

Installing and uninstalling IBM MQ 9.4 in Linux RHEL

<https://www.ibm.com/support/pages/node/7159745>

Date last updated: 06-Jul-2024

Angel Rivera
IBM MQ Support

<https://www.ibm.com/products/mq/support>

Find all the support you need for IBM MQ

+++ Objective +++

The purpose of this technical document is to show all the steps to perform the following tasks for IBM MQ in a Linux x86 64-bit machine (Red Hat Enterprise Linux RHEL 9.4), using rpm:

- 1) Installing MQ 9.4.0.0 Long Term Support (LTS)
- 2) Uninstalling MQ 9.4.0.0

Notes:

- Even though this tutorial was written for MQ 9.4, the same principles apply to all versions, release types (LTS/CD) and fix packs.
- The steps apply also to the queue manager and/or client components.
- The same principles apply to the installation and uninstallation of the MQ 9.4 Continuous Delivery (CD).

This document describes a scenario in which there is ONLY one version of MQ installed in the machine (Installation1), in the default location of: /opt/mqm

This means that the utilities "crtmqpkg" and "crtmqfp" are NOT used. These utilities are ONLY needed when installing/maintaining the 2nd or 3rd or another installation location, when there are multiple MQ installations coexisting in the same machine.

A test queue manager will be created for a quick test for putting a message into a queue via local bindings.

Starting with MQ 9.3, the MQ Explorer is NO longer included with the package file that is downloaded from IBM Passport Advantage.

For details on how to download and install the MQ Explorer see:

<https://www.ibm.com/support/pages/node/6598657>

Installing Standalone MQ Explorer 9.4 in Windows and Linux

The chapters are:

Chapter 1: Summary of commands

Chapter 2: How to get the installation images

Chapter 3: Create userid "mqm" and group "mqm"

Chapter 4: Installation of MQ 9.4.0.0 (Installation1 in /opt/mqm)

Chapter 5: Using setmqenv to setup the MQ environment variables

Chapter 6: Creating a test queue manager, quick verification test

Chapter 7: Uninstallation of the base 9.4.0.0

+ Starting with MQ 9.4 the following components are no longer provided:

<https://www.ibm.com/support/pages/node/7107744>

IBM MQ Bridge to BlockChain MQSeriesBCBridge was removed in MQ 9.4, 9.3.1 CD, Fix Packs 9.3.0.15 LTS and 9.2.0.21 LTS, how to uninstall it

<https://www.ibm.com/support/pages/node/7151798>

IBM MQ Bridge to Salesforce MQSeriesSFBridge was removed in MQ 9.4 and 9.3.1 CD - how to uninstall it

+ MQ follows Best Practice of using 2 separate file systems: one for runtime, one for data

The MQ product for UNIX (AIX, Linux, etc) follows the best practice of using 2 different directory trees:

1: A directory tree for the executables and shared libraries.

In AIX is /usr/mqm

In Linux and others is /opt/mqm

2: A directory tree for the data for the queue managers and other configuration files.

For all UNIX is /var/mqm

When you backup /var/mqm, you are backing up ONLY the data, and not the executables.

These 2 directories trees are mutually exclusive, which means:

- When you create, maintain and use queue managers, only the data in /var/mqm is touched.

There is NO altering of any of the executable files under /opt/mqm.

- When you install, uninstall, apply fix packs or interim fixes, only the files under /opt/mqm are touched.

There is NO altering of any of the files under /var/mqm.

+ If you are installing MQ for the very first time, then notice that the following directory trees are NOT yet created.

They will be created during the 1st installation of MQ!

****ROOT** RHEL94941:**

```
# ls -dl /opt/mqm
```

```
ls: cannot access '/opt/mqm': No such file or directory
```

```
# ls -dl /var/mqm
```

```
ls: cannot access '/var/mqm': No such file or directory
```

+ Related tutorials:

<https://www.ibm.com/support/pages/node/6209227>

Summary of Linux commands to administrate MQ

<https://www.ibm.com/support/pages/node/6890737>

How to compute SHA256 and check signatures of downloaded IBM MQ files: sha256sum (Linux, AIX), CertUtil (Windows)

Topic 1: Signatures

Topic 2: The Secure Hash Algorithm SHA-256 is used.

<https://www.ibm.com/support/pages/node/1135522>

Configuring MQ to use a dedicated Listener, Channel and Queue in Linux (Tutorial)

```
+++++
+++ Chapter 1: Summary of commands
+++++
```

Login as root to create group "mqm" and user "mqm".

+ Here are some Best Practices for these ids:

a) In order to use MQ Multi-Instance Queue Managers or MQ Replicated Data Queue Managers (RDQM) **it is required that the numeric groupid for "mqm" and numeric userid for "mqm" in all the hosts MUST be IDENTICAL!** Thus, if in host-1 the userid for "mqm" is 1010 but in host-2 the userid for "mqm" is 1020 (for example), then these features will NOT work: in host-2 the userid for "mqm" MUST match the userid 1010 used in host-1.

b) Note about the home directory for mqm: do NOT use "/var/mqm".
It is a Best Practice that the home directory for the user "mqm" should be "/home/mqm" or any other directory other than "/var/mqm".
Even though it is technically possible to use /var/mqm as the home directory, it is NOT optimal to designate the home directory to "/var/mqm" because it could cause unnecessary restrictions to customizations tasks such as using passwordless login (sshd may conclude that the permissions of /var/mqm are too permissive, because members of the group "mqm" need access to the directory).

++ Creating groupid "mqm" (id 500) and userid "mqm" (id 501)

The following command will create a group called "mqm", with group-id (gid) of 500:

```
# groupadd -g 500 mqm
```

The following command will create a user called "mqm", with user-id (uid) of 501 and belonging to the group "mqm", having a home directory of "/home/mqm"

```
# useradd -u 501 -g mqm -s /bin/bash -d /home/mqm -m mqm
```

Assign the proper password:

```
# passwd mqm
```

After creating the group and user, then the "id" command shows:

```
# id mqm
uid=501(mqm) gid=500(mqm) groups=500(mqm)
```

+ Installing MQ 9.4.0.0

Login as root.

