Basic usage of clustered topics in IBM MQ 9.3

https://www.ibm.com/support/pages/node/593771

Date last updated: 20-Mar-2024

Angel Rivera
IBM MQ Support
https://www.ibm.com/products/mq/support
Find all the support you need for IBM MQ

+++ Objective

This tutorial focuses on the basic usage of clustered topics in IBM MQ 9.3.

First, the scenario of a non-clustered topic is explored regarding publish / subscribe (Pub/Sub).

Then the topic is included in a cluster and the corresponding Pub/Sub changes in behavior are explored.

This tutorial is based on the small cluster that was configured in the following tutorial:
https://www.ibm.com/support/pages/node/611561
Setup of a cluster and basic usage of clustered queues in IBM MQ 9.3

The chapters are:

+++ Summary of steps to add a topic to a cluster

When using runmqsc, you need to use the CLUSTER attribute for the TOPIC object:

```
  ALTER TOPIC(T1) CLUSTER(CLUSTER1)
```

+++ References

1) The contents of the following free "redbook" is still applicable to MQ 9.3.

http://www.redbooks.ibm.com/abstracts/SG247583.html?Open
IBM MQ V7.0 Features and Enhancements (SG24-7583)

Chapter 4. Publish/Subscribe integration
4.4.1 Pub/Sub Cluster topology (Page 56)

+ Begin excerpt

A Pub/Sub Cluster uses MQ Cluster technology that connects queue managers together via cluster channels.

An MQ Cluster becomes a Pub/Sub Cluster by the definition of at least one clustered topic within the cluster.

Although the clustered topic is created on one queue manager, the definition is pushed out to all queue managers in the cluster using the same advertising method as ordinary clustered queues.

All publications that are made to the clustered topic are sent to all queue managers in the cluster that have active subscriber applications connected.

Subscribing to clustered topics
When an application subscribes to a topic that resolves to a clustered topic, MQ creates a proxy subscription and sends it from the application's connected queue manager to all other queue managers in the MQ Cluster on which the clustered topic object is defined.
If a queue manager on which the clustered topic object is defined becomes unavailable, the subscription will remain in place for up to 30 days, so that normal Pub/Sub activity is restored when the queue manager becomes available.

Publishing to clustered topics
Publications in a Publish/Subscribe cluster are sent to those queue managers for which proxy subscriptions have been received.

+ end

2) Technote:

https://www.ibm.com/support/pages/node/322483
Authorizations needed for non-mqm users to publish and subscribe to Topics in MQ

Summary

If you want to let the users in the groups "editors" and "journalists" connect to the queue manager WMQ7:

```
  setmqaut -m WMQ7 -t qmgr -g editors +connect +inq +dsp
  setmqaut -m WMQ7 -t qmgr -g journalists +connect +inq +dsp
```

If you want to let the users in the group "editors" subscribe to topic “DELI” on Queue Manager WMQ7 and to resume durable subscriptions:

```
  setmqaut -m WMQ7 -n DELI -t topic -g editors +sub +resume
```

If you want to allow users in the group "journalists" to publish to the topic:

```
  setmqaut -m WMQ7 -n DELI -t topic -g journalists +pub
```

3) Technote:

https://www.ibm.com/support/pages/node/322477
Sample scenarios to show Pub/Sub of Clustered Topics in IBM MQ

The purpose of the technote is to provide some sample scenarios for using a topic in a cluster and to perform publish and subscribe (Pub/Sub) operations on that topic, from the local queue manager and the remote queue managers.

This technote contains the following sections:

- Testcase configuration details

- Scenario 1: Publishing from QM1 (where the topic is local) - subscribing from all the Queue Managers

- Scenario 2: Publishing from QM2 (where the topic is NOT local) - subscribing from all the Queue Managers

- Looking at the TOPIC STATUS from QM1 and QM2

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++ Chapter 1: Configuration of the repositories
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

This tutorial uses the small cluster with 5 queue managers that was already created in the following tutorial:

https://www.ibm.com/support/pages/node/611561
Setup of a cluster and basic usage of clustered queues in IBM MQ 9.3

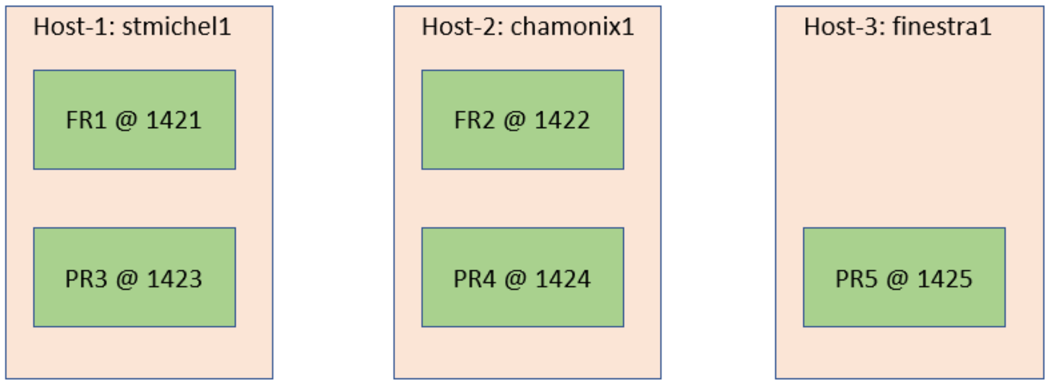
It is a best practice to have the Full Repositories (FRs) running at one of the newest version.release and a recent fix pack level, in this case 9.3.0.16 LTS.
Both FRs should be at the same version.release and fix pack level.

```
Host-1, name stmichel1, Linux, MQ 9.3.0.16 LTS
QMNAME(FR1)     Port 1421    Full Repository 1
QMNAME(PR3)     Port 1423    Partial Repository 3

Host-2, name chamonix, Linux, MQ 9.3.0.16 LTS
QMNAME(FR2)     Port 1422    Full Repository 2
QMNAME(PR4)     Port 1424    Partial Repository 4

Host-3, name finestra1, Windows, MQ 9.3.5 CD
QMNAME(PR5)     Port 1425    Partial Repository 5
```

The topology looks like this:



++

+++ Chapter 2: Adding a local topic (not clustered yet), using non-durable subscribers

++

In this scenario we will explore the addition of local topics and how local publishers and local subscribers can handle messages.

Furthermore, a variation of the scenario is shown that covers the failure from a remote subscriber to receive published messages.

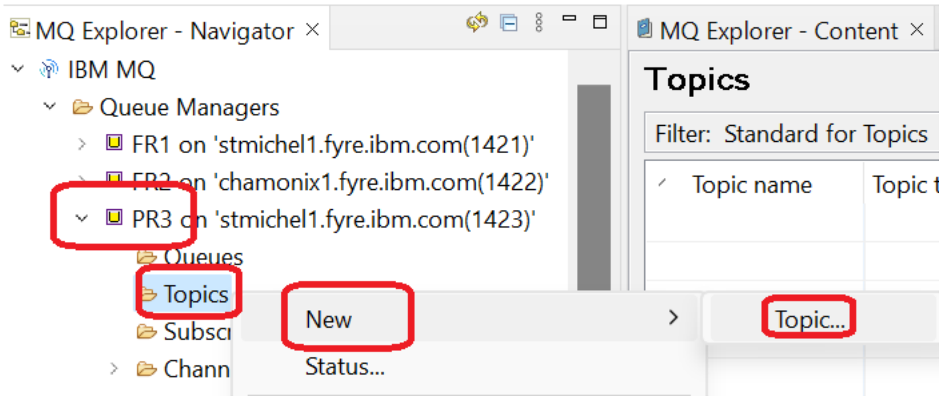
These subscribers are “non-durable” (“dynamic”, “on-the-fly”) and the tool amqssub is used to handle them. That is, they are not permanent subscribers.

In contrast, the "durable" subscribers will be defined via the administrative tools such as the MQ Explorer or runmqsc in Chapter 4.

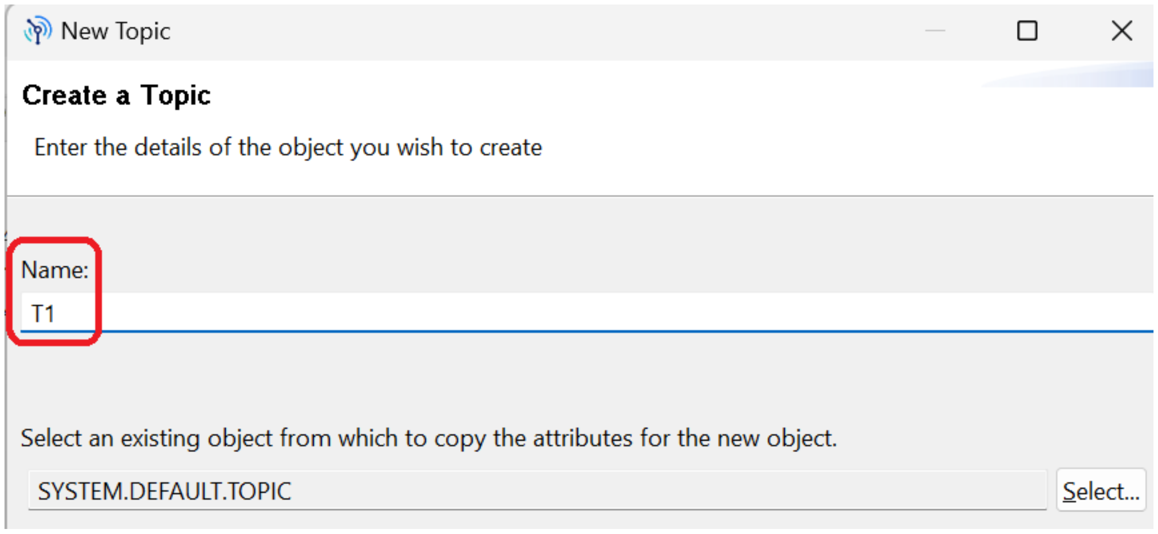
On the Partial Repository PR3, let's create 2 topic objects. For the moment, they are not yet incorporated into the cluster:
- one called "T1" that has a topic string "fruits"
- another called "T1.B" that has a topic string "fruits/oranges"

Let's start with topic object "T1".

