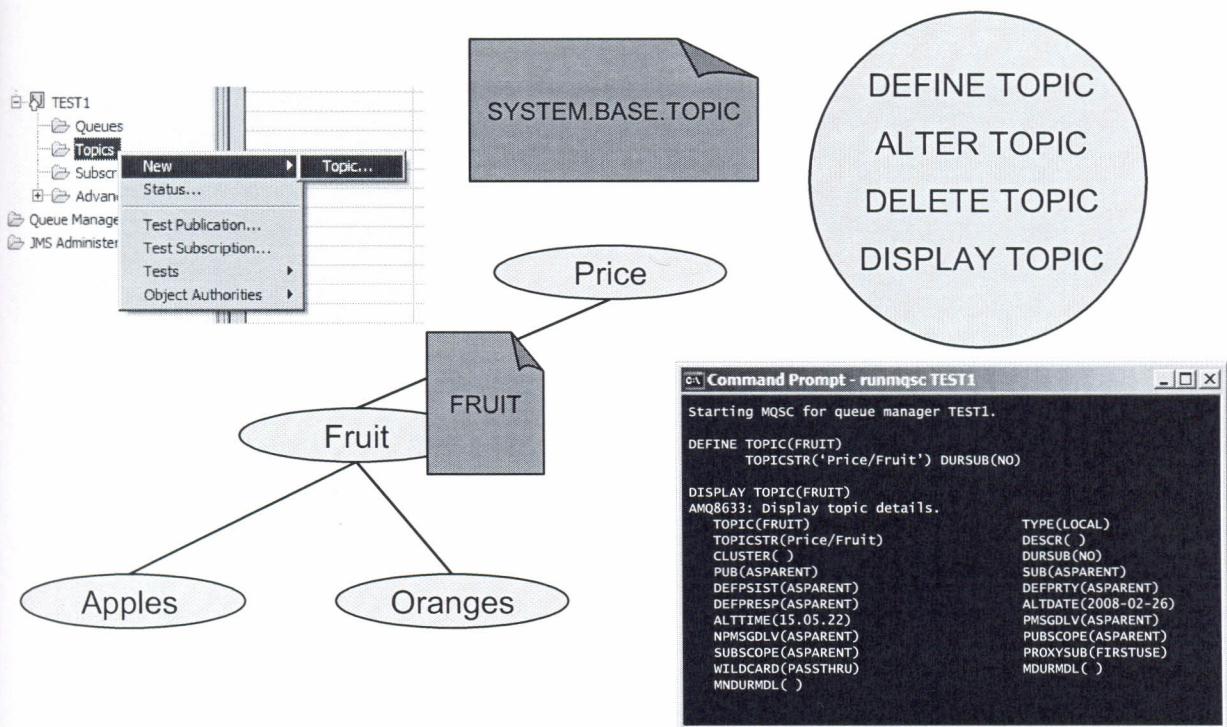
WM203 / VM2032.0

### Notes:

By default there exists a base topic object, the SYSTEM.BASE.TOPIC that contains all the settings for topic object behavior. If you define no other topic objects in your system, the behavior is taken from this topic object. 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.

If you delete this object, the queue manager acts as if the SYSTEM.BASE.TOPIC was defined with the default attributes that come out of the box. If you need to change that behavior, you first need to define this object again.

## Defining a topic object (1 of 2)



Note: the next screen provides a closer view of this command prompt.

© Copyright IBM Corporation 2008

Figure 19-10. Defining a topic object (1 of 2)

WM203 / VM2032.0

### Notes:

If you need to disallow creation of durable subscriptions for one half of the topic tree – in this case the FRUIT half of the tree. You can create one TOPIC object at the highest point where you need this behavior to start, and that behavior is 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.

The topic object can be defined using either WebSphere MQ Explorer, or using MQSC command mode.

## Defining a topic object (2 of 2)

```
D:\newqdev\q\Debug>runmqsc TEST1
5724-H72 <C> Copyright IBM Corp. 1994, 2004. ALL
RIGHTS RESERVED.
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(ASPARENT)                                  SUB(ASPARENT)
    DEFPSIST(ASPARENT)                            DEFPRTY(ASPARENT)
    DEFPRESP(ASPARENT)                           ALTDATE(2008-02-26)
    ALTTIME(15.05.22)                             PMSGDLV(ASPARENT)
    NPMMSGDLV(ASPARENT)                          PUBSCOPE(ASPARENT)
    SUBSCOPE(ASPARENT)                           PROXYSUB(FIRSTUSE)
    WILDCARD(PASSTHRU)                           MDURMDL( )
    MNDURMDL( )
```

© Copyright IBM Corporation 2008

Figure 19-11. Defining a topic object (2 of 2)

WM203 / VM2032.0

### Notes:

## DEFINE TOPIC command

```

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(ASPARENT)                               SUB(ASPARENT)
    DEFPSIST(ASPARENT)                          DEFPRTY(ASPARENT)
    DEFPRESP(ASPARENT)                          ALTDATE(2008-02-26)
    ALTTIME(15.05.22)                           PMSGDLV(ASPARENT)
    NPMMSGDLV(ASPARENT)                        PUBSCOPE(ASPARENT)
    SUBSCOPE(ASPARENT)                          PROXYSUB(FIRSTUSE)
    WILDCARD(PASSTHRU)                          MDURMDL( )
    MNDURMDL( )

