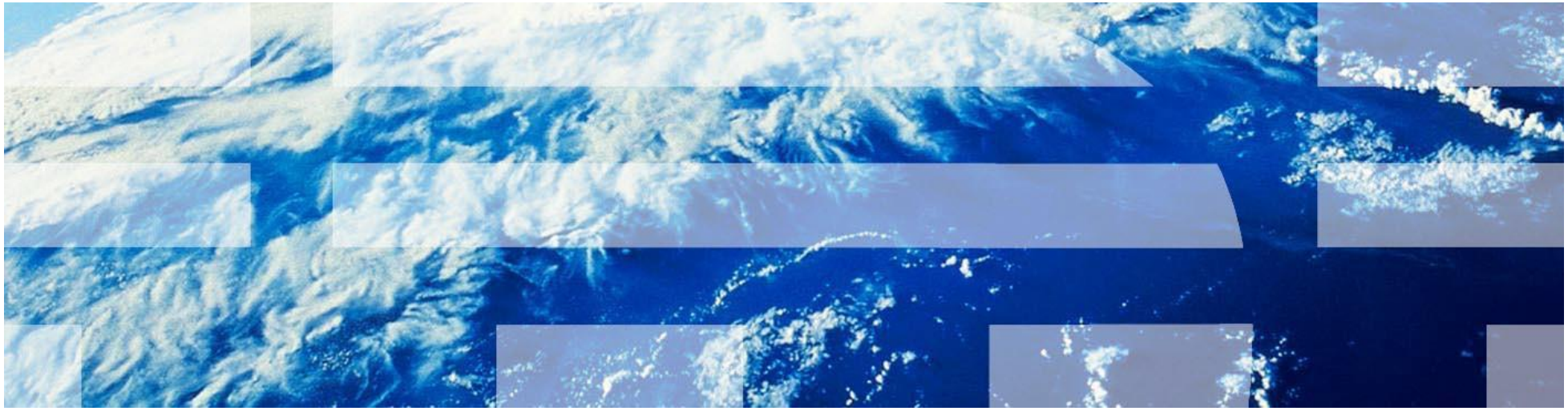
An IBM Proof of Technology

IBM MQ

Overview of Publish / Subscribe



Unit Agenda

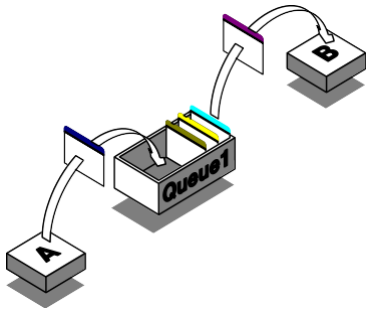
- Publish/Subscribe and IBM MQ
  - Advantages of Pub/Sub compared to point-to-point
  - Terminology - what are:
    - Topic objects, topic strings, topic space
    - Subscriptions, durable and non-durable
    - Destinations, managed and unmanaged
    - Publications, retained and otherwise
  - Topologies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

Point to Point 예시

- 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



Notes - Point to point examples

N

O

T

E

S

In contrast the other "sibling" is point to point messaging.

In the same way it is not the message content that makes an application point to point or Pub / sub, it is the infrastructure that handles the message.

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

Notes - Publish / subscribe examples

N

O

T

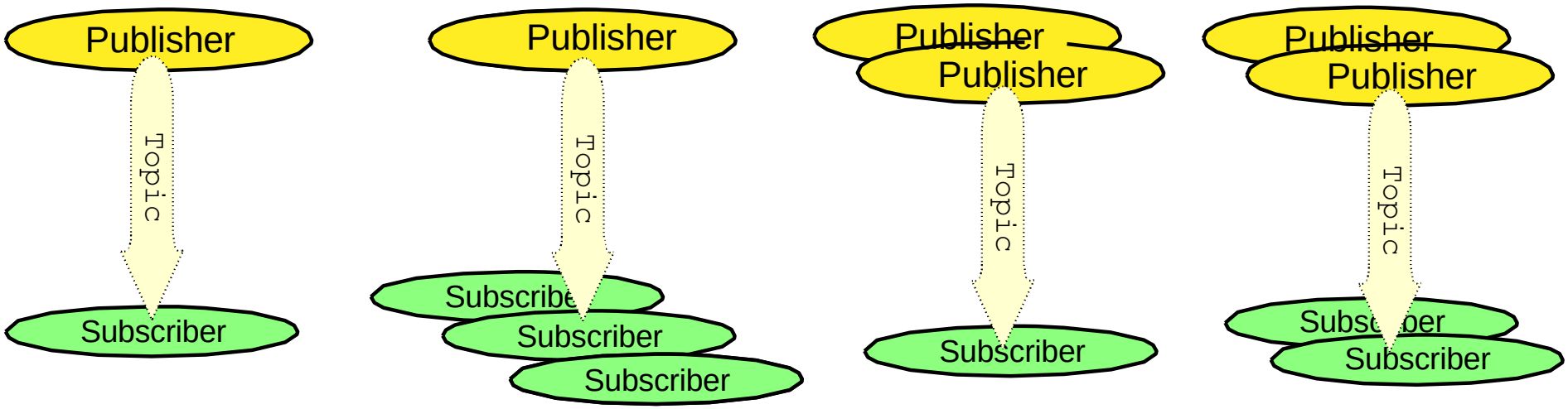
E

S

Again the infrastructure is what is pub/sub

Over 10,000 magazines in the US,
Publishers use the magazine title and advertising to indicate the contents.
We readers select which magazine to pick up based on that information.