In the left navigation panel, select queue manager "PR3", then right-click to show the context menu and select:
  Topics > New > Topics...


MQ Explorer - Navigator
IBM MQ
Queue Managers
FR1 on 'stmichel1.fyre.ibm.com(1421)'
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
Queues
Topics
Subscri
Chann
New
Status...
Topic...
MQ Explorer - Content
Topics
Filter: Standard for Topics
Topic name
Topic t

Enter the object name:
  T1
Click Next



New Topic

Create a Topic

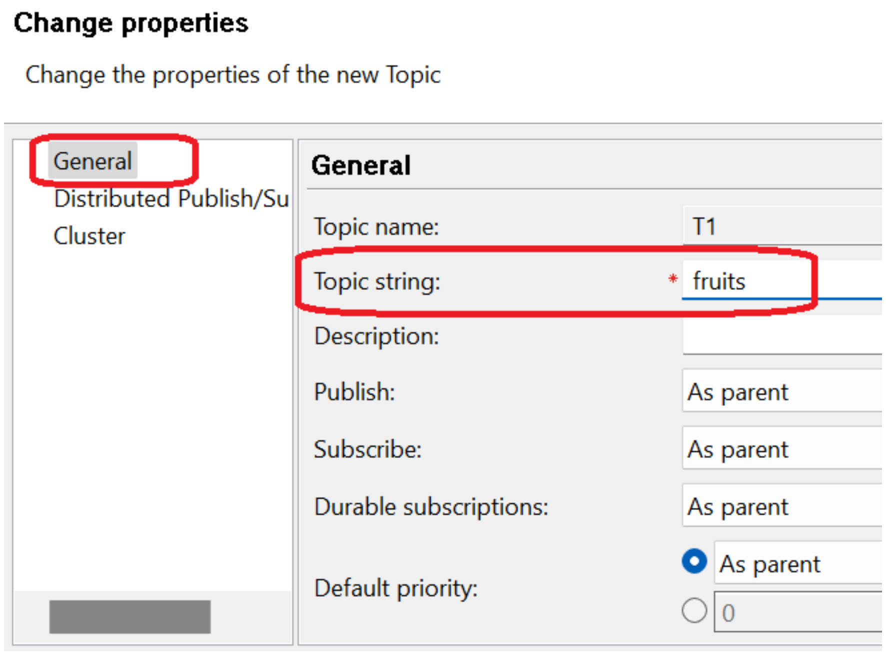
Enter the details of the object you wish to create

Name:
T1

Select an existing object from which to copy the attributes for the new object.

SYSTEM.DEFAULT.TOPIC

Enter the <u>**topic string**</u>:
  fruits



Change properties

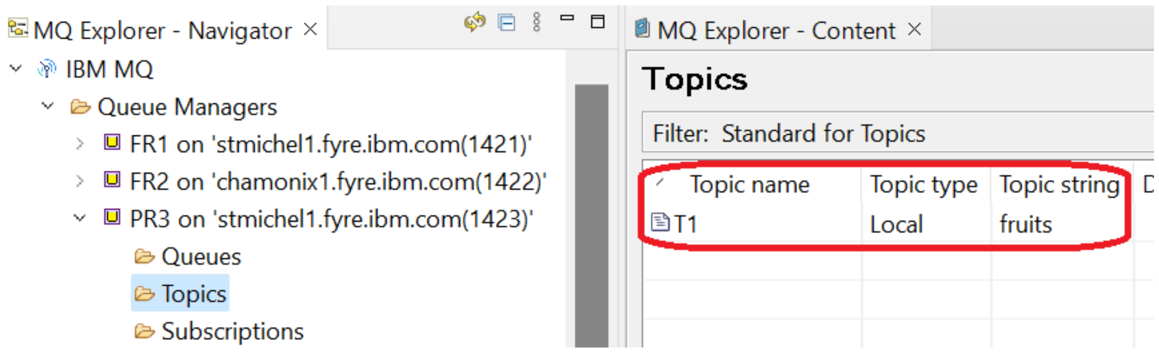
Change the properties of the new Topic

General

Topic name: T1
Topic string: * fruits
Description:
Publish: As parent
Subscribe: As parent
Durable subscriptions: As parent
Default priority: ● As parent ○ 0

Click Finish

Notice the addition of the Topic object named "T1" into the list of topics.


MQ Explorer - Navigator
IBM MQ
Queue Managers
FR1 on 'stmichel1.fyre.ibm.com(1421)'
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
Queues
Topics
Subscriptions
MQ Explorer - Content
Topics
Filter: Standard for Topics

Topic name	Topic type	Topic string
T1	Local	fruits

+ Using runmqsc

  To create topic object "T1.B" with topic string "fruits/oranges".

```
mqm@stmichel1.fyre.ibm.com: /home/mqm
$ runmqsc PR3
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager PR3.

define topic(T1.B) topicstr('fruits/oranges')
     1 : define topic(T1.B) topicstr('fruits/oranges')
AMQ8690I: IBM MQ topic created.
```

At this point the topic objects T1 and T1.B are available only on PR3 and they are not part of a cluster.

Because the topic T1 is local to PR3 and it is not yet a clustered topic, then we can only do Publish and Subscribe activities when the applications are directly connected to PR3.

If a subscriber connects to another queue manager, such as PR4 and tries to subscribe (receive messages) for this topic which are published by the Publisher connected to PR3, then the subscriber will NOT get any messages, because the topic is not a clustered topic and PR4 does NOT know about the topic.

+ Expected behavior:

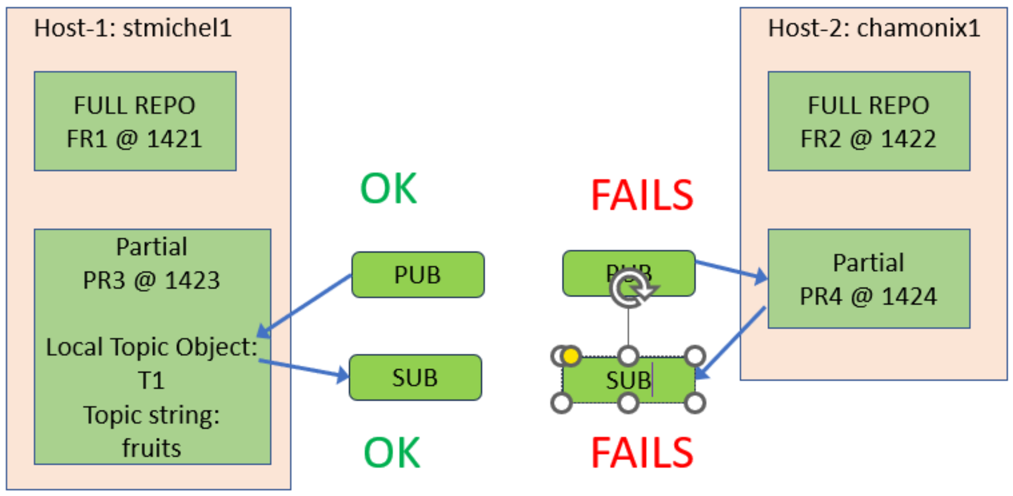
- OK. An application such as "amqspub" can publish a message to the local topic T1 when connected to PR3.

- OK. A subscriber application such as "amqssub" can get a message from the local topic T when connected to PR3.

- FAILED. Another application CAN publish a message into T1 when connected to another queue manager, such as PR4, but it is NOT really the same topic!

- FAILED. Another subscriber application CANNOT get a message from T1 when connected to another queue manager, such as PR4.

The following picture offers a good summary of the topology and the behavior:



++ Let's test the Publishing and Subscribe capabilities without a clustered topic

Open 2 command prompts and ensure that the MQ environment is properly setup for the desired version.

```
mqm@stmichel1.fyre.ibm.com: /home/mqm
$ . /opt/mqm/bin/setmqenv -n Installation1
```

+ Window 1: (Subscriber in Linux stmichel1 for PR3)

Use the amqssub sample to subscribe to the topic string "fruits" in PR3

The subscribers that are defined by this utility are “non-durable”, that is, they are “dynamic” or “on-the-fly” and when the application ends, the subscription also ends.

```
mqm@stmichel1.fyre.ibm.com: /home/mqm
$ amqssub fruits PR3
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time
```

+ Window 2: (Publisher in Linux stmichel1 - PR3)

Use the amqspub sample to publish to the topic string "fruits" in PR3.
This topic string is associated with the topic object T1.

```
$ amqspub fruits PR3
Sample AMQSPUBA start
target topic is fruits
test-1 (press ENTER)
(press ENTER to end)
Sample AMQSPUBA end
```

+ Window 1: (Subscriber in Linux stmichel1 - PR3)

Notice that the message "test-1" is received by the subscriber connected to PR3

```
$ amqssub fruits PR3
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time
message <test-1>
Calling MQGET : 30 seconds wait time
no more messages
Sample AMQSSUBA end
```

++ Variation – trying to publish in PR3 but subscribe from another host PR4

The next step in the scenario is to have a publisher or a subscriber connected to another queue manager, such as PR4.