```

© Copyright IBM Corporation 2008

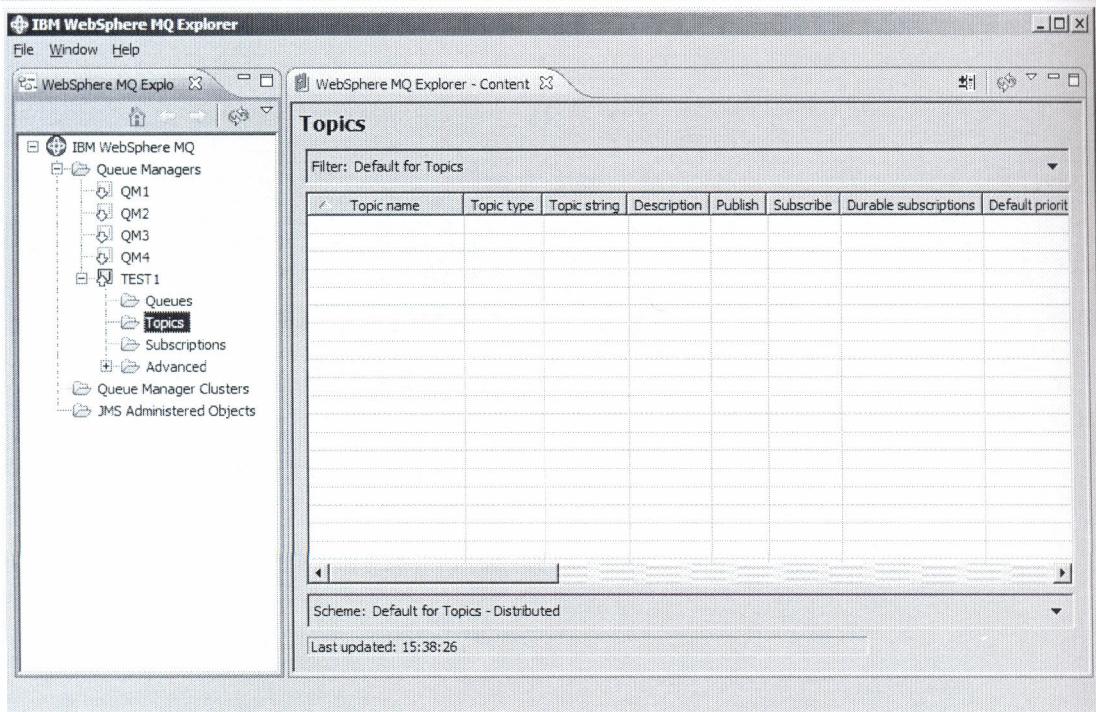
Figure 19-12. DEFINE TOPIC command

WM203 / VM2032.0

### Notes:

- The **ASPARENT** value in many of the attributes means that the value is based on the setting of the closest parent administrative topic object in the topic tree.
- DURSUB** Specifies whether applications are permitted to make durable subscriptions on this topic.
- MDURMDL** and **MNDURMDL** specify the name of the model queue to be used for durable (and non-durable) subscriptions that request that the queue manager manages the destination of its publications
- PUB** Controls whether messages can be published to this topic
- SUB** Controls whether messages are permitted to subscribe to this topic
- 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.

## Defining a topic using IBM WebSphere MQ Explorer (1 of 11)



© Copyright IBM Corporation 2008

Figure 19-13. Defining a topic using WebSphere MQ Explorer (1 of 11)