Loose coupling with Pub/Sub



IBM MQ 의 Publish/Subscribe 구현

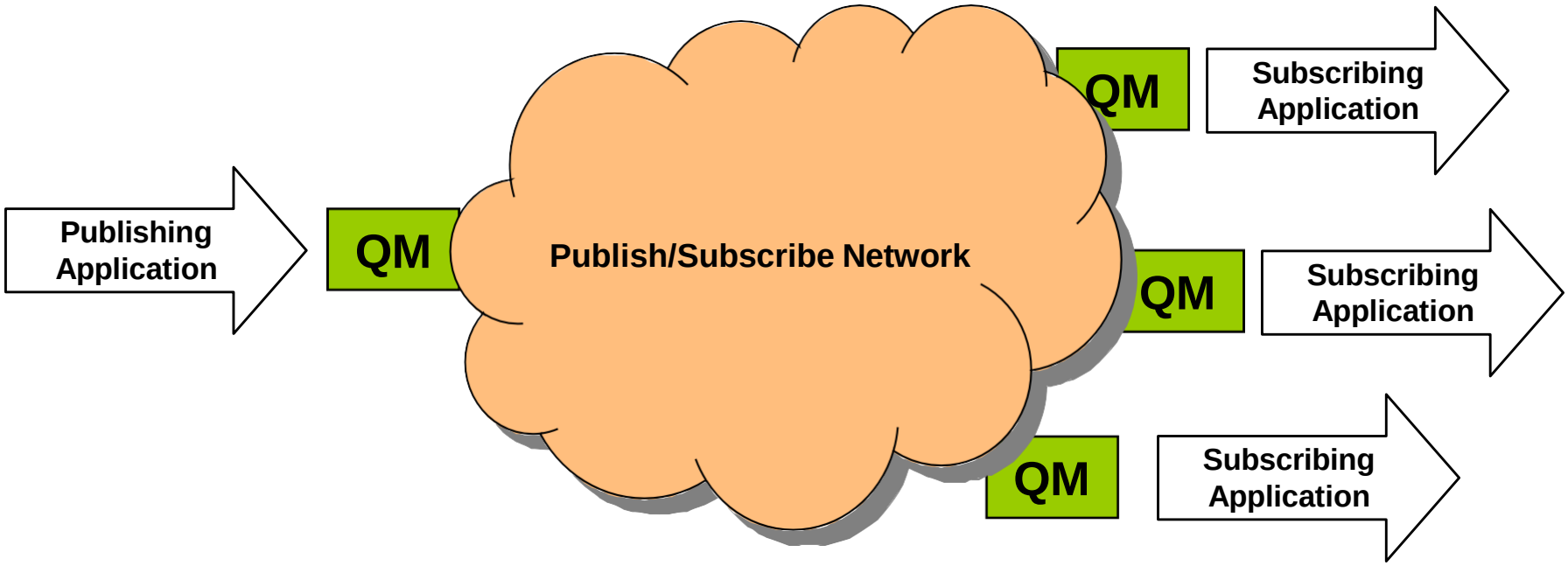
- *TOPIC STRING* 이 핵심 개념

- 메시지 게시 (Publish)
  - 각 게시 행위는 하나의 Topic String 에서만 이루어짐.

- 구독자에게 메시지 전달 (자신이 관심을 등록한 Topic String 에 게시된 메시지를 전달 받음
  - 여러 주제를 동시에 구독 가능
  - 와일드 카드를 통해 하위 주제 구독 가능
  - Selectors 를 사용하여 메시지를 필터링

- Topic String 생성 및 보안
  - 관리적으로 생성
- - 동적으로 생성

Publish/Subscribe applications

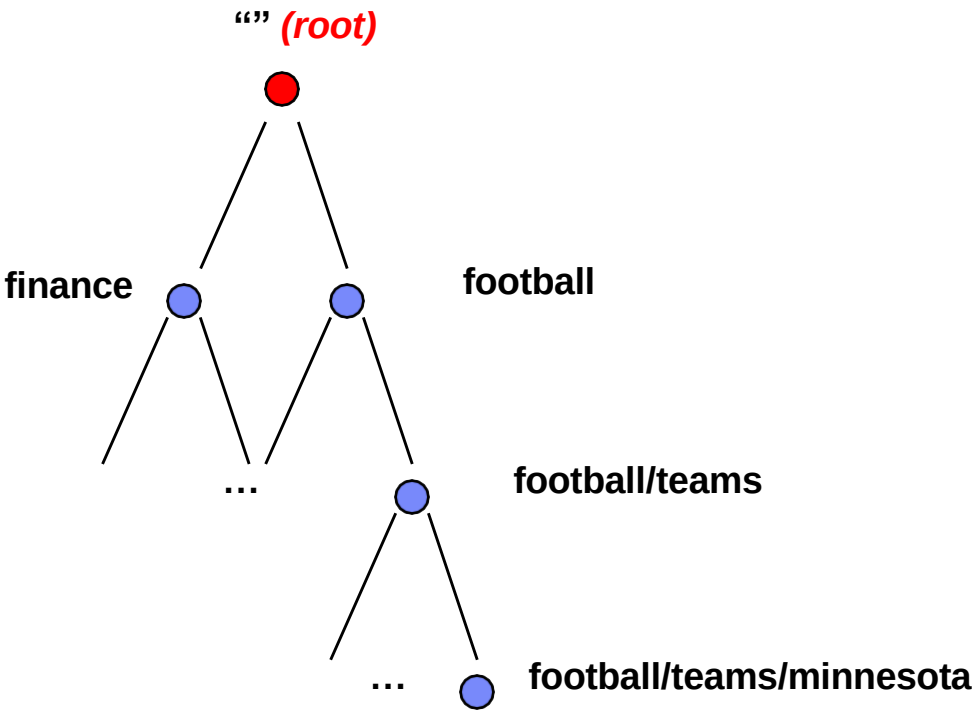


- 애플리케이션은 로컬 큐 매니저에 연결
- 네트워크로 연결된 여러 큐 매니저.
- 다양한 API 사용 가능
- *topic string* 을 통한 연결

Concepts – Topic String

- Topic *string* 무제한 길이의 문자열
- publishers 와 subscribers 를 연결
- 구조와 의미를 포함함.
  - "/" slash : 계층적 구조
  - UNIX® or Windows® 디렉토리 구조와 유사
    - e.g. ***"/news/sports/football/teams/minnesota"***
- 의미 : 구독시 와일드카드 문자를 사용하여 특정 범위의 주제 구독
  - "#" , '+'
  - e.g. ***"/news/+/football/#"***

Concepts – Topic Tree



- 토픽 계층 구조의 내부 표현
- 루트 노드에서 시작
- 사용중인 모든 토픽 문자열로부터 암시적으로 생성됨
- 토픽 Ojbect 와 Tree 노드 간의 일대일 맵핑은 필수적이지 않음.

Notes - Concepts – Topic Tree

N

O

T

E

S

A topic tree is an internal representation of the topic hierarchy – is references using the *Topic string*.

The tree has a root node at the very top. This is described using an MQ-defined base topic object (topic objects will be discussed on the next slide).

