Setup of a cluster and basic usage of clustered queues in IBM MQ 9.3

https://www.ibm.com/support/pages/node/611561

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 provides the details on how to setup a small IBM MQ cluster with:
-- 2 full repositories
-- 3 partial repositories
-- using 3 different hosts

The instructions are provided for the MQ Explorer and the runmqsc command.

The tutorial also provides the details on the basic usage of clustered queues:

Scenario 1: A local queue that is created in one of the partial repositories but not yet added to the cluster and subsequent tasks used to explore the behavior of putting and getting a message.

Scenario 2: Then the local queue is added to the cluster and the corresponding taskss are explored.

The chapters are:

Chapter 1: Creating queue managers and setup of MQ Explorer
Chapter 2: Creating a cluster with 2 Full Repositories
Chapter 3: Adding Partial Repositories
Chapter 4: Working with a local queue, not yet a clustered queue
Chapter 5: Working with a queue that was made a clustered queue

+ Companion Tutorial

A companion tutorial regarding clustered topics is available at:
https://www.ibm.com/support/pages/node/593771
Basic usage of clustered topics in IBM MQ 9.3

+++ Summary of steps to setup a new cluster

This technical document provides all the details for setting a basic cluster with 2 full repositories, adding partial repositories and adding a clustered queue.
Because some customers would like to see in a single place the summary of steps, here is the information:

It is easier to setup a new cluster and to add members to the clusters by using the wizard provided by the MQ Explorer.

If you want to use the runmqsc command, the following link provides more information.

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=cluster-setting-up-new
IBM MQ / 9.3
Setting up a new cluster

Scroll down and read the following section:

Section: Setting up a cluster using TCP/IP with a single transmission queue per queue manager

+ Summary of steps for setting up a new cluster and the 2 Full Repositories

1. Decide on the organization of the cluster and its name.
2. Decide which queue managers are to hold full repositories.
3. Alter the queue-manager definitions to add definitions for the full repositories.
   Example:
```
ALTER QMGR REPOS('ClusterName')
```
4. Define the listeners that accept network requests from other queue managers.
   Example:
```
DEFINE LISTENER(TCP.LISTENER) TRPTYPE(TCP) CONTROL(QMGR) PORT(1414)
START LISTENER(TCP.LISTENER)
```
5. Define the CLUSRCVR channel for the queue managers, with a CONNAME pointing to itself.
   Generic Example:
```
DEFINE CHANNEL(TO.channel_name) CHLTYPE(CLUSRCVR) +
  CONNAME('my_ip_name_or_address(port)') CLUSTER('ClusterName')
```
6. Define the CLUSSDR channel on the queue managers. The channel name must match that of the CLUSRCVR on the full repository.
   Generic Example:
```
DEFINE CHANNEL(channel_name) CHLTYPE(CLUSSDR) +
  CONNAME('remote_ip_name_or_address(port)') CLUSTER('ClusterName')
```

+ Summary of steps for adding a partial repository:

1. Define CLUSRCVR channel, with a CONNAME pointing to itself.
   This is the step that adds the partial repository into the cluster.
   Generic Example:

```
DEFINE CHANNEL(TO.channel_name) CHLTYPE(CLUSRCVR) +
  CONNAME('my_ip_name_or_address(port)') CLUSTER('ClusterName')
```

2. Define a CLUSSDR channel for at least one full repository. The channel name must match that of the CLUSRCVR on the full repository.
   Generic Example:

```
DEFINE CHANNEL(channel_name) CHLTYPE(CLUSSDR) +
  CONNAME('remote_ip_name_or_address(port)') CLUSTER('ClusterName')
```

+ Summary of steps for adding a clustered queue

1. Define a cluster queue.

Other queue managers in the cluster can send message to it without making remote-queue definitions for it.

Only the local queue manager can read messages from an instance of the cluster queue.

Generic Example:

```
DEFINE QLOCAL(queueName) CLUSTER('ClusterName')
```

+ Best Practice for removing a repository from the cluster

See the following section from the online manual:

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=cluster-removing-queue-manager-from-best-practice
IBM MQ / 9.3
Removing a queue manager from a cluster: best practice

++ Clarification on the role of MQ clusters in high availability configurations

A common misconception is that MQ Clusters can be used solely for providing high-availability. However, if messages are stored in a clustered queue in a queue manager in the cluster, if that queue manager is not available, then those messages in the clustered queue cannot be consumed.

Keep in mind that MQ Clusters can run on high-availability configurations, such as multi-instance queue managers or High-Availability (HA) Clusters (hardware/software products dedicated for this purpose).

For more information on High Availability, see:

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=restart-high-availability-configurations
IBM MQ / 9.3
High availability configurations

++ Best Practices

+ Clustering provides two key benefits:

- They simplify the administration of IBM MQ networks which usually require many object definitions for channels, transmit queues, and remote queues to be configured. This situation is especially true in large, potentially changing, networks where many queue managers need to be interconnected.

- Clusters can also be used to balance the workload of message traffic across queues and queue managers in the cluster.

+ One overall best practice is: the Full Repositories should run at the highest (newest) level of MQ. Thus, if you have a mix of MQ 9.1, 9.2 and 9.3 queue managers, the full repositories should be at 9.3.
The reason is that in case that there are new attributes or functions related to clustering, you want the full repositories to take advantage of them. If a full repository is at 9.0, then it will not be able to exploit new clustering features added in 9.1 and later.

+ Additional articles:

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=clusters-clustering-best-practices
Clustering: Best practices

https://www.ibm.com/support/pages/ibm-mq-clustering-best-practices
IBM MQ Clustering best practices
Author: Valerie Lampkin (2011)

++

+++ Chapter 1: Creating queue managers and setup of MQ Explorer

++

This techdoc uses 5 queue managers that were already created and that are running.

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

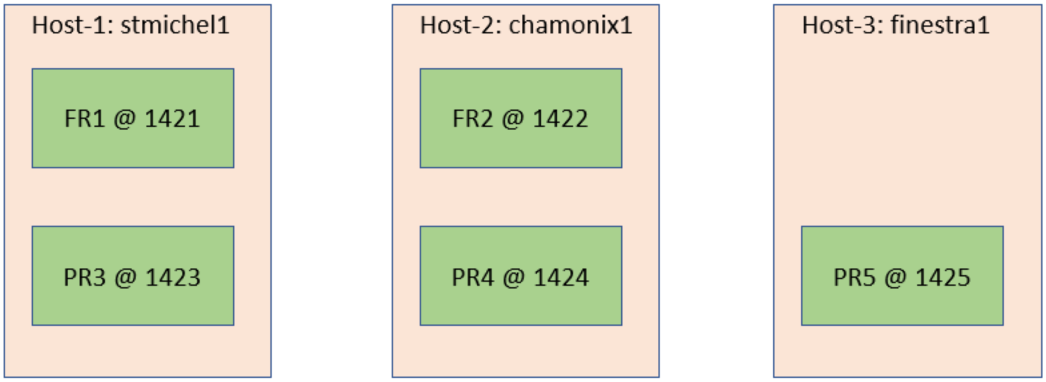
Host-2, name chamonix, Linux, MQ 9.3.0.16 LTS

QMNAME(FR2)	Port 1422	**Full Repository 2**
QMNAME(PR4)	Port 1424	Partial Repository 4

Host-3, name finestra1, Windows, MQ 9.3.5 CD

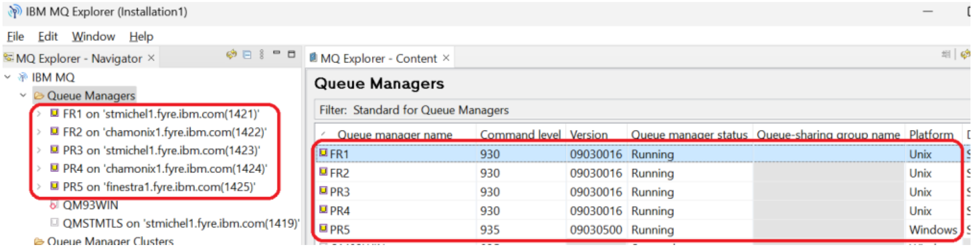
QMNAME(PR5)	Port 1425	Partial Repository 5

The topology looks like this:



+ Using MQ Explorer in Windows to remotely access the queue managers

In Host-4 Windows, the MQ Explorer (at version 9.3.5 CD which is the latest available at the time this tutorial was written) is used to remotely access the 5 queue managers:



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

Queue Managers

Filter: Standard for Queue Managers

Queue manager name	Command level	Version	Queue manager status	Queue-sharing group name	Platform
FR1	930	09030016	Running		Unix
FR2	930	09030016	Running		Unix
PR3	930	09030016	Running		Unix
PR4	930	09030016	Running		Unix
PR5	935	09030500	Running		Windows

+++
+++ Chapter 2: Creating a cluster with 2 Full Repositories
+++

A Full Repository (FR) contains a complete set of information about every queue manager and object in the cluster.

You will need at least one Full Repository, preferably two.
In this document, a cluster with 2 Full Repositories is used.

Note: Some customers think that having 3 or more FRs will be better, but actually, it is a BAD IDEA!
Why? Because each FR synchronizes ONLY with 1 FR. If there are 3 FRs, then eventually one will be out-of-sync. **Thus, do NOT have more than 2 FRs!**

In Chapter 3, three Partial Repositories will be added.

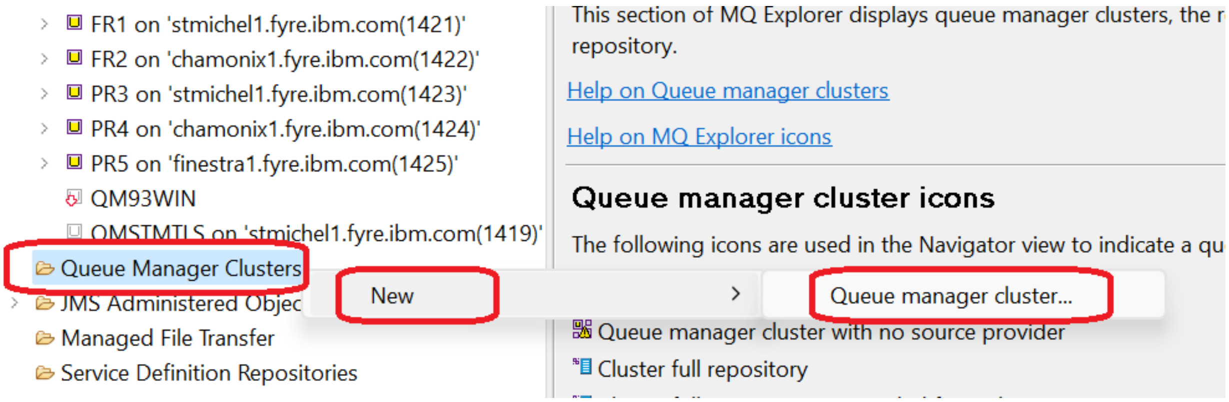
++ Using MQ Explorer (the easiest way!)

Login as an MQ Administrator.

Let's use the MQ Explorer to create a cluster by specifying 2 Full Repositories:

FR1 => Full Repository 1
FR2 => Full Repository 2

On the left navigational panel, click on "Queue Manager Clusters"
Then "New"
Then "Queue manager cluster ..."



This section of MQ Explorer displays queue manager clusters, the r
repository.

Help on Queue manager clusters

Help on MQ Explorer icons

Queue manager cluster icons

The following icons are used in the Navigator view to indicate a qu

Queue manager cluster with no source provider

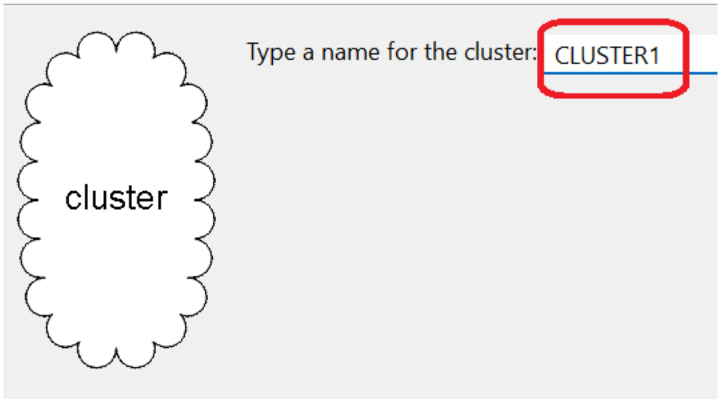
Cluster full repository

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
JMS Administered Objec
Managed File Transfer
Service Definition Repositories
New
Queue manager cluster...

You will see the dialog "Create Cluster".

Enter the name of the cluster.
In this example, we are using: "CLUSTER1"

1. Name the cluster

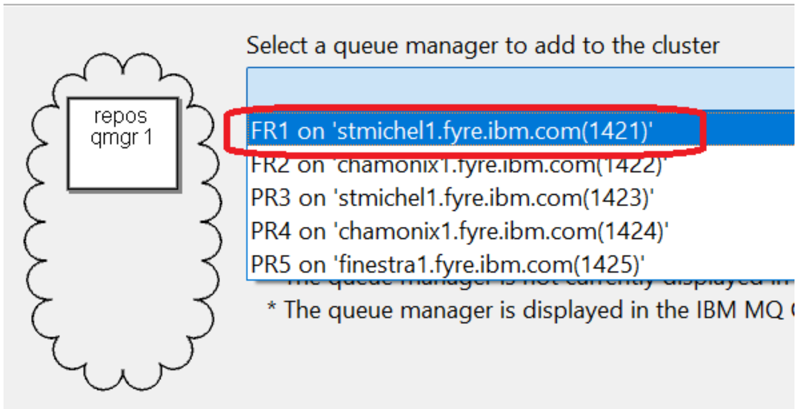


Type a name for the cluster: CLUSTER1

Click Next to go to the step "Select the first full repository queue manager".

Click the down arrow to expand the list of active queue managers, as seen by the MQ Explorer.
Let's select: FR1 and click Next.

2. Select the first full repository queue manager



Select a queue manager to add to the cluster

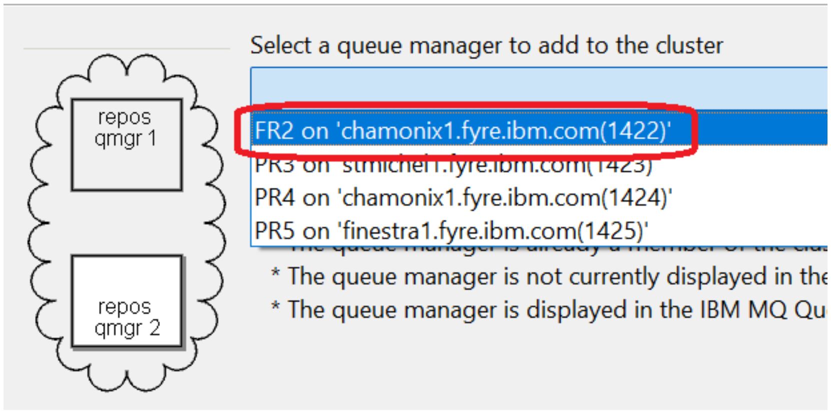
FR1 on 'stmichel1.fyre.ibm.com(1421)'
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
PR4 on 'chamonix1.fyre.ibm.com(1424)'
PR5 on 'finestra1.fyre.ibm.com(1425)'

* The queue manager is displayed in the IBM MQ

Click Next to go to the step "Select the first full repository queue manager".