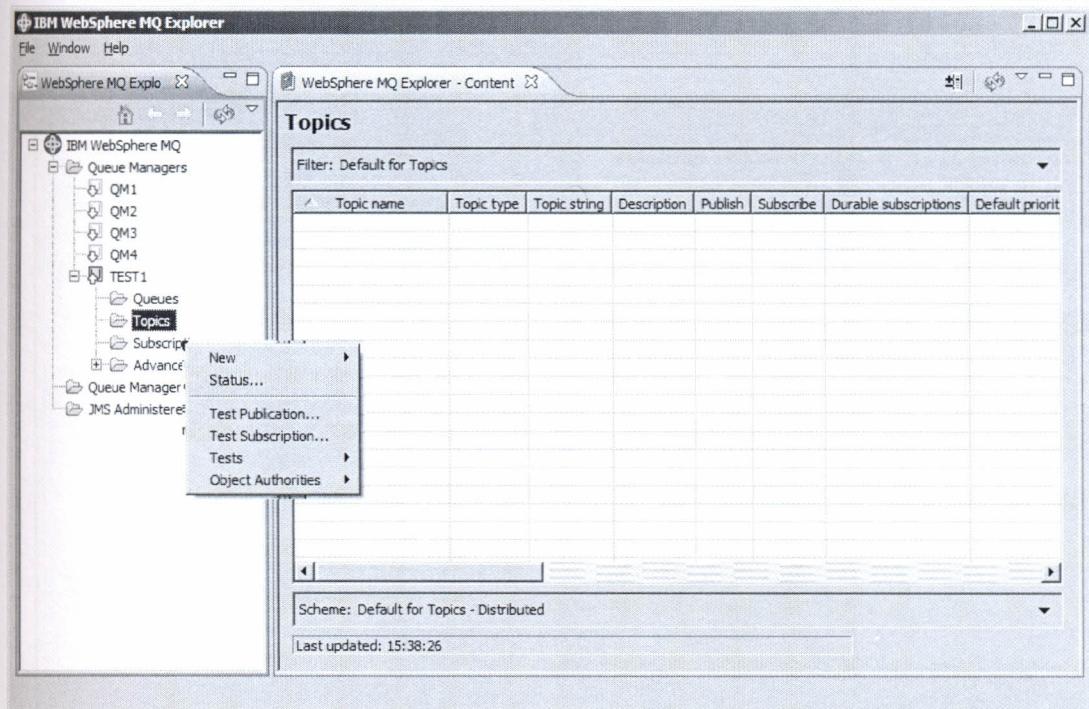
WM203 / VM2032.0

### Notes:

Looking at the DISPLAY output from the object you defined, you can see that many of the attributes that you did not 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.

(1)

## Defining a topic using IBM WebSphere MQ Explorer (2 of 11)



© Copyright IBM Corporation 2008

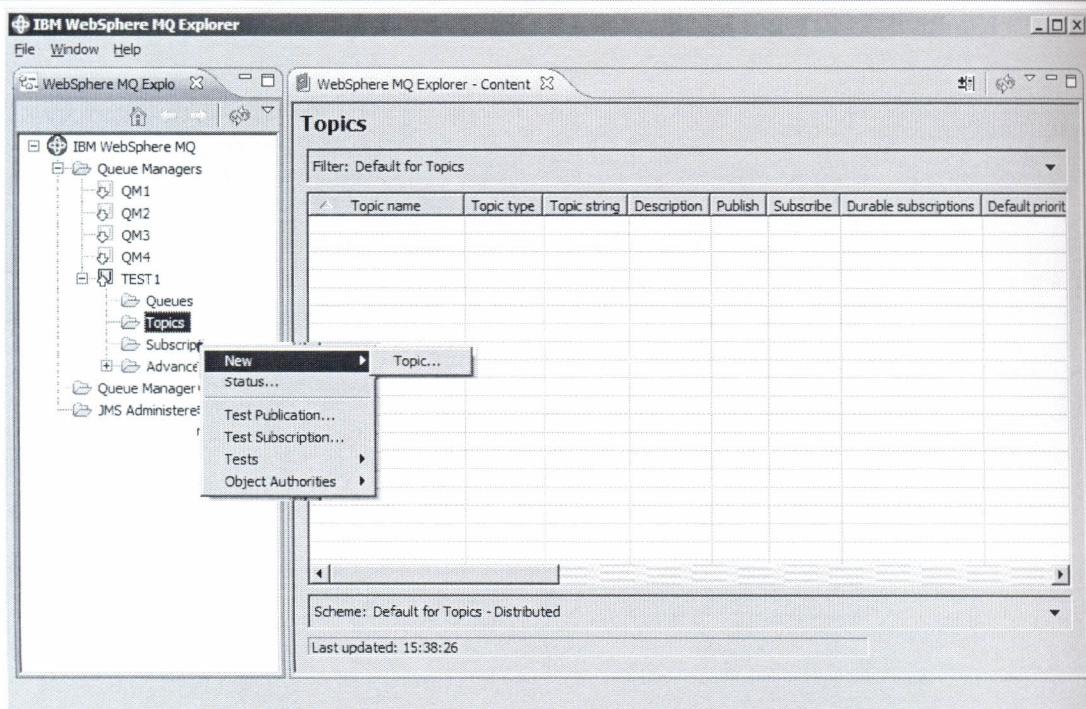
Figure 19-14. Defining a topic using WebSphere MQ Explorer (2 of 11)