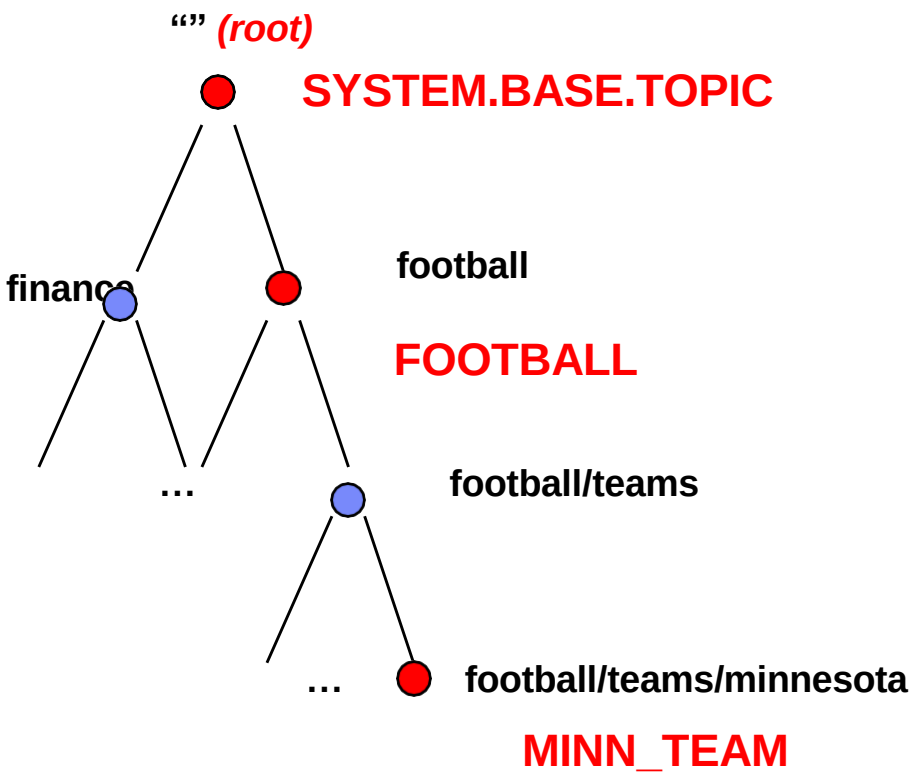
The shape of the topic tree is implied from the complete set of topic strings in use - either defined (as topic objects), published to, subscribed to.

There is ***not necessarily*** a one-to-one mapping between topic *objects* and *nodes* in the tree. This will become clear as we move to the next slide and discuss exactly what we mean by topic *objects*.

Concepts – Topic Object

- Topic *Object* 는 IBM MQ 에서 관리할 수 있는 객체
- Topic Object 의 이름 규칙 은 IBM MQ 의 다른 객체 와 동일한 이름 규칙 (예 : 큐)
- Topic String 은 Topic Object 의 속성 중 하나
  - Topic Objects 는 Topic String 외에도 다양한 속성 설정 가능
- 보안설정
- 애플리케이션에서의 사용
  - JMS programs 은 Destination 객체를 사용

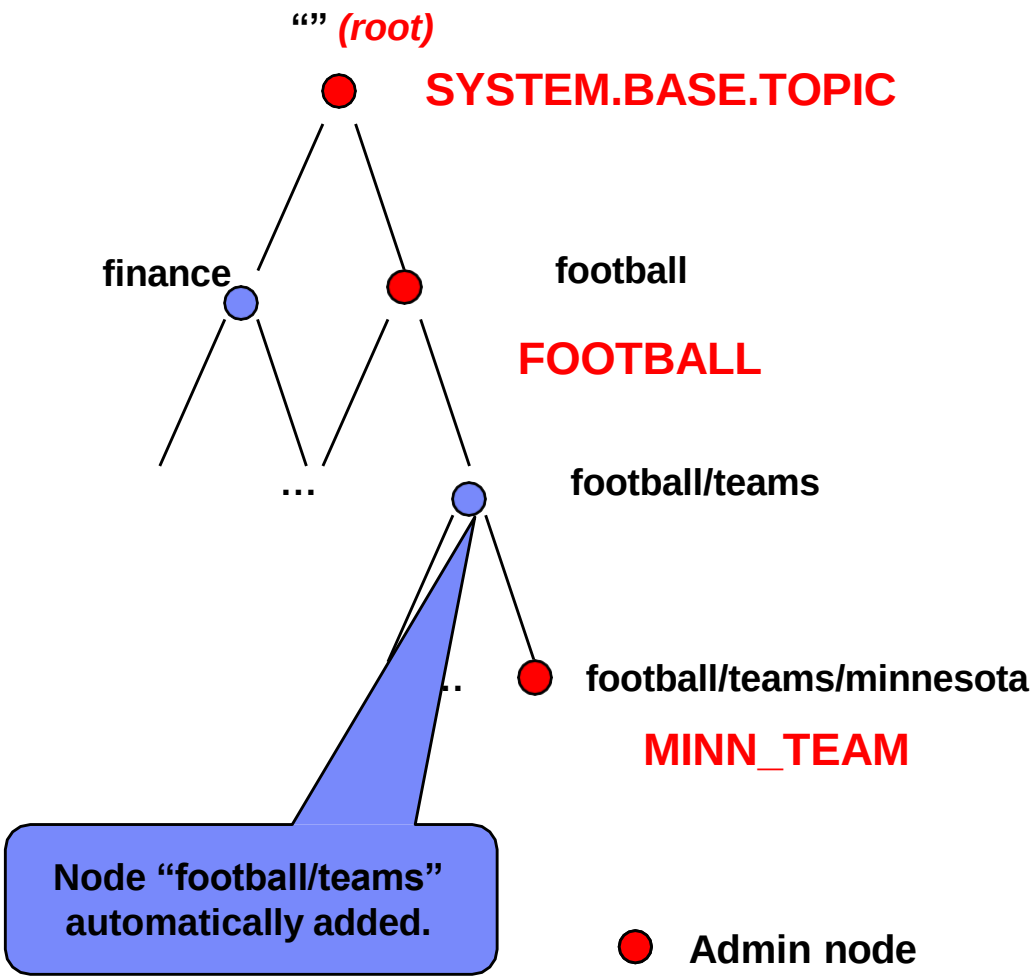
Concepts – Topic Tree



- 관리되는 노드
- 영구적 계층 구조
- 속성 지정 가능
- Topic Object 이름과 Topic String
  - 관리되는 노드에는 Topic Object 이름이 있으나 , Topic String 을 반영할 필요는 없음.