Now let's select the "second full repository queue manager" by selecting: FR2

3. Select the second full repository queue manager



Select a queue manager to add to the cluster

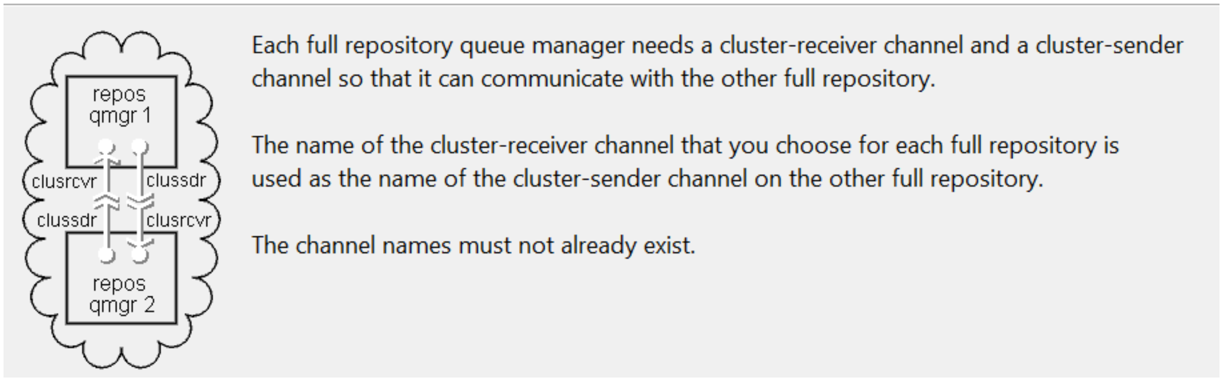
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
PR4 on 'chamonix1.fyre.ibm.com(1424)'
PR5 on 'finestra1.fyre.ibm.com(1425)'

The queue manager is already a member of the clu
* The queue manager is not currently displayed in the
* The queue manager is displayed in the IBM MQ Qu

Click Next.

You will see a dialog with the explanation of the cluster channels.

4. Creating cluster channels



Each full repository queue manager needs a cluster-receiver channel and a cluster-sender channel so that it can communicate with the other full repository.

The name of the cluster-receiver channel that you choose for each full repository is used as the name of the cluster-sender channel on the other full repository.

The channel names must not already exist.

Click Next.

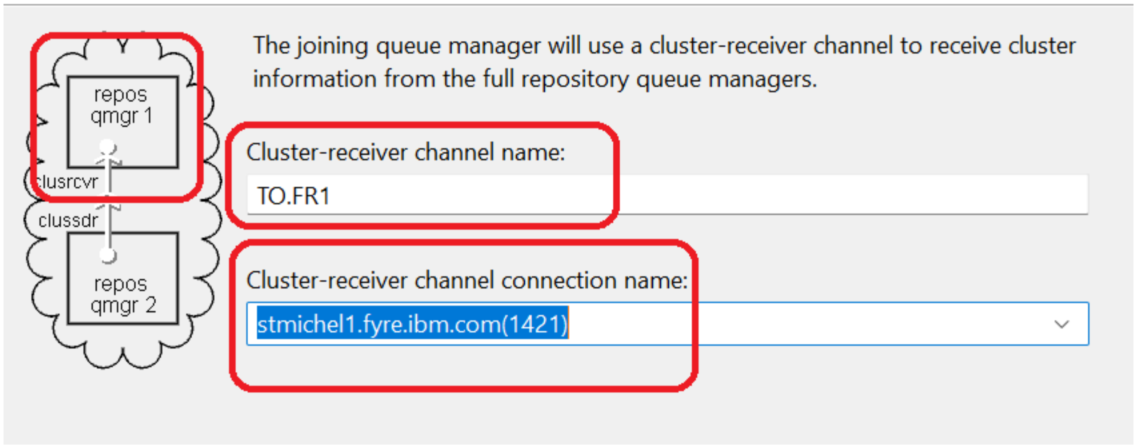
Enter the connection name (host-port) for the "cluster-receiver channel" for the queue repository FR1.

The default name follows the template:
TO.queuemanager

In this case it is:
TO.FR1

In this example it is the name of the host stmichel1 and port 1421.

4a. Name the first full repository's cluster-receiver channel



The joining queue manager will use a cluster-receiver channel to receive cluster information from the full repository queue managers.

Cluster-receiver channel name:
TO.FR1

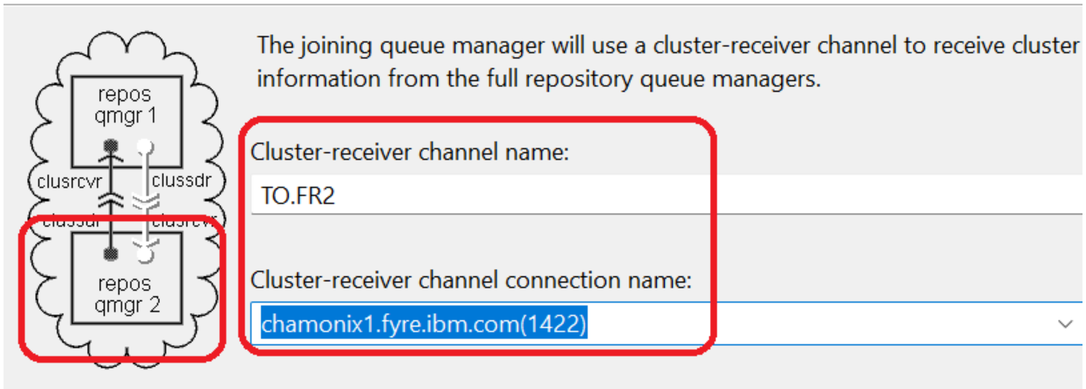
Cluster-receiver channel connection name:
stmichel1.fyre.ibm.com(1421)

Click Next.

Do the same for the queue manager FR2.
The "cluster-receiver channel" is:
TO.FR2

In this example it is the name of the host chamonix1 and port 1422.

4b. Name the second full repository's cluster-receiver channel



The joining queue manager will use a cluster-receiver channel to receive cluster information from the full repository queue managers.

Cluster-receiver channel name:

TO.FR2

Cluster-receiver channel connection name:

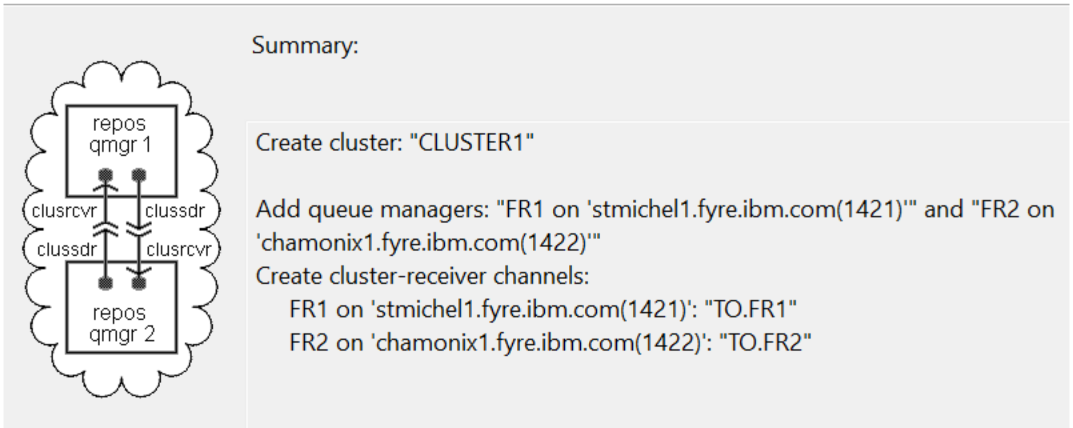
chamonix1.fyre.ibm.com(1422)

Click Next.

You will see the final screen for the dialog.

Click Finish.

5. Create the cluster



Summary:

Create cluster: "CLUSTER1"

Add queue managers: "FR1 on 'stmichel1.fyre.ibm.com(1421)'" and "FR2 on 'chamonix1.fyre.ibm.com(1422)'"
Create cluster-receiver channels:
FR1 on 'stmichel1.fyre.ibm.com(1421)': "TO.FR1"
FR2 on 'chamonix1.fyre.ibm.com(1422)': "TO.FR2"

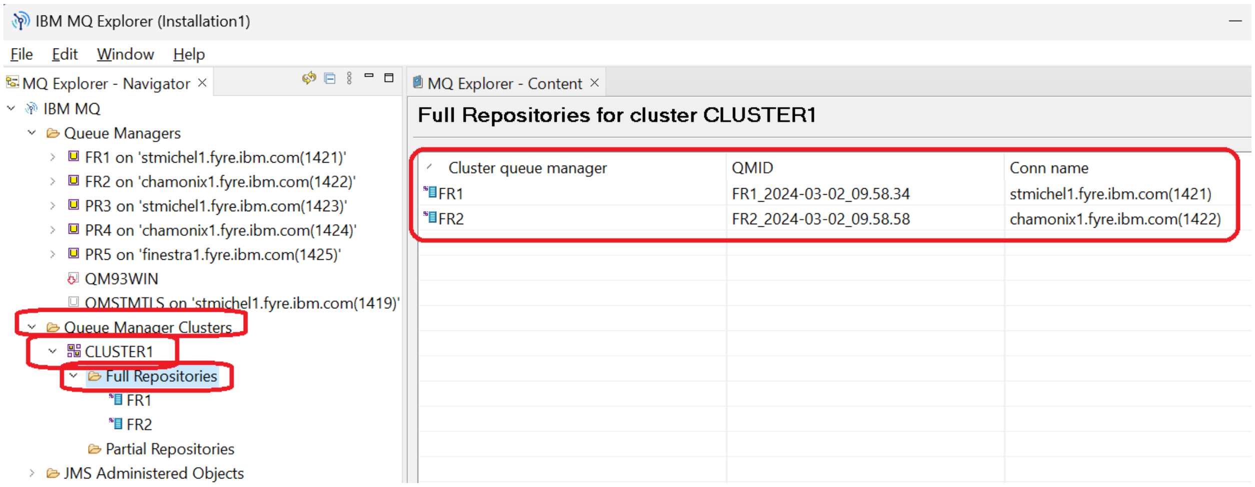
Wait a moment for the MQ Explorer to be refreshed.

Then you will see the new entry "CLUSTER1" under the "Queue Manager Clusters"

Click on the + sign to the left of CLUSTER1 to expand the folders.

Click the folder Full Repositories.

You will see that both repositories are listed on the right panel.

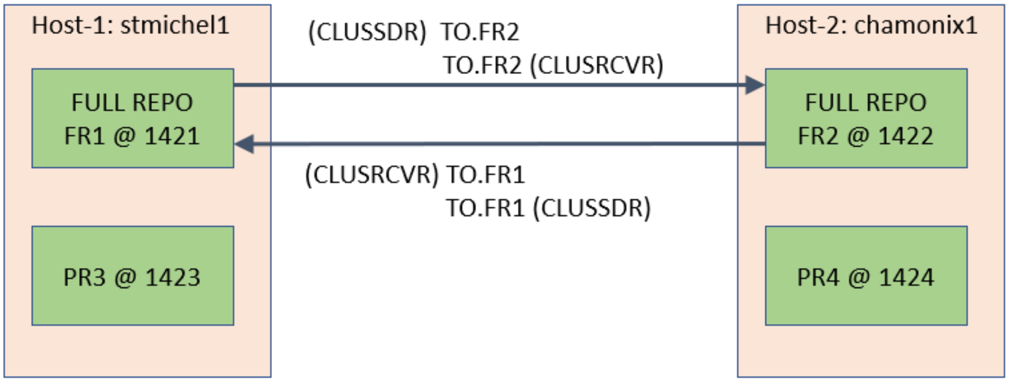


- IBM MQ
  - Queue Managers
    - FR1 on 'stmichel1.fyre.ibm.com(1421)'
    - FR2 on 'chamonix1.fyre.ibm.com(1422)'
    - PR3 on 'stmichel1.fyre.ibm.com(1423)'
    - PR4 on 'chamonix1.fyre.ibm.com(1424)'
    - PR5 on 'finestra1.fyre.ibm.com(1425)'
    - QM93WIN
    - QMSTMTLS on 'stmichel1.fyre.ibm.com(1419)'
  - Queue Manager Clusters
    - CLUSTER1
      - Full Repositories
        - FR1
        - FR2
      - Partial Repositories
  - JMS Administered Objects

Full Repositories for cluster CLUSTER1

Cluster queue manager	QMID	Conn name
FR1	FR1_2024-03-02_09.58.34	stmichel1.fyre.ibm.com(1421)
FR2	FR2_2024-03-02_09.58.58	chamonix1.fyre.ibm.com(1422)

The topology of our cluster CLUSTER1 looks like this:



Note that the Cluster Sender (CLUSSDR) named (TO.FR1) in FR2 is created by the wizard and it matches the Cluster Receiver (CLUSRCVR) of FR1 (TO.FR1).

Similarly for the other channel pair (TO.FR2).

++ Creating the cluster and the 2 Full Repositories (using line commands)

Login as an MQ Administrator.

1) Alter the queue manager to add definitions for the full repositories.
Run this runmqsc command for each queue manager that is to hold a full repository.

Generic:

```
  ALTER QMGR REPOS('ClusterName')
```

Note: This command is ONLY for the full repositories.

Concrete examples:

```
runmqsc FR1
  ALTER QMGR REPOS('CLUSTER1')
  end

runmqsc FR2
  ALTER QMGR REPOS('CLUSTER1')
  end
```

2) Create cluster receiver (CLUSRCVR) channel objects

This step is performed for all queue managers in the cluster.
Every queue manager in the cluster needs a CLUSRCVR with a connection name pointing to itself.

Every queue manager in a cluster must define a cluster-receiver channel on which it can receive messages. The cluster-receiver channel advertises the availability of the queue manager to receive messages from other queue managers in the cluster. There is no need to make definitions on other queue managers for a sending end to the cluster-receiver channel. Other definitions are made automatically when needed.

Generic:

```
  DEFINE CHANNEL(TO.channel_name) CHLTYPE(CLUSRCVR) +
    CONNAME('my_ip_name_or_address(port)') CLUSTER('ClusterName')
```

Concrete examples for the 2 Full Repositories.
Similar commands will be used in Chapter 3 for the Partial Repositories.

```
runmqsc FR1
  DEFINE CHANNEL(TO.FR1) CHLTYPE(CLUSRCVR) +
    CONNAME('stmichel1.x.com(1421)') CLUSTER('CLUSTER1')
  END
```

```
runmqsc FR2
  DEFINE CHANNEL(TO.FR2) CHLTYPE(CLUSRCVR) +
    CONNAME('chamonix1.x.com(1422)') CLUSTER('CLUSTER1')
  END
```

3) Create cluster sender (CLUSSDR) channel objects

Define one CLUSSDR to each full repository queue manager.
The channel name must match that of the cluster receiver (CLUSRCVR) on the other full repository.
DO NOT define a CLUSSDR to point to a partial repository.

Generic:

```
  DEFINE CHANNEL(channel_name) CHLTYPE(CLUSSDR) +
    CONNAME('remote_ip_name_or_address(port)') CLUSTER('ClusterName')
```

Concrete examples for the 2 Full Repositories.
Similar commands will be used in Chapter 3 for the Partial Repositories.

```
runmqsc FR1
  DEFINE CHANNEL(TO.FR2) CHLTYPE(CLUSSDR) +
    CONNAME('chamonix1.x.com(1422)') CLUSTER('CLUSTER1')
  END
```

```
runmqsc FR2
  DEFINE CHANNEL(TO.FR1) CHLTYPE(CLUSSDR) +
    CONNAME('stmichel1.x.com(1421)') CLUSTER('CLUSTER1')
  END
```

+ Where is the information about a Cluster stored in a queue manager?