WM203 / VM2032.0

### Notes:

of the  
ings –  
eans  
p the

## Defining a topic using IBM WebSphere MQ Explorer (3 of 11)



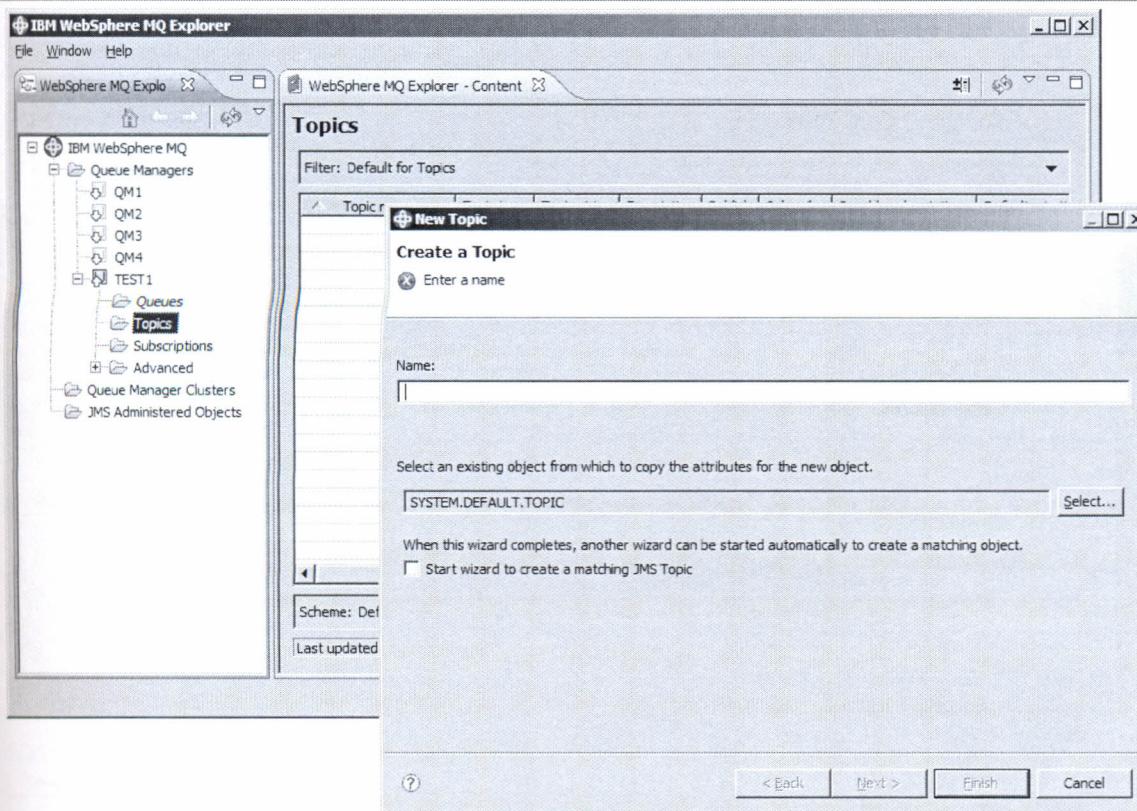
© Copyright IBM Corporation 2008

Figure 19-15. Defining a topic using WebSphere MQ Explorer (3 of 11)

WM203 / VM2032.0

### Notes:

## Defining a topic using IBM WebSphere MQ Explorer (4 of 11)



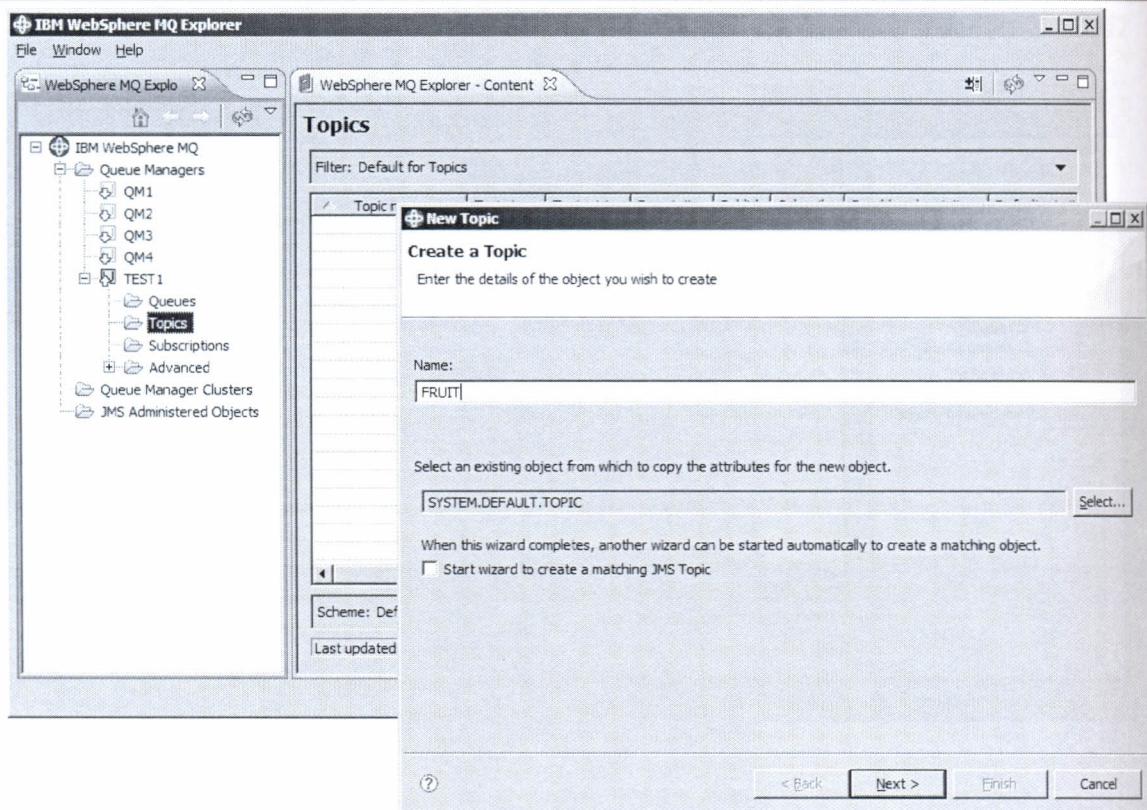
© Copyright IBM Corporation 2008

Figure 19-16. Defining a topic using WebSphere MQ Explorer (4 of 11)