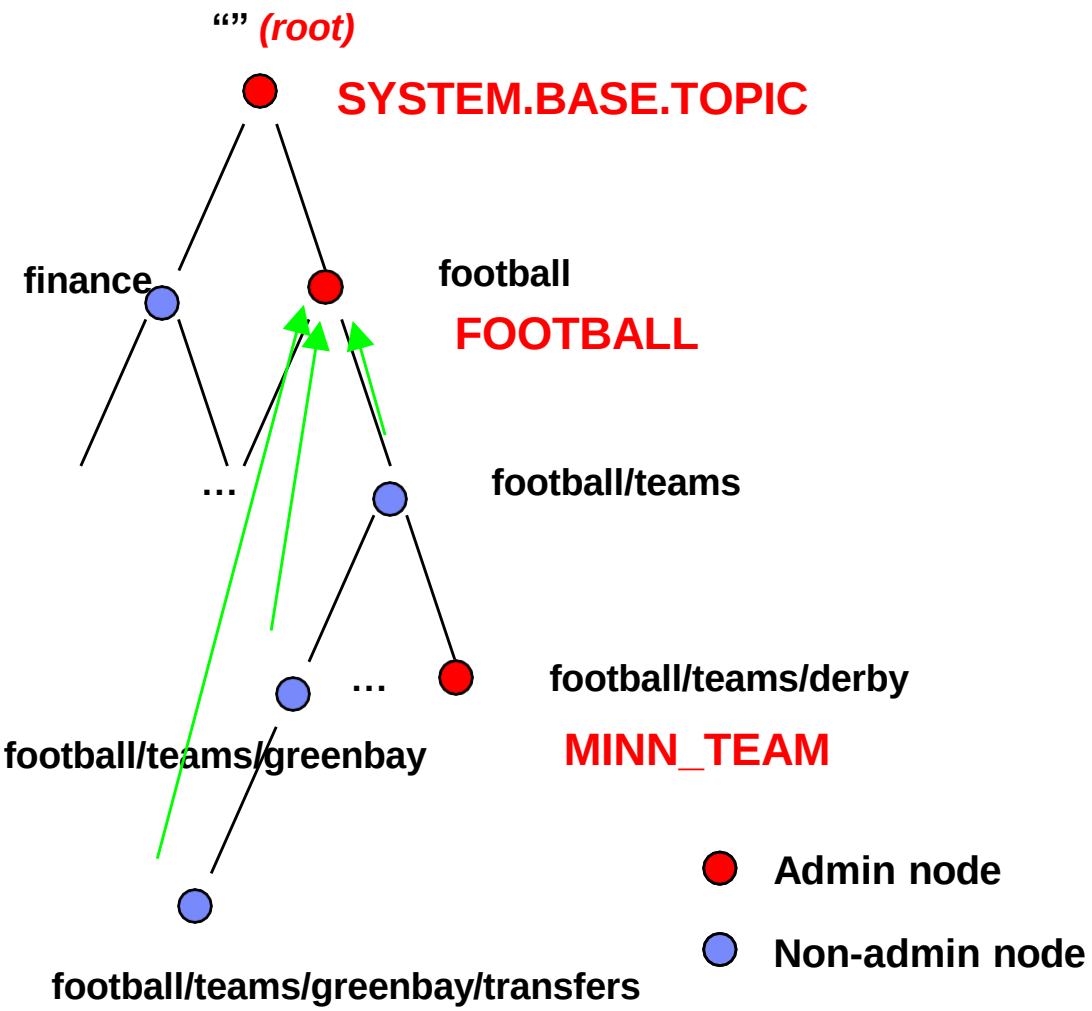
DEFINE TOPIC(FOOTBALL) TOPICSTR('football')

Concepts – Topic Tree



- 퍼블리셔, 구독자에 의한 동적 확장
- 동적으로 추가된 노드는 부모노드 (관리노드) 의 모든 속성을 상속

Concepts – Topic Tree



- 토픽트리의 확장
- 임시노드의 특성
  - 자동 제거 / 동적 트리 확장
- 속성 상속

MQSUB('/football/teams/greenbay/transfers')

Concept - Subscriptions

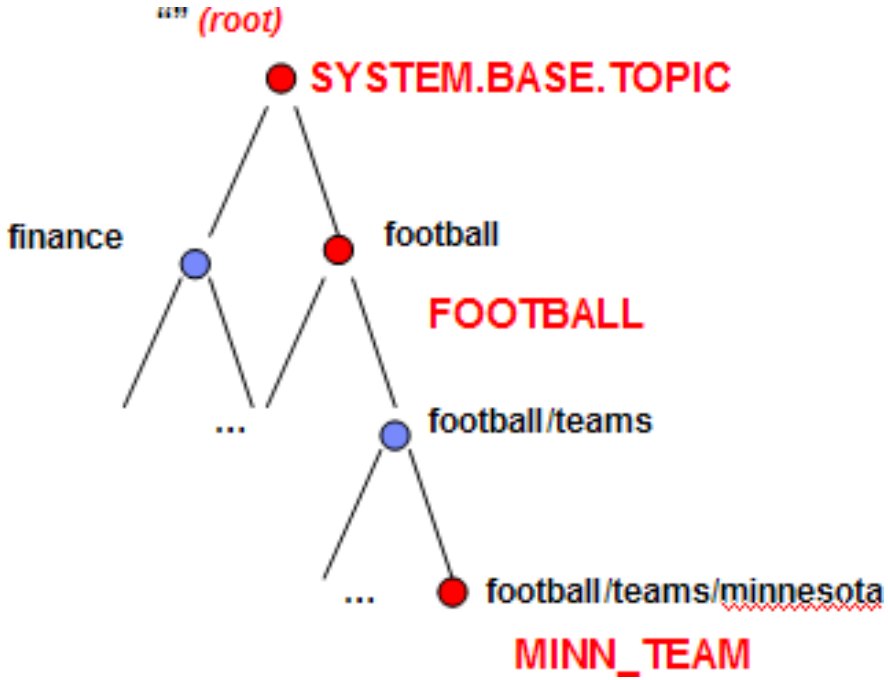
- subscription (구독) 행위는 *topic string* 에 대한 것임.
  - Wildcards can be used to set them up, but in essence they are to a topic string
  - Corresponds to a node in the topic tree
- 구독은 지속적 / 비지속적
  - Non-durable subscriptions (비지속적 구독)
  - durable subscriptions (지속적 구독)
- Subscription destinations can be *managed*, or not
  - managed destination 관리되는 구독 대상자
  - unmanaged destination 관리 되지 않는 구독 대상자
- 구독은 프로그래밍 방식 또는 관리 방식으로 등록

Concept - Retained Publications

- 특정 주제에 대한 가장 최근 발행된 메시지
  - Each node on the topic tree can have at most **one** Retained Publication
- "retained" 여부는 발행 (Publish) 시점에 결정
- 구독자 subscriber 는 언제든지 현재 Retained Publication 을 요청 가능
- *예 : StockPrice.CurrentPrice*
  - 주가 변경시 실시간 알림
  - 필요할때 언제든지 현재 주가 정보 요청