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=components-cluster-repository
IBM MQ / 9.3
Cluster repository
"

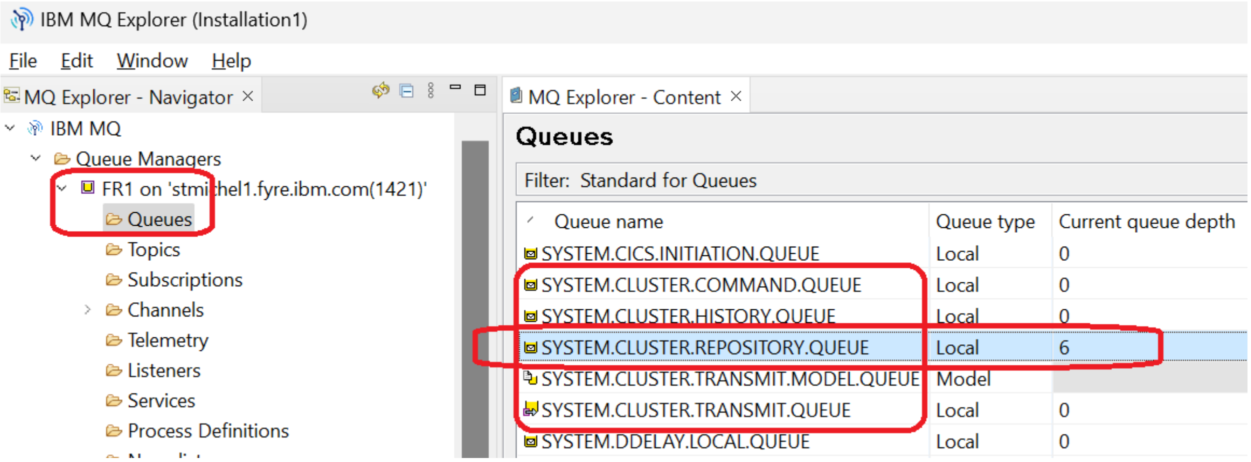
A repository is a collection of information about the queue managers that are members of a cluster.
The repository information includes queue manager names, their locations, their channels, which queues they host, and other information.
The information is stored in the form of messages on a queue called
SYSTEM.CLUSTER.REPOSITORY.QUEUE
This queue is one of the default objects.
...
The queue managers use the SYSTEM.CLUSTER.COMMAND.QUEUE queue to request and receive updates to the repositories.
"

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=cluster-working-transmission-queues-sender-channels
Working with cluster transmission queues and cluster-sender channels
"

Messages between clustered queue managers are stored on cluster transmission queues and forwarded by cluster-sender channels. At any point in time, a cluster-sender channel is associated with one transmission queue.
"

If using only one such queue, the default is:
SYSTEM.CLUSTER.TRANSMIT.QUEUE

The following shows the SYSTEM.CLUSTER.*.QUEUE for Full Repository FR1.
Notice that SYSTEM.CLUSTER.REPOSITORY.QUEUE has 6 entries: 2 for each FR, 2 for the Cluster Sender channels and 2 for the Cluster Receivers.



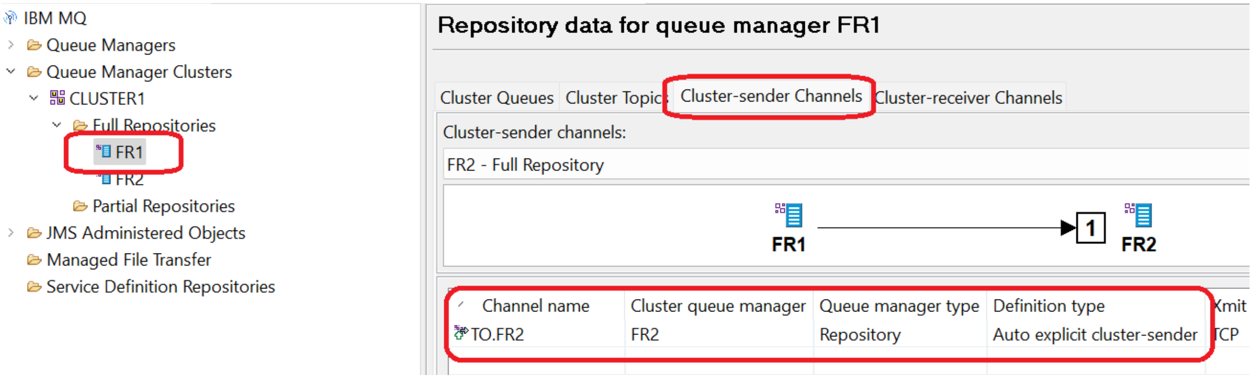
Queue name	Queue type	Current queue depth
SYSTEM.CICS.INITIATION.QUEUE	Local	0
SYSTEM.CLUSTER.COMMAND.QUEUE	Local	0
SYSTEM.CLUSTER.HISTORY.QUEUE	Local	0
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Local	6
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	Model	
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Local	0
SYSTEM.DDELAY.LOCAL.QUEUE	Local	0

++ Using MQ Explorer to verify attributes of cluster channels

Using the MQ Explorer, go to the left panel, under "Full Repositories" and click on FR1

Then on the right panel, click on the tab: Cluster-sender Channels

Notice that from FR1 to FR2 there is a cluster-sender channel called:
  TO.FR2



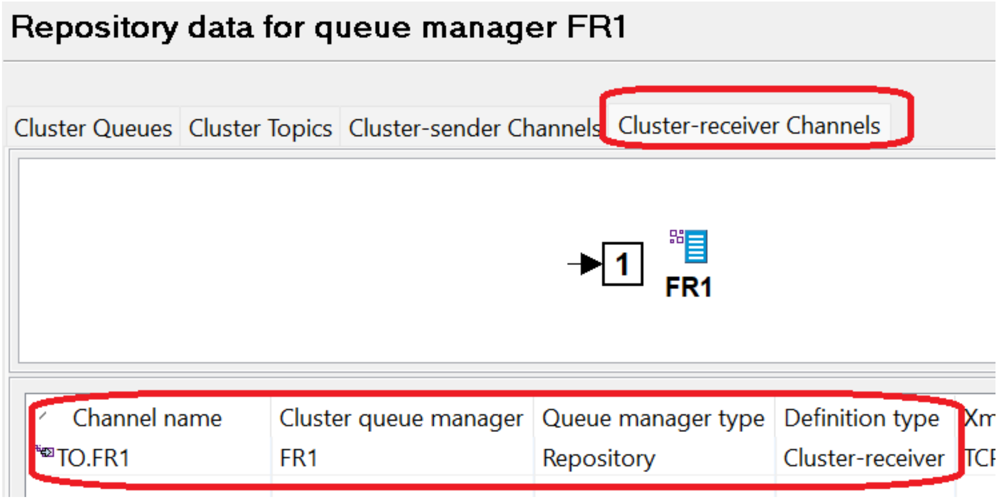
Repository data for queue manager FR1

Cluster-sender channels:

FR2 - Full Repository

Channel name	Cluster queue manager	Queue manager type	Definition type	Xmit
TO.FR2	FR2	Repository	Auto explicit cluster-sender	TCP

Click on the tab "Cluster-receiver Channels".
Notice that there is one channel called: TO.FR1



Repository data for queue manager FR1

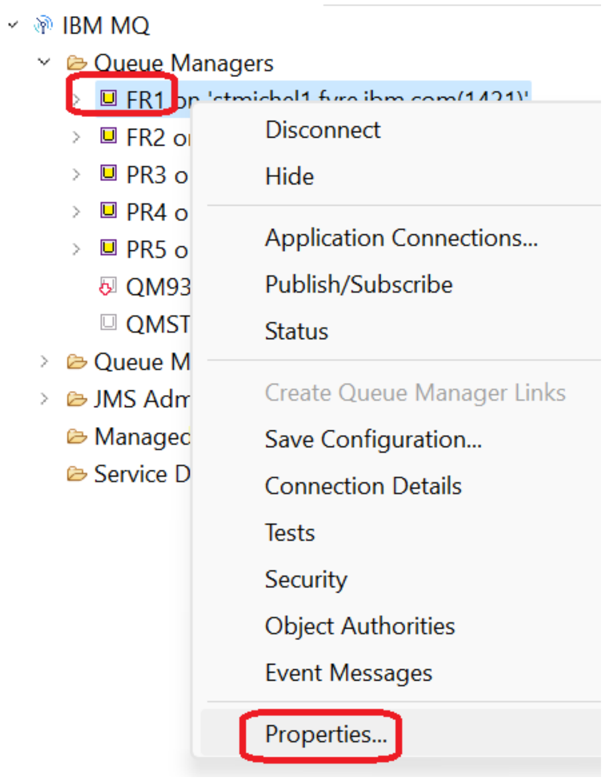
Channel name	Cluster queue manager	Queue manager type	Definition type	Xm
TO.FR1	FR1	Repository	Cluster-receiver	TCF

You can explore the same panels for the other full repository FR2.

Let's explore important attributes related to clustering.

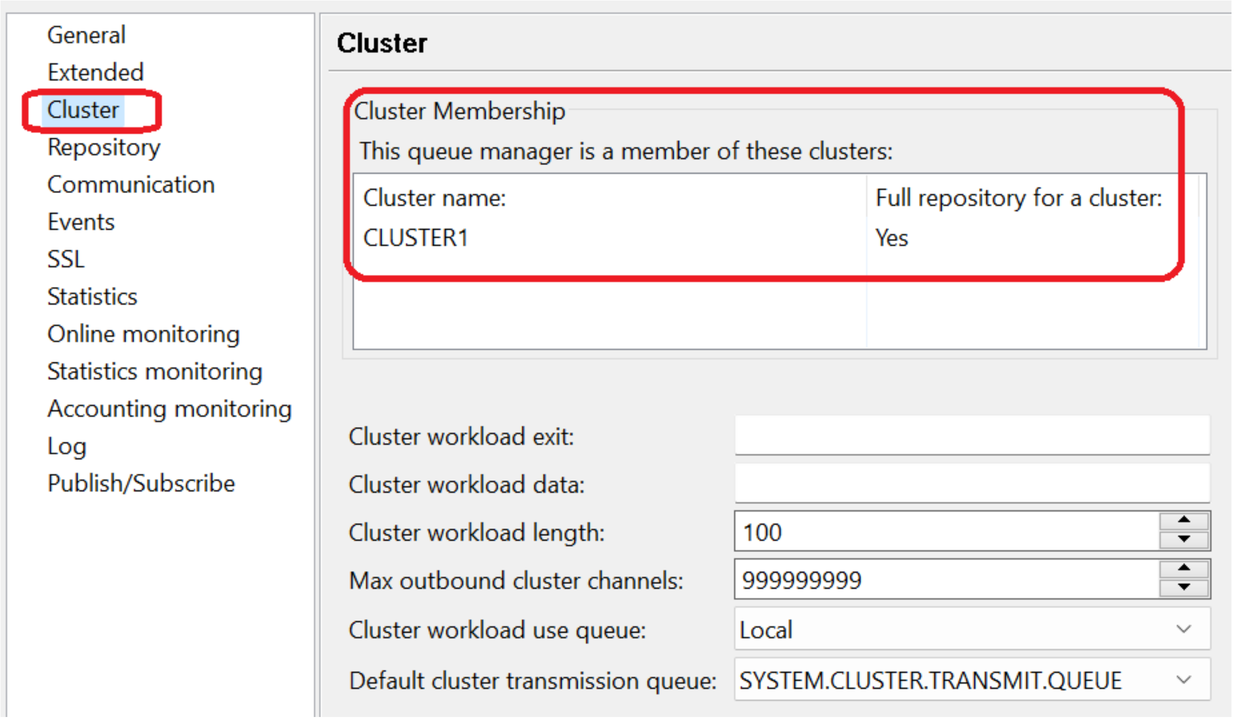
From the folder "Queue Managers" at the top of the Navigator panel, select a full repository manager, such as: FR1

Then right click "Properties..."



When you see the dialog "FR1 - Properties", click the Tab "Cluster".

Notice that this queue manager belongs to the "Cluster name": CLUSTER1
In addition, this queue manager is a "Full repository for a cluster": Yes



Cluster

Cluster Membership

This queue manager is a member of these clusters:

Cluster name:	Full repository for a cluster:
CLUSTER1	Yes

Cluster workload exit:

Cluster workload data:

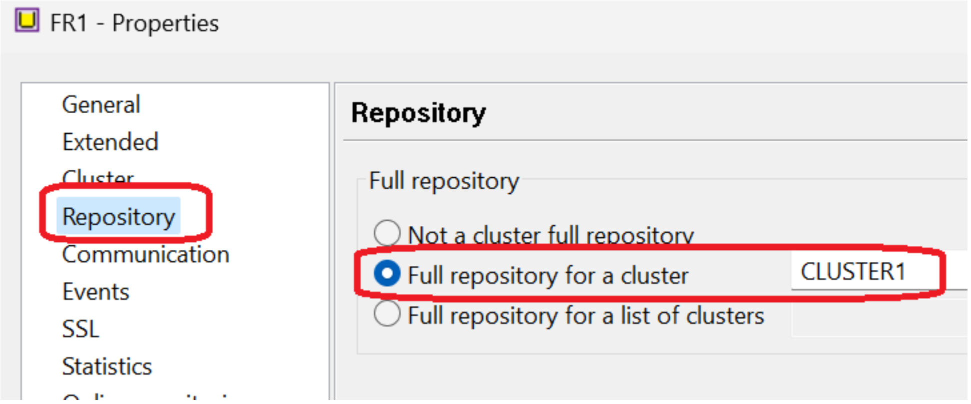
Cluster workload length: 100

Max outbound cluster channels: 999999999

Cluster workload use queue: Local

Default cluster transmission queue: SYSTEM.CLUSTER.TRANSMIT.QUEUE

Click the tab "Repository". Notice that this queue manager is a full repository for CLUSTER1.


FR1 - Properties

Repository

Full repository

○ Not a cluster full repository
● Full repository for a cluster CLUSTER1
○ Full repository for a list of clusters

++ Using line commands to verify attributes related to clustering

+ From FR1

If there is a value for REPOS attribute for the queue manager, then it indicates that the queue manager is a Full Repository for the specified cluster.

```
runmqsc FR1
```

```
display qmgr repos
AMQ8408I: Display Queue Manager details.
   QMNAME(FR1)                              REPOS(CLUSTER1)
```

The following command provides more details from the perspective of a cluster.

```
display clusqmgr(FR1)
```

At this point, the relevant attributes are (not showing the other attributes):

```
AMQ8441I: Display Cluster Queue Manager details.
   CLUSQMGR(FR1
   CHANNEL(TO.FR1)
   CLUSDATE(2024-03-13)
   CLUSTER(CLUSTER1)
   CLUSTIME(14.11.46)
   CLWLPRTY(0)
   CLWLRANK(0)
   CLWLWGHT(50)
   CONNAME(stmichel1(1421))
   DEFTYPE(CLUSRCVR)
   QMID(FR1_2024-03-02_09.58.34)
   QMTYPE(REPOS) => Indicates that this is a full repository
```

Note:
QMID(FR1_2024-03-02_09.58.34) => Unique identifier for the queue manager
  to distinguish this queue manager instance from other queue managers that
  might have previously existed in a queue manager cluster with the same name.
QMTYPE(REPOS)

Display the cluster sender and cluster receiver channels, showing the cluster name and the connection name.