The subscriber will not get a message that is published in PR3.

+ Window 3: (Subscriber in chamonix1 - PR4)

```
$ amqssub fruits PR4
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time
```

+ Window 2: (Publisher in Linux stmichel1 - PR3)

```
$ amqspub fruits PR3
Sample AMQSPUBA start
target topic is fruits
test-2 (press ENTER)
(press ENTER to end)
Sample AMQSPUBA end
```

Note that the message was published in PR3, but the message is NOT received in PR4

```
$ amqssub fruits PR4
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time
no more messages
Sample AMQSSUBA end
```

++ Variation – trying to publish in PR4 and subscribe in host PR4

Notice that because the topic T1 is NOT yet a clustered topic, then when a publisher connect to PR4 tries to publish to the topic T1 (string 'fruits') in PR4, then the publishing will be successful, but this is not the same topic that is going to be clustered.
That is, there is a local topic string "fruits" in PR3 and a totally different local topic string also called "fruits" in PR4 and these topic strings are NOT related.

A local topic definition overrides a remotely defined cluster topic definition of the same name. Creation of multiple definitions of the same cluster topic on different queue managers in a cluster is also possible. Both of these scenarios require some caution.

Having multiple definitions of the same cluster topic object in a cluster introduces the possibility of different properties defined on each.
It is not easy to determine which version of the topic definition is seen by each queue manager in the cluster and it is therefore hard to determine the expected behavior.

+++

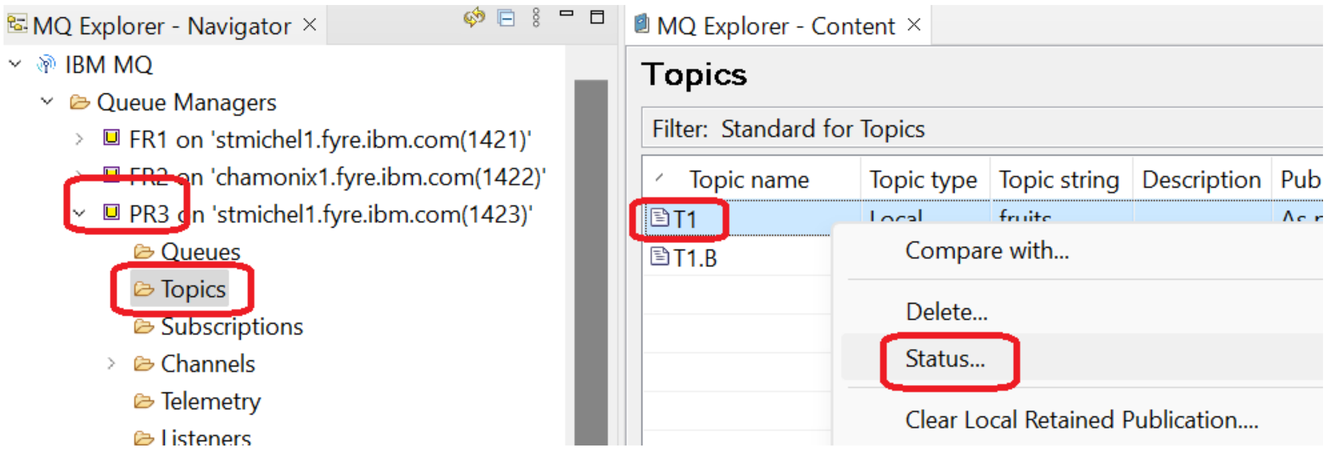
+++ Chapter 3: Looking at the status of a topic

+++

Let's take a look at the status of the Topic:

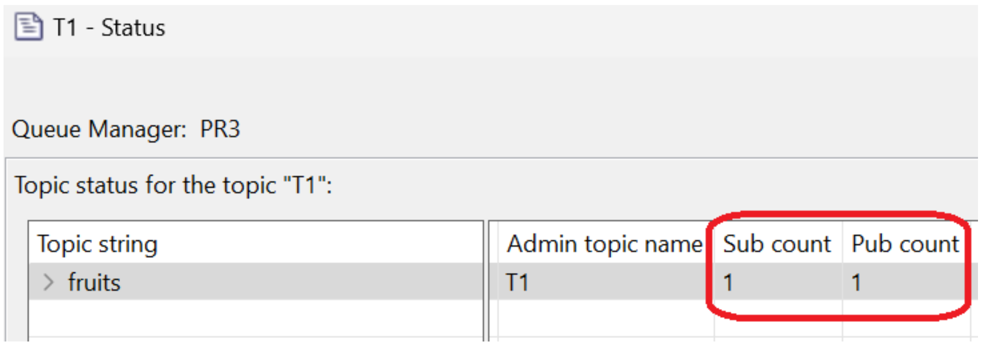
a) Using MQ Explorer

In MQ Explorer, select the desired topic, and then right click to show the context menu.
Select "Status"



You will see the status window.

Select the row for "fruits" and scroll to the right, in order to see the fields:
  Sub count
  Pub count


T1 - Status

Queue Manager: PR3

Topic status for the topic "T1":

Topic string	Admin topic name	Sub count	Pub count
fruits	T1	1	1

Notice that the "Sub count" is 1 (number of subscribers for the topic)
And that the "Pub count" is also 1 (number of publishers)

b) Via runmqsc

You can use runmqsc to show the status of the topic string.

Notice that you cannot use the Topic Object (T1), because "tpstatus" expects a Topic String.

```
display tpstatus(T1)
AMQ8472E: IBM MQ topic string not found
```

Instead, you need to use the Topic String ('fruits'):

```
display tpstatus('fruits')
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)                        ADMIN(T1)
   CLUSTER( )
   COMMINFO(SYSTEM.DEFAULT.COMMINFO.MULTICAST)
   MDURMDL(SYSTEM.DURABLE.MODEL.QUEUE)
   MNDURMDL(SYSTEM.NDURABLE.MODEL.QUEUE)
   CLROUTE(NONE)                           DEFPSIST(NO)
   DEFPRTY(0)                              DEFPRESP(SYNC)
   DURSUB(YES)                             PUB(ENABLED)
   SUB(ENABLED)                            PMSGDLV(ALLDUR)
   NPMSGDLV(ALLAVAIL)                      RETAINED(NO)
   MCAST(DISABLED)                         PUBCOUNT(1)
   SUBCOUNT(0)                             PUBSCOPE(ALL)
   SUBSCOPE(ALL)                           USEDLQ(YES)
```

The fields PUBCOUNT and SUBCOUNT show the number of publishers and subscribers.

To view more details on the publishers:

```
display tpstatus('fruits') type(pub)
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)                        LPUBDATE( )
   LPUBTIME( )
   ACTCONN(414D5143505233202020202020202020205ADCF16500B70040)
   MCASTREL( , )                           NUMPUBS(0)
```

To view more details on the subscribers:

```
display tpstatus('fruits') type(sub)
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)
   SUBID(414D5120505233202020202020202020205ADCF16502B50040)
   SUBUSER(mqm)                            RESMDATE(2024-03-20)
   RESMTIME(10:49:17)                      LMSGDATE( )
   LMSGTIME( )
   ACTCONN(414D5143505233202020202020202020205ADCF16500B50040)
   DURABLE(NO)                             SUBTYPE(API)
   MCASTREL( , )                           NUMMSGS(0)
```

Notice that this subscriber is NON-DURABLE: DURABLE(NO)
... and that it was created on-the-fly by the API calls inside an application: SUBTYPE(API)

You can use the information provided in the following technote to find the name of the application:

https://www.ibm.com/support/pages/node/620915
Identifying the name of an application that is subscribed to a topic

The summary of the technote is that you have to use the last 16 bytes of the ACTCONN in the following runmqsc command:

From the value of the ACTCONN (active connection) from the output of "tpstatus type()", copy the last 16 bytes, such as:

```
   ACTCONN(414D5143505233202020202020202020205ADCF16500B70040)
```

... and paste them inside the parenthesis for:
DISPLAY CONN(right-most 16-bytes from ACTCONN)

In this case, for the publisher:

```
DISPLAY CONN(5ADCF16500B70040)
AMQ8276I: Display Connection details.
   CONN(5ADCF16500B70040)
   EXTCONN(414D5143505233202020202020202020)
   TYPE(CONN)
   PID(538588)                             TID(1)
   APPLDESC( )                             APPLTAG(amqspub)
   APPLTYPE(USER)                          ASTATE(NONE)
   CHANNEL( )                              CLIENTID( )
   CONNAME( )                              CONNOPTS(MQCNO_SHARED_BINDING)
   USERID(mqm)                             UOWLOG( )
   UOWSTDA( )                              UOWSTTI( )
   UOWLOGDA( )                             UOWLOGTI( )
   URTYPE(QMGR)
   EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
   QMURID(0.0)                             UOWSTATE(NONE)
   CONNTAG(MQCT5ADCF16500B70040PR3_2024-03-02_09.58.47amqspub)
```

