Raspberry Pi Internet Radio

Compiling and installing the Music Player Daemon

A comprehensive guide how to compile and install the Music Player Daemon

Bob Rathbone Computer Consultancy

www.bobrathbone.com

31st of March 2026
MPD Version 0.24.9

Contents

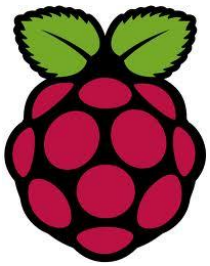
Chapter 1 - Introduction 4
Motivation for writing this procedure 4
Audience 4
Pre-requisites 4
Radio software installation 4
Supported Operating Systems 5
Conventions used in this tutorial 5
Chapter 2 - Installing the latest MPD and MPC software 7
Compiling and installing the latest Music Player Daemon 7
Upgrade the operating system 7
Downloading the source 7
Install Ninja and Meson build utilities 8
Install pre-requisite libraries 8
Build the MPD package 9
Install the MPD package 9
Configuring the new MPD package 10
Testing MPD 11
Installing MPD upgrade from a Debian package 12
Installing the latest MPC client from source 14
Installing Music Player client (mpc) from a Debian package 16
Chapter 3 - Building the MPD Debian packages 17
Building the MPD Debian package 17
Building the MPC Client Debian Package 18
Chapter 4 - Trouble shooting 20
MPD fails to start 20
Missing configuration file in the mpd.service file 20
Missing Libraries cause service start to fail 20
Cannot connect to MySql 22
Chapter 4 - Licences, disclaimer and support 24
Licences 24
Disclaimer 24
Support 24

Acknowledgements and Copyrights 24
Appendix A – The MPD service file 26
Appendix B – MPD and MPC package build example output 27
Appendix B.1 MPD package build example 27
Appendix B.2 MPC package build example 28
Appendix C – MPC commands 29

Figures

Figure 1 Music Player Daemon Website 7
Figure 2 MPC client source download 14

Chapter 1 - Introduction



This manual describes how download, compile and install the Music Player Daemon (**MPD**) for Raspberry Pi on the **Raspberry Pi OS** operating system (Formally called **Raspbian**). It is a supplement to the **Raspberry Pi Constructors Guide** for building Internet radios. However, it can be used for any package which utilises the Music Player Daemon.

The source and basic construction details for the *Raspberry Pi Internet Radio* are available from the following web site:
https://bobrathbone.com/raspberrypi/pi_internet_radio.html

Motivation for writing this procedure

The Raspberry Pi usually is installed with the **Raspberry Pi OS** operating system. At the time of writing the latest version of **Raspberry Pi OS** is called **Bullseye**. Unfortunately, **Raspberry Pi OS** is released with out-of-date versions of the Music Player daemon. To date the MPD software is never updated with the normal package upgrade releases.

The **MPD** version described in this procedure is **0.24.9**whilst the version released with **Bullseye** is **0.22.6** and is already fairly out-of-date. The version released with **Buster** is **0.21.5** which is even worse as it is years out of date. The latest version of MPD is much more stable. Due to their age, it is impossible to get support on the older versions of MPD.

Audience

This guide is intended for anyone that wishes to update their Music Player Daemon and Client (MPC) to the latest version. However, a certain amount of knowledge of the Linux Operating System is required as well as the ability to edit and change system files. This does not rule this procedure out for use by those less experienced but in such a case caution should be applied as the procedures in this manual are classed as moderately advanced.

Pre-requisites

1. **Raspberry Pi OS Trixie** (**Bookworm**, **Bullseye** or **Buster** also supported) installed and updated using **apt update** and **apt full-upgrade**.
2. The version of **MPD** package released with Raspberry Pi OS you are using should be installed first. For example: for Trixie this is version **0.24.4**.
3. A reasonably fast Raspberry Pi. The **Raspberry Pi Zero** isn't really suitable.

Note: During testing an attempt was made to run this procedure on a **Pi Zero** but without success. The problem was that the compile phase took hours to run and seemed to hang. The suggestion is to put the SD card to be upgraded into a faster Raspberry Pi and then upgrade it or run the compilation procedure on the Raspberry Pi Zero overnight.

Radio software installation

If you have reached this document whilst installing the Raspberry Pi Radio Software (**radiod**) then you should first install and configure the MPD software released with Buster and the radio software as shown in the constructor's manual.
See https://www.bobrathbone.com/raspberrypi/documents/Raspberry%20PI%20Radio.pdf

Supported Operating Systems

Three versions of the Raspberry Pi OS are supported in this tutorial.

- **Trixie** and **Bookworm** – Install version **0.24.9**
  - **mpd_0.24.9_all.deb** (for both 32 and 64-bit)
- **Bullseye** and **Bookworm** – Install version **0.23.17**
  - **mpd_0.23.17_armhf.deb** (32-bit)
  - **mpd_0.23.17_arm64.deb** (64-bit)
- **Buster** – Install version **0.23.5** (32-bit only)
  - **mpd _0.23.5_armhf.deb**

Version **0.23.17** will not run on Buster as it requires the **libicu67** library which is not available in **Buster**. If using **Buster** substitute all commands using **0.23.17** with **0.23.5.** For example:

```
$ wget http://www.musicpd.org/download/mpd/0.23/mpd-0.23.17.tar.xz
```

For **Buster**:

```
$ wget http://www.musicpd.org/download/mpd/0.23/mpd-0.23.5.tar.xz
```

Conventions used in this tutorial

Installation of the radio program requires you to enter lines at the command line prompt. This requires you to log into the Raspberry PI as user '**pi**'. The default password is **raspberry**.

Note: Don't carry out any of the following commands just yet. They are just examples.

```
Raspberrypi login: pi
Password: raspberry
pi@raspberrypi:~$ Last login: Wed Sep  9 12:17:35 2020 from 192.168.1.200
pi@raspberrypi:~$
```

The prompt line is displayed ending with a **$** sign. The **pi@raspberrypi:~** string means user 'pi' on host machine called 'raspberrypi'. The ~ character means the user 'pi' home directory **/home/pi**. In this tutorial if you are required to do something as user **pi** then only the $ sign will be shown followed by the command as shown in the example below:

```
$ mpc status
```

Copy and paste the highlighted text only to the command line. Omit the $ sign.