To keep the list short, the SYSTEM.DEF.* names are not shown.

```
display channel(*) CHLTYPE(CLUSSDR) cluster conname
AMQ8414I: Display Channel details.
   CHANNEL(TO.FR2)                        CHLTYPE(CLUSSDR)
   CLUSTER(CLUSTER1)
   CONNAME(chamonix1.fyre.ibm.com(1422))
```

```
display channel(*) CHLTYPE(CLUSRCVR) cluster conname
AMQ8414I: Display Channel details.
   CHANNEL(TO.FR1)                        CHLTYPE(CLUSRCVR)
   CLUSTER(CLUSTER1)
   CONNAME(stmichel1.fyre.ibm.com(1421))
```

Note:
You can issue the above commands for the other Full Repository FR2

++

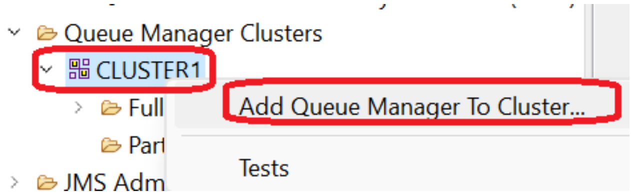
+++ Chapter 3: Adding Partial Repositories

++

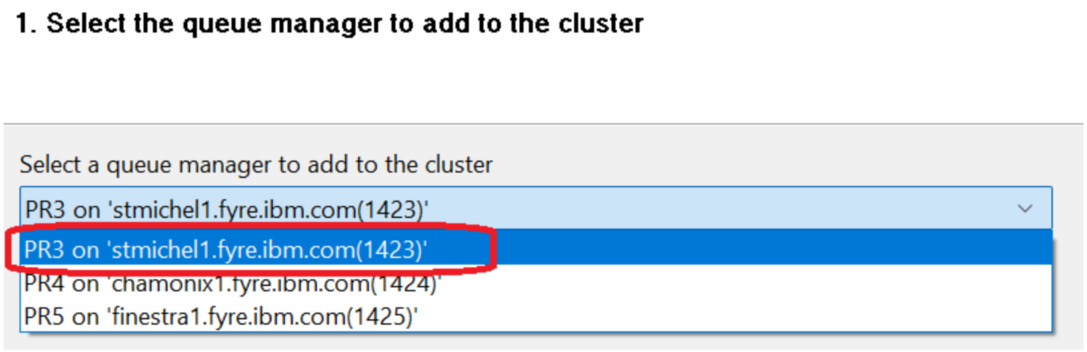
Now, let's add a Partial Repository manager to the Cluster

+ Using MQ Explorer

From the “Queue Manager Clusters” section, select: CLUSTER1
Then right click and select:
  Add Queue Manager To Cluster ...



Select PR3 and click Next.



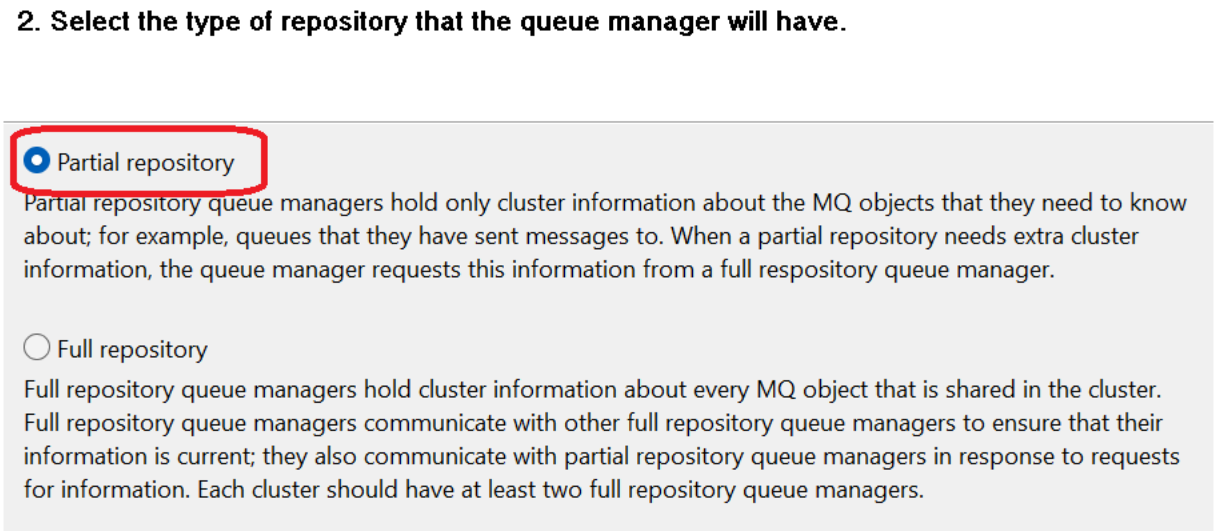
1. **Select the queue manager to add to the cluster**

Select a queue manager to add to the cluster

PR3 on 'stmichel1.fyre.ibm.com(1423)'

PR3 on 'stmichel1.fyre.ibm.com(1423)'
PR4 on 'chamonix1.fyre.ibm.com(1424)'
PR5 on 'finestra1.fyre.ibm.com(1425)'

Select "Partial repository"



2. Select the type of repository that the queue manager will have.

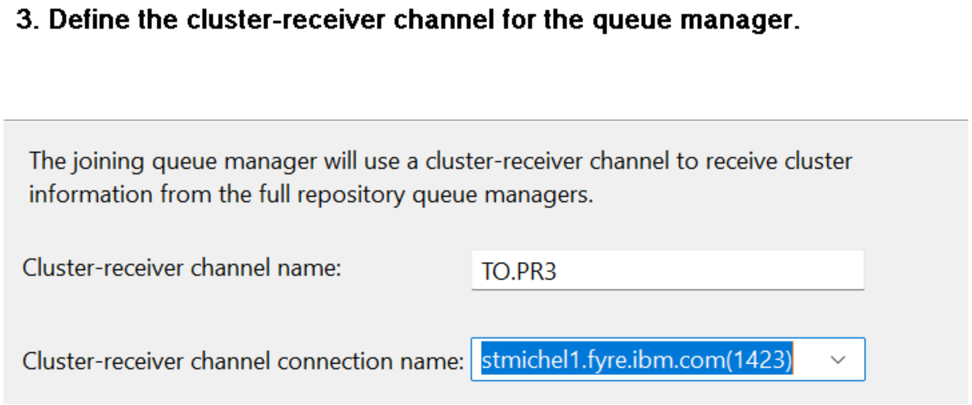
● Partial repository

Partial repository queue managers hold only cluster information about the MQ objects that they need to know about; for example, queues that they have sent messages to. When a partial repository needs extra cluster information, the queue manager requests this information from a full respository queue manager.

○ Full repository

Full repository queue managers hold cluster information about every MQ object that is shared in the cluster. Full repository queue managers communicate with other full repository queue managers to ensure that their information is current; they also communicate with partial repository queue managers in response to requests for information. Each cluster should have at least two full repository queue managers.

Accept the values provided in the screen. Click Next.



3. Define the cluster-receiver channel for the queue manager.

The joining queue manager will use a cluster-receiver channel to receive cluster information from the full repository queue managers.

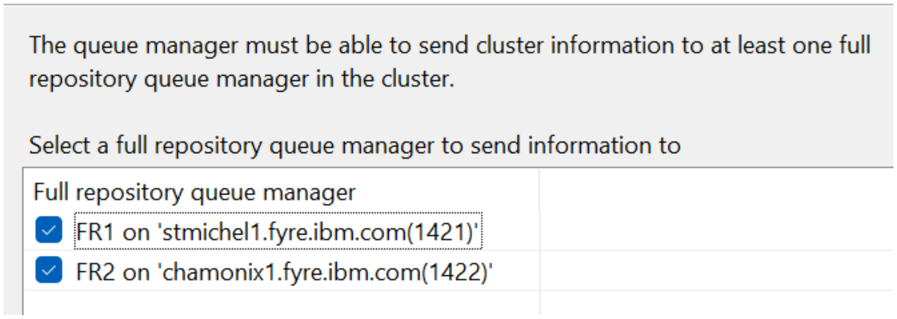
Cluster-receiver channel name: TO.PR3

Cluster-receiver channel connection name: stmichel1.fyre.ibm.com(1423)

The partial repository should be able to communicate with at least one of the full repositories. The default is the 1st in the list (in this case it is FR1).
But it is OK to connect initially to both of them.

If you select only 1 at this time, then the information for the new Partial Repository will be replicated from the initial Full Repository to the other Full Repository, and thus, the Partial Repository will be eventually connected to both Full Repositories.

4. Select the full repository queue managers.



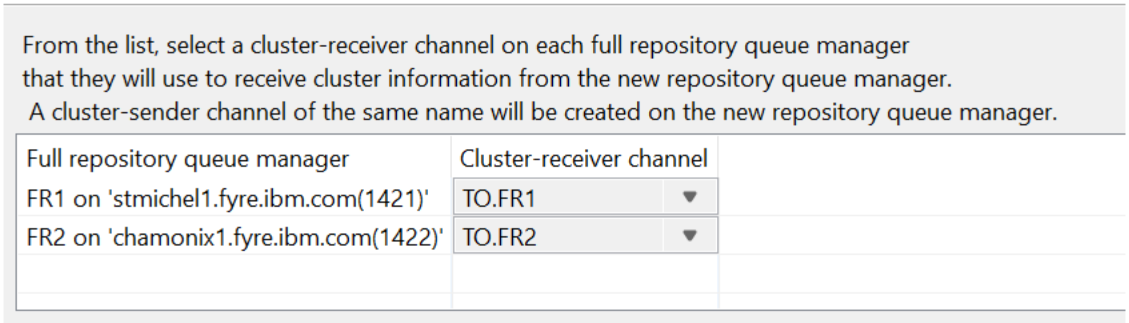
The queue manager must be able to send cluster information to at least one full repository queue manager in the cluster.

Select a full repository queue manager to send information to

Full repository queue manager
☑ FR1 on 'stmichel1.fyre.ibm.com(1421)'
☑ FR2 on 'chamonix1.fyre.ibm.com(1422)'

Accept the provided value for the cluster-receiver channel

5. Select the cluster-receiver channel to be used by the full repository queue manager.

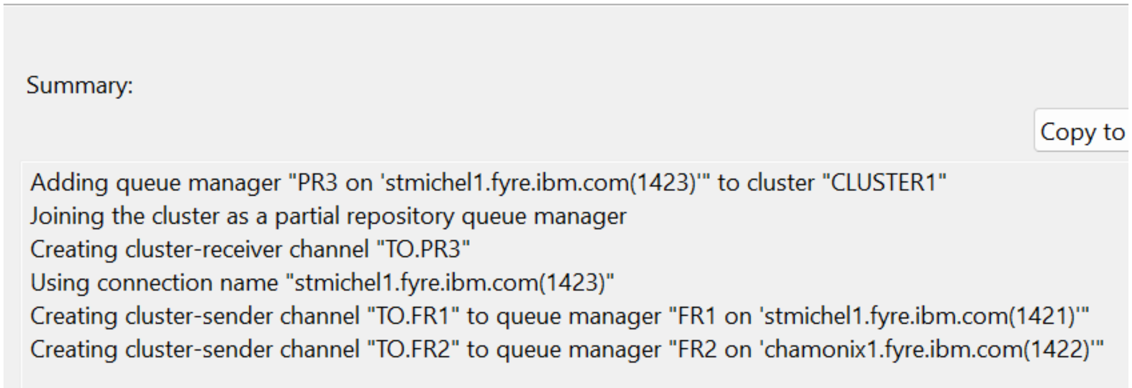


From the list, select a cluster-receiver channel on each full repository queue manager
that they will use to receive cluster information from the new repository queue manager.
A cluster-sender channel of the same name will be created on the new repository queue manager.

Full repository queue manager	Cluster-receiver channel
FR1 on 'stmichel1.fyre.ibm.com(1421)'	TO.FR1 ▾
FR2 on 'chamonix1.fyre.ibm.com(1422)'	TO.FR2 ▾

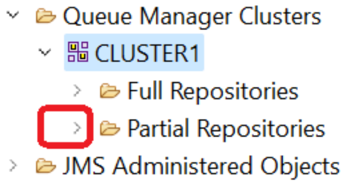
You will see the Summary and then click Finish

6. Add the queue manager to the cluster.

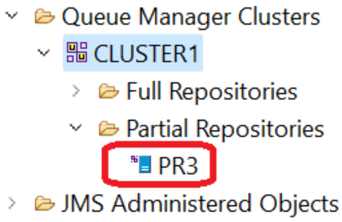


```
Adding queue manager "PR3 on 'stmichel1.fyre.ibm.com(1423)'" to cluster "CLUSTER1"
Joining the cluster as a partial repository queue manager
Creating cluster-receiver channel "TO.PR3"
Using connection name "stmichel1.fyre.ibm.com(1423)"
Creating cluster-sender channel "TO.FR1" to queue manager "FR1 on 'stmichel1.fyre.ibm.com(1421)'"
Creating cluster-sender channel "TO.FR2" to queue manager "FR2 on 'chamonix1.fyre.ibm.com(1422)'"
```

Note:
You will need to wait a little bit for the MQ Explorer to refresh the information for the Cluster.
Once you see that there is a right-pointing arrow for "Partial Repositores", then you can proceed:

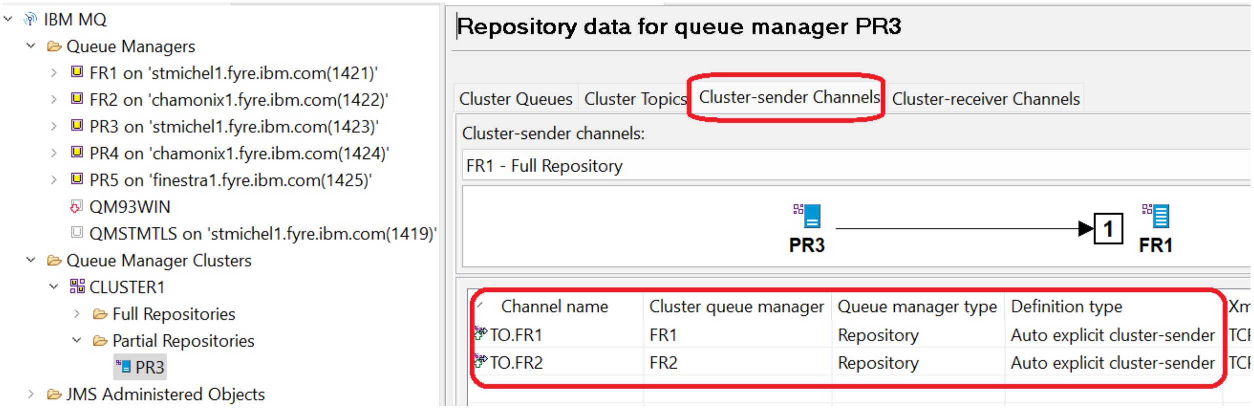


The Partial PR3 will be listed under "Partial Repositories"



Let's verify that the new Partial Repository PR3 is indeed connected to both Full Repositories:

Notice that in the Cluster-Sender Channels tab, there are 2 channels:


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
JMS Administered Objects
Repository data for queue manager PR3
Cluster Queues
Cluster Topics
Cluster-sender Channels
Cluster-receiver Channels
Cluster-sender channels:
FR1 - Full Repository
PR3
1
FR1

Channel name	Cluster queue manager	Queue manager type	Definition type	Xm
TO.FR1	FR1	Repository	Auto explicit cluster-sender	TCI
TO.FR2	FR2	Repository	Auto explicit cluster-sender	TCI

These channels were explicitly specified with the wizard, thus, under the 'Definition Type' the value is:
Auto explicit cluster-sender

Note:
If you had selected only TO.FR1
… then TO.FR2 would have been added as a result of the incorporation of the new queue manager into the cluster. Under the 'Definition Type' the value is:
auto cluster-sender (Notice the absence of the keyword 'explicit'.)

Although the above channels were created differently, both channels will have the same behavior.

Proceed to add the other Partial Repositories to the cluster via line commands.

+++ Using runmqsc commands to add PR4 into the cluster.