Go to the directory where you downloaded the installation image file, such as:

```
cd /downloads/mq9400
```

Unpack the file:

```
tar -zxvf 9.4.0.0-IBM-MQ-LinuxX64.tar.gz
```

Go to the "MQServer" directory:

```
cd ./MQServer
```

Accept the license:

```
./mqlicense.sh -accept
```

Install all the rpm files:

```
rpm -ivh MQ*.rpm
```

+ Stopping MQ queue managers (in order to apply fix pack).

The "ps -ef" command is useful to confirm that there are no running processes for MQ.

Login as an MQ administrator (user mqm).

```
$ endmqm -i QMgrName
```

```
$ ps -ef | grep -i mq
```

+ Uninstalling Base 9.4.0.0:

Login as root.

```
rpm -ev `rpm -qa | grep MQSeries | grep "9\.4\.\0\0-0"``
```

+ Uninstalling all MQ components (both Fix Pack and base)

Login as root.

```
rpm -ev `rpm -qa | grep MQSeries`
```

```
+++++
+++ Chapter 2: How to get the installation images
+++++
```

Please note that the General Availability (GA) is the same starting point for both release types:

MQ 9.4.0.0 LTS (Long Term Support) = MQ 9.4.0 CD (Continuous Delivery)

For MQ 9.4: From IBM Passport Advantage you can download

Visit IBM Passport Advantage:

https://www.ibm.com/software/passportadvantage/pao_customer.html

... search for Part Number for the single package that includes the queue manager, client and the rest of the components (except the MQ Explorer).

M0H54ML IBM MQ 9.4 Long Term Support release for Linux on x86 64-bit Multilingual

You will need around 2 GB of disk space to store the downloaded tar.gz file for the manufacturing refresh, and for the extracted rpm files.

Note for the MQ Explorer:

Starting with MQ 9.3: MQ Explorer is NO longer included with the download from Passport Advantage.

For details on how to download and install the MQ Explorer see:

<https://www.ibm.com/support/pages/node/6598657>

Installing Standalone MQ Explorer 9.4 in Windows and Linux

+ Downloaded image name for MQ 9.4.0.0:

9.4.0.0-IBM-MQ-LinuxX64.tar.gz

Local Directory:

/downloads/mq9400

++ The filesets for the RDQM feature are provided in a subdirectory

Note: This tutorial does not cover the installation of RDQM.

The rpm file for MQSeriesRDQM is stored in a deeper subdirectory:

```
# cd /downloads/mq9400/MQServer/Advanced/RDQM
```

```
# ls *.rpm
```

```
MQSeriesRDQM-9.4.0-0.x86_64.rpm
```

++ Note about using MQ in English - no need to install *Msg* rpm filesets

Messages in U.S. English are automatically installed with IBM MQ.

That is, the "message catalog for U.S. English" is always installed.

Therefore, by default, the informational, warning and error messages displayed by the MQ product will be in English.

There is no file named "MQSeriesMsg_en" because the U.S. English messages are always installed.

Message catalogs for all languages (except for English, because the English messages are included already) are included with the MQ installation image for:

- The base/rebased MQ product downloaded from IBM Passport Advantage
- The fix packs for MQ downloaded from IBM Fix Central

A best practice is to install only those message catalogs that your team is going to use and to delete those message catalogs that your team is not going to use.

Otherwise, the presence of message catalogs that are not used just take space in the disk and clutter the output of "rpm -qa" when querying for installed MQ packages.

+ Hint if you only work in English:

If you are not going to install these additional message catalogs, then you can delete the *Msg*.rpm files.

In that way, if you issue the simplest rpm command to install of available rpm filesets, these non-English message catalogs will not be installed.

For more information see:

<https://www.ibm.com/docs/en/ibm-mq/9.4?topic=linux-displaying-messages-in-your-national-language>

IBM MQ / 9.4

Displaying messages in your national language on Linux


```
+++++ Chapter 3: Create userid "mqm" and group "mqm" +++++
```

Prior to installing the MQ executable code, it is necessary to create the userid "mqm" and group "mqm". These ids will own the MQ executable and shared library files, as well as the data and recovery logs for the MQ queue managers.

The MQ code has hardcoded that any member of the group "mqm" is an MQ Administrator.

Note: If you do NOT create the group "mqm" and user "mqm" BEFORE you install MQ, then the MQ installer will create them!

One potential problem is that the MQ installer lets the OS to choose the actual gid number and uid number, thus the uid for user "mqm" in host-1 might be different than in host-2.

A Best Practice is to use the same uid and gid in all your hosts, and that means, that you need to define the group "mqm" and user "mqm" BEFORE you install MQ.

Login as root to create group "mqm" and user "mqm".

+ Here are some Best Practices for these ids:

a) In order to use MQ Multi-Instance Queue Managers or MQ Replicated Data Queue Managers (RDQM) **it is required that the numeric groupid for "mqm" and numeric userid for "mqm" in all the hosts MUST be IDENTICAL!** Thus, if in host-1 the userid for "mqm" is 1010 but in host-2 the userid for "mqm" is 1020 (for example), then these features will NOT work: in host-2 the userid for "mqm" MUST match the userid 1010 used in host-1.

b) Note about the home directory for mqm: do NOT use "/var/mqm". **It is a Best Practice that the home directory for the user "mqm" should be "/home/mqm" or any other directory other than "/var/mqm".**

Even though it is technically possible to use /var/mqm as the home directory, it is NOT optimal to designate the home directory to "/var/mqm" because it could cause unnecessary restrictions to customizations tasks such as using passwordless login (sshd may conclude that the permissions of /var/mqm are too permissive, because members of the group "mqm" need access to the directory).

The following command is useful to know the precise version of Linux.

```
**ROOT** RHEL94941: /root
# cat /etc/redhat-release
Red Hat Enterprise Linux release 9.4 (Plow)
```

The following command will create a group called "mqm", with group-id (gid) of 500:

```
# groupadd -g 500 mqm
```

The following command will create a user called "mqm", with user-id (uid) of 501 and belonging to the group "mqm":

```
# useradd -u 501 -g mqm -s /bin/bash -d /home/mqm -m mqm
```

Assign the proper password:

```
# passwd mqm
```

After creating the group and user, then the "id" command shows:

```
# id mqm
```

```
uid=501(mqm) gid=500(mqm) groups=500(mqm)
```

```
+++++
+++ Chapter 4: Installation of MQ 9.4.0.0 (Installation1 in /opt/mqm)
+++++
```

Login as root.