Some commands produce a lot output which does not need to be shown. In such a case a colon (':') is used to indicate that some output has been omitted.

```
$ mpd -V
Music Player Daemon 0.23.17(0.23.17)
Copyright 2003-2007 Warren Dukes <warren.dukes@gmail.com>
Copyright 2008-2018 Max Kellermann <max.kellermann@gmail.com>
:
Other features:
 avahi dbus udisks epoll icu inotify ipv6 systemd tcp un
```

END OF EXAMPLE COMMANDS.

Chapter 2 - Installing the latest MPD and MPC software

There are two methods of updating MPD.

1. Download the latest MPD source, then compile and install it.
2. Download a ready-made Debian package mpd_0.23.17_armhf.deb

To install directly from the Debian package (option 2) see *Installing MPD upgrade from a Debian package* on page 12.

Compiling and installing the latest Music Player Daemon

This procedure is based upon the **Installation** section of the following document.
https://mpd.readthedocs.io/en/stable/user.html#installing-on-debian-ubuntu

You are strongly advised to read both the above document and this one but with caution as either can change. However, there are a number of gaps in the document from **miscpd.org** which can lead to confusion when operating the new version of MPD. This guide has been written for the installation of Music Player Daemon version **0.24.9** dated January 2026.

Upgrade the operating system

Run the following three commands: DO NOT SKIP THIS STEP!

```
$ sudo apt update
$ sudo apt full-upgrade
$ sudo reboot
```

Downloading the source

The source is downloaded from **MusicPd** at https://www.musicpd.org/

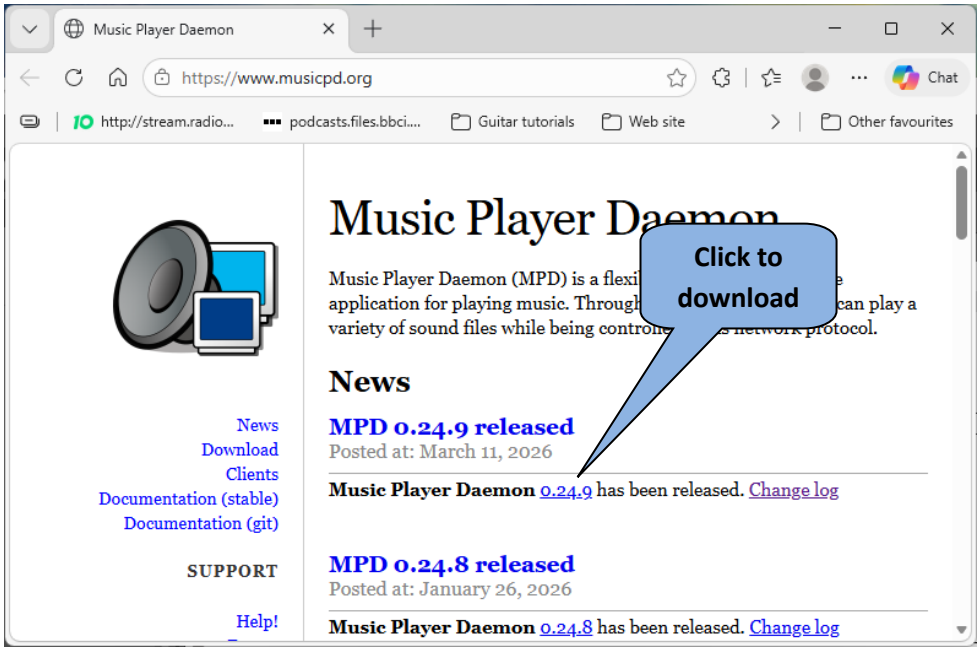


Figure 1 Music Player Daemon Website

This will list the latest version of the software; in this case it is version **0.24.9** The example above shows 0.24.9**.** If downloading a later version than shown then substitute the version number shown in the following examples with the one you are downloading. Either click on the version number to download the software to your PC, or alternatively right click the

version number to copy the link then use **wget** to download it via the command line on the Raspberry Pi. Substitute **0.24.9** with **0.23.5** in the following commands if using Buster.

```
$ cd
$ wget https://www.musicpd.org/download/mpd/0.24/mpd-0.24.9.tar.xz
```

Now unzip the archive

```
$ tar -xvf mpd-0.24.9
```

This will unzip the archive to a directory called **mpd-0.24.19**.

Install Ninja and Meson build utilities

This build uses Ninja and Meson.

```
$ sudo apt install meson ninja-build
```

Install pre-requisite libraries

Now install the pre-requisite libraries. Note that the text below is all one line. Do not copy the $ sign.

For **Trixie** and later (Builds for earlier OS versions no longer supported)

```
$ sudo apt install meson g++ pkgconf \
  libfmt-dev \
  libpcre2-dev \
  libmad0-dev libmpg123-dev libid3tag0-dev \
  libflac-dev libvorbis-dev libopus-dev libogg-dev \
  libadplug-dev libaudiofile-dev libsndfile1-dev libfaad-dev \
  libfluidsynth-dev libgme-dev libmikmod-dev libmodplug-dev \
  libmpcdec-dev libwavpack-dev libwildmidi-dev \
  libsidplay2-dev libsidutils-dev libresid-builder-dev \
  libavcodec-dev libavformat-dev \
  libmp3lame-dev libtwolame-dev libshine-dev \
  libsamplerate0-dev libsoxr-dev \
  libbz2-dev libcdio-paranoia-dev libiso9660-dev libmms-dev \
  libzzip-dev \
  libcurl4-gnutls-dev libexpat1-dev \
  nlohmann-json3-dev \
  libasound2-dev libao-dev libjack-jackd2-dev libopenal-dev \
  libpulse-dev libshout3-dev \
  libsndio-dev \
  libmpdclient-dev \
  libnfs-dev \
  libupnp-dev \
  libavahi-client-dev \
  libsqlite3-dev \
  libsystemd-dev \
  libgtest-dev \
  libicu-dev \
  libchromaprint-dev \
  libgcrypt20-dev \
  libsystemd-dev \
  libpipewire-0.3-dev
```

Press enter to install the above libraries.

Build the MPD package

The package uses a multi-platform software build system called **meson** from **mesonbuild.com**. See https://mesonbuild.com/ for further information.