Administered Subscription (MQ 에서 관리되는 구독

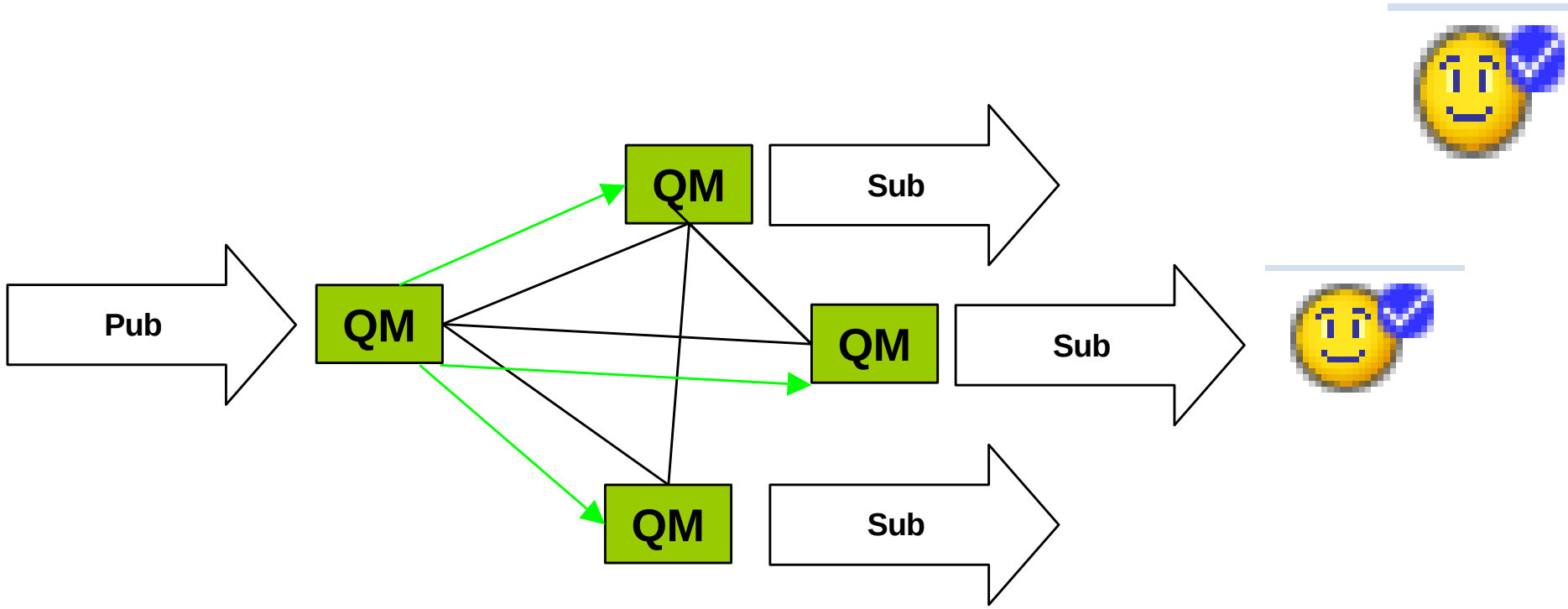
- 주요 속성
  Sample attributes
  - **TOPICSTR** – 무제한 길이
    TOPICSTR('/football/teams/#')
  - **DEST** – 구독과 관련된 메시지가 전달될 객체 이름
    DEST(MY.QUEUE)
  - **DESTQMGR** – 대상 큐 매니저
    DESTQMGR(REMOTE.QMGR)



Distributed Pub/Sub Topologies

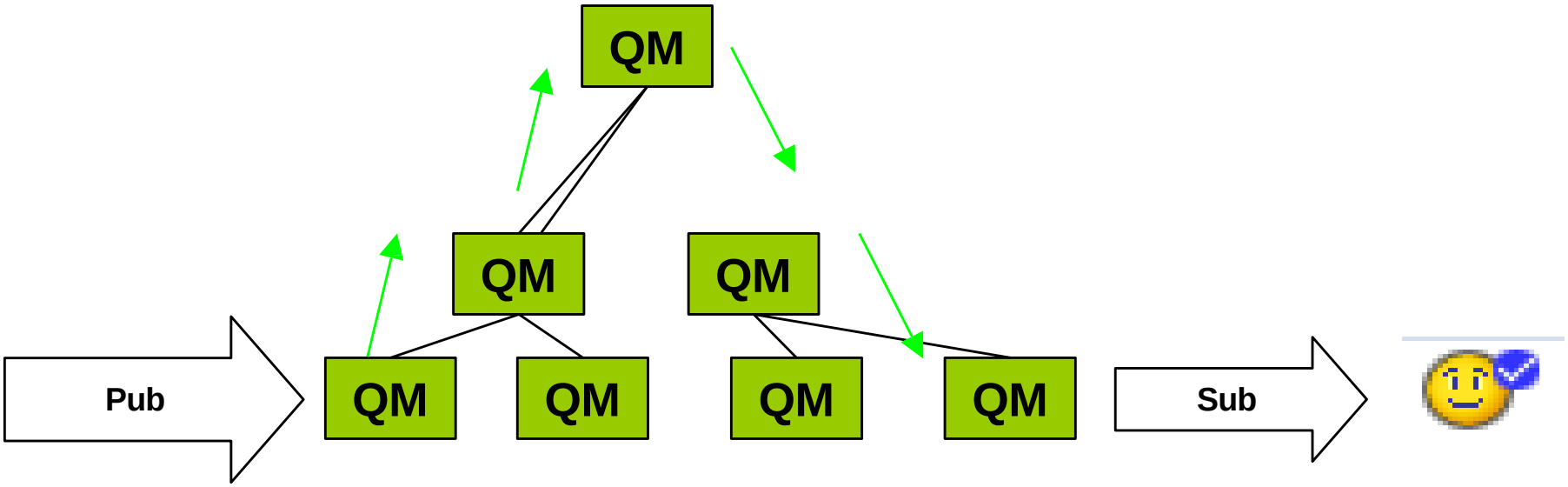
- 분산 Pub/Sub topologiy
  - Publish/Subscribe *Clusters (클러스터 방식)*
    - Queue managers are connected using a “network” or “mesh” connectivity model
    - MQ Clustering technology is used to provide connectivity between participating queue managers
    - Publications from any queue manager in the cluster may be subscribed to by any other
    - Enables multiple paths for publications, can eliminate single points of failure
  - Publish/Subscribe *Hierarchies (계층구조 방식)*
    - 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
- 혼합 토폴로지

