

Installing MQ 9.1.0.0 LTS, applying Fix Pack MQ 9.1.0.1 and uninstalling MQ in Linux RHEL

<https://www.ibm.com/support/pages/node/725975>

Date last updated: 21-Aug-2023

Angel Rivera
IBM MQ Support

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

Find all the support you need for IBM MQ

+++ Update 21-Aug-2023

A newer tutorial dedicated to MQ 9.3 was published:

<https://www.ibm.com/support/pages/node/6988681>

Installing MQ 9.3, applying Fix Pack 9.3.0.5, uninstalling in Linux RHEL

+++ Update 30-Jun-2022 for MQ 9.3:

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 9.3 in Windows and Linux

+++ 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 7.4), using rpm:

- 1) Installing MQ 9.1.0.0 Long Term Support (LTS)
- 2) Applying Fix Pack 9.1.0.1
- 3) Uninstalling the Fix Pack 9.1.0.1
- 4) Uninstalling MQ 9.1.0.0

Notes:

- Even though this tutorial was written for MQ 9.1, 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 MQ Fix Packs mentioned here were the latest at the time the tutorial was written.
- The overall recommendation is to apply the most recent Fix Packs.

- The same principles apply to the installation and uninstallation of the MQ 9.1.x Continuous Delivery (CD). Please keep in mind that the MQ 9.1.x CD releases do NOT have Fix Packs, which means that if you have installed MQ 9.1.1 CD and then you want to replace it with MQ 9.1.2 CD, then you must:

- a) Uninstall MQ 9.1.1 and then
- b) Install MQ 9.1.2.

This document describes a scenario in which there is ONLY one version of MQ installed in the machine, 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 installation coexisting in the same machine.

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

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 9.1.0.0 (Installation1 in /opt/mqm)

Chapter 5: Making this installation the Primary one and using setmqenv

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

Chapter 7: Installation of the Fix Pack 9.1.0.1

Chapter 8: Uninstallation of the Fix Pack 9.1.0.1

Chapter 9: Uninstallation of the base 9.1.0.0

+ Related tutorials:

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

Summary of Linux commands to administrate MQ

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

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

Chapter 1: User "root" adds the proper user and group in both hosts

Chapter 2: MQ administrator (user "mqm") creates a queue manager with the basic/normal objects in host-1

Chapter 3: 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.

Chapter 4: User "fulano" from host-2 puts and gets messages using the dedicated objects

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

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

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

+ The following command will create a group called "mqm", with group-id (gid) of 501:

```
# groupadd -g 501 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=501(mqm) groups=501(mqm)
```

+ Installing MQ 9.1.0.0

Login as root.

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

```
cd /downloads/mq91
```

Unpack the file:

```
tar -zxvf IBM_MQ_9.1_LINUX_X86-64.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
```

+ Installing Fix Pack MQ 9.1.0.1

Login as root.

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

```
cd /downloads/mq9101
```

Unpack the file:

```
tar -zxvf 9.1.0-IBM-MQ-LinuxX64-FP0001.tar.gz
```

Install all the rpm files:

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

+ Uninstalling Fix Pack 9.1.0.1

Login as root.

```
rpm -ev `rpm -qa | grep MQSeries | grep "9\.\1\.\0\-\1"`
```

+ Uninstalling Base 9.1.0.0:

Login as root.

```
rpm -ev `rpm -qa | grep MQSeries | grep "9\.\1\.\0\-\0"`
```

+ Uninstalling all MQ components (both Fix Pack 9.1.0.1 and base 9.1.0.0)

Login as root.

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

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

++ Update 30-Jun-2022:

For MQ 9.3: Visit IBM Passport Advantage:

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

... search for Part Number:

M053XML IBM MQ 9.3 Long Term Support release for Linux on x86 64-bit elmage

For MQ 9.2: From IBM Passport Advantage you can download the manufacturing refresh: MQ 9.2.0.5 LTS that includes 9.2.0.0 + 9.2.0.1 + ... + 9.2.0.5

Visit IBM Passport Advantage:

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

... search for Part Number:

M04V3ML Description: IBM MQ 9.2.0.5 Long Term Support for Linux on x86 64-bit elmage

For MQ 9.1 in Linux, we would recommend for your team to upgrade to the following manufacturing refresh:

9.1.0.11 LTS, which includes 9.1.0.0 + 9.1.0.1 + ... + 9.1.0.11

Visit IBM Passport Advantage:

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

... and search for Part Number:

M05BCEN IBM MQ 9.1.0.11 for Linux on x86 64-bit elmage

++ MQ 9.1.0.0 (full install) - from IBM Passport Advantage

<https://www-01.ibm.com/support/docview.wss?uid=swg24044776>

Downloading IBM MQ Version 9.1

(From the IBM Passport Advantage web site)

It is recommended that you search for Part Number, which will provide a more focused list.

