

# Clustering best practices and useful commands

ColinPaice

Published on 28/11/2017 / Updated on 28/11/2017

MQ clustering is a mature and “well known” piece of technology. As it usually works without any problems, but it is often neglected and configuration problems creep in. The “wake up” call is when you get called out because things are not working as expected, and you find you have forgotten what to do!

This blog was written because one of my colleagues in customer support was given a “wake up call” in the middle of the night when a customer had problems.

Below is a list of best practices, and commands you may find useful when checking your environment (and it would be worth checking your environment just in case someone else <not you of course> has introduced some problems):

1. You need just two full repositories per cluster; no more – no less. (If you have three or more — see Why only two full repositories)
2. For each queue manager define one cluster receiver channel
3. Define just one cluster sender channel to one full repository. Once the partial repository joins the cluster, all other cluster sender channels will be created automatically.
4. Once the partial repository has connected to the full repository, the information about connecting to the other full repository is downloaded.
5. Do not manually define a cluster sender channel pointing to a partial repository.
6. Only issue a REFRESH CLUSTER if you really need to, and understand the impact it may have. For details [see here](#). We have seen situations where people issued the command, waited a bit, and reissued it again. The refresh cluster command sends messages to other queue managers, which may be slow to process these messages and slow to send the replies back. Although nothing may appear to be happening – the queue manager may just be waiting for replies. Issuing the command multiple times caused network congestion and it took much longer for the command to complete.
7. It is better to have a few clusters than many clusters
8. You can have overlapping clusters, but these should be kept to a minimum
9. For large clusters it is better for your full repositories to be just repositories, and process no application messages
10. Be careful when doing Disaster Recovery type activities. We have seen people backup queue managers, take them off-site, restore the queue manager and restart it. It connects to the full repository and says “I am over here now”. As the queue manager thinks it is in two places, clustering gets very confused.
11. Did I mention have just two full repositories, not to define a cluster sender channel to a partial repository, and to be careful with the REFRESH CLUSTER command?

## Useful commands

Note some commands can produce a lot of output!

**DIS QMGR REPOS REPOSNL** – shows if this queue manager is a repository.

**DIS Q(\*) CLUSTER(X)** shows all queues on this queue manager which are in cluster X

**DIS Q(MYQUEUE) CLUSTER(PAYROLL) TYPE(CLUSTER)** or

**DIS QCLUSTER(MYQUEUE) CLUSTER(PAYROLL)** shows information from the cluster repository; it shows the known queues – note a Partial Repository has only a subset of the information.

**DIS CLUSQMGR(\*) WHERE(DEFTYPE,EQ,CLUSSDRB) QMTYPE**

to show which channels have manual definition and automatic definition.

## What do I need to check?

1. AMQ9430 message on distributed MQ error log. This shows a manually defined cluster sender channel to a partial repository
2. CSQX430E message on z/OS MQ job log. This shows a manually defined cluster sender channel a partial repository

On each queue manager in a cluster issue

**DIS CLUSQMGR(\*) where(DEFTYPE,eq,CLUSSDRB) qmtype**

deftype qmtype

CLUSSDRB NORMAL bad definition – has Both Manual and Automatic. You need to resolve this

CLUSSDRB REPOS OK This is for a full repository to full repository connection

CLUSSDR REPOS OK. This is a connection to a full repository

CLUSSDRA NORMAL OK automatic definition to a normal (not full repository)

CLUSSDRA REPOS OK automatic definition to a full repository

## What do I do if I have an “extra channel”

If you have found some unwanted channels, you need to delete them.

### For a full repository

1. Check you have an alternate definition to the queue manager **DIS CLUSQMGR(name)**
2. Alter the channel and set the cluster to blank ( ' ' )
3. Stop the channel
4. Delete the channel

### For a partial repository

1. Check there is another connection to a repository queue manager **dis clusqmgr(\*) where(qmtype,eq,repos) qmtype deftype**
2. Check there is a cluster receiver channel **dis clusqmgr('colin')** where colin is your local queue manager
3. Alter the channel and set the cluster to blank ( ' ' )
4. Stop the channel
5. Delete the channel
6. Issue a **REFRESH CLUSTER REPOS(YES)**

by ColinPaice

### 3 comments on "Clustering best practices and useful commands"

1. **David Ware** March 22, 2019

I would write the list of problem channels as below, anything else would be ok:  
CLUSSDRB \* -> NORMAL BAD – no need for a manual sender targeting a queue manager that is not a full repository

CLUSSDRA REPOS -> REPOS BAD – Full repositories should have manually defined paths between themselves

CLUSSDR \* -> \* probably bad – the target has not responded

2. **Shiva** February 22, 2019

CLUSSDRB of QMTYPE(REPOS) Can be from Partial Repos to Full Repos QM also.

3. **VictorE** February 08, 2018

Hi Colin,

What considerations to make when we've mixed versions of MQ in Cluster  
for ex of a 4 node cluster ,

1 FullRepo & 1 partial at MQ 9

and 1 Full repo & Partial at MQ8