Change to the recently created MPD build directory:

```
$ cd mpd-0.24.9
```

Configure the package with the following command:

```
$ meson setup . output/release --buildtype=debugoptimized -Db_ndebug=true
```

Change the sysconfdir setting to determine where mpd will look for mpd.conf; The sysconfdir prameter must be set to '/etc' i.e., not 'etc' which will result in mpd looking for /usr/local/etc/mpd.conf:

```
$ meson configure output/release |grep sysconfdir
sysconfdir                    etc
```

To change the location of the configuration file from **/etc/local/etc/mpd.conf** to **/etc/mpd.conf** enter the following instruction.

```
$ meson configure output/release -Dsysconfdir='/etc'
$ meson configure output/release |grep sysconfdir
sysconfdir                    /etc
```

If you are upgrading MPD then first save the existing /etc/mpd.conf with the following instruction:

```
$ cp /etc/mpd.conf ~/mpd.conf
```

Once you have saved your existing /etc/mpd.conf, compile and install MPD using **ninja**. For further information see https://ninja-build.org/. Run the following.

```
$ sudo ninja -C output/release
```

This will take quite a long time as it has at least 670 files to compile. On a Raspberry Pi 4 it will take about 7 minutes. On an older slower Raspberry Pi, it will be at least twice that.

Install the MPD package

Stop MPD and any software using it. In this example **radiod** is stopped.

```
$ sudo systemctl stop radiod mpd
```

Finally run the following command to actually install the package.

```
$ sudo ninja -C output/release install
```

This will install the **mpd** daemon in the **/usr/local/bin** directory.

```
ninja: Entering directory `output/release'
[1/2] Installing files
Installing mpd to /usr/local/bin
Installing /home/pi/mpd-0.24.9/output/release/systemd/system/mpd.socket
to /usr/local/lib/systemd/system
Installing /home/pi/mpd-0.24.9/output/release/systemd/system/mpd.service
to /usr/local/lib/systemd/system
Installing /home/pi/mpd-0.24.9/output/release/systemd/user/mpd.socket to
/usr/local/lib/systemd/user
Installing /home/pi/mpd-0.24.9/output/release/systemd/user/mpd.service to
/usr/local/lib/systemd/user
Installing /home/pi/mpd-0.24.9/mpd.svg to
/usr/local/share/icons/hicolor/scalable/apps
Installing /home/pi/mpd-0.24.9/AUTHORS to /usr/local/share/doc/mpd
Installing /home/pi/mpd-0.24.9/COPYING to /usr/local/share/doc/mpd
Installing /home/pi/mpd-0.24.9/NEWS to /usr/local/share/doc/mpd
Installing /home/pi/mpd-0.24.9/README.md to /usr/local/share/doc/mpd
```

Configuring the new MPD package

If you previously installed any version of MPD then the old version of **mpd** will still be present in the **/usr/bin** directory.

```
$ /usr/bin/mpd -V | head -1
Music Player Daemon 0.24.4 (0.24.4)
```

Note: your version may be different from that above.

However, the new **mpd** binary is installed in **/usr/local/bin**.

```
$ which mpd
/usr/local/bin/mpd
```

This can be confirmed with the following command:

```
$ /usr/local/bin/mpd -V | head -1
Music Player Daemon 0.24.9 (0.24.9)
```

Edit the **/usr/local/lib/systemd/system/mpd.service** system definition file (Use sudo vi or nano)
Make sure that **ExecStart** line points to /usr/local/bin/mpd

```
ExecStart=/usr/local/bin/mpd -systemd
```

Now reload system units or reboot the Raspberry Pi.

```
$ sudo systemctl daemon-reload
```

Testing MPD

Start the MPD daemon and check its status.

```
$ sudo systemctl start mpd
$ sudo systemctl status mpd
```

This should display something similar to the following:

```
● mpd.service - Music Player Daemon
     Loaded: loaded (/usr/local/lib/systemd/system/mpd.service; disabled;
vendor preset: enabled)
     Active: active (running) since Sun 2022-07-24 09:25:13 BST; 1min 4s
ago
TriggeredBy: ● mpd.socket
       Docs: man:mpd(1)
             man:mpd.conf(5)
   Main PID: 19029 (mpd)
      Tasks: 6 (limit: 2055)
        CPU: 2.534s
     CGroup: /system.slice/mpd.service
             └─19029 /usr/local/bin/mpd --systemd /etc/mpd.conf

Jul 24 09:25:13 piradio2 systemd[1]: Starting Music Player Daemon...
Jul 24 09:25:13 piradio2 mpd[19029]: Ignoring the 'pid_file' setting in
systemd mode
Jul 24 09:25:13 piradio2 systemd[1]: Started Music Player Daemon.
:
```

Note: Note ignore any warnings about wildmidi or bind errors. Also the version of **mpd** being used is in **/usr/local/bin** and not **/usr/bin**. This is confirmed with the following command:

```
$ which mpd
/usr/local/bin/mpd
```

This also applies to the service startup files which are now contained in the **/usr/local/lib/system/system** directory and not **/lib/system/system**.

If you already have a playlist and MPD has been configured for your sound device the first station should start playing. If you don't have a playlist then create one in **/var/lib/mpd/playlists/<playlist name>.m3u**
For example, **/var/lib/mpd/playlists/_Radio.m3u**

```
#EXTM3U
#EXTINF:-1,BBC Radio 1
http://sc59.lon.llnw.net:80/stream/bbcmedia_radio1_mf_p#BBC Radio 1
#EXTM3U
#EXTINF:-1,BBC Radio 2
http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio2_mf_p?s#BBC Radio 2
#EXTINF:-1,Nashville FM
http://server-27.stream-server.nl:8300/stream#Nashville FM
#EXTM3U
#EXTINF:-1,UK Country Radio
http://s3.xrad.io:8064/#UK Country Radio
#EXTM3U
#EXTINF:-1,Radio Lichtenstein
http://live.radiol.li:8000/country#Radio Lichtenstein
```

Now load the new radio playlist. Omit the **m3u** extension.

```
$ mpc load _Radio
```

Confirm that it is loaded.

```
$ mpc playlist
BBC Radio 1
BBC Radio 2
UK Country Radio
```

Now play the first station.

```
$ mpc play
BBC Radio 1
[playing] #1/143   0:00/0:00 (0%)
volume: 60%   repeat: off   random: off   single: off   consume: off
```

Adjust the volume as necessary

```
$ mpc volume 70
```

This ends the procedure. If you are running the Rathbone radio software the restart it by rebooting the Raspberry Pi or run **sudo systemctl start radiod**.