Release: 9.1.0.0

Part Number: CNRE3ML

Description: IBM MQ V9.1 for Linux on x86 64-bit elmage

Downloaded image name for rebased MQ 9.1.0.0:

IBM_MQ_9.1_LINUX_X86-64.tar.gz

Local Directory:

/downloads/mq91

++ Fix Pack 9.1.0.1 - from IBM Fix Central

The MQ Fix Packs are available from IBM Fix Central:

<http://www-01.ibm.com/support/docview.wss?uid=swg27006037>

Recommended Fixes for WebSphere MQ

Follow the prompts for MQ 9.1:

IBM MQ Version 9.1 LTS	Fix Pack 9.1.0.1
------------------------	------------------

You will be taken to another web page where you can select the desired version:

IBM WebSphere MQ Version 9.1 (latest 9.1.0.1)

Downloaded image:

9.1.0-IBM-MQ-LinuxX64-FP0001.tar.gz

Local Directory:

/downloads/mq9101

++ Note about new feature: RDQM

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

```
# cd /downloads/mq9100
# ls MQServer/Advanced/RDQM
installRDQMsupport MQSeriesRDQM-9.1.0-0.x86_64.rpm PreReqs repackage
uninstallRDQMsupport
```

++ 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.

Message catalogs for all languages 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.

+ References:

.
https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.1.0/com.ibm.mq.ins.doc/q008953_.htm

IBM MQ 9.1.x / IBM MQ / Installing and uninstalling / Installing and uninstalling IBM MQ on Linux /

Displaying messages in your national language on Linux

+ begin excerpt

To display messages from a different national language message catalog, you must install the appropriate catalog and set the LANG environment variable.

.
About this task

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

Message catalogs for all languages are installed in MQ_INSTALLATION_PATH/msg/language identifier, where language identifier is one of the identifiers in Table 1.

+ end excerpt

For example, in a test system where NONE of the message catalogs were installed, the following is the contents of the directory mentioned above (notice the subdirectory en_US for the locale English in the US).

```
$ ls -F /opt/mqm/msg
en_US/
```

In addition:

https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.1.0/com.ibm.mq.ins.doc/q008350_.htm#q008350_mcat

IBM MQ 9.1.x / IBM MQ / Installing and uninstalling / Installing and uninstalling IBM MQ on Linux / Installing IBM MQ on Linux using rpm / IBM MQ rpm components for Linux systems

+ begin excerpt

Table 2. IBM MQ message catalogs for Linux systems

Message catalog language	RPM package name
Brazilian Portuguese	MQSeriesMsg_pt
Czech	MQSeriesMsg_cs
French	MQSeriesMsg_fr
German	MQSeriesMsg_de
Hungarian	MQSeriesMsg_hu

Italian	MQSeriesMsg_it
Japanese	MQSeriesMsg_ja
Korean	MQSeriesMsg_ko
Polish	MQSeriesMsg_pl
Russian	MQSeriesMsg_ru
Spanish	MQSeriesMsg_es
Simplified Chinese	MQSeriesMsg_Zh_CN
Traditional Chinese	MQSeriesMsg_Zh_TW
U.S. English	not applicable

+ end excerpt

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


```
+++++
+++ 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.

Login as root.

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

```
+++ROOT+++ noon1.fyre.ibm.com: /downloads
# lsb_release -dr
Description: Red Hat Enterprise Linux Server release 7.4 (Maipo)
Release: 7.4
```

The following command will create a group called "mqm", with group-id (gid) of 501:

```
# groupadd -g 501 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=501(mqm) groups=501(mqm)
```

```
+++++
+++ Chapter 4: Installation of MQ 9.1.0.0 (Installation1 in /opt/mqm)
+++++
```

Login as root.

Go to the local directory where the file with the MQ 9.1.0.0 code is located (see Chapter 2)
cd /downloads/mq91

Unpack the file:
tar -zxvf IBM_MQ_9.1_LINUX_X86-64.tar.gz

Go to the newly create "MQServer" directory:
cd /downloads/mq91/MQServer

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

Note:
 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).
 This document assumes that you want to install all the rpm filesets.

Install all the rpm files:
rpm -ivh MQ*.rpm