From a command prompt in the Linux host (where FR2 and PR4 reside)

Login as an MQ Administrator.

```
$ runmqsc PR4
```

Define a Cluster Receiver channel.
Every queue manager in a cluster must define a cluster-receiver channel on which it can receive messages. The cluster-receiver channel advertises the availability of the queue manager to receive messages from other queue managers in the cluster. There is no need to make definitions on other queue managers for a sending end to the cluster-receiver channel TO.PR4. Other definitions are made automatically when needed.

Note: This command adds the Partial repository to the cluster. There is no equivalent of “alter qmgr” for adding partial repositories.

```
DEFINE CHANNEL(TO.PR4) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
  CONNAME('chamonix1.x.com(1424)') CLUSTER(CLUSTER1)
```

Define a CLUSSDR channel for at least one full repository.
Every queue manager in a cluster must define one cluster-sender channel to send messages to the full repositories.

```
DEFINE CHANNEL(TO.FR1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
  CONNAME('stmichel1.x.com(1421)') CLUSTER(CLUSTER1)
```

The full repository used above will notify eventually the other full repository and an automatic channel will be created later on between the partial and the other full repository.

However, it is OK to manually define the channel between the partial and the other full repository:

```
DEFINE CHANNEL(TO.FR2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
  CONNAME('chamonix1.x.com(1422)') CLUSTER(CLUSTER1)
```

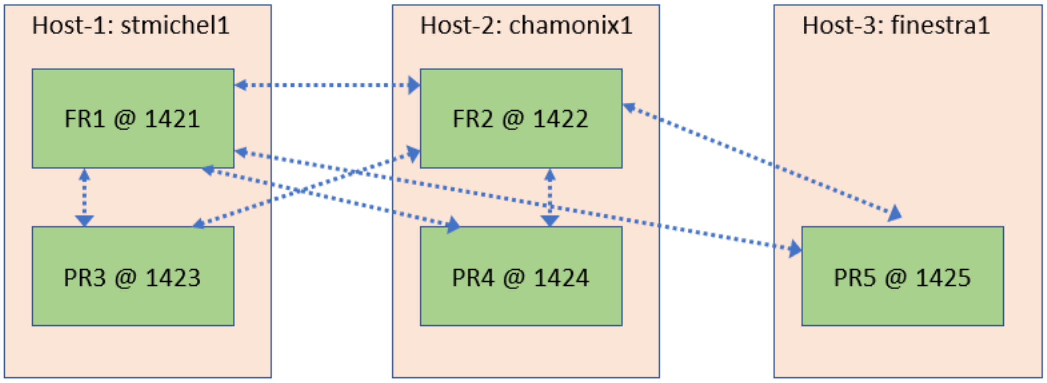
+ Repeat the process for the 3rd Partial Repository, PR5, from the Windows host

runmqsc PR5

DEFINE CHANNEL(TO.PR5) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('finestra1.x.com(1425)') CLUSTER(CLUSTER1)

DEFINE CHANNEL(TO.FR1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('stmichel1.x.com(1421)') CLUSTER(CLUSTER1)

+ The topology now looks like this:



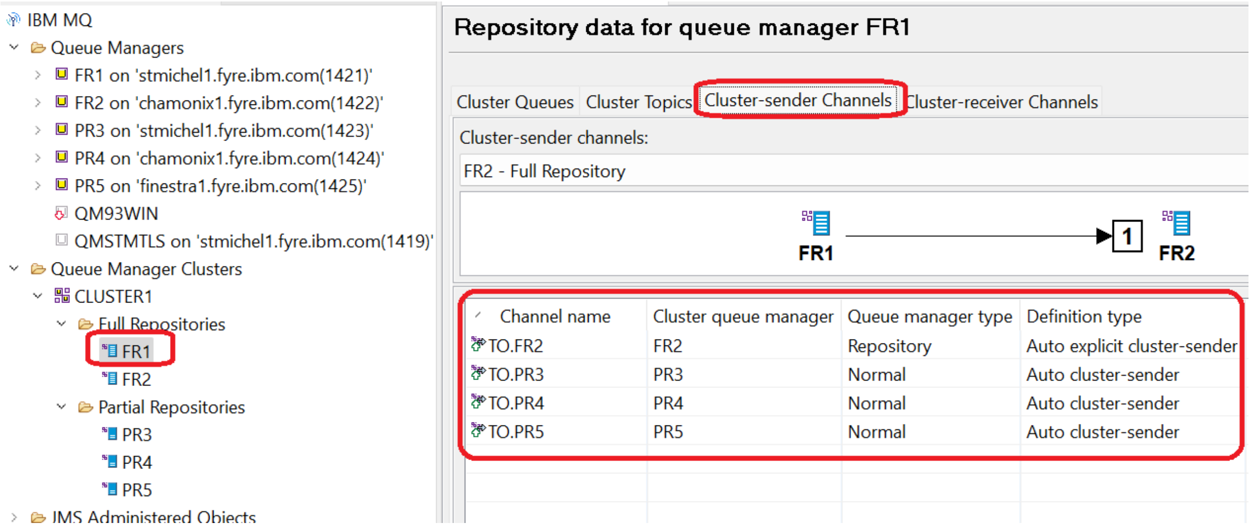
For simplicity, the connecting arrows do not have the names of the pairs of cluster-sender and cluster-receiver channels between each queue manager.

Note: At this point, there are NO channels between Partials.
There is NO direct connection time between the partial repositories PR3, PR4 and PR5. Later on, due to the usage of a clustered queue in Chapter 5, a channel between the appropriate partial repositories will be automatically created.

+ The MQ Explorer shows now our cluster:

Note: You may need to wait few minutes for the cluster information for the new PR4 and PR5 to be propagated.

The following shows the "Cluster-sender Channels" from other queue managers to FR1:



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
      - FR1
      - FR2
    - Partial Repositories
      - PR3
      - PR4
      - PR5
- JMS Administered Objects

Repository data for queue manager FR1

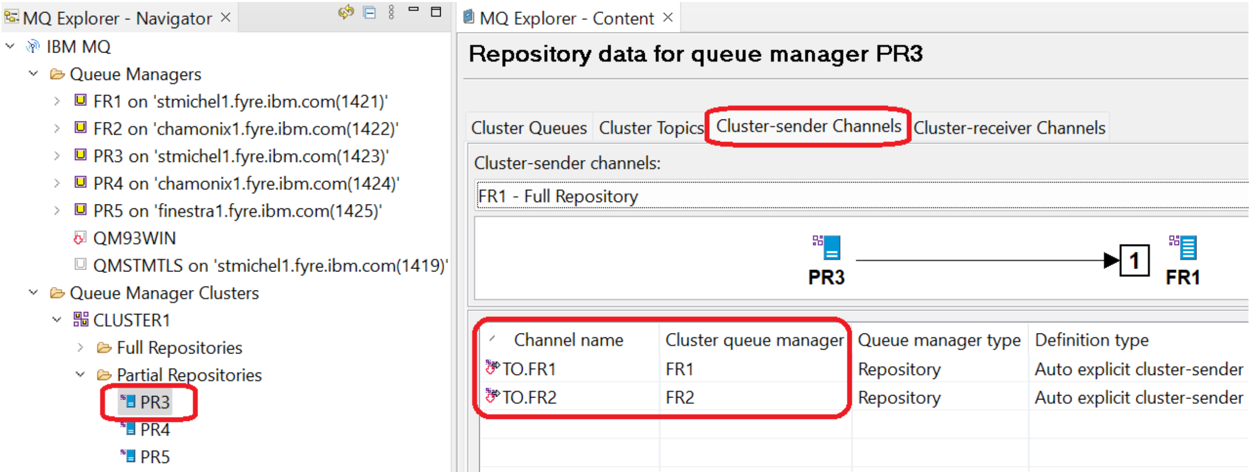
Cluster-sender channels:

FR2 - Full Repository

Channel name	Cluster queue manager	Queue manager type	Definition type
TO.FR2	FR2	Repository	Auto explicit cluster-sender
TO.PR3	PR3	Normal	Auto cluster-sender
TO.PR4	PR4	Normal	Auto cluster-sender
TO.PR5	PR5	Normal	Auto cluster-sender

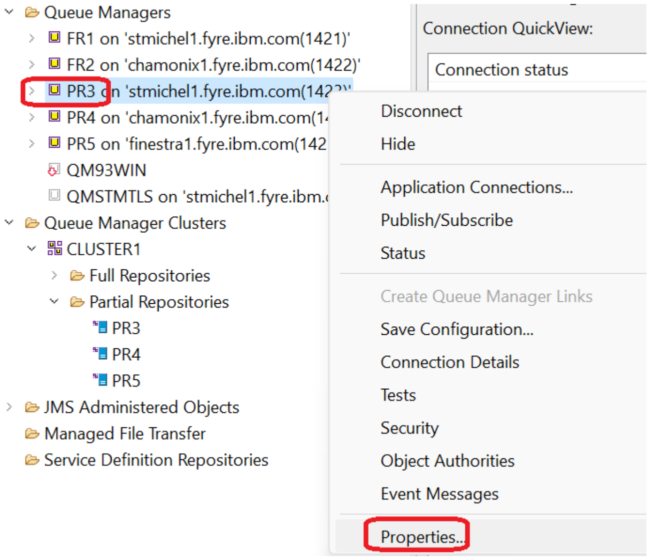
++ Let's explore some aspects of a Partial Repository:

Select Partial Repository PR3.
Click on the tab "Cluster-sender Channels" to see the 2 cluster-sender channels that connect this partial repository to the full repositories.

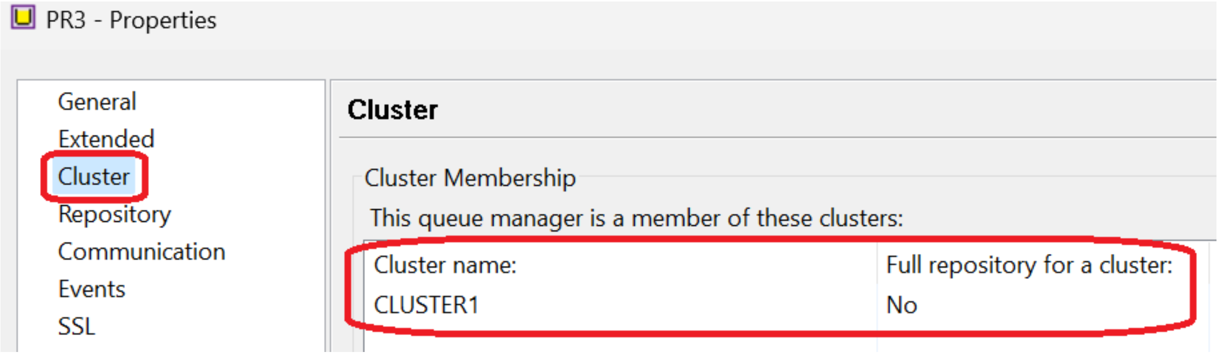

MQ Explorer - Navigator
IBM MQ
Queue Managers
FR1 on 'stmichel1.fyre.ibm.com(1421)'
FR2 on 'chamonix1.fyre.ibm.com(1422)'
PR3 on 'stmichel1.fyre.ibm.com(1423)'
PR4 on 'chamonix1.fyre.ibm.com(1424)'
PR5 on 'finestra1.fyre.ibm.com(1425)'
QM93WIN
QMSTMTLS on 'stmichel1.fyre.ibm.com(1419)'
Queue Manager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

Channel name	Cluster queue manager	Queue manager type	Definition type
TO.FR1	FR1	Repository	Auto explicit cluster-sender
TO.FR2	FR2	Repository	Auto explicit cluster-sender

From the left panel, under Queue Managers, select the entry for PR3 and right click to show the properties



Select the tab for "Clusters".
Notice that this queue manager belongs to the cluster CLUSTER1.



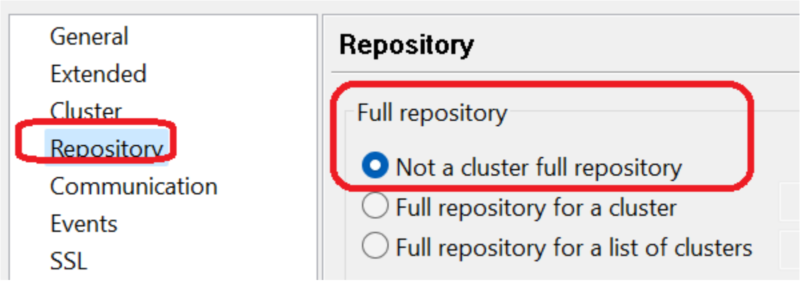
Cluster

Cluster Membership

This queue manager is a member of these clusters:

Cluster name:	Full repository for a cluster:
CLUSTER1	No

Click the tab "Repository".



Repository

Full repository

● Not a cluster full repository
○ Full repository for a cluster
○ Full repository for a list of clusters

+ Let's explore some of the Cluster attributes for the Partial Repository PR3

runmqsc PR3

Notice that in the following query, a null value in REPOS indicates that this is NOT a Full Repository and consequently, it is a Partial Repository.

```
display qmgr repos
AMQ8408I: Display Queue Manager details.
   QMNAME(PR3)                             REPOS( )
```

To find out the cluster to which this queue manager belongs to, it is necessary to use the following command.

Notice the attribute-value pair: CLUSTER(CLUSTER1)

Also, notice that the QMTYPE attribute has a value of NORMAL, which indicates that this is NOT a Full Repository and consequently, it is a Partial Repository. A Full Repository will have as value "REPOS".

```
display clusqmgr(PR3)
AMQ8441I: Display Cluster Queue Manager details.
  CLUSQMGR(PR3)              ALTDATE(2024-03-13)
  BATCHSZ(50)              CHANNEL(TO.PR3)
  CLUSDATE(2024-03-13)        CLUSTER(CLUSTER1)
  CLUSTIME(15.40.05)          CLWLPRTY(0)
  CLWLRANK(0)               CLWLWGHT(50)
  CONNAME(stmichel1.fyre.ibm.com(1423))   CONVERT(NO)
  DEFTYPE(CLUSRCVR)           DESCR( )
  QMID(PR3_2024-03-02_09.58.47)      QMTYPE(NORMAL)
  RCVDATA( )              RCVEXIT( )
  SCYDATA( )              SCYEXIT( )
  SENDDATA( )              SENDEXIT( )
  SEQWRAP(999999999)          SHORTRTY(10)
  SHORTTMR(60)              SSLCAUTH(REQUIRED)
  SSLCIPH( )             SSLPEER( )
  STATUS(INACTIVE)           SUSPEND(NO)
  TPNAME( )               TRPTYPE(TCP)
  USEDLQ(YES)              USERID( )
  VERSION(09030016)          XMITQ( )
```

+++
+++ Chapter 4: Working with a local queue, not yet a clustered queue
+++

The objective of this chapter is to show that local queues defined in one queue manager are NOT automatically made visible to the other members of the cluster.

The local queue CQ1 is defined in the queue manager PR3.

+ Expected behavior:

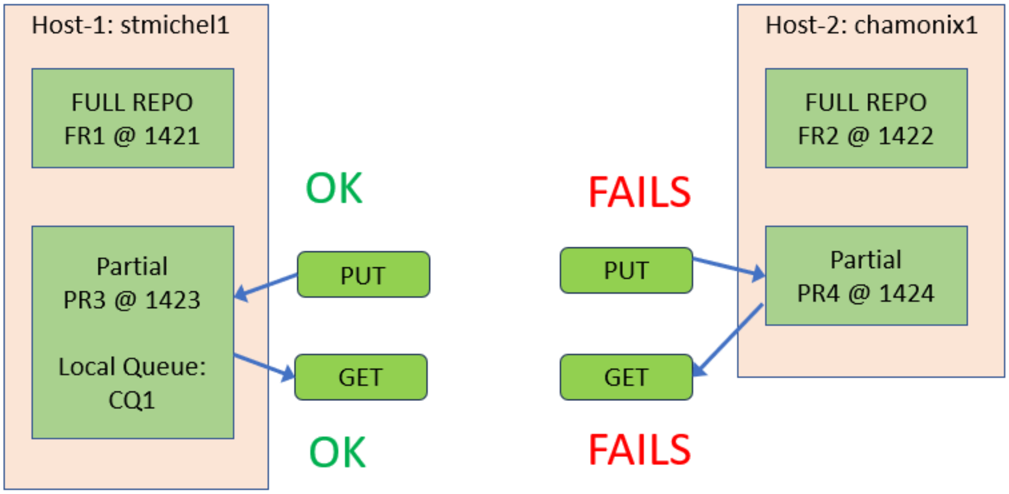
- OK. An application such as "amqsput" can put a message to the local CQ1 when connected to PR3.