Go to the local directory where the file with the MQ 9.4.0.0 code is located (see Chapter 2)
cd /downloads/mq9400

Unpack the file:
tar -zxvf 9.4.0.0-IBM-MQ-LinuxX64.tar.gz

Go to the newly created "MQServer" directory:
cd /downloads/mq9400/MQServer

Lists the *.rpm files:

```
ls -l *.rpm
MQSeriesAMQP-9.4.0-0.x86_64.rpm
MQSeriesAMS-9.4.0-0.x86_64.rpm
MQSeriesClient-9.4.0-0.x86_64.rpm
MQSeriesFTAgent-9.4.0-0.x86_64.rpm
MQSeriesFTBase-9.4.0-0.x86_64.rpm
MQSeriesFTLogger-9.4.0-0.x86_64.rpm
MQSeriesFTService-9.4.0-0.x86_64.rpm
MQSeriesFTTools-9.4.0-0.x86_64.rpm
MQSeriesGSKit-9.4.0-0.x86_64.rpm
MQSeriesJava-9.4.0-0.x86_64.rpm
MQSeriesJRE-9.4.0-0.x86_64.rpm
MQSeriesMan-9.4.0-0.x86_64.rpm
MQSeriesMsg_cs-9.4.0-0.x86_64.rpm
MQSeriesMsg_de-9.4.0-0.x86_64.rpm
MQSeriesMsg_es-9.4.0-0.x86_64.rpm
MQSeriesMsg_fr-9.4.0-0.x86_64.rpm
MQSeriesMsg_hu-9.4.0-0.x86_64.rpm
MQSeriesMsg_it-9.4.0-0.x86_64.rpm
MQSeriesMsg_ja-9.4.0-0.x86_64.rpm
MQSeriesMsg_ko-9.4.0-0.x86_64.rpm
MQSeriesMsg_pl-9.4.0-0.x86_64.rpm
MQSeriesMsg_pt-9.4.0-0.x86_64.rpm
MQSeriesMsg_ru-9.4.0-0.x86_64.rpm
MQSeriesMsg_Zh_CN-9.4.0-0.x86_64.rpm
MQSeriesMsg_Zh_TW-9.4.0-0.x86_64.rpm
MQSeriesRuntime-9.4.0-0.x86_64.rpm
MQSeriesSamples-9.4.0-0.x86_64.rpm
MQSeriesSDK-9.4.0-0.x86_64.rpm
MQSeriesServer-9.4.0-0.x86_64.rpm
MQSeriesWeb-9.4.0-0.x86_64.rpm
MQSeriesXRService-9.4.0-0.x86_64.rpm
```

Accept the license:
./mqlicense.sh -accept
Agreement accepted: Proceed with install.

Note about deleting those "Message Catalog Files" that your team does not need: Depending on what is the desired objective for this installation, you may need to select the desired rpm files to install.

Also, you may want to delete those rpm files that you do not need, such as the Message Catalog files (*Msg*.rpm) which provide support for non-English error messages, such as in French, Spanish, etc.

This document assumes that you want to install all the rpm filesets (except RDQM)

Note:

During the running of rpm, you will see in the output, something about " Header V4

RSA/SHA256 Signature, key ID 07b22880: NOKEY "

Do NOT worry, it is not an error!

For more details, see "Topic 1" from the following technote:

<https://www.ibm.com/support/pages/node/6890737>

How to compute SHA256 and check signatures of downloaded IBM MQ files: sha256sum (Linux, AIX), CertUtil (Windows)

Topic 1: Signatures

Topic 2: The Secure Hash Algorithm SHA-256 is used.

Install all the rpm files (except for RDQM, whose rpm file resides in another folder)

rpm -ivh MQ*.rpm

```
warning: MQSeriesAMQP-9.4.0-0.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID 07b22880: NOKEY
```

```
Verifying... ##### [100%]
```

```
Preparing... ##### [100%]
```

```
Updating / installing...
```

```
 1:MQSeriesRuntime-9.4.0-0 ##### [ 3%]
```

```
Warning : package "MQSeriesRuntime" is signed but key is not installed on this system.
```

```
rpm verify shows "Header V4 RSA/SHA256 Signature, key ID 07b22880: NOKEY"
```

```
rpm warning message may have been issued at install time.
```

```
See topic "IBM MQ code signatures" in the IBM MQ documentation for more
```

```
information.
```

```
 2:MQSeriesJava-9.4.0-0 ##### [ 6%]
```

```
 3:MQSeriesJRE-9.4.0-0 ##### [ 10%]
```

```
 4:MQSeriesFTBase-9.4.0-0 ##### [ 13%]
```

```
 5:MQSeriesGSKit-9.4.0-0 ##### [ 16%]
```

```
 6:MQSeriesServer-9.4.0-0 ##### [ 19%]
```

```
Updated PAM configuration in /etc/pam.d/ibmmq
```

```
 7:MQSeriesFTAgent-9.4.0-0 ##### [ 23%]
```

```
 8:MQSeriesFTService-9.4.0-0 ##### [ 26%]
```

```
Licensed entitlement 'advanced' set for installation at '/opt/mqm'.
```

```
 9:MQSeriesAMQP-9.4.0-0 ##### [ 29%]
```

```
10:MQSeriesAMS-9.4.0-0 ##### [ 32%]
```

```
Licensed entitlement 'advanced' set for installation at '/opt/mqm'.
```

```
11:MQSeriesFTLogger-9.4.0-0 ##### [ 35%]
```

```
12:MQSeriesWeb-9.4.0-0 ##### [ 39%]
```

```
13:MQSeriesXRService-9.4.0-0 ##### [ 42%]
```

```
Licensed entitlement 'advanced' set for installation at '/opt/mqm'.
```

```
14:MQSeriesClient-9.4.0-0 ##### [ 45%]
```

```
15:MQSeriesFTTools-9.4.0-0 ##### [ 48%]
```