Installing MPD upgrade from a Debian package

This procedure is is written for Trixie only. All packages mentioned previously are available in the package directory on the Bob Rathbone web site.This is an upgrade and **mpd** and **mpc** must be installed first. If you have already installed the Rathbone Internet radio software then this step isn't necessary but does no harm as mpc and mpd are already installed.

```
$ sudo apt install mpd mpc
```

Answer yes 'y' when asked to continue.

Install the required libraries. This may take some time if not already installed. Note the following instruction is all one line. Do not copy the $ character.

```
$ sudo apt install libsidutils0 libresid-builder0c2a libcurl3-gnutls
libaudiofile1 libyajl2 libiso9660-11 libzzip-0-13 libao4
```

For Bullseye

```
$ sudo apt install libadplug-2.3.3-0
```

For Buster

```
$ sudo apt install libadplug-2.2.1-0v5
```

Download the MPD updated software.

```
$ wget https://bobrathbone.com/raspberrypi/packages/mpd_0.23.17_armhf.deb
```

Now install with **dpkg**.

```
$ sudo dpkg -i mpd_0.23.17_armhf.deb
(Reading database ... 135664 files and directories currently installed.)
Preparing to unpack mpd_0.23.17_armhf.deb ...
Installing MPD
systemctl stop mpd.service mpd.socket
Warning: The unit file, source configuration file or drop-ins of
mpd.service changed on disk. Run 'systemctl daemon-reload' to reload
units.
Unpacking mpd (0.23.17) over (0.22.6-1) ...
Setting up mpd (0.23.17) ...
Updating MPD configuration
sudo chown mpd:audio /var/tmp/mpd.conf
/etc/mpd.conf exists
sudo chown mpd:audio /etc/mpd.conf
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for man-db (2.9.4-2) ...
```

Ignore the warnings for **/var/log/mpd** and **/var/lib/mpd**.
Modify **/usr/local/lib/systemd/system/mpd.service** to use **/etc/mpd.conf** as its configuration file:
Change

```
ExecStart=/usr/bin/mpd --no-daemon
```

To

```
ExecStart=/usr/bin/mpd --no-daemon /etc/mpd.conf
```

Reboot the Raspberry Pi to enable the new version.

```
$ sudo reboot
```

Note: For some unknown reason when **mpd.socket** is run from this package it changes the ownership of the **/var/run/mpd** directory from **mpd:audio** to **root:root** which causes **mpd.service** to fail with a permissions error. To correct this problem the following line has been added to **mpd.service** just before the **ExecStart** statement.

```
ExecStartPre=-/bin/chown mpd:audio /var/run/mpd
```

Installing the latest MPC client from source

The **mpc** client software is also out of date but will work fine with the latest version of MPD described in this manual. However, this can also be also be compiled and installed. Only carry out this routine after installing meson and ninja and compiling MPD as previously shown. Go to https://www.musicpd.org/ and click on "clients" in the left-had menu. Now copy the link to **mpc** as shown below.

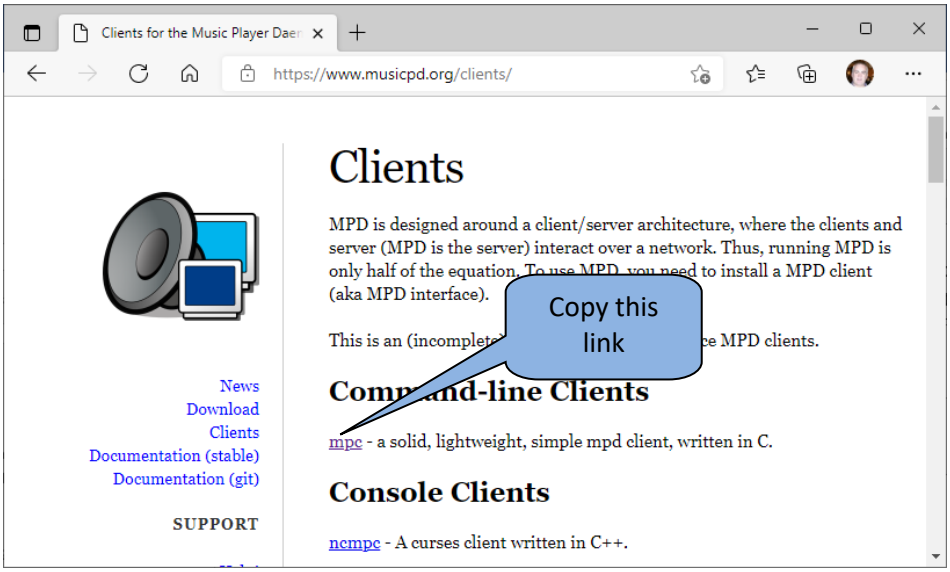


Figure 2 MPC client source download

Log into your home directory and type in **wget** and copy the link just copied from the musicpd.org site.

```
$ wget https://www.musicpd.org/download/mpc/0/mpc-0.35.tar.xz
```

Extract the **mpc** source archive.

```
$ tar -xvf mpc-0.35.tar.xz
```

Change to **mpc** the build directory.

```
$ cd mpc-0.35
```

Run meson to configure the compile process.

```
$ meson . output
```

Compile the **mpc** source code.

```
$ ninja -C output
```

Install the **mpc** client.

```
$ sudo ninja -C output install
ninja: Entering directory `output'
[0/1] Installing files
Installing mpc to /usr/local/bin
Installing /home/pi/mpc-0.35/AUTHORS to /usr/local/share/doc/mpc
Installing /home/pi/mpc-0.35/COPYING to /usr/local/share/doc/mpc
Installing /home/pi/mpc-0.35/NEWS to /usr/local/share/doc/mpc
Installing /home/pi/mpc-0.35/README.rst to /usr/local/share/doc/mpc
Installing /home/pi/mpc-0.35/contrib/mpd-m3u-handler.sh to
/usr/local/share/doc/mpc/contrib
Installing /home/pi/mpc-0.35/contrib/mpd-pls-handler.sh to
/usr/local/share/doc/mpc/contrib
Installing /home/pi/mpc-0.35/contrib/mpc-completion.bash to
/usr/local/share/doc/mpc/contrib
```