WM203 / VM2032.0

### Notes:

## Defining a topic using IBM WebSphere MQ Explorer (5 of 11)



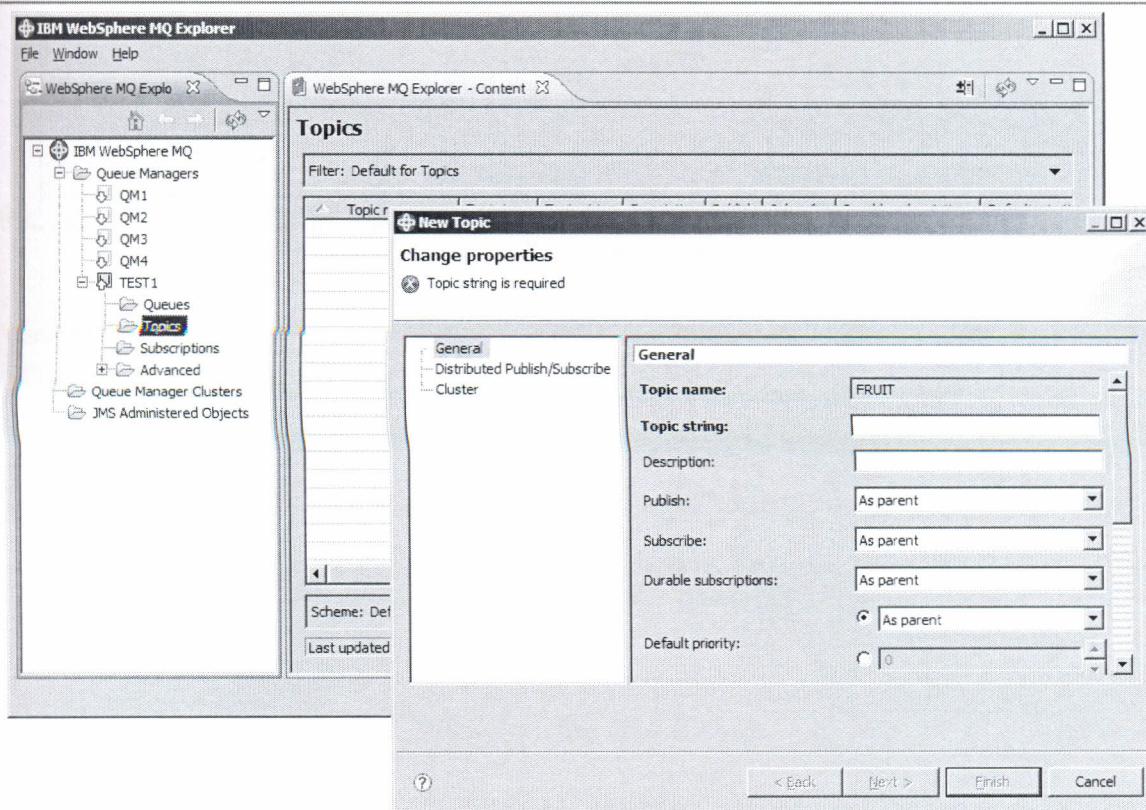
© Copyright IBM Corporation 2008

Figure 19-17. Defining a topic using WebSphere MQ Explorer (5 of 11)