++

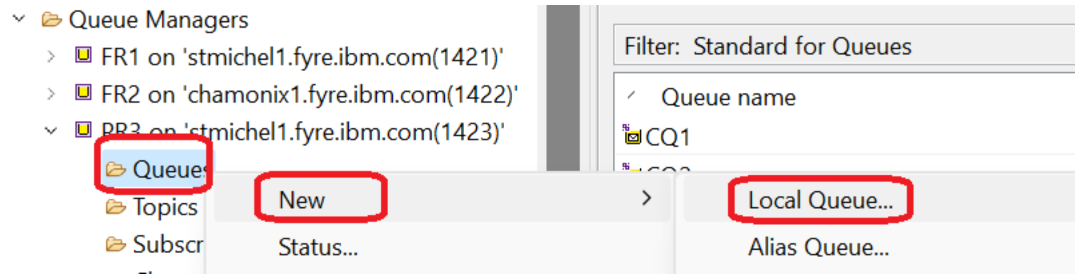
+++ Chapter 4: Creating an administrative, durable subscriber

++

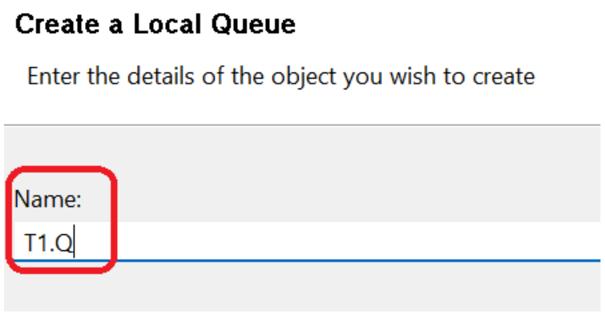
The scenario in this chapter is very similar to the one in Chapter 2.
The difference is that in this chapter we will use "durable" subscriptions that are defined administratively.

One convenient way for us to see the messages that are received by a durable subscription is to define a "provided" destination, which means that we will create a queue that will be used to store the messages received by the subscriber.

Create local queue: T1.Q


Queue Managers
FR1 on 'stmichel1.fyre.ibm.com(1421)'
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
Queues
Topics
Subscr
New
Status...
Local Queue...
Alias Queue...
Filter: Standard for Queues
Queue name
CQ1

...



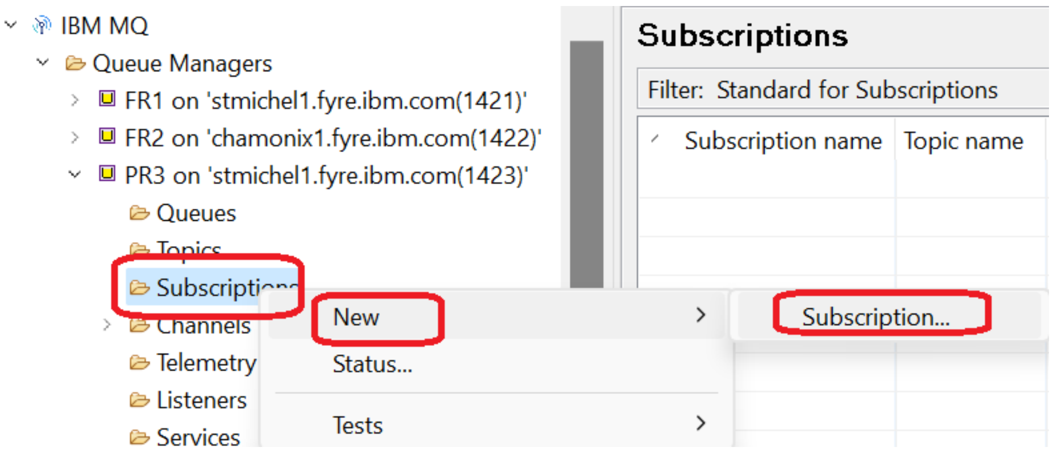
Create a Local Queue

Enter the details of the object you wish to create

Name:
T1.Q

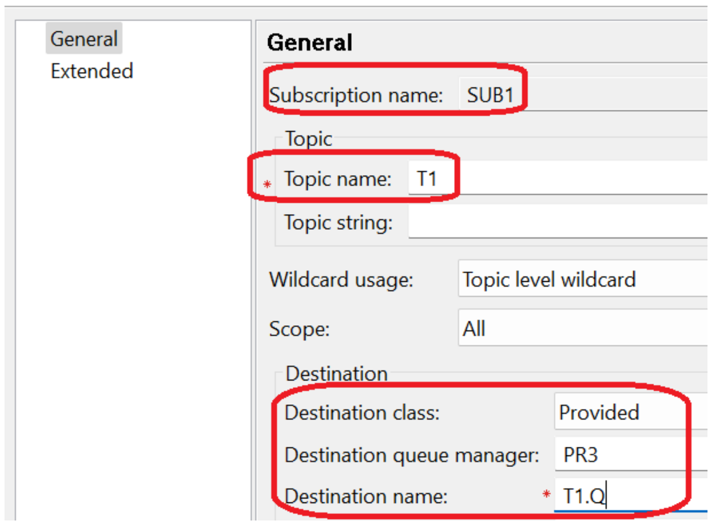
Follow the prompts and accept the defaults.

Now create durable subscriber: SUB1



Follow the prompts and specify:

Name: SUB1
Topic (object) Name: T1
Destination class: Provided
Destination queue manager: PR3
Destination name: T1.Q (this is the dedicated queue that was created in the previous step)



Subscription name: SUB1
Topic name: T1
Topic string:
Wildcard usage: Topic level wildcard
Scope: All
Destination class: Provided
Destination queue manager: PR3
Destination name: T1.Q

+ Using runmqsc

In runmqsc you can use the following to create similar objects:

```
DEFINE QLOCAL(T1.Q2)
```

```
DEFINE SUB('SUB2') TOPICSTR('fruits') DESTCLAS(PROVIDED) DEST(T1.Q2) DESTQMGR(PR3)
```

++ Let's test the Publishing and Subscribe capabilities without a clustered topic

Open 2 command prompts and ensure that the MQ environment is properly setup for the desired version.

Window 1: (Publisher in Linux stmichel1 - PR3)

Use the amqspub sample to publish to the topic string "fruits" in PR3

```
$ amqspub fruits PR3
Sample AMQSPUBA start
target topic is fruits
test-A (press ENTER)
(press ENTER to end)
Sample AMQSPUBA end
```

Window 2: (Browser in Linux stmichel1 - PR3)

Use the amqsbcg sample to browse the queue T1.Q in PR3

```
$ amqsbcg T1.Q PR3
AMQSBCG0 - starts here
**********************
 MQOPEN - 'T1.Q'
 MQGET of message number 1
****Message descriptor****
...
****   Message      ****

mqm@stmichel1.fyre.ibm.com: /home/mqm
(1007) amqsbcg T1.Q PR3

AMQSBCG0 - starts here
**********************

 MQOPEN - 'T1.Q'


 MQGET of message number 1, CompCode:0 Reason:0
****Message descriptor****

  StrucId  : 'MD  '  Version : 2
  Report   : 0  MsgType : 8
  Expiry   : -1  Feedback : 0
  Encoding : 546  CodedCharSetId : 1208
  Format : 'MQSTR   '
  Priority : 0  Persistence : 0
  MsgId : X'414D51205052332020202020202020205ADCF16501BE0040'
  CorrelId : X'414D51205052332020202020202020205ADCF16592B90040'
...
  ** Origin Context
  PutApplType    : '26'
  PutApplName    : 'PR3                         '
  PutDate  : '20240320'    PutTime  : '18093978'
  ApplOriginData : '    '
...
****   Message      ****
 length - 6 of 6 bytes
00000000:  7465 7374 2D41                           'test-A          '
```

+ Variation: defining a durable subscriber in remote queue manager

Remember that at this point, the topic 'fruits' is local to PR3 and it is not yet a clustered topic. This means that if there are subscribers (durable, non-durable) in a remote queue manager, those subscribers will NOT receive the messages published for the topic in the queue manager PR3.

In the remote queue manager PR4 (in Linux chamonix1), let's define a durable subscriber for the topic and its corresponding destination queue:

Window 1: (Linux chamonix1 - PR4)

```
runmqsc PR4
DEFINE QLOCAL(T1.Q3)
DEFINE SUB('SUB3') TOPICSTR('fruits') DESTCLAS(PROVIDED) DEST(T1.Q3) DESTQMGR(PR4)
```

Let's publish a message in PR3 for this topic.