16:MQSeriesMan-9.4.0-0	#####	[52%]
17:MQSeriesMsg_cs-9.4.0-0	#####	[55%]
18:MQSeriesMsg_de-9.4.0-0	#####	[58%]
19:MQSeriesMsg_es-9.4.0-0	#####	[61%]
20:MQSeriesMsg_fr-9.4.0-0	#####	[65%]
21:MQSeriesMsg_hu-9.4.0-0	#####	[68%]
22:MQSeriesMsg_it-9.4.0-0	#####	[71%]
23:MQSeriesMsg_ja-9.4.0-0	#####	[74%]
24:MQSeriesMsg_ko-9.4.0-0	#####	[77%]
25:MQSeriesMsg_pl-9.4.0-0	#####	[81%]
26:MQSeriesMsg_pt-9.4.0-0	#####	[84%]
27:MQSeriesMsg_ru-9.4.0-0	#####	[87%]
28:MQSeriesMsg_Zh_CN-9.4.0-0	#####	[90%]
29:MQSeriesMsg_Zh_TW-9.4.0-0	#####	[94%]
30:MQSeriesSamples-9.4.0-0	#####	[97%]
31:MQSeriesSDK-9.4.0-0	#####	[100%]

++ What are the effects of installing MQ on a pristine machine?

The rest of this chapter explains some of the effects of installing MQ.

1) The rpm database will indicate that MQ is installed:

```
# rpm -qa | grep MQSeries | sort
MQSeriesAMQP-9.4.0-0.x86_64
MQSeriesAMS-9.4.0-0.x86_64
MQSeriesClient-9.4.0-0.x86_64
MQSeriesFTAgent-9.4.0-0.x86_64
MQSeriesFTBase-9.4.0-0.x86_64
MQSeriesFTLogger-9.4.0-0.x86_64
MQSeriesFTService-9.4.0-0.x86_64
MQSeriesFTTools-9.4.0-0.x86_64
MQSeriesGSKit-9.4.0-0.x86_64
MQSeriesJava-9.4.0-0.x86_64
MQSeriesJRE-9.4.0-0.x86_64
MQSeriesMan-9.4.0-0.x86_64
MQSeriesMsg_cs-9.4.0-0.x86_64
MQSeriesMsg_de-9.4.0-0.x86_64
MQSeriesMsg_es-9.4.0-0.x86_64
MQSeriesMsg_fr-9.4.0-0.x86_64
MQSeriesMsg_hu-9.4.0-0.x86_64
MQSeriesMsg_it-9.4.0-0.x86_64
MQSeriesMsg_ja-9.4.0-0.x86_64
MQSeriesMsg_ko-9.4.0-0.x86_64
MQSeriesMsg_pl-9.4.0-0.x86_64
MQSeriesMsg_pt-9.4.0-0.x86_64
MQSeriesMsg_ru-9.4.0-0.x86_64
MQSeriesMsg_Zh_CN-9.4.0-0.x86_64
MQSeriesMsg_Zh_TW-9.4.0-0.x86_64
MQSeriesRuntime-9.4.0-0.x86_64
MQSeriesSamples-9.4.0-0.x86_64
MQSeriesSDK-9.4.0-0.x86_64
MQSeriesServer-9.4.0-0.x86_64
MQSeriesWeb-9.4.0-0.x86_64
MQSeriesXRService-9.4.0-0.x86_64
```

2) There is a new directory which has a file with the information of the MQ installations:
/etc/opt/mqm

```
# ls -dl /etc/opt/mqm
drwxr-xr-x 2 root root 24 Jun 18 12:16 /etc/opt/mqm
```

and a new file:

```
# ls -dl /etc/opt/mqm/*
-rw-r--r-- 1 root root 153 Jun 18 12:16 /etc/opt/mqm/mqinst.ini
```

The contents of the file is:

```
# cat /etc/opt/mqm/mqinst.ini
Installation:
  Name=Installation1
  Description=
  Identifier=1
  FilePath=/opt/mqm
```

3) There is a new directory that has the MQ executable code and libraries:
/opt/mqm

```
# ls -dl /opt/mqm
dr-xr-xr-x 19 mqm mqm 4096 Jun  5 06:54 /opt/mqm
```

This is the list of the subdirectories:

```
# ls -dl /opt/mqm/*
dr-xr-xr-x 6 mqm mqm      57 Jun 18 12:16 /opt/mqm/amqp
dr-xr-xr-x 3 mqm mqm    8192 Jun 18 12:17 /opt/mqm/bin
dr-xr-xr-x 7 mqm mqm   4096 Jun 18 12:16 /opt/mqm/gskit8
dr-xr-xr-x 4 mqm mqm   4096 Jun 18 12:17 /opt/mqm/inc
-r--r--r-- 1 mqm mqm 16419 Jun  5 06:54 /opt/mqm/instinfo.tsk
dr-xr-xr-x 7 mqm mqm     65 Jun  5 06:54 /opt/mqm/java
dr-xr-xr-x 3 mqm mqm     40 Jun 18 12:16 /opt/mqm/lap
dr-xr-xr-x 4 mqm mqm   4096 Jun 18 12:17 /opt/mqm/lib
dr-xr-xr-x 3 mqm mqm   4096 Jun 18 12:17 /opt/mqm/lib64
dr-xr-xr-x 2 mqm mqm   4096 Jun 18 12:16 /opt/mqm/licenses
dr-xr-xr-x 4 mqm mqm     30 Jun 18 12:17 /opt/mqm/man
dr-xr-xr-x 8 bin bin     78 Jun  5 06:56 /opt/mqm/mqft
-r--r--r-- 1 mqm mqm     0 Jun  5 06:54 /opt/mqm/mqpatch.dat
dr-xr-xr-x 7 mqm mqm     68 Jun 18 12:17 /opt/mqm/mqxr
dr-xr-xr-x 16 mqm mqm    188 Jun 18 12:17 /opt/mqm/msg
dr-xr-xr-x 2 mqm mqm   4096 Jun 18 12:16 /opt/mqm/README
dr-xr-xr-x 14 mqm mqm   4096 Jun 18 12:17 /opt/mqm/samp
dr-xr-xr-x 2 mqm mqm   4096 Jun 18 12:17 /opt/mqm/swidtag
dr-xr-x--- 9 mqm mqm    152 Jun 18 12:17 /opt/mqm/web
```