Amend the **ExecStart** statement in **/usr/local/lib/systemd/system/mpd.service** to include the configuration file name (/etc/mpd.conf).

```
ExecStart=/usr/local/bin/mpd --no-daemon /etc/mpd.conf
```

The new **mpc** client will be found in

```
$ which mpc
/usr/local/bin/mpc
```

Move the old version of **mpc** out the way and link the newly compiled client to **/usr/bin/mpc**

```
$ cd /usr/bin/
$ sudo mv mpc mpc.old
$ sudo ln -s /usr/local/bin/mpc
```

Restart the MPD

```
$ sudo systemctl restart mpd
```

Test **mpc**. Check the version first.

```
$ mpc help | head -3 | grep version
mpc version: 0.35
```

Note that at the time of writing the **mpc version** command gives the wrong version number. i.e. 0.34 instead of 0.35. Use the above instruction to display the version number.

Run mpc. There is no need to reboot the Raspberry Pi.

```
Radio Caroline: Sweet Little Machine - We Used To Own These Playgrounds
[playing] #18/116 262:59/0:00 (0%)
volume: 58%   repeat: off   random: off   single: off   consume: off
```

Note: There are new commands for the new version of **mpc**. A complete list of mpc commands is shown in *Appendix C – MPC commands* on page 29.

Installing Music Player client (mpc) from a Debian package

The **mpc** client package is currently available in the package directory on the Bob Rathbone web site.
This is an upgrade and the latest **mpd** as shown in the section called *Installing MPD upgrade from a Debian package* on page 12 must be installed first.

Download the MPD update software.

```
$ wget https://bobrathbone.com/raspberrypi/packages/mpc_0.35_armhf.deb
```

Now install with **dpkg**.

```
$ sudo dpkg -i mpc_0.35_armhf.deb
(Reading database ... 107174 files and directories currently installed.)
Preparing to unpack mpc_0.35_armhf.deb ...
Installing MPC
Unpacking mpc (0.35) over (0.34) ...
Removing MPC
sudo rm /usr/bin/mpc
sudo mv /usr/bin/mpc.old /usr/bin/mpc
Setting up mpc (0.35) ...
Updating MPC configuration
sudo rm /usr/bin/mpc
sudo ln -s /usr/local/bin/mpc /usr/bin/mpc
```

Test **mpc**.

```
$ mpc
BBC Radio: BBC Radio 2
[playing] #2/114  30:52/0:00 (0%)
volume: 85%   repeat: off   random: off   single: off   consume: off
```

Chapter 3 - Building the MPD Debian packages

Building the MPD Debian package

This chapter shows how to create the MPD Debian package using the script provided from the following GitHub Repository.
https://github.com/bobrathbone/mpdpackage

Before this can be done you must first download and compile the MPD package from GitHub as shown in *Chapter 2 - Installing the latest MPD* on page 7. If this has already been done, download the MPD Debian package scripts from GitHub.

```
$ cd
$ git clone https://github.com/bobrathbone/mpdpackage
```

Now change directory to **mpdpackage**.

```
$ cd mpdpackage
```

Edit the build script **build.sh**. Find the **BUILD_DIR** definition. Amend this to point to the actual MPD build directory that was used in *Chapter 2 - Installing the latest MPD* on page 7.

```
BUILD_DIR=/home/pi/mpd-0.24.9
```

Now run the **build.sh** script.

```
$ ./build.sh
```

If you see an error similar to below, check that you have correctly specified the **BUILD_DIR** definition.

```
Building MPD version 0.24.9
ERROR: /home/pi/mpd-0.23.17/output not found
Check that BUILD_DIR is correctly specified in this script
Aborting
```

If you have correctly specified the **BUILD_DIR** definition the you should see the output similar to that shown in *Appendix B – MPD and MPC package build* example output on page 27. The package **mpd_0.24.9_all.deb** should also have been created. This can now be installed with **dpkg**.

Installation for the 32-bit package

```
$ sudo dpkg -i mpd_0.24.9_all.deb
```

If you require package symbols for debugging install **mpd-dbgsym_0.23.17_armhf.deb**

```
$ sudo dpkg -i mpd-dbgsym_0.24.9_all.deb
```

Installation for both 32 and 64-bit systems

```
$ sudo dpkg -i mpd_0.24.9_all.deb
```

If you require package symbols for debugging install **mpd-dbgsym_0.23.17_arm64.deb**

```
$ sudo dpkg -i mpd-dbgsym_0.24.9_all.deb
```

Building the MPC Client Debian Package

This chapter shows how to create the MPC Client Debian package using the script provided from the following GitHub Repository.
https://github.com/bobrathbone/mpcpackage

Before this can be done you must first download and compile the MPD and MPC packages as shown in *Chapter 2 - Installing the latest MPD* on page 7 and *Installing the latest MPC client from source* on page 14.

First download the MPC Debian package scripts from GitHub.

```
$ cd
$ git clone https://github.com/bobrathbone/mpcpackage
```

Now change directory to **mpcpackage**.

```
$ cd mpcpackage
```

Edit the build script **build.sh**. Find the **BUILD_DIR** definition. Amend this to point to the actual MPC build directory that was used in *Installing the latest MPC client from source* on page 14.

```
BUILD_DIR=/home/pi/mpc-0.35
```

Now run the appropriate **build.sh** script.
For 32-bit both 64-bit systems

```
$ ./build.sh
```

If you see an error similar to below, check that you have correctly specified the **BUILD_DIR** definition.

```
$ ./build.sh
Building MPC version 0.35
ERROR: /home/pi/mpc-0.35/output not found
Check that BUILD_DIR is correctly specified in this script
```

```
Aborting
```

If you have correctly specified the **BUILD_DIR** definition the you should see the output similar to that shown in *Appendix B – MPD and MPC package build* example output on page 27. The package **mpd_0.35_all.deb** should also have been created depending on the architecture. This can now be installed with **dpkg**.