Window 1: (Publisher in Linux stmichel1 - PR3)

Use the amqspub sample to publish to the topic string "fruits" in PR3

```
$ amqspub fruits PR3
Sample AMQSPUBA start
target topic is fruits
test-B (press ENTER)
(press ENTER to end)
Sample AMQSPUBA end
```

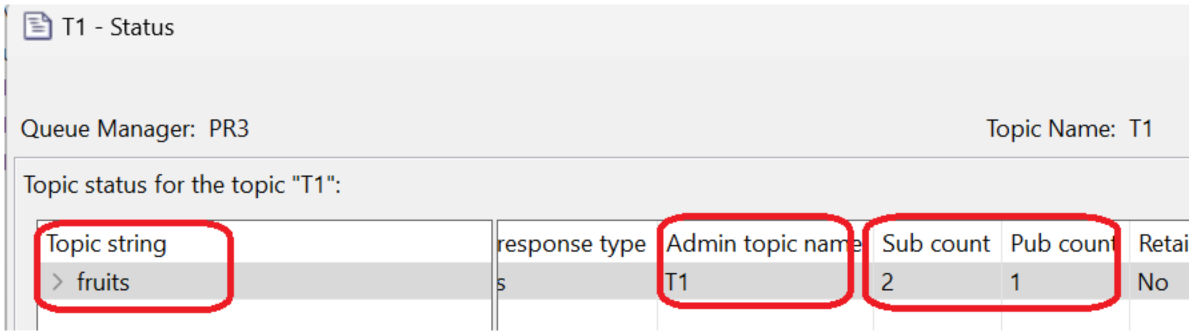
Window 3: (Browser in Linux - PR4)

Use the amqsbcg sample to browse the queue T1.Q3 in PR4

```
$ amqsbcg T1.Q3 PR4
AMQSBCG0 - starts here
**********************
 MQOPEN - 'T1.Q3'
 No more messages
 MQCLOSE
```

+ Let's review the status

The status for the Topic 'fruits' indicates that there are 2 subscribers and 1 publisher:



T1 - Status

Queue Manager: PR3 Topic Name: T1

Topic status for the topic "T1":

Topic string	response type	Admin topic name	Sub count	Pub count	Retai
› fruits	s	T1	2	1	No

Let's display those subscribers who received messages:

```
DISPLAY SBSTATUS(*) WHERE(NUMMSGS GT 0)
```

Ignore the following subscription:
  SUB(QMgrName SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS

```
AMQ8099I: IBM MQ subscription status inquired.
   SUB(SUB1)
   SUBID(414D5120505233202020202020202020205ADCF16592B90040)
   NUMMSGS(2)

AMQ8099I: IBM MQ subscription status inquired.
   SUB(SUB2)
   SUBID(414D5120505233202020202020202020205ADCF16508B20040)
   NUMMSGS(3)
```

To find out more details on the destination queue for the subscription SUB1:

```
DISPLAY SUB(SUB1) DEST
AMQ8096I: IBM MQ subscription inquired.
   SUBID(414D5120505233202020202020202020205ADCF16592B90040)
   SUB(SUB1)                                DEST(T1.Q)
```

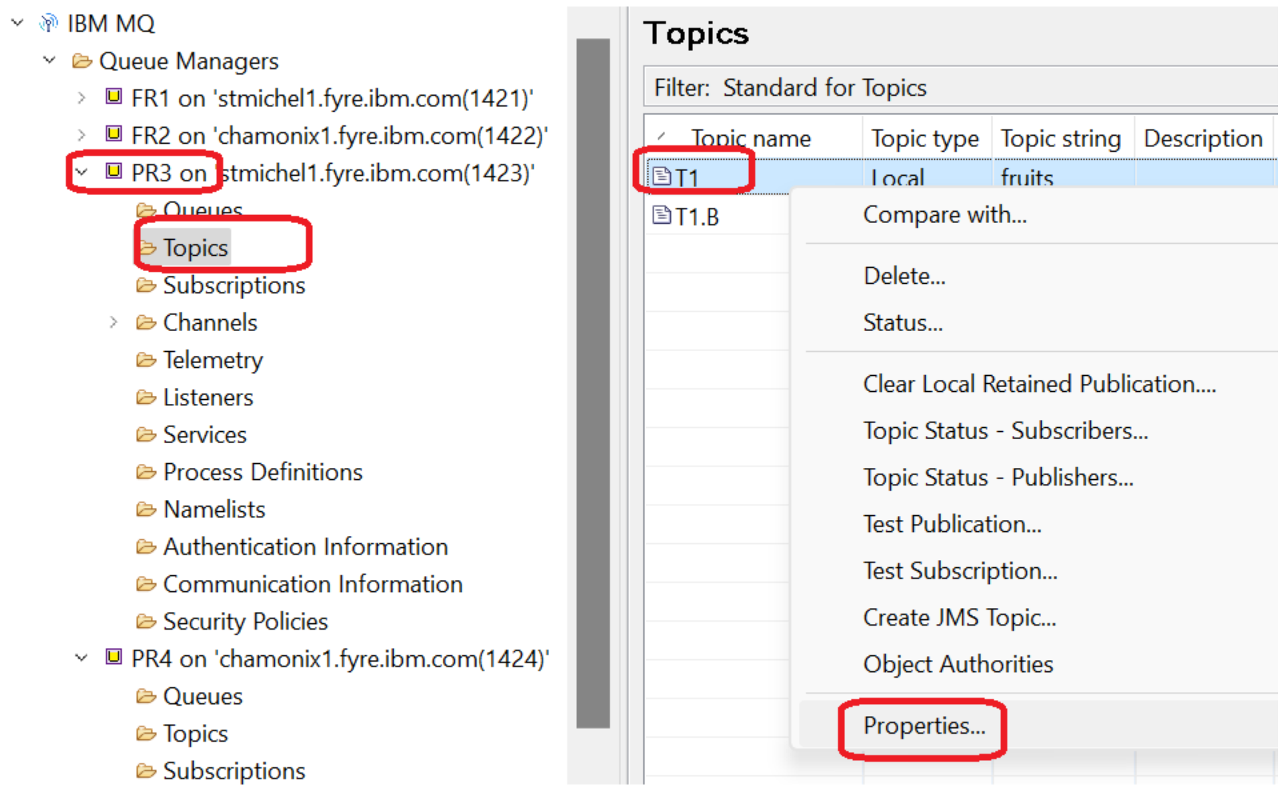
Now you can display attributes for the destination queue:

```
DISPLAY QL(T1.Q) CURDEPTH
AMQ8409I: Display Queue details.
   QUEUE(T1.Q)                              TYPE(QLOCAL)
   CURDEPTH(1)
```

++
+++ Chapter 5: Altering the local topic to be a clustered topic
++

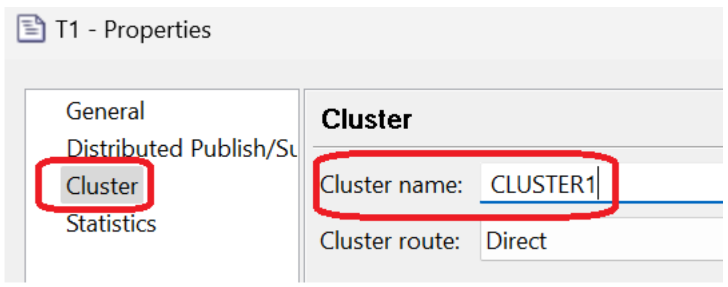
In order for a subscriber that is connected to PR4, to get a message published by a publisher connected to PR3, it is necessary to add the topic T1 into the cluster.

Select the topic "T1" from the partial repository queue manager PR3, right click to show the Properties.


IBM MQ
Queue Managers
FR1 on 'stmichel1.fyre.ibm.com(1421)'
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
Queues
Topics
Subscriptions
Channels
Telemetry
Listeners
Services
Process Definitions
Namelists
Authentication Information
Communication Information
Security Policies
PR4 on 'chamonix1.fyre.ibm.com(1424)'
Queues
Topics
Subscriptions
Topics
Filter: Standard for Topics
Compare with...
Delete...
Status...
Clear Local Retained Publication....
Topic Status - Subscribers...
Topic Status - Publishers...
Test Publication...
Test Subscription...
Create JMS Topic...
Object Authorities
Properties...

Topic name	Topic type	Topic string	Description
T1	Local	fruits	
T1.B			

In the tab "Cluster" for the Properties, enter the cluster name: CLUSTER1
Click OK.



Cluster name: CLUSTER1
Cluster route: Direct

If you want to use runmqsc, then you will need to alter the topic and specify the cluster name, as follows:

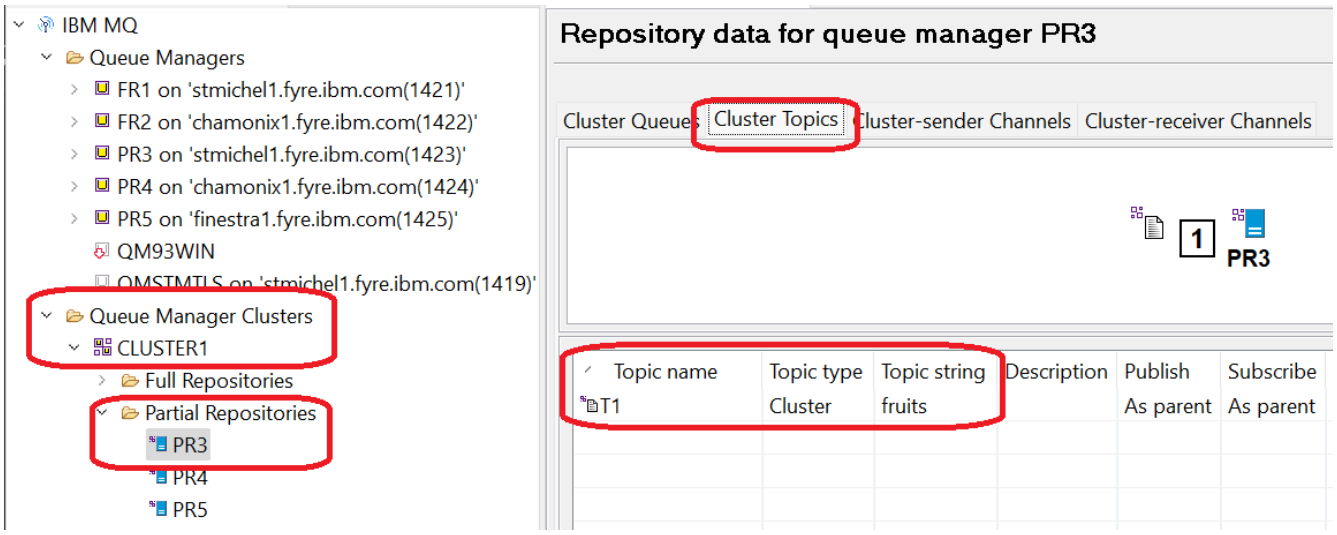
```
ALTER TOPIC(T1) CLUSTER(CLUSTER1)
```

Let's take a look at the Cluster objects.