4) There is a new directory structure for the MQ data:
/var/mqm

```
# ls -dl /var/mqm
```

```
drwxrwsr-x 14 mqm mqm 4096 Jun 18 12:16 /var/mqm
```

Here is the list of the subdirectories:

```
# ls -dl /var/mqm/*
```

```
drwxrwsr-x 2 mqm mqm 6 Jun 18 12:16 /var/mqm/config
drwxrwsr-x 3 mqm mqm 19 Jun 18 12:16 /var/mqm/conv
drwxrwsrwx 2 mqm mqm 66 Jun 18 12:16 /var/mqm/errors
drwxrwsr-x 3 mqm mqm 27 Jun 18 12:16 /var/mqm/exits
drwxrwsr-x 3 mqm mqm 27 Jun 18 12:16 /var/mqm/exits64
drwxrwsr-x 2 mqm mqm 6 Jun 18 12:16 /var/mqm/log
-rw-rw-r-- 1 mqm mqm 661 Jun 18 12:16 /var/mqm/mqclient.ini
drwxrwsr-x 5 mqm mqm 53 Jun 18 12:16 /var/mqm/mqft
-rw-rw-r-- 1 mqm mqm 2475 Jun 18 12:16 /var/mqm/mqs.ini
drwxrwsr-x 3 mqm mqm 21 Jun 18 12:16 /var/mqm/qmgrs
-rw-rw-r-- 1 mqm mqm 1941 Jun 18 12:16 /var/mqm/service.env
drwxrwsr-x 4 mqm mqm 32 Jun 18 12:16 /var/mqm/shared
drwxrwsr-x 3 mqm mqm 21 Jun 18 12:16 /var/mqm/sockets
drwxrwsrwx 2 mqm mqm 6 Jun 18 12:16 /var/mqm/trace
drwxrwsr-x 3 mqm mqm 27 Jun 18 12:16 /var/mqm/web
```

Note:

a) The queue manager data will be located under:

```
/var/mqm/qmgrs
```

b) The recovery logs (transaction logs) for the queue manager will be located under:

```
/var/mqm/log
```

5) There are NO symbolic links in /usr/bin for the MQ executable code.

You can specify later that this Installation should be the 'Primary' and then the symbolic links will be created.

```
# ls -l /usr/bin/*mq*
```

```
(no entries for MQ code)
```

6) Because there are no symbolic links under /usr/bin, the MQ commands are not under the default PATH, which means that if you try go run "dspmqver" to show the version of MQ, the command will not be found:

```
# dspmqver
```

```
-bash: dspmqver: command not found
```



```
+++++ Chapter 5: Using setmqenv to setup the MQ environment variables +++++
```

Note that by default, the MQ installer does NOT make the installation the Primary one.

There is a feature called "multi-version" installation, in which several different versions of MQ can coexist in the same host.

One important detail is that when MQ is installed in the host, there are NO symbolic links created in /usr/bin or /usr/lib.

Why? Because the installation software does not know if you want to have this new installation the Primary one or if you want to use as Primary another installation of MQ in the same host.

There are 2 approaches to set the MQ environment variables to use MQ commands.

In this tutorial we will proceed with the 1st approach (using setmqenv).

+ Option 1) No Primary installation (the default), using setmqenv

If you decide to have multiple installations of MQ in the same host and you do not want for none of them to be the Primary, then you will have to issue the following command every time that you login. The command can be added to your .bashrc or .profile file.

Specify the MQ environment. In Linux/AIX, you MUST "source" (inline) the command "setmqenv". There are 2 options:

a) You can explicitly use the command "source":

```
source /opt/mqm/bin/setmqenv -n Installation1
```

b) Or you can do the following, which is tricky for new users: dot, then space, then the rest.

```
. /opt/mqm/bin/setmqenv -n Installation1
| |
| +--> space
+--> dot
```

RECOMMENDATION to create a script:

As user "root" create a shell script that other local users could use to setup the MQ environment variables.

Login as root

Go to the directory /usr/local/bin

```
# cd /usr/local/bin
```

Create script:

```
# vi set-mq-inst1
```

Or you can use a name such as:

```
# vi set-mq-94
```

Notice that the installation name is: Installation1

```
===== begin script
# Name: set-mq-inst1
# Purpose: to setup the environment to run MQ
. /opt/mqm/bin/setmqenv -n Installation1
# Additional MQ directories for the PATH
export
PATH=$PATH:$MQ_INSTALLATION_PATH/java/bin:$MQ_INSTALLATION_PATH/samp/bin:$MQ_INSTALLATION_PATH/samp/jms/samples:
# Add local directory for running Java/JMS programs
export CLASSPATH=$CLASSPATH:.
# Display the full fix pack level
dspmqver -f 2
# end
===== end of script
```

Specify the script as executable:

```
# chmod 755 /usr/local/bin/set-mq-inst1
```

All local users can now execute this script from their command prompt or within the .profile or .bashrc

a) You can explicitly use the command "source":

```
source /usr/local/bin/set-mq-inst1
```

b) Or you can do the following, which is tricky for new users: dot, then space, then the rest.

```
. /usr/local/bin/set-mq-inst1
| |
| +--> space
+--> dot
```

+ Option 2) If there is only 1 installation of MQ in your system, the one with 9.4, then you can explicitly make this installation to be the "Primary". In that way, there will be symbolic links in /usr/bin that point to /opt/mqm.