For 32-bit and 64-bit systems

```
$ sudo dpkg -i mpc_0.34_all.deb
```

Chapter 4 - Trouble shooting

MPD fails to start

Missing configuration file in the mpd.service file

```
$ sudo systemctl start mpd
Job for mpd.service failed because the control process exited with error
code.
See "systemctl status mpd.service" and "journalctl -xe" for details.
```

Check the status

```
$ systemctl status mpd.service
● mpd.service - Music Player Daemon
   Loaded: loaded (/usr/local/lib/systemd/system/mpd.service; disabled;
vendor preset: enabled)
   Active: failed (Result: exit-code) since Fri 2020-05-29 19:30:19 BST;
1min 41s ago
     Docs: man:mpd(1)
           man:mpd.conf(5)
  Process: 1060 ExecStart=/usr/local/bin/mpd --no-daemon (code=exited,
status=1/FAILURE)
 Main PID: 1060 (code=exited, status=1/FAILURE)

May 29 19:30:19 buster03 systemd[1]: Starting Music Player Daemon...
May 29 19:30:19 buster03 mpd[1060]: exception: No configuration file
found
May 29 19:30:19 buster03 systemd[1]: mpd.service: Main process exited,
code=exited, status=1/FAILURE
May 29 19:30:19 buster03 systemd[1]: mpd.service: Failed with result
'exit-code'.
May 29 19:30:19 buster03 systemd[1]: Failed to start Music Player Daemon.
```

In this case **/usr/local/bin/mpd** cannot find **/etc/mpd.conf**. Check that the **ExecStart** statement in **/usr/local/lib/systemd/system/mpd.service** contains the configuration file name. You should not need this for **Trixie**.

```
ExecStart=/usr/local/bin/mpd --no-daemon /etc/mpd.conf
```

Missing Libraries cause service start to fail

A message similar to the following shows that one or more required libraries are missing.

```
$ sudo systemctl status mpd.service
● mpd.service - Music Player Daemon
:
Apr 12 01:54:59 buster04 mpd[3460]: /usr/local/bin/mpd: error while loading
shared libraries: libao.
Apr 12 01:54:59 buster04 systemd[1]: mpd.service: Main process exited,
code=exited, status=127/n/a
Apr 12 01:54:59 buster04 systemd[1]: mpd.service: Failed with result 'exit-
code'.
Apr 12 01:54:59 buster04 systemd[1]: Failed to start Music Player Daemon.
```

Solution: Install the pre-requisite libraries. Note that the text below is all one line.
Do not copy the $. Press enter to install the libraries.

```
$ sudo apt install libpcre3 libao4 libmad0 libmpg123-0 libid3tag0
libflac8 libvorbis0a libopus0 libogg0 libsndfile1 libfaad2 libfluidsynth1
libgme0 libmikmod3 libmodplug1 libmpcdec6 libwavpack1 libwildmidi2
libsidplay2 libsidutils0 libresid-builder0c2a libavcodec58 libavformat58
libmp3lame0 libtwolame0 libshine3 libsamplerate0 libsoxr0 libbz2-1.0
libcdio-paranoia2 libiso9660-11 libmms0 libzzip-0-13 libyajl2 libexpat1
libasound2 libaom0 libjack-jackd2-0 libopenal1 libpulse0 libshout3
libsndio7.0 libmpdclient2 libnfs12 libupnp13 libavahi-client3 libsqlite3-
0 libsystemd0 libicu63 libchromaprint1 libgcrypt20 libcurl3-gnutls
libadplug-2.2.1-0v5 libaudiofile1
```

Reset failed units **mpd** service units. and recheck.

```
$ sudo systemctl reset-failed mpd.socket mpd.service
```

Restart **mpd** service.

```
$ sudo systemctl start mpd.service
```

Re-check the mpd services.

```
$ sudo systemctl status mpd.*
● mpd.service - Music Player Daemon
     Loaded: loaded (/usr/local/lib/systemd/system/mpd.service; disabled;
vendor preset: enabled)
     Active: active (running) since Sun 2022-03-20 08:39:11 GMT; 23min
ago
TriggeredBy: ● mpd.socket
       Docs: man:mpd(1)
             man:mpd.conf(5)
   Main PID: 2419 (mpd)
      Tasks: 6 (limit: 4915)
        CPU: 28.860s
     CGroup: /system.slice/mpd.service
             └─2419 /usr/local/bin/mpd --systemd /etc/mpd.conf

:
Mar 20 08:39:11 pi400 systemd[1]: Started Music Player Daemon.

● mpd.socket
     Loaded: loaded (/usr/local/lib/systemd/system/mpd.socket; disabled;
vendor preset: enabled)
     Active: active (running) since Sun 2022-03-20 08:37:20 GMT; 25min
ago
   Triggers: ● mpd.service
     Listen: /run/mpd/socket (Stream)
             [::]:6600 (Stream)
      Tasks: 0 (limit: 4915)
        CPU: 2ms
     CGroup: /system.slice/mpd.socket
:
```

Finally remove any redundant libraries

```
$ sudo apt autoremove
```

Cannot connect to MySql

O!MPD reports the following:

ERROR

Failed to connect to MySQL server on:
127.0.0.1

Check the status of the MySql service:

I traced the problem to no user name provided to access MySql

```
$ sudo systemctl status mysql.service
● mariadb.service - MariaDB 10.11.3 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled;
preset: enabled)
     Active: active (running) since Fri 2023-12-01 06:20:10 GMT; 18min
ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
:
socket: '/run/mysqld/mysqld.sock'  port: 3306  Ra>
Dec 01 06:20:10 bookworm32 systemd[1]: Started mariadb.service - MariaDB
10.11.3 database server.
Dec 01 06:20:10 bookworm32 /etc/mysql/debian-start[1120]: Upgrading MySQL
tables if necessary.
Dec 01 06:24:56 bookworm32 mariadbd[870]: 2023-12-01  6:24:56 54
[Warning] Access denied for user ''@'localhost'
Dec 01 06:33:29 bookworm32 mariadbd[870]: 2023-12-01  6:33:29 58
[Warning] Access denied for user ''@'localhost'
Dec 01 06:33:39 bookworm32 mariadbd[870]: 2023-12-01  6:33:39 59
[Warning] Access denied for user ''@'localhost'
```

The above shows that no user has been set up for MySql. Edit **ompd/include/config.inc.php** and add user pi and password.

```
//  +------------------------------------------------------------------------
--+
//  | MySQL configuration
|
//  +------------------------------------------------------------------------
--+
$cfg['mysqli_host']                 = '127.0.0.1';
$cfg['mysqli_db']                   = 'ompd';
$cfg['mysqli_user']                 = 'pi';
$cfg['mysqli_password']             = 'raspberry';
$cfg['mysqli_port']                 = '3306';
$cfg['mysqli_socket']               = '';
$cfg['mysqli_auto_create_db']       = true;
```

Restart apache2:

```
$ sudo systemctl restart apache2
```

Chapter 4 - Licences, disclaimer and support

Licences

The software and documentation for this project is released under the GNU General Public Licence.