Example of a Publish/Subscribe Cluster

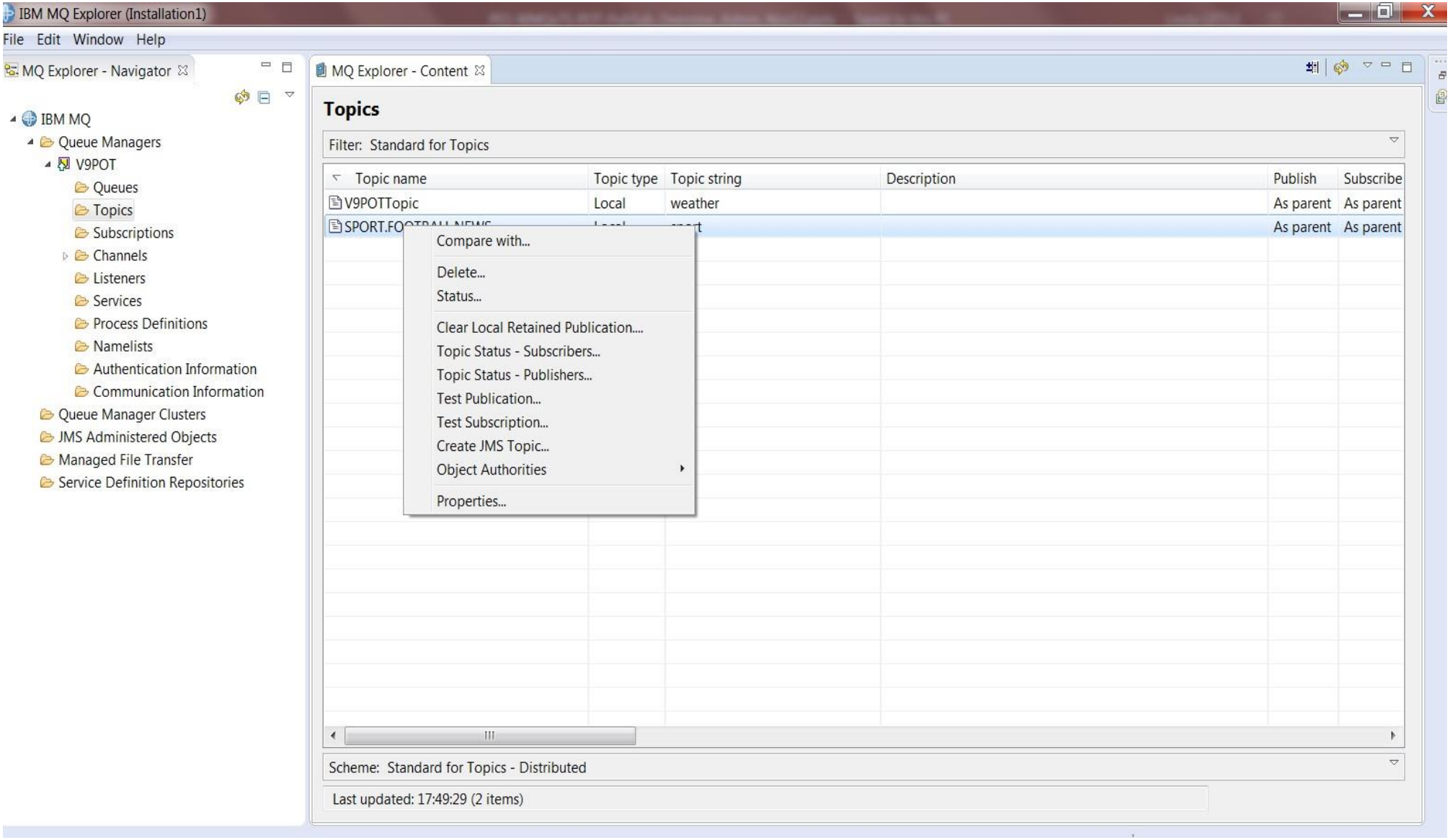


- **Consistent definitions in cluster**
- **Multiple routes across cluster**

Example of a Publish/Subscribe Hierarchy

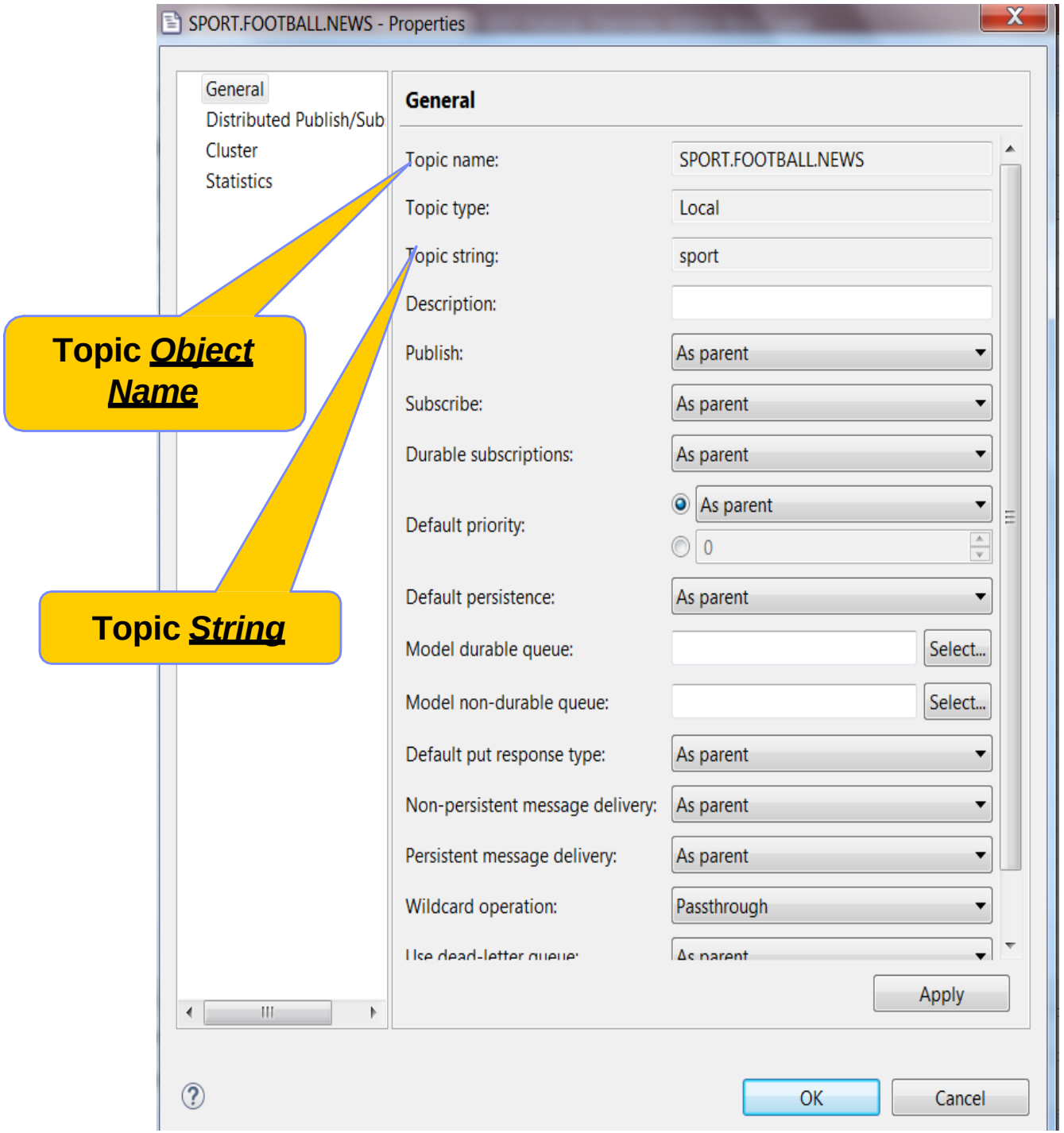


Publish/Subscribe in the IBM MQ Explorer



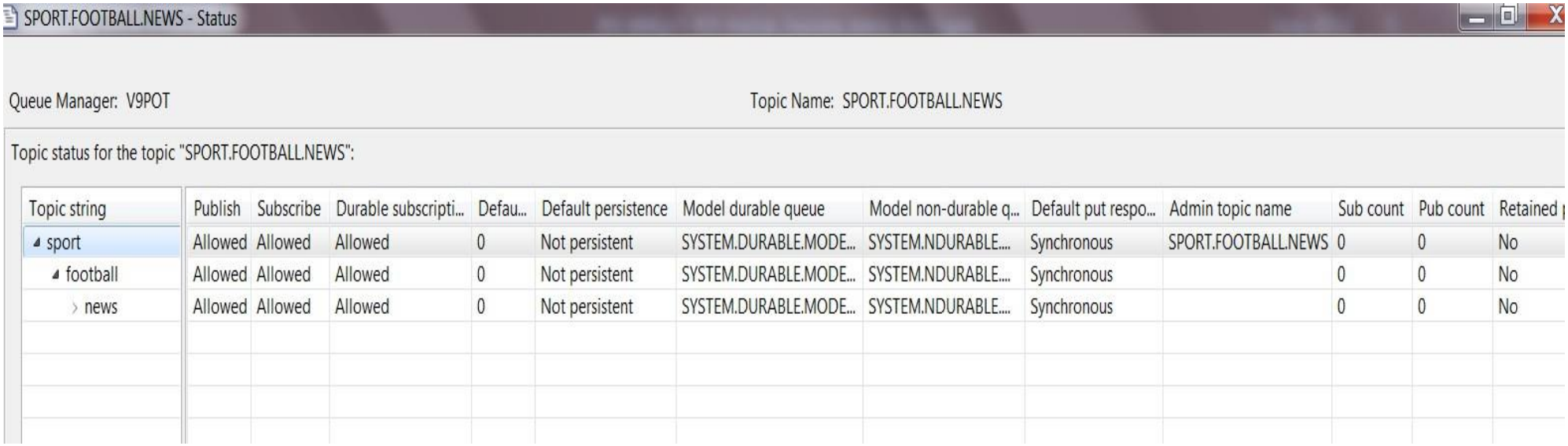
Topic Objects

- Topic Object 는 IBM MQ 에서 게시 / 구독 작업을 관리하는 중요한 객체
- Topic String 은 게시자와 구독자를 연결하며, 길이에 제한이 없는 문자열로 구성
- Topic Object 는 다양한 속성을 통해 메시지 전달 방식을 제어, Topic Tree 구조와 밀접하게 연관



Topic Status – current usage of topics

- DISPLAY TPSTATUS(*topic-string*) 명령을 사용하여 특정 Topic 의 상태를 확인
  - 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



Topic string	Publish	Subscribe	Durable subscripti...	Defau...	Default persistence	Model durable queue	Model non-durable q...	Default put respo...	Admin topic name	Sub count	Pub count	Retained
sport	Allowed	Allowed	Allowed	0	Not persistent	SYSTEM.DURABLE.MODE...	SYSTEM.NDURABLE....	Synchronous	SPORT.FOOTBALL.NEWS	0	0	No
football	Allowed	Allowed	Allowed	0	Not persistent	SYSTEM.DURABLE.MODE...	SYSTEM.NDURABLE....	Synchronous		0	0	No
news	Allowed	Allowed	Allowed	0	Not persistent	SYSTEM.DURABLE.MODE...	SYSTEM.NDURABLE....	Synchronous		0	0	No

Notes - Topic Status – current usage of topics