Expand the folder for "Queue Manager Clusters" then the cluster "CLUSTER1".

Expand "Partial Repositories" and select "PR3"

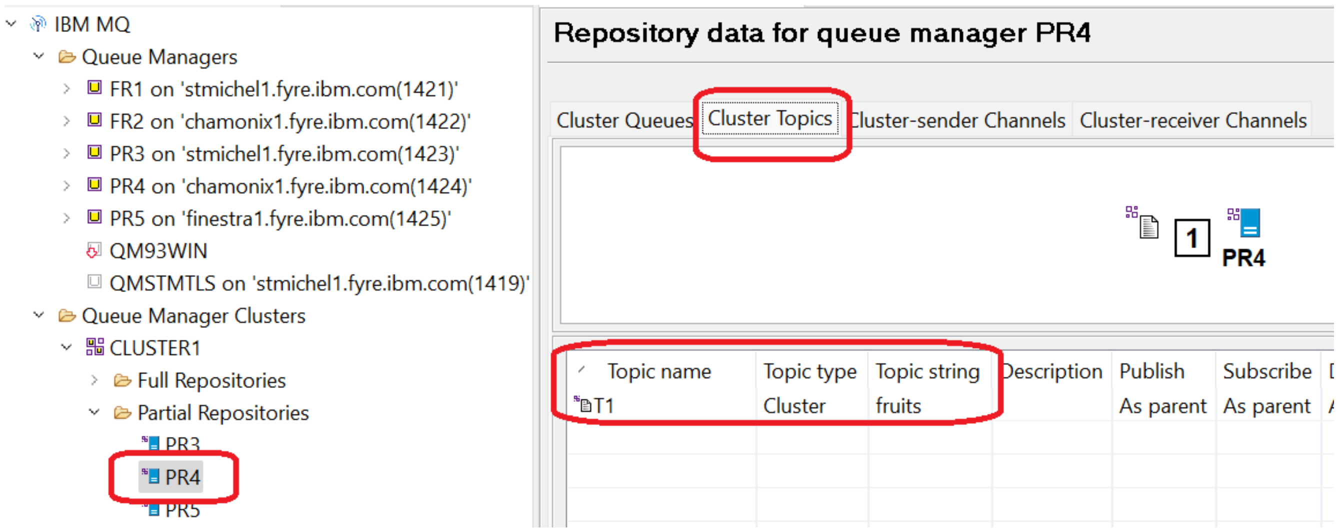
In the right panel click on the tab "Cluster Topics"
Notice that the topic name "T1" refers to the Topic Object, which has a topic string of "fruits".


IBM MQ
Queue Managers
FR1 on 'stmichel1.fyre.ibm.com(1421)'
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
PR4 on 'chamonix1.fyre.ibm.com(1424)'
PR5 on 'finestra1.fyre.ibm.com(1425)'
QM93WIN
QMSTMTLS on 'stmichel1.fyre.ibm.com(1419)'
Queue Manager Clusters
CLUSTER1
Full Repositories
Partial Repositories
PR3
PR4
PR5
Repository data for queue manager PR3
Cluster Queues
Cluster Topics
Cluster-sender Channels
Cluster-receiver Channels
1
PR3

Topic name	Topic type	Topic string	Description	Publish	Subscribe
T1	Cluster	fruits		As parent	As parent

To verify that this topic T1 is now available in the cluster, in the left panel, under the "Partial Repositories" select PR4.

Notice that the topic name "T1" is shown as available in PR4, even though we did not do an explicit definition of this topic in PR4:


IBM MQ
Queue Managers
FR1 on 'stmichel1.fyre.ibm.com(1421)'
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
PR4 on 'chamonix1.fyre.ibm.com(1424)'
PR5 on 'finestra1.fyre.ibm.com(1425)'
QM93WIN
QMSTMTLS on 'stmichel1.fyre.ibm.com(1419)'
Queue Manager Clusters
CLUSTER1
Full Repositories
Partial Repositories
PR3
PR4
PR5
Repository data for queue manager PR4
Cluster Queues
Cluster Topics
Cluster-sender Channels
Cluster-receiver Channels
1
PR4

Topic name	Topic type	Topic string	Description	Publish	Subscribe
T1	Cluster	fruits		As parent	As parent

NOTE:
When handling clustered topics, the topic definition is fanned out to ALL the queue managers in the cluster, whether or not there are subscriptions for the topics in the remote queue managers.

Let's explore the definitions in the queue managers:

Use runmqsc on PR3, where the topic T1 was defined:

runmqsc PR3

display topic(T1) all

```
AMQ8633I: Display topic details.
   TOPIC(T1)                          TYPE(LOCAL)
   TOPICSTR(fruits)                     DESCR( )
   CLUSTER(CLUSTER1)                       CLROUTE(DIRECT)
   DURSUB(ASPARENT)                        PUB(ASPARENT)
   SUB(ASPARENT)                         DEFPSIST(ASPARENT)
   DEFPRTY(ASPARENT)                       DEFPRESP(ASPARENT)
   ALTDATE(2024-03-20)                    ALTTIME(11.23.36)
   PMSGDLV(ASPARENT)                       NPMSGDLV(ASPARENT)
   PUBSCOPE(ASPARENT)                       SUBSCOPE(ASPARENT)
   PROXYSUB(FIRSTUSE)                      WILDCARD(PASSTHRU)
   MDURMDL( )                           MNDURMDL( )
   MCAST(ASPARENT)                         COMMINFO( )
   USEDLQ(ASPARENT)                        CUSTOM( )
```

Display the details on the clustered topic T1

```
display tcluster(T1)
AMQ8633I: Display topic details.
   TOPIC(T1)                              TYPE(CLUSTER)
   TOPICSTR(fruits)                       DESCR( )
   CLUSTER(CLUSTER1)                      CLUSQMGR(PR3)
   CLROUTE(DIRECT)                        CLSTATE(ACTIVE)
   QMID(PR3_2024-03-02_09.58.47)          DURSUB(ASPARENT)
   PUB(ASPARENT)                          SUB(ASPARENT)
   DEFPSIST(ASPARENT)                     DEFPRTY(ASPARENT)
   DEFPRESP(ASPARENT)                     CLUSDATE(2024-03-20)
   CLUSTIME(11.23.36)                     ALTDATE(2024-03-20)
   ALTTIME(11.23.36)                      PMSGDLV(ASPARENT)
   NPMSGDLV(ASPARENT)                     PUBSCOPE(ASPARENT)
   SUBSCOPE(ASPARENT)                     PROXYSUB(FIRSTUSE)
   WILDCARD(PASSTHRU)                     MDURMDL( )
   MNDURMDL( )
```

Repeat the above 2 commands from the other queue manager, PR4:

Notice that the topic T1 has NOT been defined as local to PR4, and thus, the local non-clustered topic T1 is NOT found in PR4:

```
runmqsc PR4
```

```
display topic(T1) all
AMQ8147E: IBM MQ object T1 not found.
```

But the clustered topic T1 is found!

```
display tcluster(T1)
AMQ8633I: Display topic details.
   TOPIC(T1)                              TYPE(CLUSTER)
   TOPICSTR(fruits)                       DESCR( )
   CLUSTER(CLUSTER1)                      CLUSQMGR(PR3)
   CLROUTE(DIRECT)                        CLSTATE(ACTIVE)
   QMID(PR3_2024-03-02_09.58.47)          DURSUB(ASPARENT)
   PUB(ASPARENT)                          SUB(ASPARENT)
   DEFPSIST(ASPARENT)                     DEFPRTY(ASPARENT)
   DEFPRESP(ASPARENT)                     CLUSDATE(2024-03-20)
   CLUSTIME(11.23.37)                     ALTDATE(2024-03-20)
   ALTTIME(11.23.36)                      PMSGDLV(ASPARENT)
   NPMSGDLV(ASPARENT)                     PUBSCOPE(ASPARENT)
   SUBSCOPE(ASPARENT)                     PROXYSUB(FIRSTUSE)
   WILDCARD(PASSTHRU)                     MDURMDL( )
   MNDURMDL( )
```

Repeat the steps for the rest of the queue managers in the cluster.
Notice that all of them will have a definition for the clustered topic T1.