```
Preparing...                               ##### [100%]
Updating / installing...
 1:MQSeriesRuntime-9.1.0-0                 ##### [ 3%]
 2:MQSeriesJRE-9.1.0-0                    ##### [ 6%]
 3:MQSeriesJava-9.1.0-0                   ##### [ 9%]
 4:MQSeriesServer-9.1.0-0                 ##### [12%]
 5:MQSeriesFTBase-9.1.0-0                 ##### [15%]
 6:MQSeriesFTAgent-9.1.0-0                ##### [18%]
 7:MQSeriesFTService-9.1.0-0              ##### [21%]
 8:MQSeriesFTLogger-9.1.0-0               ##### [24%]
 9:MQSeriesFTTools-9.1.0-0                ##### [26%]
10:MQSeriesAMQP-9.1.0-0                   ##### [29%]
11:MQSeriesAMS-9.1.0-0                    ##### [32%]
12:MQSeriesWeb-9.1.0-0                     ##### [35%]
13:MQSeriesXRService-9.1.0-0              ##### [38%]
14:MQSeriesBCBridge-9.1.0-0               ##### [41%]
15:MQSeriesSFBridge-9.1.0-0               ##### [44%]
16:MQSeriesExplorer-9.1.0-0               ##### [47%]
```

17:MQSeriesClient-9.1.0-0	##### [50%]
18:MQSeriesGSKit-9.1.0-0	##### [53%]
19:MQSeriesMan-9.1.0-0	##### [56%]
20:MQSeriesMsg_cs-9.1.0-0	##### [59%]
21:MQSeriesMsg_de-9.1.0-0	##### [62%]
22:MQSeriesMsg_es-9.1.0-0	##### [65%]
23:MQSeriesMsg_fr-9.1.0-0	##### [68%]
24:MQSeriesMsg_hu-9.1.0-0	##### [71%]
25:MQSeriesMsg_it-9.1.0-0	##### [74%]
26:MQSeriesMsg_ja-9.1.0-0	##### [76%]
27:MQSeriesMsg_ko-9.1.0-0	##### [79%]
28:MQSeriesMsg_pl-9.1.0-0	##### [82%]
29:MQSeriesMsg_pt-9.1.0-0	##### [85%]
30:MQSeriesMsg_ru-9.1.0-0	##### [88%]
31:MQSeriesMsg_Zh_CN-9.1.0-0	##### [91%]
32:MQSeriesMsg_Zh_TW-9.1.0-0	##### [94%]
33:MQSeriesSamples-9.1.0-0	##### [97%]
34:MQSeriesSDK-9.1.0-0	##### [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
MQSeriesFTAgent-9.1.0-0.x86_64
MQSeriesBCBridge-9.1.0-0.x86_64
MQSeriesMsg_es-9.1.0-0.x86_64
MQSeriesMsg_ru-9.1.0-0.x86_64
MQSeriesRuntime-9.1.0-0.x86_64
MQSeriesFTTools-9.1.0-0.x86_64
MQSeriesClient-9.1.0-0.x86_64
MQSeriesMsg_it-9.1.0-0.x86_64
MQSeriesSamples-9.1.0-0.x86_64
MQSeriesFTLogger-9.1.0-0.x86_64
MQSeriesExplorer-9.1.0-0.x86_64
MQSeriesMsg_hu-9.1.0-0.x86_64
MQSeriesMsg_Zh_TW-9.1.0-0.x86_64
MQSeriesJava-9.1.0-0.x86_64
MQSeriesFTService-9.1.0-0.x86_64
MQSeriesAMS-9.1.0-0.x86_64
MQSeriesSFBridge-9.1.0-0.x86_64
MQSeriesMan-9.1.0-0.x86_64
MQSeriesMsg_fr-9.1.0-0.x86_64
MQSeriesMsg_ko-9.1.0-0.x86_64
```

```
MQSeriesMsg_Zh_CN-9.1.0-0.x86_64
MQSeriesJRE-9.1.0-0.x86_64
MQSeriesAMQP-9.1.0-0.x86_64
MQSeriesGSKit-9.1.0-0.x86_64
MQSeriesMsg_ja-9.1.0-0.x86_64
MQSeriesSDK-9.1.0-0.x86_64
MQSeriesFTBase-9.1.0-0.x86_64
MQSeriesXRService-9.1.0-0.x86_64
MQSeriesMsg_de-9.1.0-0.x86_64
MQSeriesMsg_pt-9.1.0-0.x86_64
MQSeriesServer-9.1.0-0.x86_64
MQSeriesWeb-9.1.0-0.x86_64
MQSeriesMsg_cs-9.1.0-0.x86_64
MQSeriesMsg_pl-9.1.0-0.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 Aug  1 07:22 /etc/opt/mqm
```

and a new file:
/etc/opt/mqm/mqinst.ini

```
# ls -dl /etc/opt/mqm/*
-rw-r--r-- 1 root root 153 Aug  1 07:22 /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 262 Aug  1 07:22 /opt/mqm
```

This is the list of the subdirectories:

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

```
dr-xr-xr-x 6 mqm mqm 57 Aug 1 07:22 /opt/mqm/amqp
dr-xr-xr-x 3 mqm mqm 8192 Aug 1 07:23 /opt/mqm/bin
dr-xr-xr-x 16 mqm mqm 188 Aug 1 07:22 /opt/mqm/doc
dr-xr-xr-x 7 mqm mqm 245 Aug 1 07:23 /opt/mqm/gskit8
dr-xr-xr-x 4 mqm mqm 4096 Aug 1 07:23 /opt/mqm/inc
-r--r--r-- 1 mqm mqm 15999 Jul 5 09:55 /opt/mqm/instinfo.tsk
dr-xr-xr-x 7 mqm mqm 65 Jul 5 09:56 /opt/mqm/java
dr-xr-xr-x 4 mqm mqm 4096 Aug 1 07:23 /opt/mqm/lib
dr-xr-xr-x 3 mqm mqm 4096 Aug 1 07:23 /opt/mqm/lib64
dr-xr-xr-x 2 mqm mqm 4096 Aug 1 07:22 /opt/mqm/licenses
dr-xr-xr-x 4 mqm mqm 30 Aug 1 07:23 /opt/mqm/man
dr-xr-xr-x 5 mqm mqm 44 Aug 1 07:23 /opt/mqm/mqbc
dr-xr-xr-x 3 mqm mqm 46 Aug 1 07:23 /opt/mqm/mqexplorer
dr-xr-xr-x 8 bin bin 78 Jul 5 10:00 /opt/mqm/mqft
-r--r--r-- 1 mqm mqm 0 Jul 5 09:55 /opt/mqm/mqpatch.dat
dr-xr-xr-x 5 mqm mqm 44 Aug 1 07:23 /opt/mqm/mqsf
dr-xr-xr-x 7 mqm mqm 68 Aug 1 07:23 /opt/mqm/mqxr
dr-xr-xr-x 16 mqm mqm 188 Aug 1 07:23 /opt/mqm/msg
dr-xr-xr-x 2 mqm mqm 335 Aug 1 07:22 /opt/mqm/README
dr-xr-xr-x 12 mqm mqm 4096 Aug 1 07:23 /opt/mqm/samp
dr-xr-xr-x 2 mqm mqm 4096 Aug 1 07:23 /opt/mqm/swidtag
dr-xr-x--- 9 mqm mqm 171 Aug 1 07:23 /opt/mqm/web
```

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

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

```
drwxrwsr-x 13 mqm mqm 4096 Jul 31 11:30 /var/mqm
```

Here is the list of the subdirectories:

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

```
drwxrwsr-x 2 mqm mqm 6 Aug 1 07:22 /var/mqm/config
drwxrwsr-x 3 mqm mqm 19 Aug 1 07:22 /var/mqm/conv
drwxrwsrwx 2 mqm mqm 87 Aug 1 07:22 /var/mqm/errors
drwxrwsr-x 3 mqm mqm 27 Aug 1 07:22 /var/mqm/exits
drwxrwsr-x 3 mqm mqm 27 Aug 1 07:22 /var/mqm/exits64
drwxrwsr-x 3 mqm mqm 19 Aug 1 07:22 /var/mqm/log
-rw-rw-r-- 1 mqm mqm 637 Aug 1 07:22 /var/mqm/mqclient.ini
drwxrwsr-x 5 mqm mqm 53 Aug 1 07:22 /var/mqm/mqft
-rw-rw-r-- 1 mqm mqm 2704 Aug 1 07:22 /var/mqm/mqs.ini
drwxrwsr-x 5 mqm mqm 47 Aug 1 07:22 /var/mqm/qmgrs
-rw-rw-r-- 1 mqm mqm 1941 Aug 1 07:22 /var/mqm/service.env
drwxrwsr-x 4 mqm mqm 32 Aug 1 07:22 /var/mqm/shared
```

```
drwxrwsr-x 5 mqm mqm 47 Aug 1 07:22/var/mqm/sockets
drwxrwsrwx 2 mqm mqm 24 Aug 1 07:22/var/mqm/trace
drwxrwsr-x 3 mqm mqm 27 Aug 1 07:22/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:

```
# 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 to run "dspmqver" to show the version of MQ, the command will not be found:

```
# dspmqver
```

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

```
+++++
+++ Chapter 5: Making this installation the Primary one
+++++
```

Because we will have ONLY one installation of MQ in this machine, it is more efficient to make this installation the Primary one.

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

Starting with MQ 7.1, there is a new 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 7.1 or 7.5 or 8.0 or 9.0 or 9.1 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 take in order to properly MQ commands:

1) 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 .profile file.