- OK. An application such as "amqsget" can get a message from the local CQ1 when connected to PR3.

- FAILED. Another application CANNOT put a message into CQ1 when connected to another queue manager, such as PR4 (because CQ1 is NOT known to PR4).

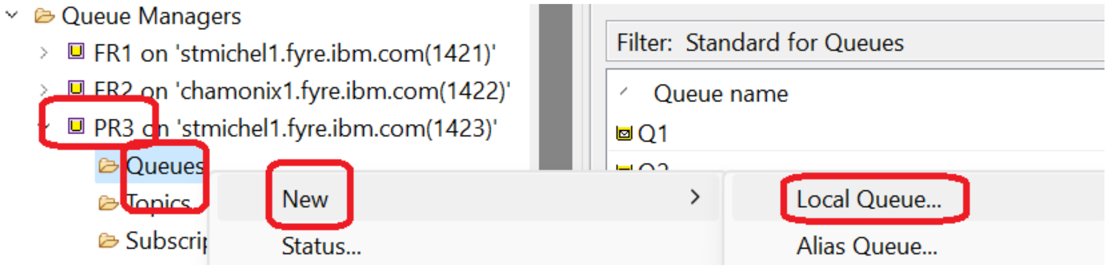
- FAILED. Another application CANNOT get a message from CQ1 when connected to another queue manager, such as PR4 (because CQ1 is NOT known to PR4).

These scenarios are illustrated in the following diagram:

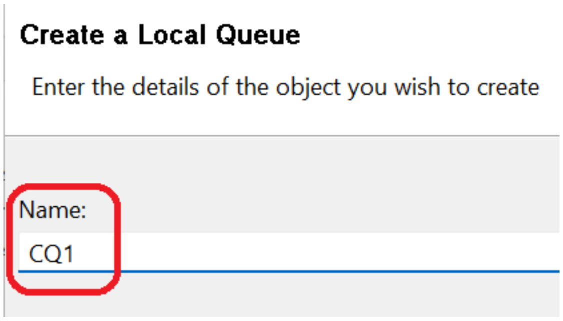


+ Using the MQ Explorer to create a local queue CQ1 in PR3

Select the queue manager PR3 and expand it.
Then right-click on the folder "Queues", select "New" then "Local Queue ..."


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
New
Status...
Local Queue...
Alias Queue...
Filter: Standard for Queues
Queue name
Q1

Enter the name "CQ1" and click Next.

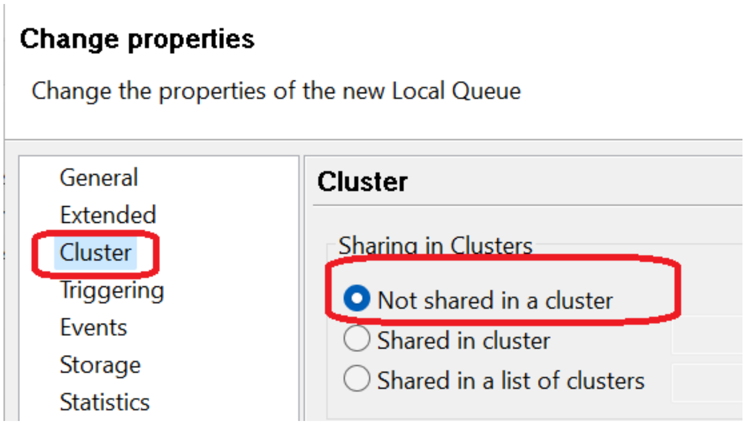


Create a Local Queue

Enter the details of the object you wish to create

Name: CQ1

You will see the Properties for the queue.
Click the tab "Cluster".
Notice that this queue is NOT in the cluster at this moment (we will do it later on).



Change properties

Change the properties of the new Local Queue

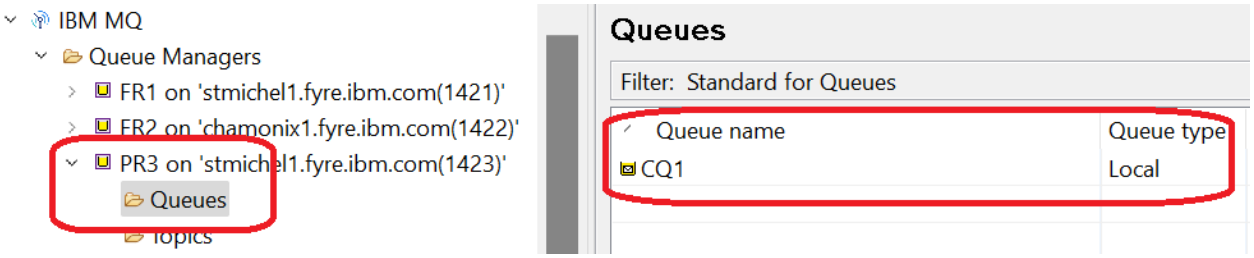
Cluster

Sharing in Clusters

● Not shared in a cluster
○ Shared in cluster
○ Shared in a list of clusters

Click Finish to create the queue.

We have now a local queue called: CQ1


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

Queues

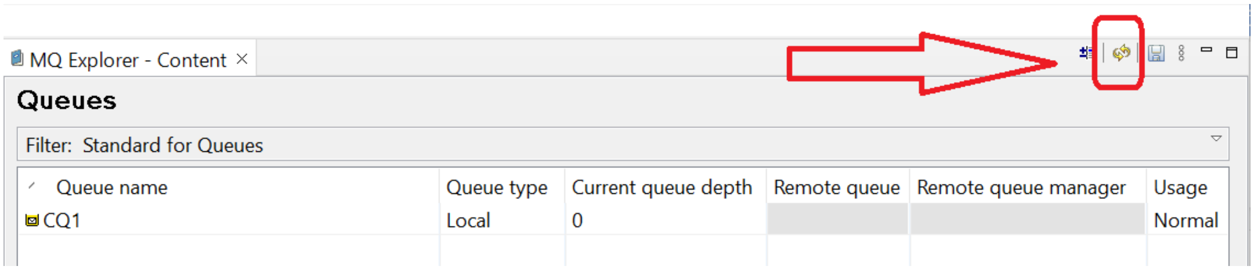
Filter: Standard for Queues

Queue name	Queue type
CQ1	Local

+ Using runmqsc to create a local queue "CQ2" in PR3

```
runmqsc PR3
  define qlocal(CQ2)
  end
```

Then let's go back to the MQ Explorer to see the new queue.
In case that you do not see right away the new queue in MQ Explorer, then you need to click the Refresh icon (shown on the right corner of the following picture)

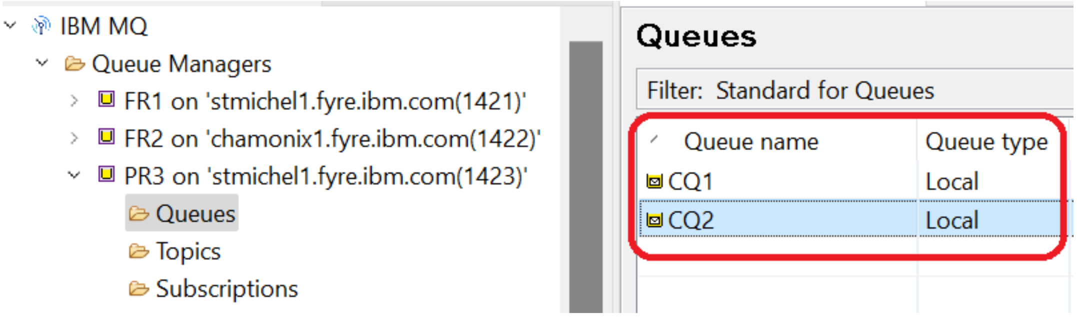


Queues

Filter: Standard for Queues

Queue name	Queue type	Current queue depth	Remote queue	Remote queue manager	Usage
CQ1	Local	0			Normal

Now you should see the 2 queues that were created:


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

Queues

Filter: Standard for Queues

Queue name	Queue type
CQ1	Local
CQ2	Local

+ Using runmqsc to look at the cluster properties for a queue:

Notice that the CLUSTER attribute is null, which indicates that this is NOT a clustered queue.

```
runmqsc PR3
```

```
display ql(CQ1) cluster
AMQ8409I: Display Queue details.
   QUEUE(CQ1)                              TYPE(QLOCAL)
   CLUSTER( )
```

++ Testing the Put and Get samples on a local queue (not clustered yet)

Open 3 command prompts and ensure that the MQ environment is properly setup for the desired version.
Then specify the MQSERVER environment variable to connect to the proper queue manager.

+ Successful Put and Get using a local queue in PR3

Login into the host of PR3 as a user that has proper authority to connect to the queue manager and to put messages.

Use the amqsput sample to put a message into CQ1 in PR3

```
mqm@stmichel1.fyre.ibm.com: /home/mqm
$ amqsput CQ1 PR3
Sample AMQSPUT0 start
target queue is CQ1
TEST-PUT-1 (then press ENTER)
(do not type anything, then press ENTER to exit)
Sample AMQSPUT0 end
```

Use the amqsget sample to get a message from CQ1 in PR3

```
amqsget CQ1 PR3
Sample AMQSGET0 start
message <TEST-PUT-1>
no more messages
Sample AMQSGET0 end
```

Let's put another message, to be ready for the next test case.

```
amqsput CQ1 PR3
Sample AMQSPUT0 start
target queue is CQ1
TEST-PUT-2 (then press ENTER)
(do not type anything, then press ENTER to exit)
Sample AMQSPUT0 end
```

+ Unsuccessful Get from local queue CQ1 in PR3 when using another queue manager from the cluster (PR4)

Notice that the queue CQ1 is NOT defined in PR4.

Login into the host of PR4

```
mqm@chamonix1.fyre.ibm.com: /home/mqm
$ amqsget CQ1 PR4
Sample AMQSPUT0 start
target queue is CQ1
MQOPEN ended with reason code 2085
unable to open queue for input
Sample AMQSPUT0 end
```

The return code 2085 indicates that the queue manager PR4 does not have a queue CQ1.

```
$ mqrc 2085
     2085  0x00000825  MQRC_UNKNOWN_OBJECT_NAME
```

+ Unsuccessful Put from another queue manager (PR4) from the cluster into a queue that is local CQ1 to PR3

Notice that the queue CQ1 is NOT defined in PR4, and it is not in the cluster.

```
$ amqsput CQ1 PR4
Sample AMQSPUT0 start
target queue is CQ1
MQOPEN ended with reason code 2085
unable to open queue for output
Sample AMQSPUT0 end
```

Notice the same problem as before, the local queue CQ1 in PR3 is not a clustered queue and thus, it is not available in the other queue managers in the cluster.

++

+++ Chapter 5: Working with a queue that was made a clustered queue

++

The objective of this chapter is to show that local queues defined in one queue manager and which are added to the cluster, will be automatically made visible to the other members of the cluster.

The local queue CQ1 which was defined in the queue manager PR3 in Chapter 4, is now added to the cluser.

+ Expected behavior now that the queue is clustered:

- OK. An application such as "amqsput" can put a message to the local CQ1 when connected to PR3.

- OK. An application such as "amqsget" can get a message from the local CQ1 when connected to PR3.

- OK. Another application can put a message into the clustered queue CQ1 when connected to another queue manager, such as PR4.

- FAILED. Another application CANNOT get a message from the clustered queue CQ1 when connected to another queue manager, such as PR4.
  NOTE: This is a restriction of a clustered queue: the GET operation can ONLY be done with the local queue of the queue manager (where the physical storage for the queue is available)

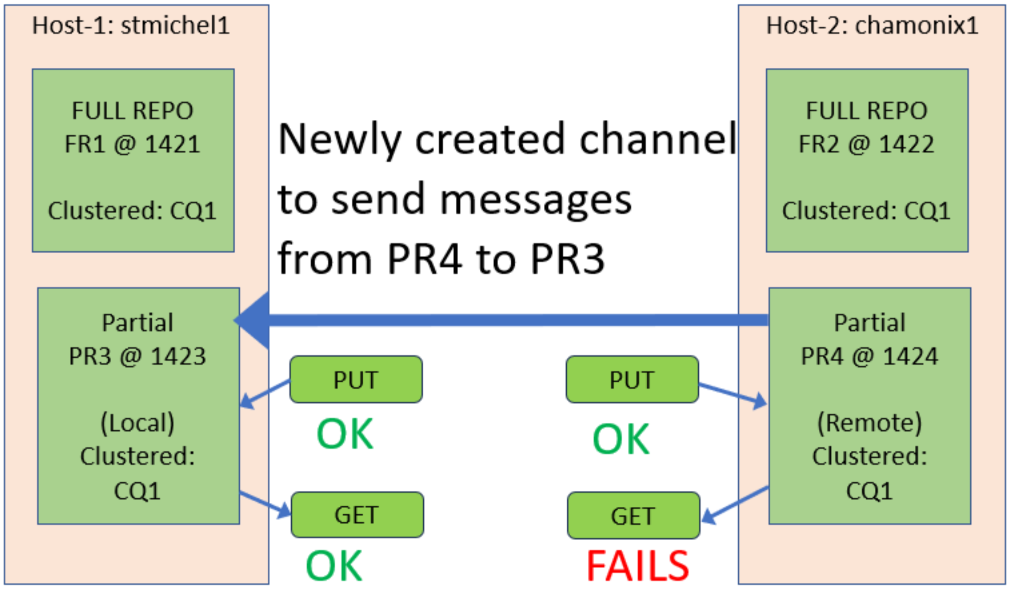
When the local queue CQ1 in PR3 is added to the cluster, then PR3 notifies FR1 about it, and FR1 synchronizes with FR2. Thus, both FR1 and FR2 indicate that CQ1 is now a clustered queue in PR3.

When an application connects to PR4 and wants to put a message into CQ1, the first reaction of PR4 is to say "I do not have CQ1 in my inventory of queues, BUT because I am in a cluster, I am going to check with a Full Repository (FR2)".

Then FR2 tells PR4: "yes, CQ1 is a clustered queue and the physical queue resides in PR3, thus, you need to create a NEW CLUSTER SENDER and connect with the CLUSTER RECEIVER from PR3".

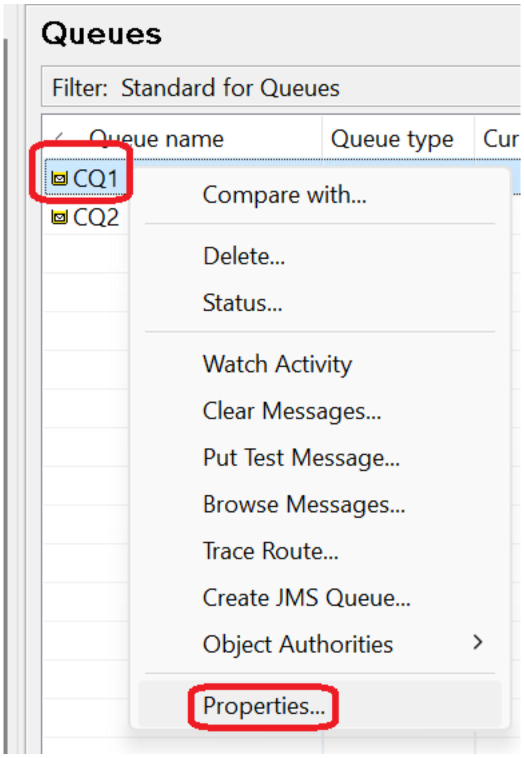
In addition, FR2 tells PR4: "you need to create a remote queue definition for CQ1 that uses the SYSTEM.CLUSTER.TRANSMIT.QUEUE for PR3, using the new cluster sender-receiver pair between PR4 and PR3".