As user "root" you will need to run the following command.

```
/opt/mqm/bin/setmqinst -i -p /opt/mqm
```

Some of the side-effects for making an installation a Primary are:

A) There are 2 lines appended to /etc/opt/mqm/mqinst.ini to indicate that this installation is a Primary one.

```
# cat /etc/opt/mqm/mqinst.ini
```

```
Installation:
  Name=Installation1
  Description=
  Identifier=1
  FilePath=/opt/mqm
```

```
Primary:
```

```
  Name=Installation1
```

B) There will be symbolic links in /usr/lib

```
# ls -l /usr/bin/*mq*
```

```
lrwxrwxrwx 1 root root 21 Jul 31 11:38 /usr/bin/addmqinf -> /opt/mqm/bin/addmqinf
...
```

+ After performing one of the above options, now "dspmqver" can be found in the PATH:

```
**ROOT** RHEL94941: /downloads/mq9400/MQServer
```

```
# . /opt/mqm/bin/setmqenv -n Installation1
```

```
# dspmqver
```

```
Name:          IBM MQ
Version:       9.4.0.0
Level:        p940-L240605.1
BuildType:    IKAP - (Production)
Platform:     IBM MQ for Linux (x86-64 platform)
Mode:        64-bit
O/S:         Linux 5.14.0-427.20.1.el9_4.x86_64
O/S Details:  Red Hat Enterprise Linux 9.4 (Plow)
InstName:     Installation1
InstDesc:
Primary:      No
InstPath:    /opt/mqm
DataPath:    /var/mqm
MaxCmdLevel: 940
LicenseType:  Production
ReleaseType:  Long Term Support (LTS) and Continuous Delivery (CD)
```

```
+++++ Chapter 6: Creating a test queue manager, quick verification test +++++
```

Login as user "mqm" (MQ Administrator)

Notice that there are NO environment variables for MQ.

```
$ set | grep -i mq
HISTFILE=/home/mqm/.bash_history
HOME=/home/mqm
...
```

It is necessary to run an MQ script in order to set up needed environment variables. You can add the following into your profile or bashrc.

Notice that you MUST include the dot, then the space and then the rest of text. This is called "to source" the script, that is, to make it "inline".

```
mqm@ RHEL94941: /home/mqm
$ source /opt/mqm/bin/setmqenv -n Installation1
```

Note: If root created the script set-mq-inst1 in the previous chapter, then you can issue:

```
$ source /usr/local/bin/set-mq-inst1
Version: 9.4.0.0
```

These are the environment variables set by setmqenv:

```
$ set | grep -i mq
MANPATH=/opt/mqm/man::/usr/man
MQ_DATA_PATH=/var/mqm
MQ_ENV_MODE=64
MQ_INSTALLATION_NAME=Installation1
MQ_INSTALLATION_PATH=/opt/mqm
MQ_JAVA_DATA_PATH=/var/mqm
MQ_JAVA_INSTALL_PATH=/opt/mqm/java
MQ_JAVA_LIB_PATH=/opt/mqm/java/lib64
MQ_JRE_PATH=/opt/mqm/java/jre64/jre
MQ_RETVAL=0
PATH=/opt/mqm/bin:/opt/ibm/java-x86_64-70/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/usr/local/bin:/home/mqm/bin:./sbin:/opt/mqm/java/bin:/opt/mqm/samp/bin:/opt/mqm/samp/jms/samples::/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin:/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/bin
PWD=/home/mqm
USER=mqm
```

The file /var/mqm/mqs.ini has the master index for the queue managers.

```
mqm@ RHEL94941: /home/mqm
```

```
$ cd /var/mqm
```

```
$ cat mqs.ini
```

```
## Module Name: mqs.ini                                ##
## Type       : MQ Machine-wide Configuration File    ##
## Function   : Define MQ resources for an entire machine ##
AllQueueManagers:
    DefaultPrefix=/var/mqm

LogDefaults:
    LogDefaultPath=/var/mqm/log
```

The data for the queue managers will be located at:

/var/mqm/qmgrs

This is the location for the recovery/transaction logs:

/var/mqm/log

The "general" error logs (not associated with a queue manager) and the FDC files that might be generated at runtime, will be located at:

/var/mqm/errors

++ Creating a queue manager for testing and initial customization to overcome typical security situations (not recommended for production)

<https://www.ibm.com/support/pages/node/1135522>

Configuring MQ to use a dedicated Listener, Channel and Queue in Linux (Tutorial)

- User "root" adds the proper user and group in both hosts
- MQ administrator (user "mqm") creates a queue manager with the basic/normal objects in host-1
- MQ administrator adds the dedicated objects.
 - Listener (such as MY.LISTENER in port 1420)
 - Server-Connection Channel (such as MY.CHANNEL)
 - Channel Authentication Record (CHLAUTH) for this server-connection channel that allows only the user "fulano" who is a member of the group "mqusers".
 - Queue (such as MY.Q)
 - Authority records for group "mqusers" to display, put, get, browse, etc. from the dedicated queue.
- User "fulano" from host-2 puts and gets messages using the dedicated objects

The following command specifies a Dead Letter Queue and port 1414

```
$ crtmqm -u SYSTEM.DEAD.LETTER.QUEUE -p 1414 TEST94
```

```
IBM MQ queue manager 'TEST94' created.  
Directory '/var/mqm/qmgrs/TEST94' created.  
The queue manager is associated with installation 'Installation1'.  
Creating or replacing default objects for queue manager 'TEST94'.  
Default objects statistics : 84 created. 0 replaced. 0 failed.  
Completing setup.  
Setup completed.
```

```
$ strmqm TEST94
```

```
IBM MQ queue manager 'TEST94' starting.  
The queue manager is associated with installation 'Installation1'.  
6 log records accessed on queue manager 'TEST94' during the log replay phase.  
Log replay for queue manager 'TEST94' complete.  
Transaction manager state recovered for queue manager 'TEST94'.  
Plain text communication is enabled.  
IBM MQ queue manager 'TEST94' started using V9.4.0.0.
```

```
$ runmqsc TEST94
```

```
## Define a channel to be used by a remote MQ Explorer  
define channel(SYSTEM.ADMIN.SVRCONN) chltype(SVRCONN)
```

```
## Define test queue:  
define qlocal (Q1)
```

```
end
```

Put a message using "bindings" (via local shared memory, not using TCP network) to the queue Q1:

\$ amqspout Q1 TEST94

```
Sample AMQSPUT0 start
target queue is Q1
this is a test
<press Enter without any further text to exit>
Sample AMQSPUT0 end
```

Now you can issue the sample that reads the message from the queue:

\$ amqsget Q1 TEST94

```
Sample AMQSGET0 start
message <this is a test>
no more messages
Sample AMQSGET0 end
```

Result: The use of bindings samples to put a message and to get that message, worked fine.