The GNU General Public License (GNU GPL or GPL) is the most widely used free software license, which guarantees end users (individuals, organizations, companies) the freedoms to use, study, share (copy), and modify the software. Software that ensures that these rights are retained is called free software. The license was originally written by Richard Stallman of the Free Software Foundation (FSF) for the GNU project.

The GPL grants the recipients of a computer program the rights of the Free Software Definition and uses *copyleft* to ensure the freedoms are preserved whenever the work is distributed, even when the work is changed or added to. The GPL is a *copyleft* license, which means that derived works can only be distributed under the same license terms. This is in distinction to permissive free software licenses, of which the BSD licenses are the standard examples. GPL was the first *copyleft* license for general use. This means that you may modify and distribute the software and documentation subject to the conditions of the licences.

See http://www.gnu.org/licenses for up-to-date information on the GNU General Public License.

The licences for the source and documentation for this project are:
GNU General Public License. See http://www.gnu.org/licenses/gpl.html
GNU AFFERO General Public License. See http://www.gnu.org/licenses/agpl.html
GNU Free Documentation License. See http://www.gnu.org/licenses/fdl.html
MPD uses an older version of the license at https://www.gnu.org/licenses/gpl-2.0.txt

Disclaimer

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BELIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE OR DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Support

The author does not directly support the Music Player Daemon software. You are advised to contact the MPD forum at https://forum.musicpd.org/. However, if you have any comments or feedback about this procedure or document, please contact bob@bobrathbone.com

Acknowledgements and Copyrights

Warren Dukes warren.dukes@gmail.com and Max Kellermann max.kellermann@gmail.com and the **musicpd.org** community for their excellent work on the Music Player Daemon.

Copyrights (Music Player Damon Software)
Copyright 2003-2007 Warren Dukes <warren.dukes@gmail.com>
Copyright 2008-2018 Max Kellermann <max.kellermann@gmail.com>

Appendix A – The MPD service file

Below is the service unit for MPD which is used by the **systemctl** command to stop and start **MPD**.

File **/usr/local/lib/systemd/system/mpd.service**

```
[Unit]
Description=Music Player Daemon
Documentation=man:mpd(1) man:mpd.conf(5)
After=network.target sound.target

[Service]
Type=notify
ExecStart=/usr/local/bin/mpd --no-daemon /etc/mpd.conf

# Enable this setting to ask systemd to watch over MPD, see
# systemd.service(5).  This is disabled by default because it causes
# periodic wakeups which are unnecessary if MPD is not playing.
#WatchdogSec=120

# allow MPD to use real-time priority 50
LimitRTPRIO=50
LimitRTTIME=infinity

# disallow writing to /usr, /bin, /sbin, ...
ProtectSystem=yes

# more paranoid security settings
NoNewPrivileges=yes
ProtectKernelTunables=yes
ProtectControlGroups=yes
ProtectKernelModules=yes
# AF_NETLINK is required by libsmbclient, or it will exit() .. *sigh*
RestrictAddressFamilies=AF_INET AF_INET6 AF_UNIX AF_NETLINK
RestrictNamespaces=yes

[Install]
WantedBy=multi-user.target
Also=mpd.socket
```

If installing from the MPD Debian package the following line is added to **mpd.service** during installation just before the **ExecStart** statement. There is no need to add this line yourself.

```
ExecStartPre=-/bin/chown mpd:audio /var/run/mpd
```

Appendix B – MPD and MPC package build example output

Appendix B.1 MPD package build example

```
$ ./build.sh
Building MPD version 0.23.17
Linking output directory to MPD build directory /home/pi/mpd-
0.23.17/output
rm -rf output
ln -s /home/pi/mpd-0.23.17/output
/home/pi/mpdpackage

Copying MPD files from /home/pi/mpd-0.23.17
cp /home/pi/mpd-0.23.17/output/release/mpd .
cp /home/pi/mpd-0.23.17/mpd.svg .
cp /home/pi/mpd-0.23.17/AUTHORS .
cp /home/pi/mpd-0.23.17/COPYING .
cp /home/pi/mpd-0.23.17/NEWS .
cp /home/pi/mpd-0.23.17/README.md .
Corrected mpd.service ExecStart statement
ExecStart=/usr/local/bin/mpd --no-daemon /etc/mpd.conf
Set permissions for mpd.conf
sudo chown mpd:audio mpd.conf
sudo apt-get -y install equivs apt-file lintian
Reading package lists... Done
Building dependency tree
Reading state information... Done
apt-file is already the newest version (3.2.2).
equivs is already the newest version (2.2.0).
lintian is already the newest version (2.15.0).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

Building package mpd version 0.23.17
from input file mpdpkg
dh_testdir
dh_testroot
dh_prep
dh_testdir
dh_testroot
dh_install
dh_installdocs
dh_installchangelogs
dh_compress
dh_fixperms
dh_installdeb
dh_gencontrol
dh_md5sums
dh_builddeb
dpkg-deb: building package 'mpd' in '../mpd_0.23.17_armhf.deb'.

The package has been created.
Attention, the package has been created in the current directory,
not in ".." as indicated by the message above!
Check using Lintian y/n: n
```

When asked to check using **Lintian** answer **n** (no) as any errors displayed are beyond our control and are primarily due to the locations chosen for MPD.