Notice that you have to use: dot + space + setmqenv command

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

2) If there is only 1 installation of MQ in your system, the one with 9.0, 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
```

In this tutorial we will proceed with the 2nd approach (marking the installation as Primary).

Note: user root is the one authorized to run it.

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

```
143 of 143 tasks have been completed successfully.
```

```
'Installation1' (/opt/mqm) set as the primary installation.
```

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 are now 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
```

```
...
```

3) Now "dspmqver" can be found in the PATH:

```
# dspmqver
```

```
Name: IBM MQ
```

```
Version: 9.1.0.0
```

```
Level: p910-L180705
```

```
BuildType: IKAP - (Production)
```

```
Platform: IBM MQ for Linux (x86-64 platform)
```

```
Mode: 64-bit
```

```
O/S: Linux 3.10.0-862.2.3.el7.x86_64
```

```
InstName: Installation1
```

```
InstDesc:
```

```
Primary: Yes
```

```
InstPath: /opt/mqm
```

```
DataPath: /var/mqm
```

```
MaxCmdLevel: 910
```

```
LicenseType: Production
```


+ RECOMMENDATION:

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-91

Notice that the installation name is: Installation1

```
===== begin script
```

```
# Name: set-mq-91
```

```
# Purpose: to setup the environment to run MQ 9.1
```

```
. /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_IN
```

```
STALLATION_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-91

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

They will need to invoke the script with the leading dot!

```
.    /usr/local/bin/set-mq-91
|   |
|   +--- blank
+--- dot
```

```

+++++ 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
INPUTRC=/home/mqm/.inputrc
LOGNAME=mqm
MAIL=/var/spool/mail/mqm
PATH=/opt/ibm/java-x86_64-70/bin:/home/mqm/bin:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/X11R6/bin:/usr/games:/usr/lib/mit/bin:/usr/lib/mit/sbin:/usr/local/bin:/home/mqm/bin:./sbin:/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin:/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/bin
PWD=/home/mqm
USER=mqm

```

Even though the directory for Installation1 /opt/mqm is the Primary, still 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@noon1.fyre.ibm.com: /home/mqm
$ . /opt/mqm/bin/setmqenv -n Installation1

```

Note: If root created the script set-mq-90 in the previous chapter, then you can issue:

```

mqm@noon1.fyre.ibm.com: /home/mqm
$ . /usr/local/bin/set-mq-91
Version: 9.1.0.0

```

These are the environment variables set by setmqenv:

```

$ set | grep -i mq
CLASSPATH=/opt/mqm/java/lib/com.ibm.mq.jar:/opt/mqm/java/lib/com.ibm.mqjms.jar:/opt/mqm/java/lib/com.ibm.mq.allclient.jar:/opt/mqm/samp/wmqjava/samples:/opt/mqm/samp/jms/samples:
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=/usr/local/bin:/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:/home/mqm/.local/bin:/home/mqm/bin
```

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

```
mqm@noon1.fyre.ibm.com: /home/mqm
$ cd /var/mqm
$ cat mqs.ini
## Module Name: mqs.ini                *#
## Type      : WebSphere MQ Machine-wide Configuration File      *#
## Function   : Define WebSphere MQ resources for an entire machine *#
AllQueueManagers:
  DefaultPrefix=/var/mqm

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

Notice the current contents for the directory where the data for the queue managers is located:

```
$ cd /var/mqm/qmgrs
$ ls -al
drwxrwsr-x 13 mqm mqm 4096 Aug  1 07:22 @SYSTEM

$ cd /var/mqm/log
$ ls -al
```

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

The following post from the MQ forum in dwAnswers is very helpful:

<https://developer.ibm.com/answers/questions/229920/what-are-the-minimum-steps-to-create-a-new-test-qu.html>

What are the minimum steps to create a new test queue manager, allowing remote access by MQ Administrators and non-administrators?

Caveat: You must define the queue DLQ later. The above parameter does NOT actually create the queue, it just tells the queue manager which is the name of the dlq.

Note: Many MQ Explorer users hide the SYSTEM* queues and thus, if you use as the dlq the SYSTEM.DEAD.LETTER.QUEUE, then it will be hidden and you may not notice if there are messages in the dlq.

Login as the MQ Administrator.

```
$ crtmqm -u DLQ TEST_91
```

```
IBM MQ queue manager created.
```

```
Directory '/var/mqm/qmgrs/TEST_91' created.
```

```
The queue manager is associated with installation 'Installation1'.
```

```
Creating or replacing default objects for queue manager 'TEST_91'.
```

```
Default objects statistics : 84 created. 0 replaced. 0 failed.
```

```
Completing setup.
```

```
Setup completed.
```

```
$ strmqm TEST_91
```

```
IBM MQ queue manager 'TEST_91' starting.
```

```
The queue manager is associated with installation 'Installation1'.
```

```
5 log records accessed on queue manager 'TEST_91' during the log replay phase.
```

```
Log replay for queue manager 'TEST_91' complete.
```

```
Transaction manager state recovered for queue manager 'TEST_91'.
```

```
IBM MQ queue manager 'TEST_91' started using V9.1.0.0.
```

```
$ runmqsc TEST_91
```

```
## Define a listener. The default is port 1491. (The default is 1414, but here another port is used, just to illustrate that it is possible to use another port other than 1414)
```

```
define listener(LISTENER) trptype(tcp) control(qmgr) port(1491)
start listener(LISTENER)
```

```
## Define a channel to be used by a remote MQ Explorer
```

```
define channel(SYSTEM.ADMIN.SVRCONN) chlname(SVRCONN)
```

```
## Define the DLQ
define qlocal(DLQ) like(SYSTEM.DEAD.LETTER.QUEUE)

## Define test queue:
define qlocal (Q1)

end
```

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

```
$ amqsput Q1 TEST_91
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 TEST_91
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.