Setup the following environment variable to exercise the client TCP network connection:

```
$ export MQSERVER='SYSTEM.ADMIN.SVRCONN/TCP/localhost(1414)'
```

Notice that the sample executable name is different, it ends with "c" (for client network connection).

Notice that because we have not done further configuration to allow the remote access for MQ Administrators, the default behavior is to throw a security error, but at least we have done a successful test of the MQ Listener and the default security.

\$ amqspoutc Q1 TEST94

```
Sample AMQSPUT0 start
MQCONNX ended with reason code 2035
```

Use the following command to find out the name of the error:

\$ mqrc 2035

```
2035 0x000007f3 MQRC_NOT_AUTHORIZED
```

Let's find the error log of the queue manager:

```
$ cd /var/mqm/qmgrs/TEST94/errors
```

Look at the bottom of the file: AMQERR01.LOG

Notice that the 1st message is the important one.

The 2nd message is a side-effect and it can be ignored.

If you really want to relax the security rules, you can take a look at:

<https://www.ibm.com/support/pages/node/1135522>

Configuring MQ to use a dedicated Listener, Channel and Queue in Linux (Tutorial)

```
06/18/2024 12:50:09 PM - Process(10085.3) User(mqm) Program(amqrmppa)
Host(RHEL94941.fyre.ibm.com) Installation(Installation1)
VRMF(9.4.0.0) QMgr(TEST94)
Time(2024-06-18T19:50:09.144Z)
RemoteHost(127.0.0.1)
CommentInsert1(SYSTEM.ADMIN.SVRCONN)
CommentInsert2(127.0.0.1)
CommentInsert3(MCAUSER(mqm) CLNTUSER(mqm) ADDRESS(localhost))
```

AMQ9776E: Channel was blocked by userid

EXPLANATION:

The inbound channel 'SYSTEM.ADMIN.SVRCONN' was blocked from address '127.0.0.1' because the active values of the channel were mapped to a userid which should be blocked. The active values of the channel were 'MCAUSER(mqm) CLNTUSER(mqm) ADDRESS(localhost)'.

ACTION:

Contact the systems administrator, who should examine the channel authentication records to ensure that the correct settings have been configured. The ALTER QMGR CHLAUTH switch is used to control whether channel authentication records are used. The command DISPLAY CHLAUTH can be used to query the channel authentication records.

```
----- cmqxrmsa.c : 1557 -----
06/18/2024 12:50:09 PM - Process(10085.3) User(mqm) Program(amqrmppa)
Host(RHEL94941.fyre.ibm.com) Installation(Installation1)
VRMF(9.4.0.0) QMgr(TEST94)
Time(2024-06-18T19:50:09.144Z)
CommentInsert1(SYSTEM.ADMIN.SVRCONN)
CommentInsert2(10085)
CommentInsert3(127.0.0.1)
```

AMQ9999E: Channel 'SYSTEM.ADMIN.SVRCONN' to host '127.0.0.1' ended abnormally.

EXPLANATION:

The channel program running under process ID 10085 for channel 'SYSTEM.ADMIN.SVRCONN' ended abnormally. The host name is '127.0.0.1'; in some cases the host name cannot be determined and so is shown as '????'.

ACTION:

Look at previous error messages for the channel program in the error logs to determine the cause of the failure. Note that this message can be excluded completely or suppressed by tuning the "ExcludeMessage" or "SuppressMessage" attributes under the "QMErrorLog" stanza in qm.ini. Further information can be found in the System Administration Guide.

We can see that a new stanza was added to the master index file “mqc.ini” to register the new queue manager:

```
$ tail /var/mqm/mqc.ini
```

```
...
QueueManager:
  Name=TEST94
  Prefix=/var/mqm
  Directory=TEST94
  InstallationName=Installation1
```

The following command shows the runtime status of the queue manager:

```
$ dspmq
```

```
QMNAME (TEST94)                                STATUS (Running)
```

```
$ dspmq -o installation -o status
```

```
QMNAME (TEST94)                                STATUS (Running)
INSTNAME (Installation1) INSTPATH (/opt/mqm) INSTVER (9.4.0.0)
```

The following shows the individual processes for the queue manager:

```
$ ps -ef | grep TEST94
```

```
mqm          9953          1  0 12:47 ?                00:00:00 /opt/mqm/bin/amqzxxma0 -m TEST94
-u mqm
mqm          9963          9953  0 12:47 ?                00:00:00 /opt/mqm/bin/amqzfuma -m TEST94
mqm          9967          9953  0 12:47 ?                00:00:00 /opt/mqm/bin/amqzmgr0 -m TEST94
mqm          9970          9953  0 12:47 ?                00:00:00 /opt/mqm/bin/amqzmuc0 -m TEST94
mqm          9985          9953  0 12:47 ?                00:00:00 /opt/mqm/bin/amqzmur0 -m TEST94
mqm         10000          9953  0 12:47 ?                00:00:00 /opt/mqm/bin/amqzmuf0 -m TEST94
mqm         10003          9953  0 12:47 ?                00:00:00 /opt/mqm/bin/amqrrmfa -m TEST94
-t2332800 -s2592000 -p2592000 -g5184000 -c3600
mqm         10009          9967  0 12:47 ?                00:00:00 /opt/mqm/bin/amqpcsea TEST94
mqm         10010          9967  0 12:47 ?                00:00:00 /opt/mqm/bin/runmqchi -m TEST94
-q SYSTEM.CHANNEL.INITQ -r
mqm         10020          9953  0 12:47 ?                00:00:00 /opt/mqm/bin/amqzlaa0 -mTEST94
-fip0
mqm         10030          10000  0 12:47 ?                00:00:00 /opt/mqm/bin/amqfcpub -mTEST94
mqm         10041          9967  0 12:47 ?                00:00:00 /opt/mqm/bin/runmqslr -r -m
TEST94 -t TCP -p 1414
mqm         10057          10030  0 12:47 ?                00:00:00 /opt/mqm/bin/amqfcxba -m TEST94
mqm         10085          10041  0 12:50 ?                00:00:00 /opt/mqm/bin/amqrmppa -m TEST94
mqm         10096          9786  0 12:53 pts/0           00:00:00 grep --color=auto TEST94
```

Verifying that the port is being used. Notice the following line from the output of "ps". It means that the port number 1414 is by used by runmqsr, the MQ Listener for this queue manager.

```
mqm      10041    9967  0 12:47 ?          00:00:00 /opt/mqm/bin/runmqsr -r -m
TEST94 -t TCP -p 1414
```

The networking command "netstat" can be used to query the status of a port. The -p flag indicates which is the process that is using it. In some older Linux, this flag can only be used by user root.