N

O

T

E

S

Use DISPLAY TPSTATUS to display the status of one or more topic nodes in a topic tree.

The value of TPSTATUS determines which topic nodes are returned on the call to DISPLAY TPSTATUS. The TPSTATUS attribute requires a topic string value. This value may be one of the following:

A specific topic string value, e.g. DIS TPS('Sports/Football') – returns just the 'Sports/Football' node

A topic string containing a '+' wildcard, e.g. DIS TPS('Sports/Football/+') – returns all immediate children of the 'Sports/Football' node

A topic string containing a '#' wildcard, e.g. DIS TPS('Sports/Football/#') – returns all descendants of 'Sports/Football' in the topic tree, plus the 'Sports/Football' node itself

A topic string containing multiple wildcards, e.g. DIS TPS('Sports/+/teams/#') – returns any immediate child of 'Sports' which has a child of 'teams', plus the all descendants of that node

Note that the use of wildcards should follow the rules as laid out in INSERT REFERENCE TO BOOK CONTAINING INFO ON HOW WILDCARD CHARS WORK AS IN 99966. An asterisk is not a supported wildcard for the TPSTATUS attribute.

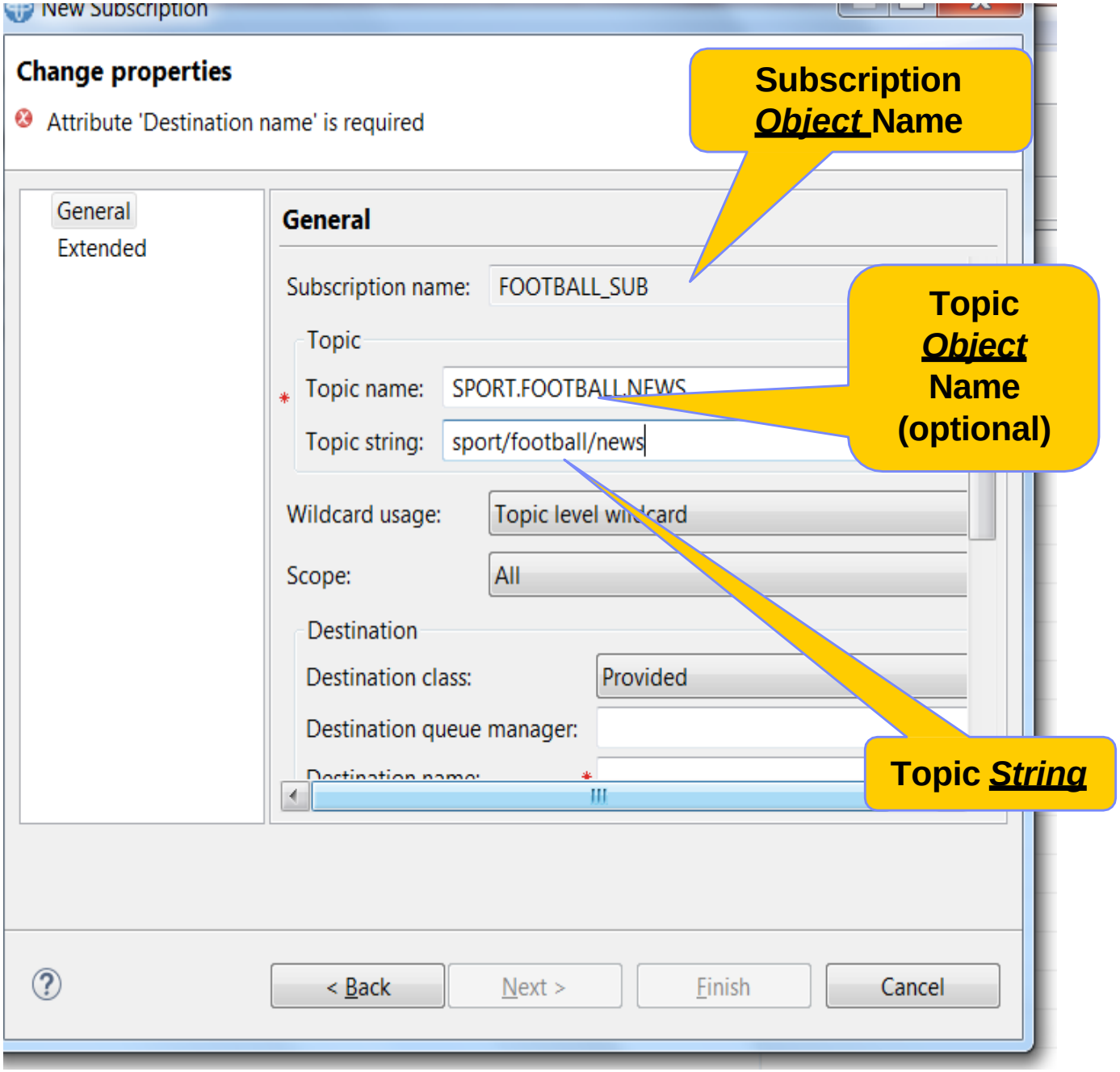
To return a list of all root-level topics, use DIS TPS('+')