+ It is important to know the location of the error logs for the queue manager:

```
$ cd /var/mqm/qmgrs/TEST_91/errors
$ ls
AMQERR01.LOG AMQERR02.LOG AMQERR03.LOG amqp_0.log
```

Sometimes, there will be FDC files generated by MQ processes. They will be located in another directory, which is:

```
$ cd /var/mqm/errors
$ ls
AMQERR01.LOG AMQERR02.LOG AMQERR03.LOG
```

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

```
$ cd /var/mqm
$ cat mqm.ini
...
QueueManager:
  Name=TEST_91
  Prefix=/var/mqm
  Directory=TEST_91
  InstallationName=Installation1
```

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

```
$ dspmq
QMNAME(TEST_91)                STATUS(Running)

$ dspmq -o installation
QMNAME(TEST_91)                INSTNAME(Installation1)
INSTPATH(/opt/mqm) INSTVER(9.1.0.0)
```

The following shows the individual processes for the queue manager:

```
$ ps -ef | grep TEST_91
mqm    26413    1 0 07:45 ?        00:00:00 /opt/mqm/bin/amqzma0 -m TEST_91 -u mqm
mqm    26421 26413  0 07:45 ?        00:00:00 /opt/mqm/bin/amqzfuma -m TEST_91
mqm    26424 26413  0 07:45 ?        00:00:00 /opt/mqm/bin/amqzmuc0 -m TEST_91
mqm    26443 26413  0 07:45 ?        00:00:00 /opt/mqm/bin/amqzmur0 -m TEST_91
mqm    26457 26413  0 07:45 ?        00:00:00 /opt/mqm/bin/amqzmuf0 -m TEST_91
mqm    26462 26413  0 07:45 ?        00:00:00 /opt/mqm/bin/amqrrmfa -m TEST_91 -
t2332800 -s2592000 -p2592000 -g5184000 -c3600
mqm    26475 26413  0 07:45 ?        00:00:00 /opt/mqm/bin/amqzmgr0 -m TEST_91
```

```

mqm 26481 26457 0 07:45 ? 00:00:00 /opt/mqm/bin/amqfqpub -mTEST_91
mqm 26490 26475 0 07:45 ? 00:00:00 /opt/mqm/bin/runmqchi -m TEST_91 -q
SYSTEM.CHANNEL.INITQ -r
mqm 26492 26475 0 07:45 ? 00:00:00 /opt/mqm/bin/amqpcsea TEST_91
mqm 26493 26475 0 07:45 ? 00:00:00 /bin/sh /opt/mqm//bin/amqp.sh start -m
TEST_91 -d /var/mqm/qmgrs/TEST_91//. -g /var/mqm//.
mqm 26494 26413 0 07:45 ? 00:00:00 /opt/mqm/bin/amqzlaa0 -mTEST_91 -fip0
mqm 26502 26481 0 07:45 ? 00:00:00 /opt/mqm/bin/amqfcxba -m TEST_91
mqm 26503 26493 0 07:45 ? 00:00:00 /bin/sh
/opt/mqm/amqp/bin/controlAMQP_mqm.sh start -m TEST_91 -d
/var/mqm/qmgrs/TEST_91//. -g /var/mqm//.
mqm 26545 26503 2 07:45 ? 00:00:05
/opt/mqm/amqp/bin/../../../../java/jre64/jre/bin/java -
Dcom.ibm.mq.mqxr.service.type=amqp -
Dcom.ibm.msg.client.config.location=file:///opt/mqm/amqp/bin/./config/amqp_trace.c
onfig -Xoptionsfile=/opt/mqm/amqp/bin/./config/amqp_java.properties -
Dcom.ibm.msg.client.commonservices.wmq.logdir=/var/mqm/qmgrs/TEST_91//./errors -
Dcom.ibm.msg.client.commonservices.wmq.tracedir=/var/mqm//./trace -
Dcom.ibm.msg.client.commonservices.wmq.ffdcdir=/var/mqm//./errors
com.ibm.mq.MQXRService.RunMQXRService -t /opt/mqm/amqp/bin/./config -m TEST_91 -
d /var/mqm/qmgrs/TEST_91//. -g /var/mqm//.
mqm 26618 26475 0 07:46 ? 00:00:00 /opt/mqm/bin/runmqslr -r -m TEST_91 -t TCP
-p 1491
mqm 26812 25900 0 07:49 pts/0 00:00:00 grep --color=auto TEST_91