These scenarios are illustrated in the following diagram:

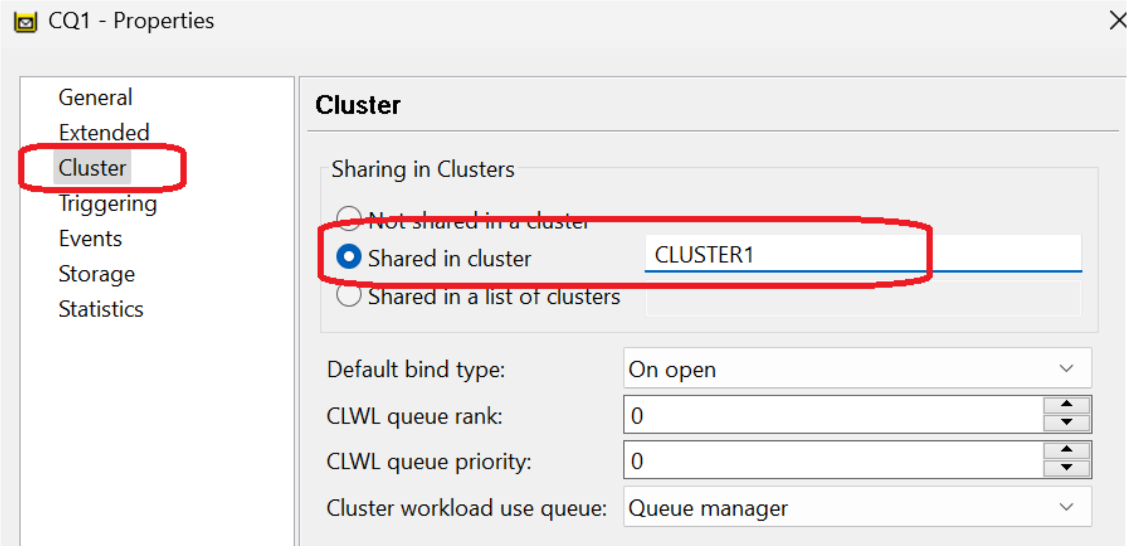


+ Using MQ Explorer to add the local queue CQ1 from PR3 into the cluster CLUSTER1. This will make the queue a "clustered queue" or a "cluster queue".

In PR3 select the queue CQ1, right click and select "Properties..."



Select the tab "Cluster" in the Properties dialog.
Click on the radio button: Shared in cluster
Then specify the name of the cluster: CLUSTER1
Click OK



○ Not shared in a cluster
● Shared in cluster CLUSTER1
○ Shared in a list of clusters

Default bind type: On open
CLWL queue rank: 0
CLWL queue priority: 0
Cluster workload use queue: Queue manager

The clustering code will notice that this queue is now a clustered queue and will propagate the information thru the cluster.

++ Using runmqsc to alter a queue to be a clustered queue

runmqsc PR3

ALTER QLOCAL(CQ2) CLUSTER('CLUSTER1')
AMQ8008I: IBM MQ queue changed.

++ Displaying the cluster attributes for a queue:

Notice that this queue CQ1 now belongs to **CLUSTER(CLUSTER1).**

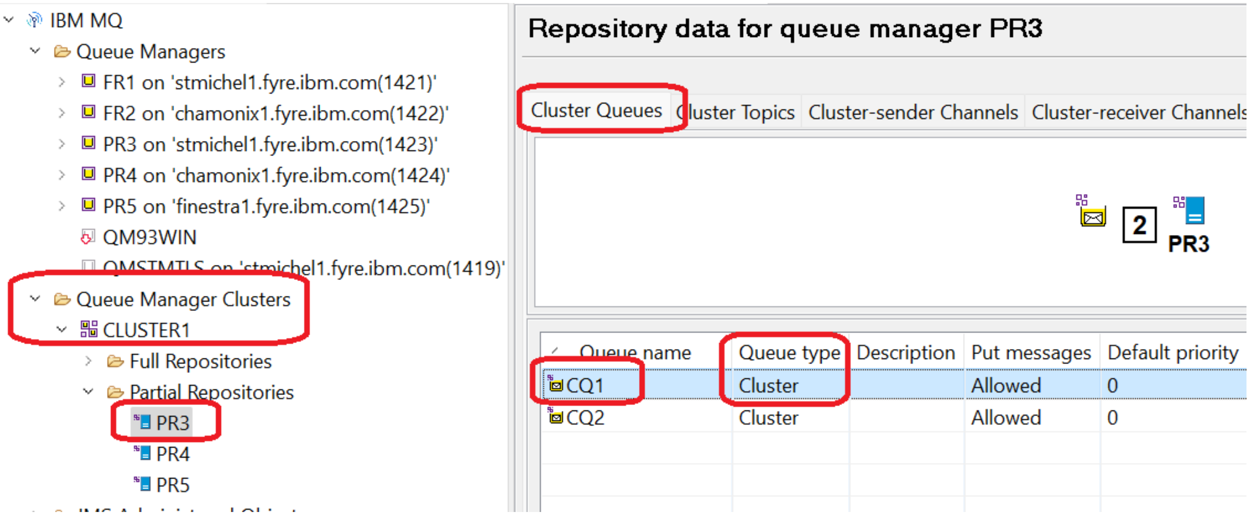
runmqsc PR3

```
display qlocal(CQ1) cluster
AMQ8409I: Display Queue details.
   QUEUE(CQ1)                              TYPE(QLOCAL)
   CLUSTER(CLUSTER1)
```

+ In MQ Explorer, let's look at the details from the cluster perspective

In the left Navigator, open the folder "Queue Manager Clusters" and select:
  CLUSTER1
Expand the "Partial Repositories" and select:
  PR3

In the right panel, click the tab "Cluster Queues"
Notice that there is a Queue named "CQ1" which has a Queue type of "Cluster".



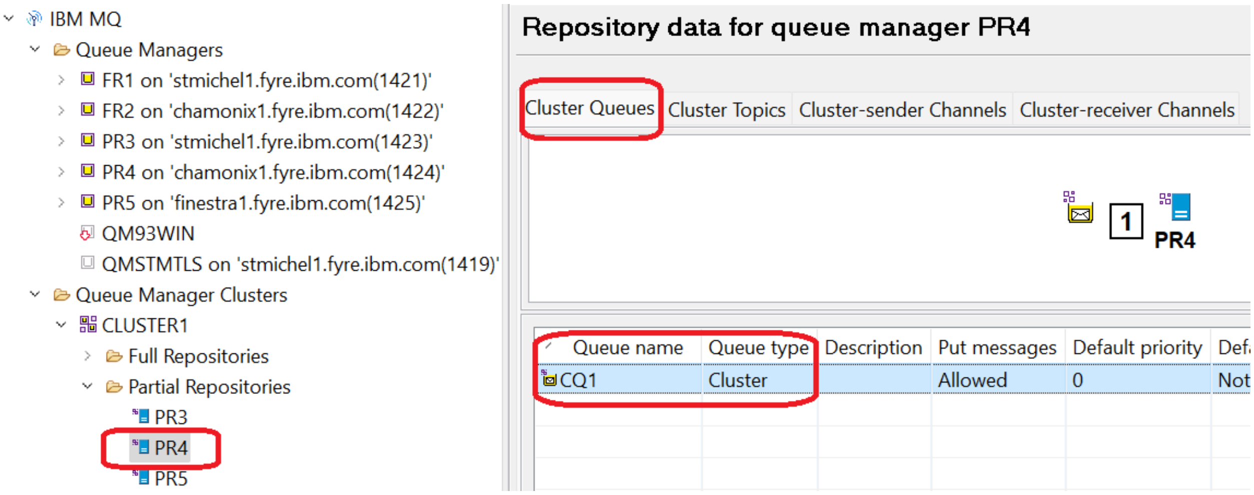
IBM MQ
- Queue Managers
  - FR1 on 'stmichel1.fyre.ibm.com(1421)'
  - FR2 on 'chamonix1.fyre.ibm.com(1422)'
  - PR3 on 'stmichel1.fyre.ibm.com(1423)'
  - PR4 on 'chamonix1.fyre.ibm.com(1424)'
  - PR5 on 'finestra1.fyre.ibm.com(1425)'
  - QM93WIN
  - QMSTMTLS on 'stmichel1.fyre.ibm.com(1419)'
- Queue 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

Queue name	Queue type	Description	Put messages	Default priority
CQ1	Cluster		Allowed	0
CQ2	Cluster		Allowed	0

Click on the other partial repository, PR4, and verify that it has an entry for that queue.


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
Cluster Queues
Cluster Topics
Cluster-sender Channels
Cluster-receiver Channels
1
PR4

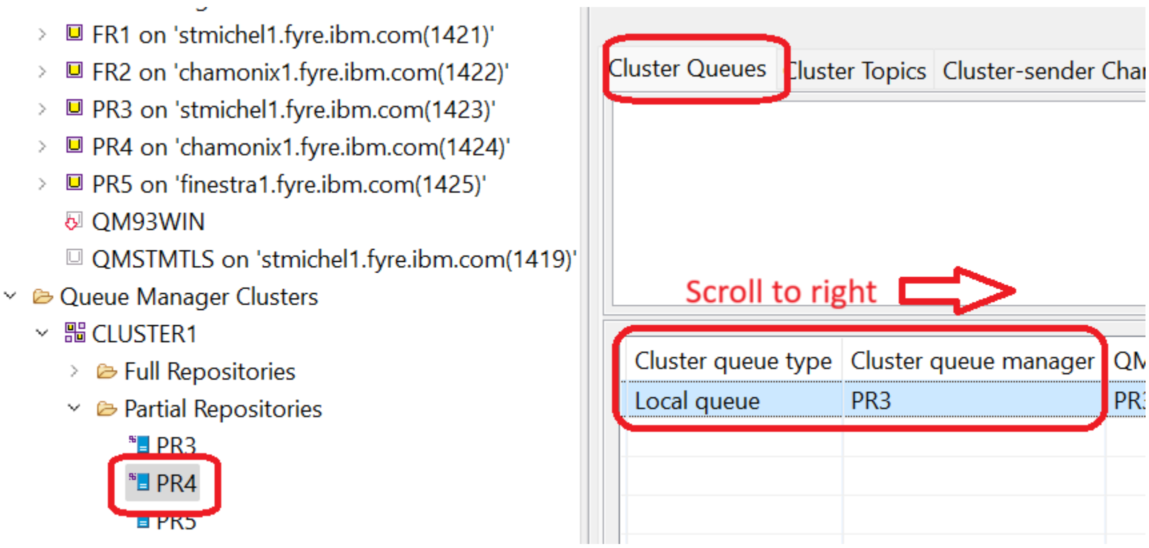
Repository data for queue manager PR4

Queue name	Queue type	Description	Put messages	Default priority	Def
CQ1	Cluster		Allowed	0	Not

Given a clustered queue, you can find out where is the local queue.
This is important because an MQGET operation can ONLY be done against the local queue.

From the above view, you can scroll to the right to see the columns:
Cluster queue type
Cluster queue manager

In this case, the clustered queue CQ1 is mapped to the "local queue CQ1" residing in PR3.


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
Cluster Queues
Cluster Topics
Cluster-sender Char

Scroll to right

Cluster queue type	Cluster queue manager	QM
Local queue	PR3	PR:

In PR4 you can use runmqsc to find out this information:

```
display qcluster(CQ1)
AMQ8409I: Display Queue details.
   QUEUE(CQ1)                              TYPE(QCLUSTER)
   ALTDATE(2024-03-14)                     ALTTIME(07.33.58)
   CLUSDATE(2024-03-14)                    CLUSTER(CLUSTER1)
   CLUSQMGR(PR3)                           CLUSQT(QLOCAL)
   CLUSTIME(07.33.58)                      CLWLPRTY(0)
   CLWLRANK(0)                             DEFBIND(OPEN)
   DEFPRTY(0)                              DEFPSIST(NO)
   DEFPRESP(SYNC)                          DESCR( )
   PUT(ENABLED)                            QMID(PR3_2024-03-02_09.58.47)
```

++ Putting a message into a clustered queue

+ Successful put to CQ1 from PR4

Notice that the rc 2085 (about object not defined) is not returned, as in the corresponding test in Chapter 4.

Login as an MQ Administrator in the host that has PR4

```
mqm@chamonix1.fyre.ibm.com: /home/mqm
$ amqsput CQ1 PR4
Sample AMQSPUT0 start
target queue is CQ1
TEST-PUT-3
Sample AMQSPUT0 end
```

Notice that the message with text "TEST-PUT-3" was successfully placed in the queue CQ1 in the remote partial repository PR4.

But this queue manager does NOT have a local queue named CQ1!

Instead, this queue manager has a "cluster queue" CQ1 that points to the location of where a local queue is located and which can store messages.

Let's explore this in detail:

```
$ runmqsc PR4
```

Notice that there is no local queue CQ1

```
display qlocal(CQ1)
AMQ8147E: IBM MQ object CQ1 not found.
```

We need to use the following command instead:

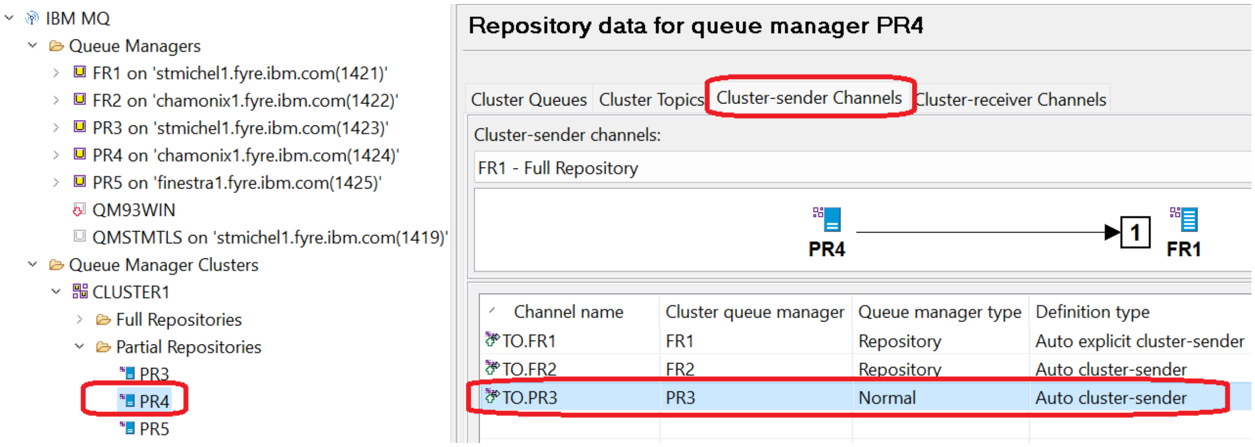
```
display qcluster(CQ1)
AMQ8409I: Display Queue details.
   QUEUE(CQ1)                            TYPE(QCLUSTER)
   ALTDATE(2024-03-14)                   ALTTIME(07.33.58)
   CLUSDATE(2024-03-14)                  CLUSTER(CLUSTER1)
   CLUSQMGR(PR3)                         CLUSQT(QLOCAL)
   CLUSTIME(07.33.58)                    CLWLPRTY(0)
   CLWLRANK(0)                           DEFBIND(OPEN)
   DEFPRTY(0)                            DEFPSIST(NO)
   DEFPRESP(SYNC)                        DESCR( )
   PUT(ENABLED)                          QMID(PR3_2024-03-02_09.58.47)
```

Notice that the following attributes indicate that this clustered queue CQ1 is a local queue in queue manager PR3:
CLUSQMGR(PR3) **CLUSQT(QLOCAL)**

During the MQ PUT to CQ1, the queue manager PR4 (a partial repository) could not find a local queue CQ1. Thus, it decided to ask one of the full repositories if CQ1 was a clustered queue. The answer was positive: the local queue for CQ1 is located in the other partial repository PR3.