Appendix B.2 MPC package build example

```
$ ./build.sh
Building MPC version 0.34
Linking output directory to MPC build directory /home/pi/mpc-0.34/output
rm -rf output
ln -s /home/pi/mpc-0.34/output
/home/pi/develop/pi/mpcpackage

Copying MPC files from /home/pi/mpc-0.34
cp /home/pi/mpc-0.34/output/mpc .
cp /home/pi/mpc-0.34/AUTHORS .
cp /home/pi/mpc-0.34/COPYING .
cp /home/pi/mpc-0.34/NEWS .
cp /home/pi/mpc-0.34/README.rst .
cp /home/pi/mpc-0.34/contrib/mpd-m3u-handler.sh .
cp /home/pi/mpc-0.34/contrib/mpd-pls-handler.sh .
cp /home/pi/mpc-0.34/contrib/mpc-completion.bash .
sudo apt-get -y install equivs apt-file lintian
Reading package lists... Done
Building dependency tree
Reading state information... Done
apt-file is already the newest version (3.2.2).
equivs is already the newest version (2.2.0).
lintian is already the newest version (2.15.0).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

Building package mpc version 0.34
from input file mpcpkg
dh_testdir
dh_testroot
dh_prep
dh_testdir
dh_testroot
dh_install
dh_installdocs
dh_installchangelogs
dh_compress
dh_fixperms
dh_installdeb
dh_gencontrol
dh_md5sums
dh_builddeb
dpkg-deb: building package 'mpc' in '../mpc_0.34_armhf.deb'.

The package has been created.
Attention, the package has been created in the current directory,
not in ".." as indicated by the message above!

Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
:
```

Appendix C – MPC commands

The following are the commands for **mpc** version 0.34 or later.

```
$ mpc help
Usage: mpc [options] <command> [<arguments>]
mpc version: 0.34

Options:
  -v, --verbose                  Give verbose output
  -q, --quiet                    Suppress status message
  --no-status                    synonym for --quiet
  -h, --host=<host>              Connect to server on <host>
  -P, --password=<password>      Connect to server using password
<password>
  -p, --port=<port>              Connect to server port <port>
  -f, --format=<format>          Print status with format <format>
  -w, --wait                     Wait for operation to finish (e.g.
database update)
  -r, --range=[<start>]:[<end>]  Operate on a range (e.g. when loading a
playlist)

Commands:
  mpc                                                  Display status
  mpc add <uri>                                        Add a song to the
queue
  mpc crop                                             Remove all but
the currently playing song
  mpc current                                          Show the
currently playing song
  mpc del <position>                                   Remove a song
from the queue
  mpc play [<position>]                                Start playing at
<position>
  mpc next                                             Play the next
song in the queue
  mpc prev                                             Play the previous
song in the queue
  mpc pause                                            Pauses the
currently playing song
  mpc pause-if-playing                                 Pauses the
currently playing song; exits with failure if not playing
  mpc toggle                                           Toggles
Play/Pause, plays if stopped
  mpc cdprev                                           Compact disk
player-like previous command
  mpc stop                                             Stop playback
  mpc seek [+-][HH:MM:SS]|<0-100>%                     Seeks to the
specified position
  mpc seekthrough [+-][HH:MM:SS]                       Seeks by an
amount of time within the song and playlist
  mpc clear                                            Clear the queue
  mpc outputs                                          Show the current
outputs
  mpc enable [only] <output # or name> [...]           Enable output(s)
  mpc disable [only] <output # or name> [...]          Disable output(s)
  mpc toggleoutput <output # or name> [...]            Toggle output(s)
  mpc outputset <output # or name> <name>=<value>      Set output
attributes
  mpc queued                                           Show the next
queued song
  mpc shuffle                                          Shuffle the queue
  mpc move <from> <to>                                 Move song in
queue
  mpc playlist [<playlist>]                            Print <playlist>
  mpc listall [<file>]                                 List all songs in
the music dir
```

```
  mpc ls [<directory>]                                    List the contents
of <directory>
  mpc lsplaylists                                         List currently
available playlists
  mpc load <file>                                         Load <file> into
the queue
  mpc insert <uri>                                        Insert a song to
the queue after the current track
  mpc prio <prio> <position/range> ...                    Change song
priorities in the queue
  mpc save <file>                                         Save a queue as
<file>
  mpc rm <file>                                           Remove a playlist
  mpc volume [+-]<num>                                    Set volume to
<num> or adjusts by [+-]<num>
  mpc repeat <on|off>                                     Toggle repeat
mode, or specify state
  mpc random <on|off>                                     Toggle random
mode, or specify state
  mpc single <on|once|off>                                Toggle single
mode, or specify state
  mpc consume <on|off>                                    Toggle consume
mode, or specify state
  mpc search <type> <query>                               Search for a song
  mpc searchadd <type> <query>                            Search songs and
add them to the queue
  mpc find <type> <query>                                 Find a song
(exact match)
  mpc findadd <type> <query>                              Find songs and
add them to the queue
  mpc searchplay <pattern>                                Find and play a
song in the queue
  mpc list <type> [<type> <query>]                        Show all tags of
<type>
  mpc crossfade [<seconds>]                               Set and display
crossfade settings
  mpc clearerror                                          Clear the current
error
  mpc mixrampdb [<dB>]                                    Set and display
mixrampdb settings
  mpc mixrampdelay [<seconds>]                            Set and display
mixrampdelay settings
  mpc update [<path>]                                     Scan music
directory for updates
  mpc rescan [<path>]                                     Rescan music
directory (including unchanged files)
  mpc sticker <uri> <get|set|list|delete|find> [args..] Sticker
management
  mpc stats                                               Display
statistics about MPD
  mpc version                                             Report version of
MPD
  mpc idle [events]                                       Idle until an
event occurs
  mpc idleloop [events]                                   Continuously idle
until an event occurs
  mpc replaygain [off|track|album]                        Set or display
the replay gain mode
  mpc channels                                            List the channels
that other clients have subscribed to.
  mpc sendmessage <channel> <message>                     Send a message to
the specified channel.
  mpc waitmessage <channel>                               Wait for at least
one message on the specified channel.
  mpc subscribe <channel>                                 Subscribe to the
specified channel and continuously receive messages.
  mpc listneighbors                                       List neighbors.
```

```
  mpc mount [<mount-path> <storage-uri>]                List mounts or
add a new mount.
  mpc unmount <mount-path>                              Remove a mount.

See man 1 mpc for more information about mpc commands and options
```

Run **man mpc** for further information.

```
$ man mpc
```