++ Expected behavior:

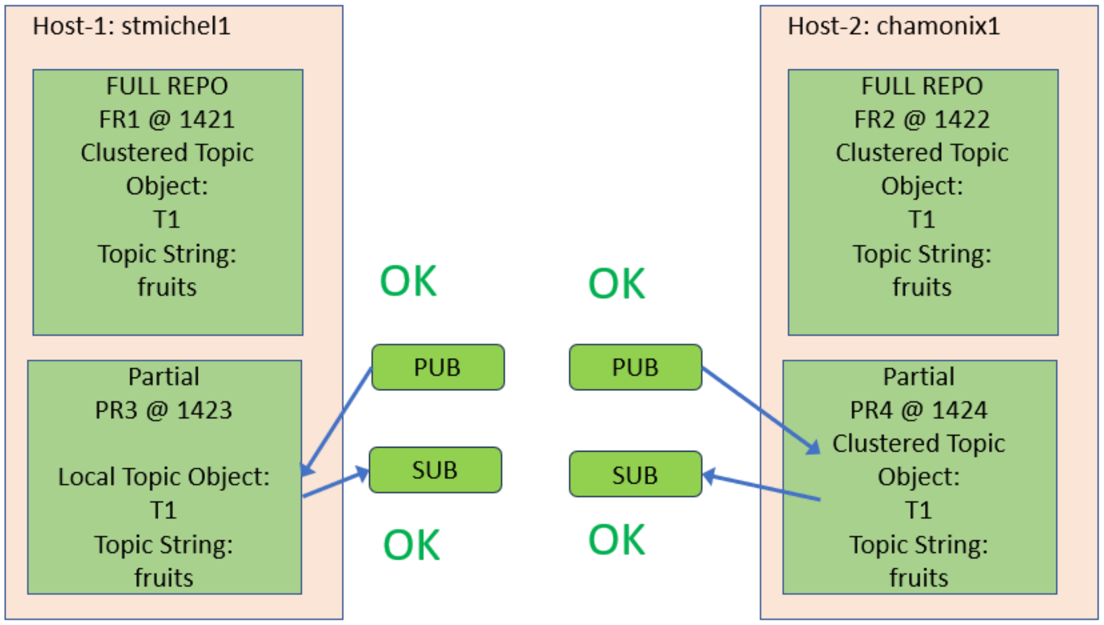
- OK. An application such as "amqspub" can publish a message to the local topic T1 when connected to PR3.

- OK. A subscriber application such as "amqssub" can get a message from the local topic T when connected to PR3.

- OK. Another application can publish a message into T1 when connected to another queue manager, such as PR4.

- OK. Another subscriber application can get a message from T1 when connected to another queue manager, such as PR4.

The topology looks like this:



++ Testing:

We would like to have a subscriber running the same Linux host as PR4.

Now that the topic is a clustered topic, this means that the subscriber in PR4 will be able to get a published message done in PR3.

+ Window 3: (Non-durable Subscriber in Linux chamonix1 - PR4)

```
mqm@chamonix1.fyre.ibm.com: /home/mqm
$ amqssub fruits PR4
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time
```

+ Window 2: (Publisher in Linux stmichel1 - PR3)

```
mqm@stmichel1.fyre.ibm.com: /home/mqm
$ amqspub fruits PR3
Sample AMQSPUBA start
target topic is fruits
test-4 (press ENTER)
(press ENTER to end)
Sample AMQSPUBA end
```

+ Window 3: (Non-durable Subscriber in Linux chamonix1 - PR4)

Message is received!

```
mqm@chamonix1.fyre.ibm.com: /home/mqm
$ amqssub fruits PR4
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time
message <test-4>
Calling MQGET : 30 seconds wait time
```

+ Window 2: (Publisher in Linux stmichel1 - PR3)

Display the Topic Status information (needs to be done on the Topic String)

```
runmqsc PR3
```

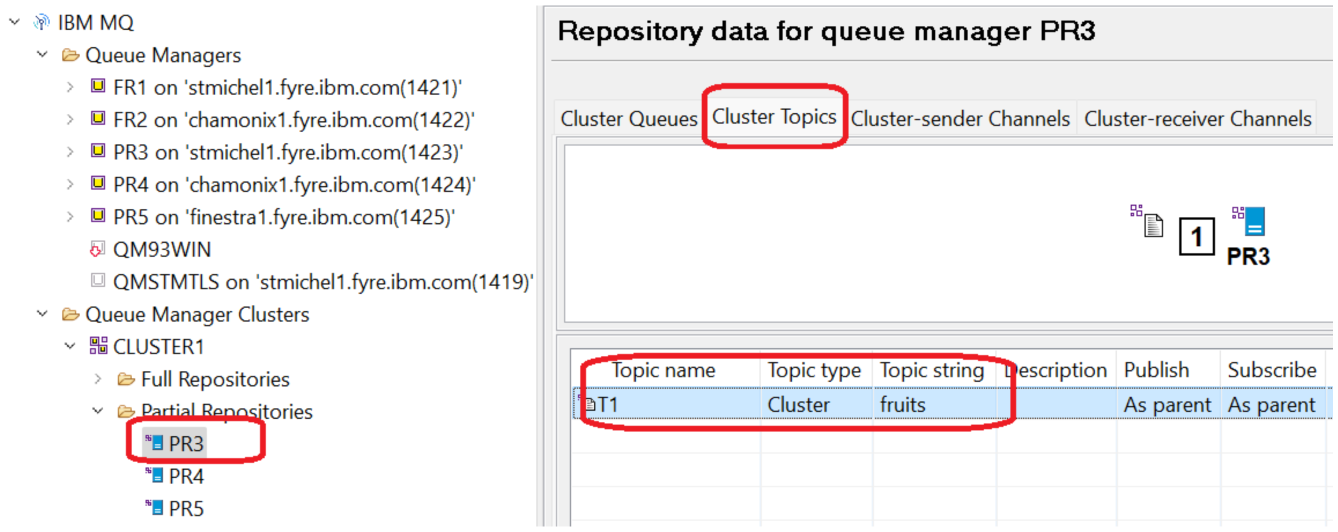
```
display tpstatus('fruits')
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)                        ADMIN(T1)
   CLUSTER(CLUSTER1)
   COMMINFO(SYSTEM.DEFAULT.COMMINFO.MULTICAST)
   MDURMDL(SYSTEM.DURABLE.MODEL.QUEUE)
   MNDURMDL(SYSTEM.NDURABLE.MODEL.QUEUE)
   CLROUTE(DIRECT)                         DEFPSIST(NO)
   DEFPRTY(0)                              DEFPRESP(SYNC)
   DURSUB(YES)                              PUB(ENABLED)
   SUB(ENABLED)                              PMSGDLV(ALLDUR)
   NPMSGDLV(ALLAVAIL)                          RETAINED(NO)
   MCAST(DISABLED)                           PUBCOUNT(0)
   SUBCOUNT(3)                              PUBSCOPE(ALL)
   SUBSCOPE(ALL)                             USEDLQ(YES)
```

```
display tpstatus('fruits') type(pub)
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)                         LPUBDATE( )
   LPUBTIME( )
   ACTCONN(414D5143505233202020202020202020205ADCF16500D30040)
   MCASTREL( , )                            NUMPUBS(0)
```

The following command shows that some of the subscribers are "proxy subscriptions"
    SUBTYPE(PROXY)

```
display tpstatus('fruits') type(sub)
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)
   SUBID(414D5120505233202020202020202020205ADCF16592B90040)
   SUBUSER(mqm)                              RESMDATE(2024-03-20)
   RESMTIME(11:08:55)                         LMSGDATE(2024-03-20)
   LMSGTIME(11:41:31)
   ACTCONN(000000000000000000000000000000000000000000000000)
   DURABLE(YES)                              SUBTYPE(ADMIN)
   MCASTREL( , )                            NUMMSGS(3)
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)
   SUBID(414D5120505233202020202020202020205ADCF16508B20040)
   SUBUSER(mqm)                              RESMDATE(2024-03-20)
   RESMTIME(11:06:22)                         LMSGDATE(2024-03-20)
   LMSGTIME(11:41:31)
   ACTCONN(000000000000000000000000000000000000000000000000)
   DURABLE(YES)                              SUBTYPE(ADMIN)
   MCASTREL( , )                            NUMMSGS(4)
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)
   SUBID(414D5120505233202020202020202020205ADCF16501120040)
   SUBUSER(mqm)                              RESMDATE(2024-03-20)
   RESMTIME(11:23:37)                         LMSGDATE(2024-03-20)
   LMSGTIME(11:41:31)
   ACTCONN(000000000000000000000000000000000000000000000000)
   DURABLE(YES)                              SUBTYPE(PROXY)
   MCASTREL( , )                            NUMMSGS(1)
```

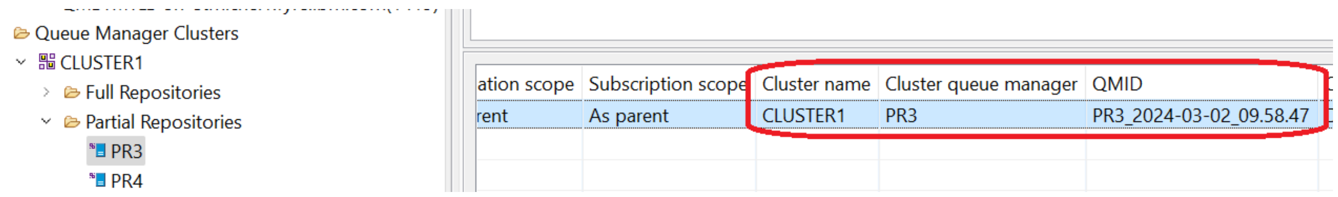
Let's see the Clustered Topic from the perspective of PR3 (where the topic is defined)