To return a list of all topics in the topic tree (depending on your configuration, this may return a large volume of data), use DIS TPS('#')

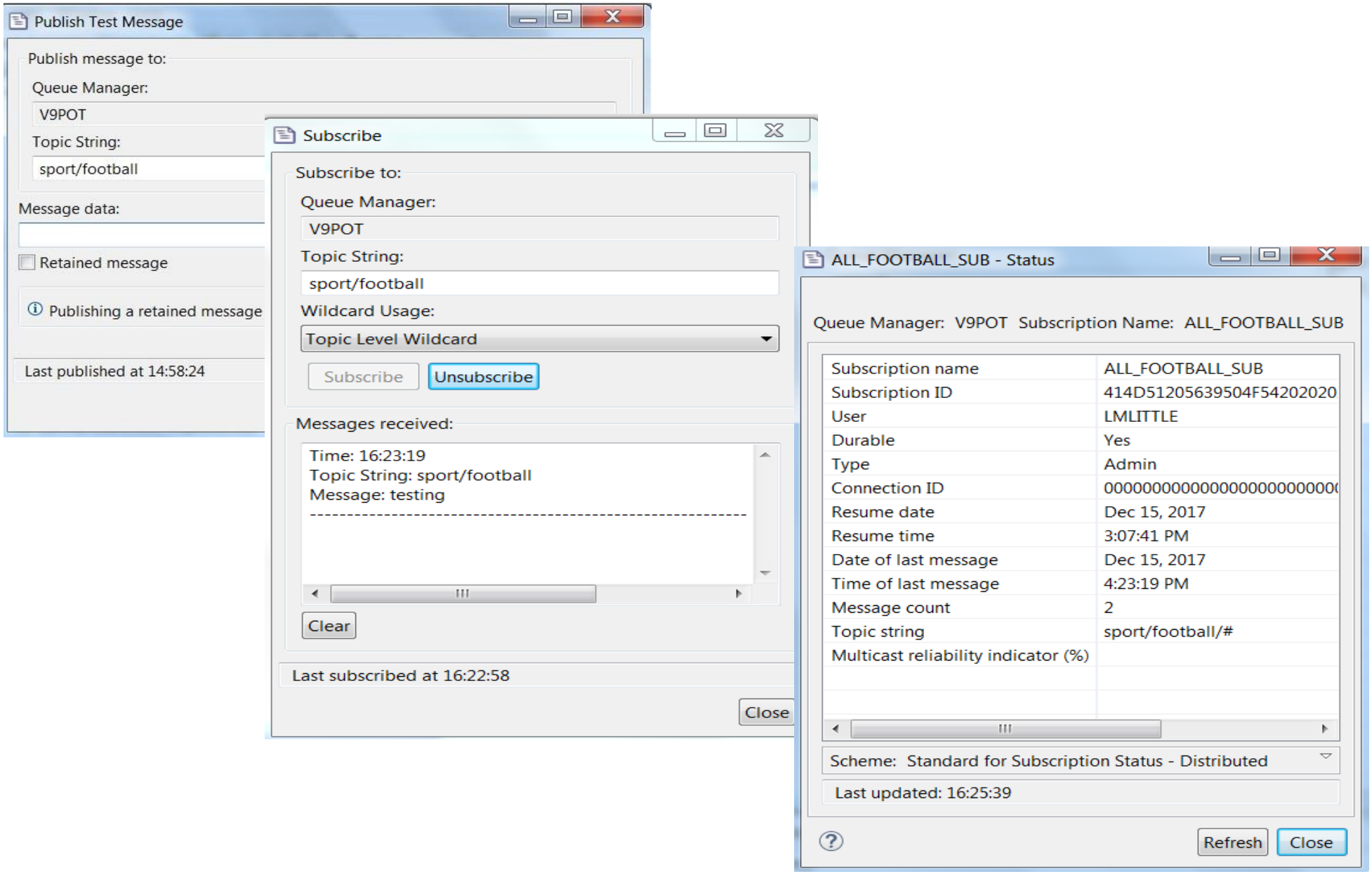
The list of topics that is returned may be further modified by the use of a filter on TOPICSTR, e.g. DIS TPS('Sports/Football/+') WHERE(TOPICSTR LK 'Sports/Football/L*') will return all immediate child nodes of the node 'Sports/Football/', which begin with the letter 'L'.

Administered Subscriptions

- 반드시 이름을 가져야 함.
- 선택적으로 특정 Topic Object 를 참조
- 반드시 Topic String 이 필요
- 와일드 카드 사용
- 지속성 여부
- Destination 관리
  - Managed / Unmanaged



Testing Publish and Subscribe



Benefits of Publish/Subscribe with IBM MQ

- 애플리케이션 간의 느슨한 결합
- 메시지 교환 패턴의 유연성
  - One-to-one
  - One-to-many
  - Many-to-one
  - Many-to-many
- 동적 또는 관리형 Topic 생성
- 구독자의 유연한 메시지 수신
  - 다중 주제 등록
  - 와일드 카드 사용
  - 셀렉터 사용