However, PR4 does not have a channel that connects directly to PR3 in order to send the message. In this case, the MQ clustering function will AUTOMATICALLY create a cluster sender channel between PR4 and PR3, using the existing cluster receiver channel TO.PR3 already defined in PR3 (this information was provided to PR4 from the full repository).

This can be seen in the MQ Explorer:


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
Cluster Queues
Cluster Topics
Cluster-sender Channels
Cluster-receiver Channels
PR4
1
FR1

Repository data for queue manager PR4

Cluster-sender channels:

FR1 - Full Repository

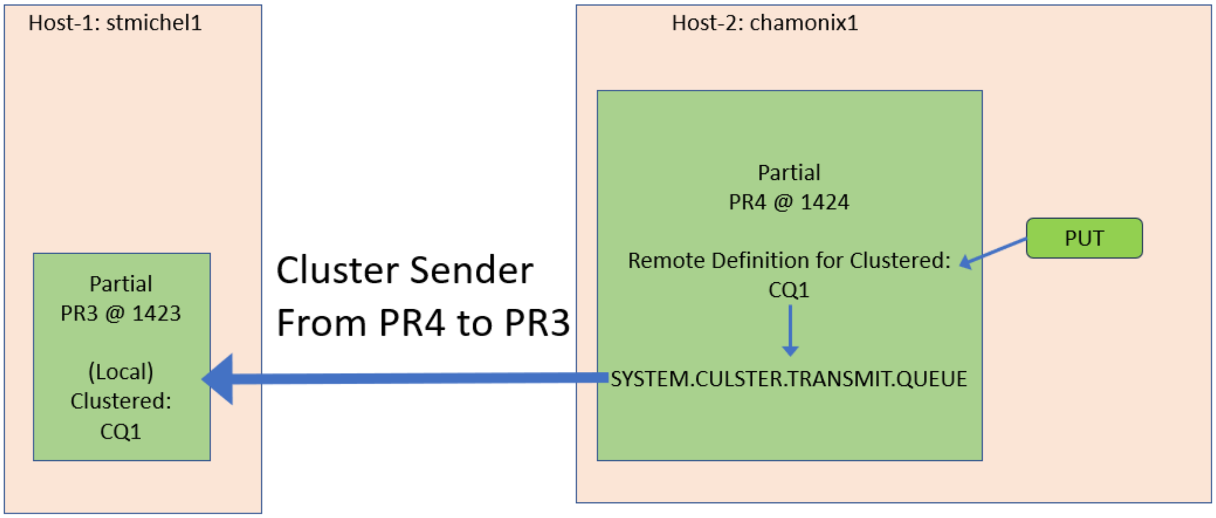
Channel name	Cluster queue manager	Queue manager type	Definition type
TO.FR1	FR1	Repository	Auto explicit cluster-sender
TO.FR2	FR2	Repository	Auto cluster-sender
TO.PR3	PR3	Normal	Auto cluster-sender

The MQ clustering code will create an auto-defined Cluster Sender channel from PR4 to PR3. This channel is called TO.PR3.

In addition, the MQ clustering code will create an internal remote queue definition in PR4 for CO1 which uses the SYSTEM.CLUSTER.TRANSMIT.QUEUE and the Cluster Sender TO.PR3.

When the application puts the message into the remote queue definition of the clustered queue CQ1, the message is forwarded to a special clustering queue called:
SYSTEM.CLUSTER.TRANSMIT.QUEUE

Then the message travels via the newly created Cluster Sender channel TO.PR3 to Partial PR3 which has the local (physical) queue CQ1.



There is a usage note regarding DISPLAY CHANNEL that is relevant for this situation:

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=cluster-working-auto-defined-sender-channels
Working with auto-defined cluster-sender channels
After you introduce a queue manager to a cluster by making its initial CLUSSDR and CLUSRCVR definitions, IBM® MQ automatically makes other cluster-sender channel definitions when required to move messages to another queue manager in the cluster. You can view information about auto-defined cluster-sender channels, but you cannot modify them.

+ You cannot see "auto-defined channels" using the runmqsc command:

```
  DISPLAY CHANNEL
```

Because the newly created channel TO.PR3 is an "auto cluster-sender" (not defined manually), then the DISPLAY CHANNEL will NOT show it!

Note that sender channel to the Full Repository TO.FR1 is shown because it was manually created (that is, it is NOT an auto-defined channel)

runmqsc PR4

```
display channel(*) CHLTYPE(CLUSSDR)
AMQ8414I: Display Channel details.
   CHANNEL(SYSTEM.DEF.CLUSSDR)             CHLTYPE(CLUSSDR)
AMQ8414I: Display Channel details.
   CHANNEL(TO.FR1)                         CHLTYPE(CLUSSDR)
```

+ To see the channel to be used for a remote Queue Manager, use the command:

```
  DISPLAY CLUSQMGR(remoteQMgrName) CHANNEL XMITQ
```

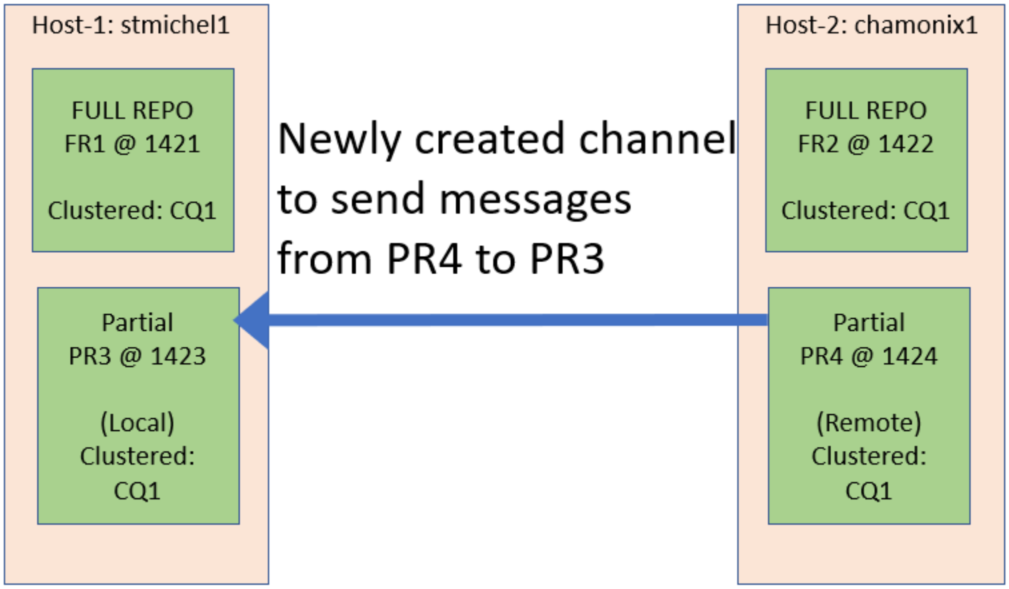
runmqsc PR4

```
display clusqmgr(PR3) channel
AMQ8441I: Display Cluster Queue Manager details.
   CLUSQMGR(PR3)                           CHANNEL(TO.PR3)
   CLUSTER(CLUSTER1)
```

+ To display the status of the auto-defined CLUSSDR channel corresponding to the CLUSRCVR channel definition you created, use the command:
DISPLAY CHSTATUS(channelname)

```
display chstatus(TO.PR3)
display chstatus(TO.PR3)
AMQ8417I: Display Channel Status details.
   CHANNEL(TO.PR3)                   CHLTYPE(CLUSSDR)
   CONNAME(9.46.98.124(1423))              CURRENT
   RQMNAME(PR3)                     STATUS(RUNNING)
   SUBSTATE(MQGET)                    XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

Thus, the topology needs to be updated to show (in a Blue arrow) the new cluster channel between PR3 and PR4.



Notice that the auto defined channel was added “on-demand” between PR4 and PR3 because PR4 needed to send a message to PR3 via the clustered queue CQ1 whose local (physical) location is in PR3.

+ At this point, there are NO auto defined channels regarding PR5.

Because we have NOT tried yet to put a message to CQ1 from the Partial PR5, then PR5 does NOT know yet about the clustered queue CQ1 and does NOT have yet a channel from PR5 to PR3.

Login into the host of PR5

```
display qlocal(CQ1)
AMQ8147E: IBM MQ object CQ1 not found.
display qcluster(CQ1)
AMQ8147E: IBM MQ object CQ1 not found.
```

+++

+++ Chapter 6: Scenario when non MQ Administrators do not have proper authority

+++

This chapter explores 1 scenario for an unsuccessful put due to lack of proper authority (non MQ administrator), in which the reason code 2035 (MQRC_NOT_AUTHORIZED) is issued.

+ Scenario using local authority for SYSTEM.CLUSTER.TRANSMIT.QUEUE in PR4

The following figure illustrates the overall scenario:
The local clustered queue CQ1 resides in PR3.
There is no local clustered queue CQ1 in PR4: only the definition that CQ1 is a clustered queue.

Reference:

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=clusters-access-control-multiple-cluster-transmission-queues
IBM MQ / 9.3
Access control and multiple cluster transmission queues

+ begin

Choose between three modes of checking when an application puts messages to remote cluster queues. The modes are:
- checking remotely against the cluster queue,
- checking locally against SYSTEM.CLUSTER.TRANSMIT.QUEUE, or
- checking against local profiles for the cluster queue, or cluster queue manager.

+ end

Let's begin.

++ Need setmqaut for non-MQ administrator to connect to the queue manager

The non-MQ administrator will receive a reason code of 2035 when trying to connect to the queue manager.

1) Login as an MQ Administrator and verify that at this point there are no authorities in the queue manager PR4 for the userid "fulano" member of the group "mquser".

```
mqm@chamonix1.fyre.ibm.com: /home/mqm
$ dmpmqaut -m PR4 -g mquser
No matching authority records.
```

2) Login as a user that is not an MQ Administrator. In this case it is user: fulano

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

At this time, this user does NOT have any authorizations with the queue manager.

```
fulano@chamonix1.fyre.ibm.com: /home/fulano
$ amqsput CQ1 PR4
Sample AMQSPUT0 start
MQCONNX ended with reason code 2035
```

The error 2035 means:

```
fulano@chamonix1:~> mqrc 2035
      2035  0x000007f3  MQRC_NOT_AUTHORIZED
```

The reason for this rc 2035 during MQCONN is that the user must have authorization to connect to the queue manager.

3) Login as an MQ Administrator and give authority to the userid "fulano" to connect to the queue manager PR4.

```
$ setmqaut -m PR4 -t qmgr -g mquser +connect +inq +dsp
The setmqaut command completed successfully.
```

```
$ dmpmqaut -m PR4 -g mquser
profile:     self
object type: qmgr
entity:      mquser
entity type: group
authority:   inq connect dsp
- - - - - - - -
profile:     @class
object type: qmgr
entity:      mquser
entity type: group
authority:   none
```

++ Need setmqaut for non-MQ administrator to put into the cluster transmit queue

4) Login again as "fulano" and try to put the message again.

```
fulano@chamonix1.fyre.ibm.com: /home/fulano
$ amqsput CQ1 PR4
Sample AMQSPUT0 start
target queue is CQ1
MQOPEN ended with reason code 2035
unable to open queue for output
Sample AMQSPUT0 end
```

Note that there are more entries in the output, such as:

```
  target queue is CQ1
  MQOPEN ended with reason code 2035
```

Now the question regarding the rc 2035 is: what type of authority is needed and on which object?

5) As an MQ Administrator for PR4 give "+put" authority to the proper queue:

View the bottom of the error log for the queue manager:

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

See the error message:

```
03/14/2024 12:47:16 PM - Process(409848.37) User(mqm) Program(amqzlaa0)
                    Host(chamonix1.fyre.ibm.com) Installation(Installation1)
                    VRMF(9.3.0.16) QMgr(PR4)
                    Time(2024-03-14T19:47:16.336Z)
                    CommentInsert1(fulano)
                    CommentInsert2(SYSTEM.CLUSTER.TRANSMIT.QUEUE [queue])
                    CommentInsert3(put)
AMQ8077W: Entity 'fulano' has insufficient authority to access object
SYSTEM.CLUSTER.TRANSMIT.QUEUE [queue].
EXPLANATION:
The specified entity is not authorized to access the required object. The
following requested permissions are unauthorized: put
```

Thus, you need to give access to this queue:

```
$ setmqaut -m PR4 -t queue -n SYSTEM.CLUSTER.TRANSMIT.QUEUE -g mquser +put
```

From the online manual regarding the following topic:
"Access control and multiple cluster transmission queues"
(The URL was provided earlier in this document)

+ begin excerpt

The local checks are made against the queue that the application opens.
In distributed queuing, the application usually opens a remote queue definition, and access checks are made against the remote queue definition.
If the message is put with a full routing header, the checks are made against the transmission queue.

If an application opens a cluster queue that is not on the local queue manager, there is no local object to check. **The access control checks are made against the cluster transmission queue, SYSTEM.CLUSTER.TRANSMIT.QUEUE.**

Even with multiple cluster transmission queues, local access control checks for remote cluster queues are made against SYSTEM.CLUSTER.TRANSMIT.QUEUE.

+ end excerpt

6) Now, let's try again as user "fulano":

```
fulano@chamonix1.fyre.ibm.com: /home/fulano
$ amqsput CQ1 PR4
Sample AMQSPUT0 start
target queue is CQ1
TEST-PUT-AS-FULANO-FROM-PR4
Sample AMQSPUT0 end
```

7) Login into the host that has PR3, which is the queue manager that has the physical queue CQ1.

Let's verify that the local queue CQ1 in PR3 has indeed received the message that was placed in PR4.

```
mqm@stmichel1.fyre.ibm.com: /home/mqm
$ amqsget CQ1 PR3
Sample AMQSGET0 start
message <TEST-PUT-AS-FULANO-FROM-PR4>
no more messages
Sample AMQSGET0 end
```

+ Unsuccessful **get** from a remote queue (working as designed)

This is just to illustrate again that it is not possible to get a message from a remote queue. **The get action is only successful on local queues.**

Let's try to get a message for CQ1 in PR4.
Remember that in PR4 this queue is a clustered queue and not a local queue.
Because the Get operations can only be done to local queues and not clustered queues, **the attempt will fail**:

```
mqm@chamonix1.fyre.ibm.com: /home/mqm
$ amqsget CQ1 PR4
Sample AMQSGET0 start
MQOPEN ended with reason code 2085
unable to open queue for input
Sample AMQSGET0 end

$ mqrc 2085
     2085  0x00000825  MQRC_UNKNOWN_OBJECT_NAME
```

The UNKNOWN_OBJECT_NAME refers to the "local" queue CQ1 in PR4.

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=clusters-clustering-tips
IBM MQ / 9.3
Clustering tips

+ begin excerpt

- You can only get a message from a local cluster queue, but you can put a message to any queue in a cluster. If you open a queue to use the MQGET command, the queue manager opens the local queue.

+ end excerpt

+++ end +++