```

Verifying that the port is being used. Notice the following line from the output of "ps". It means that the port number 1491 is by used by runmqslr, the MQ Listener for this queue manager.

```

mqm 26618 26475 0 07:46 ? 00:00:00 /opt/mqm/bin/runmqslr -r -m TEST_91 -t TCP
-p 1491

```

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 1491
```

(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 :::1491 :::* LISTEN 26618/runmqslr
```


Let's take a look at the version information held by the queue manager.
Notice that the string under the attribute VERSION is 09000001
which represents: 09 00 00 01 => 09.00.00.01 => 9.1.0.0

```
$ runmqsc TEST_91
display qmgr version
AMQ8408: Display Queue Manager details.
  QMNAME(TEST_91)                VERSION(09010000)
end
```

++++
Chapter 7: Installation of the Fix Pack 9.1.0.1
++++

Before attempting to install the Fix Pack, as an MQ Administrator, you need to ensure that there is no MQ queue manager activity. If there are MQ processes being used at the time the Fix Pack is attempted to be installed, then the operating system will prevent the installer from succeeding, and the attempt will fail.

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

```
$ endmqm -i QmgrName
```

Login as root

Go to the directory where the installation image for the Fix Pack was downloaded:

```
# cd /downloads/mq9101
```

Unpack the tar.gz file:

```
# tar -zxvf 9.1.0-IBM-MQ-LinuxX64-FP0001.tar.gz
```

List the rpm files.

Notice that the file names for the rpm contain the identifier for the Fix Pack:
U9101-9.1.0-1

```
# ls *.rpm  
MQSeriesAMQP-U9101-9.0.0-1.x86_64.rpm  
...
```

Proceed to install these rpm files:

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

Let's list all the rpms installed for MQ (to keep this doc shorter, only few items will be shown, otherwise the list could span a couple of pages)

Notice that you will have 2 types of entries:

Base 9.1.0.0: MQSeries*-9.1.0-0
Fix Pack 9.1.0.1: MQSeries*-U9101-9.1.0-1

```
# rpm -qa | grep MQSeries  
MQSeriesRuntime-9.1.0-0.x86_64  
MQSeriesRuntime-U9101-9.1.0-1.x86_64  
...
```

The following command can be used to show ONLY the rpms for 9.1.0.1:

```
# rpm -qa | grep MQSeries | grep "9\.\.1\.\.0\.-1"
MQSeriesServer-U9101-9.1.0-1.x86_64
MQSeriesXRService-U9101-9.1.0-1.x86_64
...
```

Let's display the output of dspmqver to show the updated version "9.1.0.1":

```
$ dspmqver
Name:      IBM MQ
Version:   9.1.0.1
Level:     p910-001-181108
BuildType: IKAP - (Production)
Platform:  IBM MQ for Linux (x86-64 platform)
Mode:      64-bit
O/S:       Linux 3.10.0-957.1.3.el7.x86_64
InstName:  Installation1
InstDesc:
Primary:   No
InstPath:  /opt/mqm
DataPath:  /var/mqm
MaxCmdLevel: 910
LicenseType: Production
```

++++
Chapter 8: Uninstallation of the Fix Pack 9.1.0.1
++++

Shortcut: Uninstalling all MQ components (both Fix Pack 9.1.0.1 and base 9.1.0.0)

Login as root.

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

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.

You will need to uninstall MQ, in the REVERSE order of the installation, otherwise you will get errors.

This means that for this example the sequence is:

- Uninstall the Fix Pack 9.1.0.1

- Uninstall the Base 9.1.0.0

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 user mqm and stop the queue managers:

```
$ endmqm -i QmgrName
```

There are different approaches for uninstalling MQ.
Here is one of them.

Login as root

Issue the following command, which will ONLY uninstall those MQ rpm's that have 9.1.0.1 in their name.

```
# rpm -ev `rpm -qa | grep MQSeries | grep "9\.\.1\.\.0\1"`  
Preparing packages...  
MQSeriesFTService-U9101-9.1.0-1.x86_64  
...
```

Let's explain a little bit the above 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 9.1.0.1:

```
rpm -qa | grep MQSeries | grep "9\.\.1\.\.0\1"
```

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

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

Issue now the following command, and you will see that there filesets for 9.1.0.1 are no longer shown:

```
# rpm -qa | grep MQSeries | grep "9\.\.1\.\.0\1"
```

++++
Chapter 9: Uninstallation of base 9.1.0.0
++++

Shortcut: Uninstalling all MQ components (both Fix Pack 9.1.0.1 and base 9.1.0.0)
Login as root.
rpm -ev `rpm -qa | grep MQSeries`

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.

You will need to uninstall MQ in the REVERSE order of the installation (first uninstall Fix Packs, then the base), otherwise you will get errors.