WM203 / VM2032.0

### Notes:

## Defining a topic using IBM WebSphere MQ Explorer (6 of 11)



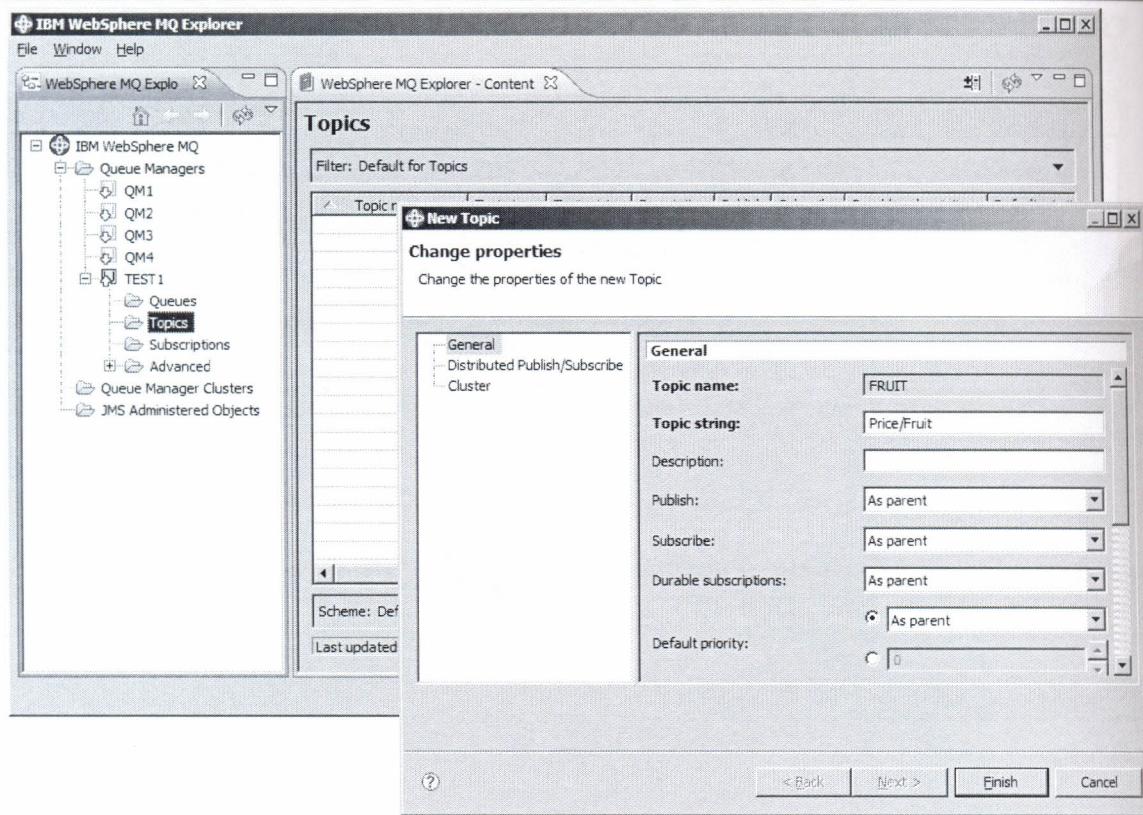
© Copyright IBM Corporation 2008

Figure 19-18. Defining a topic using WebSphere MQ Explorer (6 of 11)

WM203 / VM2032.0

### Notes:

## Defining a topic using IBM WebSphere MQ Explorer (7 of 11)



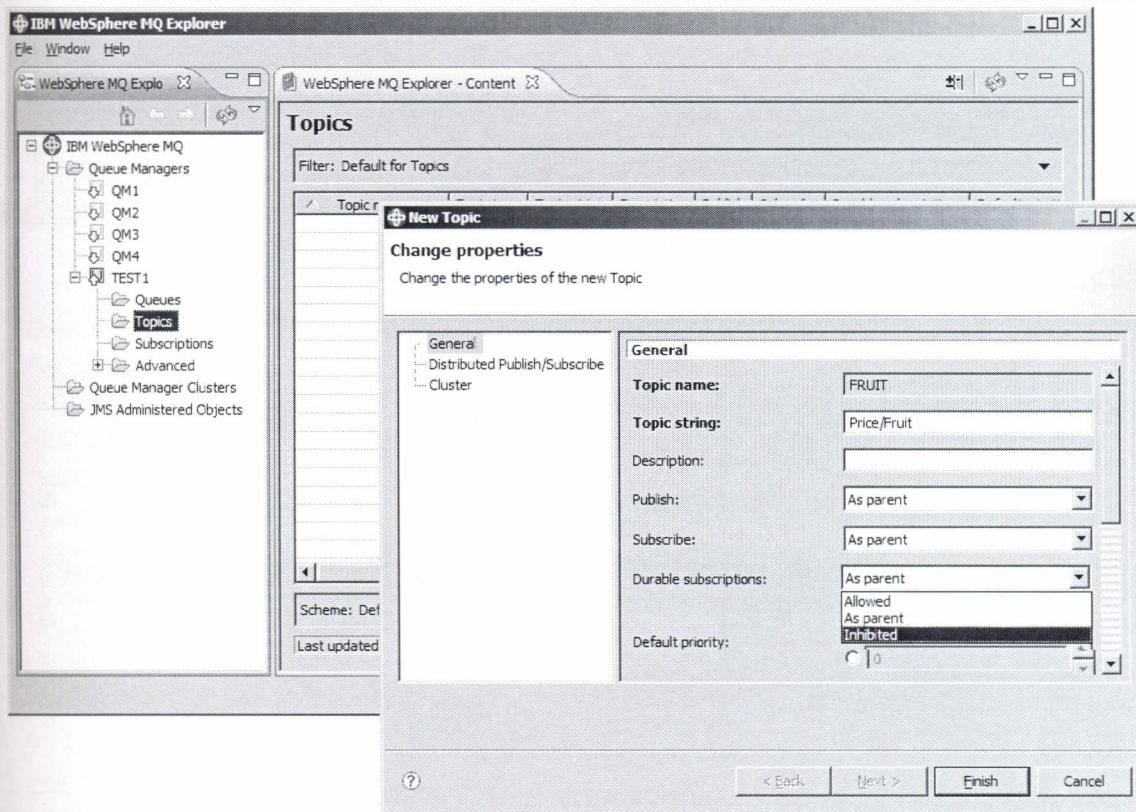
© Copyright IBM Corporation 2008

Figure 19-19. Defining a topic using WebSphere MQ Explorer (7 of 11)