IBM MQ
Queue Managers
FR1 on 'stmichel1.fyre.ibm.com(1421)'
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
PR4 on 'chamonix1.fyre.ibm.com(1424)'
PR5 on 'finestra1.fyre.ibm.com(1425)'
QM93WIN
QMSTMTLS on 'stmichel1.fyre.ibm.com(1419)'
Queue Manager Clusters
CLUSTER1
Full Repositories
Partial Repositories
PR3
PR4
PR5
Repository data for queue manager PR3
Cluster Queues
Cluster Topics
Cluster-sender Channels
Cluster-receiver Channels
1
PR3

Topic name	Topic type	Topic string	Description	Publish	Subscribe
T1	Cluster	fruits		As parent	As parent

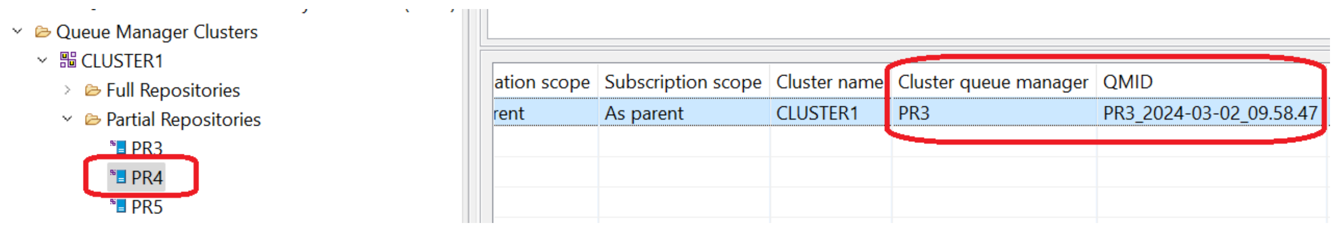
Select the row for T1 and scroll to the right: we can see that the topic T1 shows that it belongs to the cluster CLUSTER1 and it is associated with the queue manager PR3.

Notice that there is an attribute called: QMID
For this instance, the value for QMID is: PR3_2024-03-02_09.58.47



ation scope	Subscription scope	Cluster name	Cluster queue manager	QMID
rent	As parent	CLUSTER1	PR3	PR3_2024-03-02_09.58.47

Let's do the same for PR4:



ation scope	Subscription scope	Cluster name	Cluster queue manager	QMID
rent	As parent	CLUSTER1	PR3	PR3_2024-03-02_09.58.47

Let scroll to the right.
Notice that these values are the same as above:
Cluster queue manager: PR3
QMID: PR3_2024-03-02_09.58.47

+ Let's explore PR5, which does not have an explicit subscriber.
But it has a "proxy" subscriber.

In Windows host "finestra1"

```
C:\> runmqsc PR5

DISPLAY SBSTATUS(*) WHERE(NUMMSGS GT 0)
AMQ8099I: IBM MQ subscription status inquired.
   SUB(PR5 SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
414D5159010100000000000000000000000000000000 SYSTEM.BROKER.ADMIN.STREAM
MQ/PR5                                       /StreamSupport)
   SUBID(414D51205052352020202020202020202035DDF165061D0040)
   NUMMSGS(1)

display tpstatus('fruits')
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)                        ADMIN(T1)
   CLUSTER(CLUSTER1)
   COMMINFO(SYSTEM.DEFAULT.COMMINFO.MULTICAST)
   MDURMDL(SYSTEM.DURABLE.MODEL.QUEUE)
   MNDURMDL(SYSTEM.NDURABLE.MODEL.QUEUE)
   CAPEXPRY(NOLIMIT)                        CLROUTE(DIRECT)
   DEFPSIST(NO)                            DEFPRTY(0)
   DEFPRESP(SYNC)                           DURSUB(YES)
   PUB(ENABLED)                            SUB(ENABLED)
   PMSGDLV(ALLDUR)                          NPMSGDLV(ALLAVAIL)
   RETAINED(NO)                            MCAST(DISABLED)
   PUBCOUNT(0)                             SUBCOUNT(2)
   PUBSCOPE(ALL)                            SUBSCOPE(ALL)
   USEDLQ(YES)

display tpstatus('fruits') type(sub)
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)
   SUBID(414D5120505235202020202020202020290AFB6501130040)
   SUBUSER(MUSR_MQADMIN)                    RESMDATE(2024-03-20)
   RESMTIME(11:23:37)                       LMSGDATE( )
   LMSGTIME( )
   ACTCONN(000000000000000000000000000000000000000000000000)
   DURABLE(YES)                            SUBTYPE(PROXY)
   MCASTREL( , )                           NUMMSGS(0)
```

```
AMQ8754I: Display topic status details.
   TOPICSTR(fruits)
   SUBID(414D5120505235202020202020202020290AFB6504130040)
   SUBUSER(MUSR_MQADMIN)                    RESMDATE(2024-03-20)
   RESMTIME(11:23:37)                       LMSGDATE( )
   LMSGTIME( )
   ACTCONN(000000000000000000000000000000000000000000000000)
   DURABLE(YES)                             SUBTYPE(PROXY)
   MCASTREL( , )                            NUMMSGS(0)

display tpstatus('fruits') type(pub)
AMQ8472E: IBM MQ topic string not found
```

+++
+++ Chapter 6: Doing pub/sub as non-MQ administrator
+++

In the previous chapters, the pub/sub commands were issued by an MQ administrator.

If you are going to do pub/sub activities as a non-MQ administrator, then you will need to provide the necessary authorities.

Let's use the following id (which is NOT an MQ administrator)

```
mqm@chamonix1.fyre.ibm.com: /home/mqm
$ id fulano
uid=1021(fulano) gid=1005(mquser) groups=1005(mquser)
```

The target queue manager is going to be PR4.

Let's ensure that there are no authorities already defined for this user in PR4.
Actually, in UNIX, the authorities are controlled at the UNIX group level and not at the UNIX userid. Thus, we need to specify the group “mquser” instead of “fulano”:

```
dmpmqaut -m PR4 -g mquser
```

User fulano tries to publish a message into the topic, but fails with 2035 at MQCONN.

Use the amqspub sample to publish to the topic string "fruits" in PR3.
This topic string is associated with the topic object T1.

Login as fulano.

```
fulano@chamonix1.fyre.ibm.com: /home/fulano
$ amqspub fruits PR4
Sample AMQSPUBA start
MQCONN ended with reason code 2035
```

The short name for the reason code 2035 can be found by issuing:

```
$ mqrc 2035
    2035  0x000007f3  MQRC_NOT_AUTHORIZED
```

The user fulano will NOT be able to find out more details on this error.
Instead, the MQ Administrator needs to look at the error logs of the queue manager for more details.

Login as MQ administrator

Notice that the reason code was for MQCONN, in this case, as MQ administrator review the recent entries at the bottom of the error log AMQERR01.LOG file in:
   /var/mqm/qmgrs/PR4/errors

The problem is that the user did not have the "connect" authority for the queue manager.

Add the authority to the group "mquser" to connect to the queue manager

```
mqm@chamonix1.fyre.ibm.com: /var/mqm/qmgrs/PR4/errors
$ setmqaut -m PR4 -t qmgr -g mquser +connect +inq +dsp
```

User fulano tries again to publish a message, but fails again with 2035, but this time it fails at MQOPEN.

```
fulano@chamonix1.fyre.ibm.com: /home/fulano
$ amqspub fruits PR4
Sample AMQSPUBA start
target topic is fruits
MQOPEN ended with reason code 2035
unable to open topic for publish
Sample AMQSPUBA end
```

Login as MQ administrator and look at the bottom of the error log:

```
mqm@chamonix1.fyre.ibm.com: /var/mqm/qmgrs/PR4/errors
$ vi AMQERR01.LOG

03/20/2024 12:39:37 PM - Process(409848.68) User(mqm) Program(amqzlaa0)
                    Host(chamonix1.fyre.ibm.com) Installation(Installation1)
                    VRMF(9.3.0.16) QMgr(PR4)
                    Time(2024-03-20T19:39:37.855Z)
                    CommentInsert1(fulano)
                    CommentInsert2(fruits)
                    CommentInsert3(pub)
AMQ8009W: Entity 'fulano' has insufficient authority to access topic string
'fruits'.
EXPLANATION:
The specified entity is not authorized to access the required topic. The
following permissions were requested: pub
```

As MQ Administrator provide the proper authority:

```
mqm@chamonix1.fyre.ibm.com: /var/mqm/qmgrs/PR4/errors
$ setmqaut -m PR4 -n T1 -t topic -g mquser +pub +sub +resume
The setmqaut command completed successfully.
```

Now the user fulano tries again to publish a message, and this time it succeeds.

```
fulano@chamonix1.fyre.ibm.com: /home/fulano
$ amqspub fruits PR4
Sample AMQSPUBA start
target topic is fruits
this is a test message during pub
Sample AMQSPUBA end
```

+++ end +++