This means that for this example the sequence is:

- Uninstall the Fix Pack 9.1.0.x
- Uninstall the Base 9.1.0.0

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 TEST_91  
IBM MQ queue manager 'TEST_91' ending.  
IBM MQ queue manager 'TEST_91' 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 9.1.0.0:

```
# rpm -qa | grep MQSeries
MQSeriesFTAgent-9.1.0-0.x86_64
MQSeriesBCBridge-9.1.0-0.x86_64
MQSeriesMsg_es-9.1.0-0.x86_64
MQSeriesMsg_ru-9.1.0-0.x86_64
MQSeriesRuntime-9.1.0-0.x86_64
MQSeriesFTTools-9.1.0-0.x86_64
MQSeriesClient-9.1.0-0.x86_64
MQSeriesMsg_it-9.1.0-0.x86_64
MQSeriesSamples-9.1.0-0.x86_64
MQSeriesFTLogger-9.1.0-0.x86_64
MQSeriesExplorer-9.1.0-0.x86_64
MQSeriesMsg_hu-9.1.0-0.x86_64
MQSeriesMsg_Zh_TW-9.1.0-0.x86_64
MQSeriesJava-9.1.0-0.x86_64
MQSeriesFTService-9.1.0-0.x86_64
MQSeriesAMS-9.1.0-0.x86_64
MQSeriesSFBridge-9.1.0-0.x86_64
MQSeriesMan-9.1.0-0.x86_64
MQSeriesMsg_fr-9.1.0-0.x86_64
MQSeriesMsg_ko-9.1.0-0.x86_64
MQSeriesMsg_Zh_CN-9.1.0-0.x86_64
MQSeriesJRE-9.1.0-0.x86_64
MQSeriesAMQP-9.1.0-0.x86_64
MQSeriesGSKit-9.1.0-0.x86_64
MQSeriesMsg_ja-9.1.0-0.x86_64
MQSeriesSDK-9.1.0-0.x86_64
MQSeriesFTBase-9.1.0-0.x86_64
MQSeriesXRService-9.1.0-0.x86_64
MQSeriesMsg_de-9.1.0-0.x86_64
MQSeriesMsg_pt-9.1.0-0.x86_64
MQSeriesServer-9.1.0-0.x86_64
MQSeriesWeb-9.1.0-0.x86_64
MQSeriesMsg_cs-9.1.0-0.x86_64
MQSeriesMsg_pl-9.1.0-0.x86_64
```

Proceed to uninstall MQ 9.1.0.0.

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 9.1.0.0:

```
rpm -qa | grep MQSeries | grep "9\.\.1\.\.0\-\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\.\.1\.\.0\-\0"`
```

```
Preparing packages...
```

```
MQSeriesFTService-9.1.0-0.x86_64  
MQSeriesFTLogger-9.1.0-0.x86_64  
MQSeriesFTAgent-9.1.0-0.x86_64  
MQSeriesWeb-9.1.0-0.x86_64  
MQSeriesFTTools-9.1.0-0.x86_64  
MQSeriesFTBase-9.1.0-0.x86_64  
MQSeriesXRService-9.1.0-0.x86_64  
MQSeriesAMQP-9.1.0-0.x86_64  
MQSeriesSFBridge-9.1.0-0.x86_64  
MQSeriesBCBridge-9.1.0-0.x86_64  
MQSeriesExplorer-9.1.0-0.x86_64  
MQSeriesAMS-9.1.0-0.x86_64  
MQSeriesServer-9.1.0-0.x86_64  
MQSeriesJRE-9.1.0-0.x86_64  
MQSeriesJava-9.1.0-0.x86_64  
MQSeriesMsg_pl-9.1.0-0.x86_64  
MQSeriesMsg_cs-9.1.0-0.x86_64  
MQSeriesMsg_pt-9.1.0-0.x86_64  
MQSeriesMsg_de-9.1.0-0.x86_64  
MQSeriesSDK-9.1.0-0.x86_64  
MQSeriesMsg_ja-9.1.0-0.x86_64  
MQSeriesGSKit-9.1.0-0.x86_64  
MQSeriesMsg_Zh_CN-9.1.0-0.x86_64  
MQSeriesMsg_ko-9.1.0-0.x86_64  
MQSeriesMsg_fr-9.1.0-0.x86_64  
MQSeriesMan-9.1.0-0.x86_64  
MQSeriesMsg_Zh_TW-9.1.0-0.x86_64  
MQSeriesMsg_hu-9.1.0-0.x86_64  
MQSeriesSamples-9.1.0-0.x86_64  
MQSeriesMsg_it-9.1.0-0.x86_64  
MQSeriesClient-9.1.0-0.x86_64  
MQSeriesMsg_ru-9.1.0-0.x86_64
```



```
MQSeriesMsg_es-9.1.0-0.x86_64  
MQSeriesRuntime-9.1.0-0.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)

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

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
```