\$ netstat -anp | grep 1414

```
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp6      0          0 :::1414          :::*              LISTEN         10041/runmqsr
```

Let's take a look at the version information held by the queue manager. Notice that the string under the attribute VERSION is 09040000 which represents: 09 04 00 00 => 09.04.00.00 => 9.4.0.0

\$ runmqsc TEST94

display qmgr version cmdlevel

AMQ8408I: Display Queue Manager details.

QMNAME (TEST94)

CMDLEVEL (940)

VERSION (09040000)

end

```
+++++
Chapter 7: Uninstallation of base 9.4.0.0
+++++
```

Shortcut: Uninstalling all MQ components (Fix Pack and base)

Login as root.

```
rpm -ev `rpm -qa | grep MQSeries`
```

Before trying to uninstall the MQ code, you will need to stop all the MQ queue managers and MQ clients in the host.

There should NOT be any type of MQ activity.

+ Login as the MQ administrator “mqm” and terminate the queue manager:

```
$ endmqm -i TEST94
```

```
IBM MQ queue manager 'TEST94' ending.
```

```
IBM MQ queue manager 'TEST94' ended.
```

```
$ ps -ef | grep -i mq
```

```
mqm      9434  9432  0 11:44 pts/0    00:00:00 -bash
```

```
mqm      10940 9434  0 12:07 pts/0    00:00:00 ps -ef
```

+ Login as root

Issue the following command to find out the filesets for MQ:

```
# rpm -qa | grep MQSeries | sort
```

```
MQSeriesAMQP-9.4.0-0.x86_64
```

```
MQSeriesAMS-9.4.0-0.x86_64
```

```
...
```

```
MQSeriesRuntime-9.4.0-0.x86_64
```

```
...
```

Proceed to uninstall MQ 9.4.0.0.

Let's explain a little bit the following command. There are 2 phases:

Phase-1: Notice that the command exploits the back-quote or back-tick to issue a command that identifies those file sets that are associated with MQ 9.4.0.0:

```
rpm -qa | grep MQSeries | grep "9\.4\.0\-"
```

Phase-2: Uninstall the list of rpm obtained in Phase-1:

```
rpm -ev list-from-Phase-1
```

```
# rpm -ev `rpm -qa | grep MQSeries | grep "9\.4\.0-0"`
Preparing packages...
MQSeriesFTService-9.4.0-0.x86_64
MQSeriesFTLogger-9.4.0-0.x86_64
MQSeriesFTAgent-9.4.0-0.x86_64
MQSeriesFTTools-9.4.0-0.x86_64
MQSeriesXRService-9.4.0-0.x86_64
MQSeriesWeb-9.4.0-0.x86_64
MQSeriesAMQP-9.4.0-0.x86_64
MQSeriesFTBase-9.4.0-0.x86_64
MQSeriesClient-9.4.0-0.x86_64
MQSeriesJRE-9.4.0-0.x86_64
MQSeriesJava-9.4.0-0.x86_64
MQSeriesSDK-9.4.0-0.x86_64
MQSeriesSamples-9.4.0-0.x86_64
MQSeriesMsg_Zh_TW-9.4.0-0.x86_64
MQSeriesMsg_Zh_CN-9.4.0-0.x86_64
MQSeriesMsg_ru-9.4.0-0.x86_64
MQSeriesMsg_pt-9.4.0-0.x86_64
MQSeriesMsg_pl-9.4.0-0.x86_64
MQSeriesMsg_ko-9.4.0-0.x86_64
MQSeriesMsg_ja-9.4.0-0.x86_64
MQSeriesMsg_it-9.4.0-0.x86_64
MQSeriesMsg_hu-9.4.0-0.x86_64
MQSeriesMsg_fr-9.4.0-0.x86_64
MQSeriesMsg_es-9.4.0-0.x86_64
MQSeriesMsg_de-9.4.0-0.x86_64
MQSeriesMsg_cs-9.4.0-0.x86_64
MQSeriesMan-9.4.0-0.x86_64
MQSeriesAMS-9.4.0-0.x86_64
MQSeriesServer-9.4.0-0.x86_64
MQSeriesGSKit-9.4.0-0.x86_64
MQSeriesRuntime-9.4.0-0.x86_64
```

Issue now the following command, and you will see that there are no more filesets for MQ

```
# rpm -qa | grep MQSeries
```

The absence of MQ filesets means that MQ is no longer installed in the host (location /opt/mqm)

```
# ls -dl /opt/mqm
```

```
ls: cannot access '/opt/mqm': No such file or directory
```

The MQ data under /var/mqm is left intact (it was not touched by the uninstallation).

```
# ls -dl /var/mqm
```

```
drwxrwsr-x 14 mqm mqm 4096 Jun 18 12:46 /var/mqm
```

Let's verify the contents of the mqinst.ini file, which keeps track of the installations of MQ in the host.

Notice that the entry Installation1 was NOT deleted.

This is working as designed.

If you want to delete that entry, then it is necessary to use the MQ command "dltmqinst".

```
# cat /etc/opt/mqm/mqinst.ini
```

```
Installation:
```

```
  Name=Installation1
```

```
  Description=
```

```
  Identifier=1
```

```
  FilePath=/opt/mqm
```

```
+++ end +++
```