WM203 / VM2032.0

### Notes:

## Defining a topic using IBM WebSphere MQ Explorer (8 of 11)



© Copyright IBM Corporation 2008

Figure 19-20. Defining a topic using WebSphere MQ Explorer (8 of 11)

WM203 / VM2032.0

### Notes:

# Defining a topic using IBM WebSphere MQ Explorer (9 of 11)

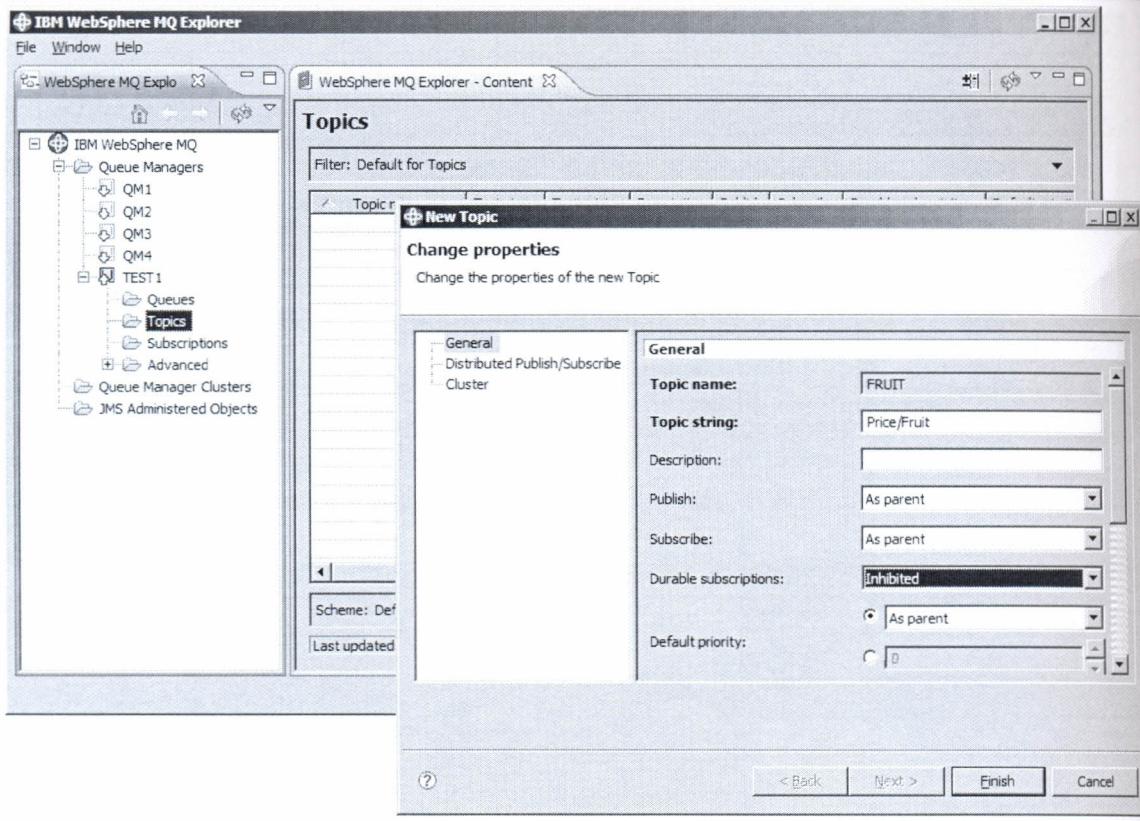
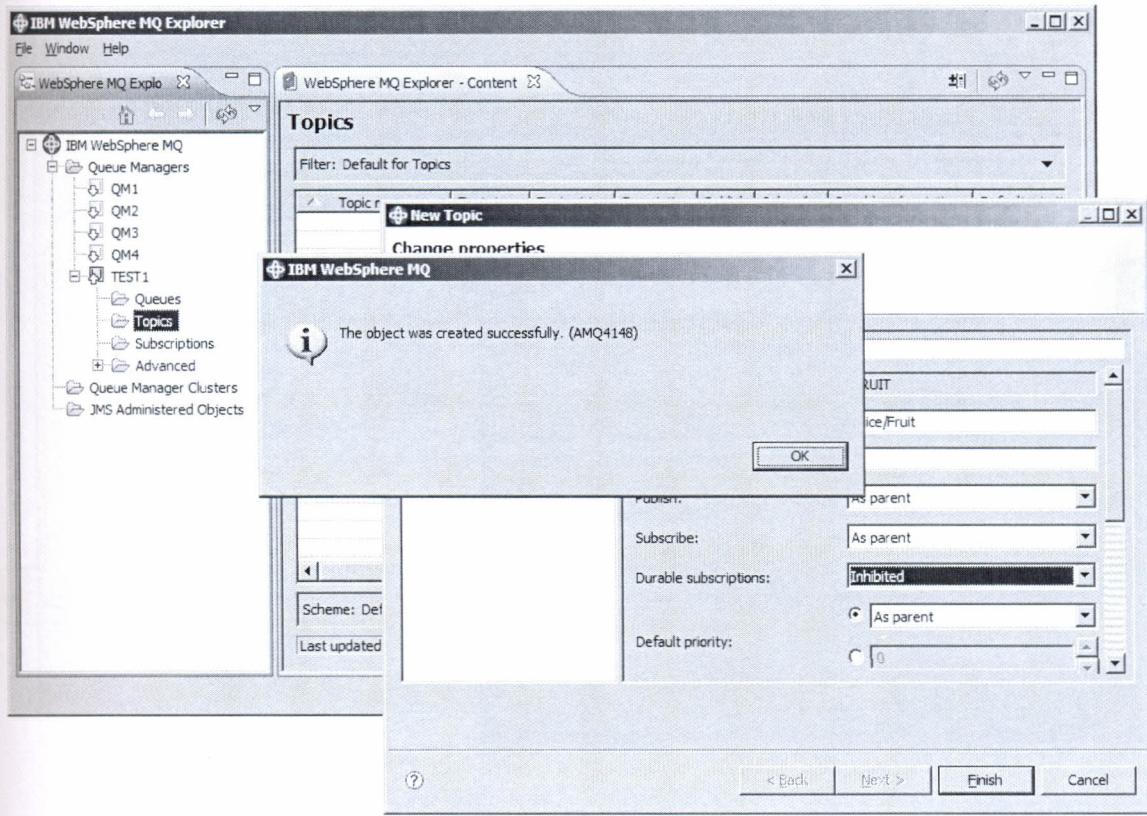


Figure 19-21. Defining a topic using WebSphere MQ Explorer (9 of 11)

WM203 / VM2032.0

## Notes:

## 9 Defining a topic using IBM WebSphere MQ Explorer (10 of 11)



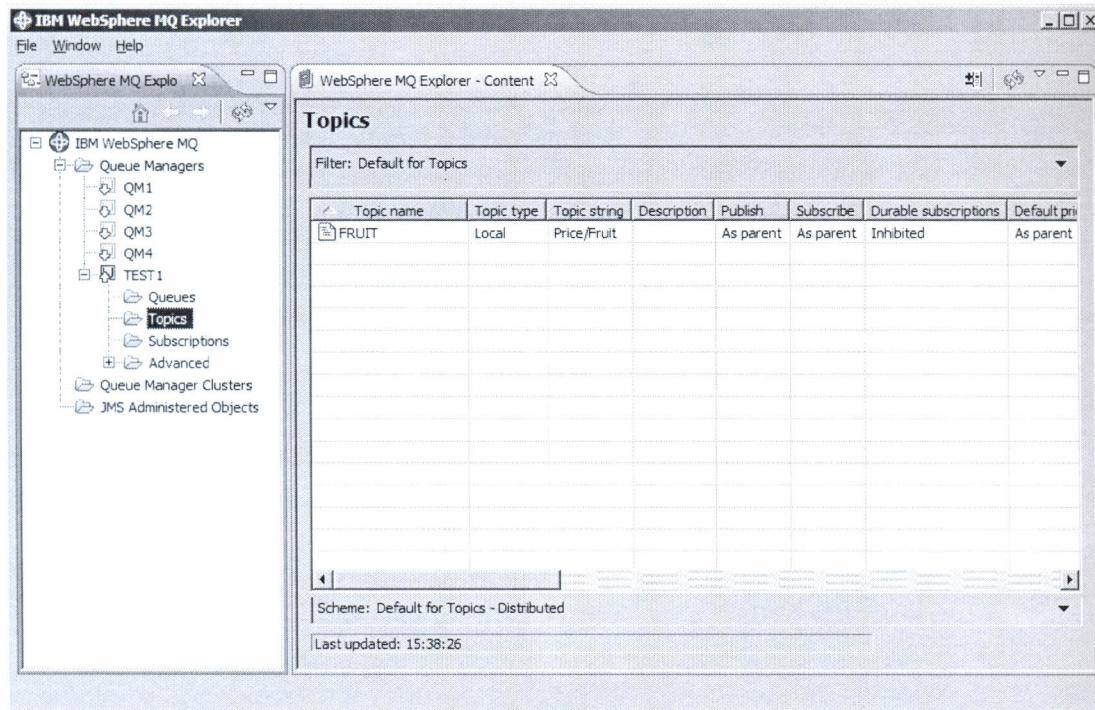
© Copyright IBM Corporation 2008

Figure 19-22. Defining a topic using WebSphere MQ Explorer (10 of 11)

WM203 / VM2032.0

### Notes:

# Defining a topic using IBM WebSphere MQ Explorer (11 of 11)



© Copyright IBM Corporation 2008

Figure 19-23. Defining a topic using WebSphere MQ Explorer (11 of 11)

WM203 / VM2032.0

## Notes: