
Minecraft Bedrock Launcher Documentation

MrARM & contributors

Nov 21, 2020

Contents:

1	Getting started	1
1.1	A NOTE FOR NEW USERS	1
1.2	MacOS	1
1.3	Linux Mint	1
1.4	Ubuntu 19.04	2
1.5	Ubuntu 18.04 and 18.10	2
1.6	Ubuntu 16.04	2
1.7	Arch	2
1.8	AppImage	3
1.9	Flatpak	3
1.10	Source build	3
1.11	Getting help	3
2	Frequently Asked Questions	5
2.1	Do I need to own Minecraft: Pocket Edition to play?	5
2.2	Does it work on Windows 7 / 8?	5
2.3	How does it work?	5
2.4	Where are my worlds?	6
2.5	Can I use resource packs?	6
2.6	Why would I want to use this projects server, instead of the official one?	6
3	Troubleshooting	7
3.1	Extracting APKs	8
3.2	Appimage	8
3.3	Controller	8
3.4	Game launcher	9
3.5	Qt launcher UI	11
4	Building from sources	13
4.1	Introduction (read first)	13
4.2	Compiling MSA (for Xbox Live)	14
4.3	Compiling the game launcher	15
4.4	Compiling the Qt UI	16
4.5	Compiling the server launcher	17
5	Extract utility	19
5.1	Usage	19

5.2	Build instructions	19
6	Dedicated server	21
6.1	Server Properties	21
7	About	23



1.1 A NOTE FOR NEW USERS

This repository is currently not compatible with the latest versions of Minecraft: Bedrock Edition. However, there is an active fork of this repo that does run the latest versions. Both repositories are also undergoing a rewrite to support the new 64Bit versions of Bedrock Edition (Hooray Catalina users!). You can find links For Linux and Mac OS builds below.

Linux - <https://github.com/ChristopherHX/linux-packaging-scripts/releases>

MacOS - <https://github.com/ChristopherHX/mcpelauncher-manifest/releases>

1.2 MacOS

A prebuilt **macOS binary** is [available here](#).

If you want to compile from sources on macOS [go here](#).

1.3 Linux Mint

Please note that Mint 19 **DOES NOT** equal Ubuntu 19.04. Use the `Ubuntu 18.04` instructions for Mint 19.

1.4 Ubuntu 19.04

```
sudo dpkg --add-architecture i386
wget -O - https://mcpelauncher.mrarm.io/apt/conf/public.gpg.key | sudo apt-key add -
sudo add-apt-repository 'deb http://mcpelauncher.mrarm.io/apt/ubuntu/ disco main'
sudo apt-get install msa-daemon msa-ui-qt mcpelauncher-client mcpelauncher-ui-qt
```

You will need to install 32 bit graphics drivers - for integrated graphics and most AMD GPUs libegl1-mesa-dev:i386 will work.

1.5 Ubuntu 18.04 and 18.10

```
sudo dpkg --add-architecture i386
wget -O - https://mcpelauncher.mrarm.io/apt/conf/public.gpg.key | sudo apt-key add -
sudo add-apt-repository 'deb http://mcpelauncher.mrarm.io/apt/ubuntu/ bionic main'
sudo apt install msa-daemon msa-ui-qt mcpelauncher-client mcpelauncher-ui-qt
```

You will need to install 32 bit graphics drivers - for integrated graphics and most AMD GPUs libegl1-mesa:i386 libegl1-mesa-dev:i386 will work.

If you want to compile from sources on Ubuntu 18.04 or 18.10 [go here](#).

1.6 Ubuntu 16.04

```
sudo dpkg --add-architecture i386
sudo add-apt-repository -y ppa:beineri/opt-qt596-xenial
wget -O - https://mcpelauncher.mrarm.io/apt/conf/public.gpg.key | sudo apt-key add -
sudo add-apt-repository 'deb http://mcpelauncher.mrarm.io/apt/ubuntu/ xenial main'
sudo apt update
sudo apt install msa-daemon msa-ui-qt mcpelauncher-client mcpelauncher-ui-qt
# To launch do: . /opt/qt59/bin/qt59-env.sh && mcpelauncher-ui-qt
```

You will need to install 32 bit graphics drivers - for integrated graphics and most AMD GPUs libegl1-mesa:i386 libegl1-mesa-drivers:i386 will work.

If you want to compile from sources on Ubuntu 16.04 [go here](#).

1.7 Arch

There are **Arch AUR packages** available: mcpelauncher-msa-git mcpelauncher-msa-ui-qt-git for Xbox Live support (you need to install both), mcpelauncher-linux-git for the actual launcher and mcpelauncher-ui-git for the metalauncher (recommended). You will need to enable the [multilib repository](#). For audio support, you need to install lib32-libpulse and lib32-alsa-plugins.

So, summing it up you should install: mcpelauncher-msa-git mcpelauncher-msa-ui-qt-git mcpelauncher-linux-git mcpelauncher-ui-git lib32-libpulse lib32-alsa-plugins

1.8 AppImage

A prebuilt **AppImage binary** is [available here](#). After downloading simply open terminal, `chmod +x Minecraft_Bedrock_Launcher.AppImage` and run it.

This is generally the preferred way if your OS is not one of the ones listed above.

1.9 Flatpak

You can also install mcpelauncher via [Flatpak on Flathub](#) To install it, first [setup Flatpak](#) then run

```
sudo flatpak install flathub io.mrarm.mcpelauncher
```

To run it, run

```
flatpak run io.mrarm.mcpelauncher
```

1.10 Source build

If there are no packages available for your distribution, check out the [Source build guide](#).

You can also use the [Linux build script](#).

1.11 Getting help

We have a Discord chatroom, which you can join using the following link: <https://discord.gg/TaUNBXr>

Frequently Asked Questions

2.1 Do I need to own Minecraft: Pocket Edition to play?

Yes. To get the required x86 .apk, the Google Play account you sign in with needs to own the game on the Play Store.

The launcher **cannot** download the game from:

- The Amazon App store
- The Apple iOS App store
- by redeeming a gift code for Minecraft: Windows 10 Edition

2.2 Does it work on Windows 7 / 8?

No, due to the amount of work required to get this running on Windows. Play the Windows 10 Edition instead if you can. Note that you can not run this in the current version of [Windows Subsystem for Linux \(WSL\)](#), as it does not support 32-bit applications.

2.3 How does it work?

The project runs the native libraries from the Android version directly on your computer. This is accomplished by fixing the incompatibilities between the libc used on Android and the one used on desktop Linux or OS X (Android - Bionic; Linux - glibc). This is a simple compatibility layer which doesn't impact performance in any significant way; it's similar to Wine, but much more lightweight and simple. Additionally, all Android-specific code has been rewritten to run on Linux (AppPlatform, Store, Xbox Live, etc.).

2.4 Where are my worlds?

Linux: `~/.local/share/mcpelauncher/games/com.mojang/minecraftWorlds` Linux (Flatpak): `~/.var/app/io.mrarm.mcpelauncher/data/mcpelauncher/games/com.mojang/minecraftWorlds`

Mac OS X: `~/Library/Application Support/mcpelauncher/games/com.mojang/minecraftWorlds`

Each world has its own directory. If you have multiple worlds, you can identify them by their name in `levelname.txt` found in each world.

The `mcpelauncher-server` creates and expects its world files in `world`.

2.5 Can I use resource packs?

Yes, put them in `~/.local/share/mcpelauncher/games/com.mojang/resource_packs`. (For Flatpak) `~/.var/app/io.mrarm.mcpelauncher/data/mcpelauncher/games/com.mojang/resource_packs`

For Mac OS X, put them in `~/Library/Application Support/mcpelauncher/games/com.mojang/resource_packs`.

Shaders are also resource packs. (Shaders must be GLSL based)

You may need to extract `.zip` and `.mcpack` files into their own subdirectory for them to work properly.

2.6 Why would I want to use this projects server, instead of the official one?

If...

- the official one doesn't work for you
- you need scripting on the server
- some rarely needed particular modding capabilities

- *Extracting APKs*
 - *I downloaded a Minecraft APK from the internet and it doesn't work!*
 - *I copied the Minecraft APK from my phone over to my computer, but it doesn't extract*
 - *Can somebody give me the APK / extracted files?*
- *Appimage*
 - *Appimage crashes when I try to sign in to Google*
- *Controller*
 - *The player's view drifts by itself when a controller is plugged in*
- *Game launcher*
 - *Graphics performance issues (software rendering) - EGLUT: failed to initialize EGL display...*
 - *MSA daemon could not be found*
 - *No audio / Music*
 - *Updating the launcher*
 - *MacOS X Catalina*
 - *macOS X Mojave - This browser or app may not be secure. Try using a different browser*
 - *File picking doesn't work or crashes*
 - *I compiled and/or installed everything, but Minecraft doesn't start*
 - *I Used the Qt UI (metalauncher) to download the game, but mcpelauncher-client still doesn't work*

- *I run into lagspikes during gameplay*
- *Qt launcher UI*
 - *Running the troubleshooter*
 - *Could not find the game launcher*
 - *The launcher crashes when I press Download and Play*
 - *Clearing the launcher UI data*
 - *Obtaining the game log*

3.1 Extracting APKs

3.1.1 I downloaded a Minecraft APK from the internet and it doesnt work!

APKs acquired from anything other than the Google Play store is unsupported by us. It might be possible, but we will not help you.

3.1.2 I copied the Minecraft APK from my phone over to my computer, but it doesnt extract

Your APK is made for the ARM architecture. You need an x86 (32-bit) or x86_64 (64-bit) APK to play on your computer. If you have an ARM laptop (most chromebooks, pinebook) it should work.

3.1.3 Can somebody give me the APK / extracted files?

NO. That is Piracy and we **DO NOT** support it on this project. Anyone caught asking this in the support server will face consequences.

3.2 Appimage

3.2.1 Appimage crashes when I try to sign in to Google

The current workaround is to use LD_PRELOAD for nss3, eg. `LD_PRELOAD=/usr/lib/libnss3.so ./Minecraft_Bedrock_Launcher.AppImage` Make sure you the libnss3 package installed.

3.3 Controller

3.3.1 The player's view drifts by itself when a controller is plugged in

This happens when the controller's Deadzones are not set properly for the game. You must manually set the deadzones for your controller.

- **Linux:** https://wiki.archlinux.org/index.php/Gamepad#evdev_API_deadzones or if you have an Xbox360 Controller, open a terminal and run `sudo xboxdrv --detach-kernel-driver --deadzone 6000 --silent --type xbox360 --mimic-xpad` while you are **ingame**.

- Mac OS X: TBA

3.4 Game launcher

3.4.1 Graphics performance issues (software rendering) - EGLUT: failed to initialize EGL display...

If the game is running poorly, it might be using software rendering (identified by the `llvmpipe` string in the renderer). If you have been redirected here by a link in the launcher this is the case.

You should make sure to install the proper 32-bit graphic drivers for your hardware.

- For integrated graphics and most AMD GPUs (and maybe Nvidia on nouveau) - `sudo apt-get install libegl1-mesa:i386 libegl1-mesa-drivers:i386`
- For some Nvidia cards (eg. GeForce series), assuming proprietary driver is already installed, install the associated `libnvidia-gl-xxx` package (where `xxx` = driver version for your hardware). For example, if the `nvidia-driver-390` metapackage is installed, then `sudo apt-get install libnvidia-gl-390`.

You may need to reinstall the proprietary drivers if you had installed them manually before.

3.4.2 MSA daemon could not be found

Make sure you have properly installed the following packages: `msa-daemon msa-ui-qt` If compiling from sources, make sure to install the MSA component.

3.4.3 No audio / Music

Please note that the in-game Music is downloaded separately as a free item in the Marketplace.

- **Ubuntu:** Make sure `libasound2:i386` is installed. If it is, you may need to install `libpulse0:i386`.
- **Arch:** Make sure `lib32-libpulse` and `lib32-alsa-plugins` are installed on your system.
- **Other distros:** Make sure to install `alsa` (`asound`) and/or `pulseaudio` depending on your system configuration.

3.4.4 Updating the launcher

to update the launcher on newer versions of the launcher Press the gear icon and then check for updates

if your launcher is too old or you cant find these buttons in Ubuntu prebuilt packages do `sudo apt-get update && sudo apt-get upgrade`

and on Mac OS - Redownload the package

3.4.5 MacOS X Catalina

Currently, macOS Catalina does not support 32-bit applications at this time, so you will need to use the NG version of the launcher <https://github.com/ChristopherHX/osx-packaging-scripts/releases/tag/ng.dmg> which is not fully supported and may be behind on certain updates

Alternatively, you can:

- Downgrade to Mojave
- Find a suitable Linux Distro (anything Ubuntu-based will be the easiest)

3.4.6 macOS X Mojave - This browser or app may not be secure. Try using a different browser

Try using [ChristopherHX's fork](<https://github.com/ChristopherHX/mcpelauncher-manifest/releases>) which uses a different workaround for 1.13+ support than the standard launcher. For support, contact @ChristopherHX on the [Discord chat room](<https://discord.gg/TaUNBXr>).

3.4.7 File picking doesn't work or crashes

You need to install the `zenity` utility:

- Ubuntu - `sudo apt-get install zenity`

3.4.8 I compiled and/or installed everything, but Minecraft doesn't start

Chances are, you don't have the actual game yet. This project does **not** provide MCPE/Minecraft Bedrock Edition itself.

The easiest way to download and start the game is through the graphical Qt UI (`mcpelauncher-ui-qt`, sometimes called `metalauncher`). This requires to log into a Google Play Account with Minecraft purchased.

Otherwise, obtain a valid Minecraft x86 .apk file and use the *extract utility*.

3.4.9 I Used the Qt UI (metalauncher) to download the game, but mcpelauncher-client still doesn't work

To start a given version of Minecraft you can run `mcpelauncher-client` with the `-dg` argument, eg. `mcpelauncher-client -dg ~/.local/share/mcpelauncher/versions/DESIRED_VERSION`.

`ls ~/.local/share/mcpelauncher/versions/` will list all versions you have installed.

3.4.10 I run into lagspikes during gameplay

This can be fixed by starting the game manually via the command line. To do this, open a terminal and run the following:

Linux:

- `mcpelauncher-client -dg ~/.local/share/mcpelauncher/versions/1.2.3.4` - Replace 1.2.3.4 with the version of Minecraft you want to run.

Mac OS X:

- `cd /Applications/Minecraft\ Bedrock\ Launcher.app/Contents/MacOS`
- `./mcpelauncher-client -dg ~/Library/Application\ Support/mcpelauncher/versions/1.2.3.4` - Replace 1.2.3.4 with the version of Minecraft you want to run.

3.5 Qt launcher UI

3.5.1 Running the troubleshooter

Click the gear icon (settings) in the top right corner of the Qt launcher window, and press the [Run troubleshooter] button.

3.5.2 Could not find the game launcher

This may happen when the game launcher has not been installed or set properly. The launcher (mcpelauncher-client) must be available in the PATH variable for the launcher to work. If you are compiling from sources and do not want to install the launcher, please set the GAME_LAUNCHER_PATH at compile time with cmake (eg. `cmake -DGAME_LAUNCHER_PATH=/home/paul/mcpelauncher/build/mcpelauncher-client ..`, make sure that this is the path to the directory containing the binary, and not the binary itself). Otherwise make sure the launcher is properly installed in your system.

3.5.3 The launcher crashes when I press Download and Play

First of all, make sure a version name is displayed under the Download and Play button. If it's not, please wait some time and/or try restarting the launcher. A window asking you to accept Google Play TOS should appear first.

If the launcher still fails to download, it's possible you are trying to download a beta version of the game. Make sure to register in the beta first at <https://play.google.com/apps/testing/com.mojang.minecraftpe>.

You must have purchased Minecraft on the account you're trying to use.

In some cases, you may need to *clear the launcher data*.

3.5.4 Clearing the launcher UI data

Run the following commands:

```
rm -rf ~/.config/Minecraft\ Linux\ Launcher
rm -rf ~/.local/share/Minecraft\ Linux\ Launcher
rm -rf ~/.cache/Minecraft\ Linux\ Launcher
```

3.5.5 Obtaining the game log

In order to be able to view the game log, in the launcher press the gear in the top right corner and check the “Show log when starting the game” option. This will show a log and update it in realtime. You can copy it by pressing the icon in the top-right corner of the log window. Additionally, the log will be shown if the game crashes.

4.1 Introduction (read first)

If you want to compile from sources make sure to install the basic prerequisites from below.

4.1.1 Basic prerequisites

- **Ubuntu** - `sudo apt-get install git cmake pkg-config`
- **Fedora** - `sudo dnf install git make cmake pkg-config`

4.1.2 What to build

- You will want to build MSA for Xbox Live support (for the client). Without it the game will not support signing in to Xbox Live.
- The game launcher to run the client.
- The Qt launcher UI if you want to easily download the .apk from Google Play and/or easily manage multiple Minecraft versions.
- The extract utility if you don't want to use the Qt launcher UI.

Note: When building the server you don't need to build MSA or the game launcher - only build the server and the Qt launcher or the extract utility.

4.1.3 Updating

To update one of the components (so, msa, mcpelauncher or mcpelauncher-ui), cd to it's directory and run: `git pull && git submodule update`. Then follow the same compile instructions as when doing a normal

build, but skip the `git clone` line. You'll still need to follow the install steps again afterwards.

4.1.4 Uninstalling

If you installed using `make install` in order to uninstall do the following **as root** (escalate using `sudo su`):

```
rm /usr/local/bin/msa-daemon
rm /usr/local/bin/msa-ui-qt
rm /usr/local/bin/msa-ui-gtk
rm /usr/local/bin/mcpelauncher-client
rm -r /usr/local/share/mcpelauncher
rm /usr/local/bin/mcpelauncher-ui-qt
rm /usr/local/share/applications/mcpelauncher-ui-qt.desktop
rm /usr/local/share/pixmaps/mcpelauncher-ui-qt.png
```

Some of the commands may return errors - that's fine as some modules simply could have been not installed.

4.2 Compiling MSA (for Xbox Live)

4.2.1 Prerequisites

- **Ubuntu 18.04+** - `sudo apt-get install libssl-dev libcurl4-openssl-dev qtbase5-dev qtwebengine5-dev`
- **Ubuntu 16.04** - You must add a Qt 5.9+ repository first (`add-apt-repository ppa:beineri/opt-qt596-xenial && apt-get update`) from which you should install `apt-get install qt59base qt59webengine`; also install `apt-get install libssl-dev libcurl4-openssl-dev`
- **Fedora** - `sudo dnf install openssl-devel libcurl-devel qt5-qtbase-devel qt5-qtwebengine-devel`
- **macOS** - `brew install cmake qt`

4.2.2 Build instructions

```
git clone --recursive https://github.com/minecraft-linux/msa-manifest.git msa && cd msa
↪msa
mkdir -p build && cd build
cmake -DENABLE_MSA_QT_UI=ON ..
make -j12
```

macOS: replace the `cmake` line with `cmake -DCMAKE_PREFIX_PATH=$(brew --prefix qt) -DENABLE_MSA_QT_UI=ON ..`

4.2.3 Installation

You can now optionally install the MSA daemon system-wise. If you don't, you'll need to specify the path to MSA later (and the resulting binary will only work on your system).

- **Generic instructions** - Run `sudo make install`. Note that this doesn't make use of your system package manager, and therefore if possible, it's generally not recommended if there are better alternatives available for your system.

- **Ubuntu** - You can create a .deb file and install it using the following commands:

```

cpack --config msa-daemon/CPackConfig.cmake
sudo dpkg -i ./msa-daemon-*-Linux.deb && sudo apt-get install -f
cpack --config msa-ui-qt/CPackConfig.cmake
sudo dpkg -i ./msa-ui-qt-*-Linux.deb && sudo apt-get install -f

```

Important Note: Before continuing to the next step, make sure to go to the parent directory with `cd ../`. (make sure to return from the build directory and then from the msa directory). This generally applies to the following steps as well.

4.3 Compiling the game launcher

4.3.1 Prerequisites

- **Ubuntu** - you'll need to `sudo dpkg --add-architecture i386`, then install the required packages: `sudo apt-get install g++-multilib libpng-dev:i386 libx11-dev:i386 libxi-dev:i386 libcurl4-openssl-dev:i386 libudev-dev:i386 libevdev-dev:i386 libegl1-mesa-dev:i386 libasound2:i386`
- **Fedora** - you do not need to change the architecture, just run `sudo dnf install gcc-c++ libstdc++.i686 glibc-devel.i686 libpng-devel.i686 zlib-devel.i686 libX11-devel.i686 libXi-devel.i686 libcurl-devel.i686 systemd-devel.i686 libevdev-devel.i686 mesa-libEGL-devel.i686 alsa-lib.i686 pulseaudio-libs.i686 mesa-dri-drivers.i686 systemd-devel libXtst-devel.i686`
- **macOS** - `brew install cmake libpng`

4.3.2 Build instructions

```

git clone --recursive https://github.com/minecraft-linux/mcpelauncher-manifest.git
↪mcpelauncher && cd mcpelauncher
mkdir -p build && cd build
cmake ..
make -j12

```

Important note: Please note that you may need to replace `cmake ..` with `cmake -DMSA_DAEMON_PATH=/absolute/path/to/daemon/build/dir/msa-daemon ..` if you didn't install the MSA daemon (e.g. if you ran the previous command in `/home/paul/`, you'd have to use `/home/paul/msa/build/msa-daemon` as the path). Note the `..` is preceded by a space and is not part of the path to the daemon.

4.3.3 Installation

You can now optionally install the launcher system-wide. If you don't, you'll need to specify the path to the meta-launcher later (and the resulting binary will only work on your system).

- **Generic instructions** - Run `sudo make install`. Note that this doesn't make use of your system package manager, and therefore if possible, it's generally not recommended if there are better alternatives available for your system.
- **Ubuntu** - You can create a .deb file and install it using the following commands:

```

cpack --config mcpelauncher-client/CPackConfig.cmake
sudo dpkg -i ./mcpelauncher-client-*-Linux.deb && sudo apt-get install -f

```

4.4 Compiling the Qt UI

4.4.1 Prerequisites

- **Ubuntu 18.04+** - `sudo apt-get install libssl-dev libcurl4-openssl-dev libuv1-dev libzip-dev libprotobuf-dev protobuf-compiler qtbase5-dev qtwebengine5-dev qtdeclarative5-dev libqt5svg5-dev qml-module-qtquick2 qml-module-qtquick-layouts qml-module-qtquick-controls qml-module-qtquick-controls2 qml-module-qtquick-window2 qml-module-qtquick-dialogs qml-module-qt-labs-settings qml-module-qt-labs-folderlistmodel`
- **Ubuntu 16.04** - You must add a Qt 5.9+ repository first (add-apt-repository ppa:beineri/opt-qt596-xenial && apt-get update, you probably have done this when compiling MSA already) from which you should install `apt-get install qt59base qt59declarative qt59quickcontrols qt59quickcontrols2 qt59webengine`; also install `apt-get install libssl-dev libcurl4-openssl-dev libuv1-dev libzip-dev libprotobuf-dev protobuf-compiler`
- **Fedora** - `sudo dnf install libuv-devel libzip-devel protobuf-devel protobuf-compiler qt5-qtbase-devel qt5-qtwebengine-devel qt5-qtdeclarative-devel qt5-qtsvg-devel qt5-qtquickcontrols qt5-qtquickcontrols2`
- **macOS** - `brew install cmake qt libzip libuv protobuf`
- *the game launcher*

4.4.2 Build instructions

```
git clone --recursive https://github.com/minecraft-linux/mcpelauncher-ui-manifest.git
↪mcpelauncher-ui
cd mcpelauncher-ui && mkdir -p build && cd build
cmake ..
make -j12
```

macOS: replace the `cmake` line with `cmake -DCMAKE_PREFIX_PATH=$(brew --prefix qt) ..`

If you haven't installed the launcher system-wide, please replace `cmake ..` with `cmake -DGAME_LAUNCHER_PATH=/absolute/path/to/mcpelauncher/build/dir/mcpelauncher-client ..` (if you compiled the `mcpelauncher-manifest` in `/home/paul`, then you'd have to use `cmake -DGAME_LAUNCHER_PATH=/home/paul/mcpelauncher-manifest/build/mcpelauncher-client ..` as the command).

4.4.3 Installation

You can now optionally install the launcher system-wise.

- **Generic instructions** - Run `sudo make install`. Note that this doesn't make use of your system package manager, and therefore if possible, it's generally not recommended if there are better alternatives available for your system.
- **Ubuntu** - You can create a `.deb` file and install it using the following commands:

```
cpack --config mcpelauncher-ui-qt/CPackConfig.cmake
sudo dpkg -i ./mcpelauncher-ui-qt-*-Linux.deb && sudo apt-get install -f
```

4.5 Compiling the server launcher

4.5.1 Prerequisites

- **Ubuntu** - `sudo apt-get install git cmake pkg-config`
- **Fedora** - `sudo dnf install git make cmake pkg-config gcc-c++ libstdc++.i686 glibc-devel.i686`
- **macOS** - `brew install cmake`

4.5.2 Requirements

- **Ubuntu** - you'll need to `sudo dpkg --add-architecture i386`, then install the required packages:
`sudo apt-get install g++-multilib`
- **macOS** - none

4.5.3 Build instructions

```
git clone --recursive https://github.com/minecraft-linux/mcpelauncher-manifest.git_
↪mcpelauncher && cd mcpelauncher
mkdir -p build && cd build
cmake -DBUILD_CLIENT=OFF ..
make -j12
```

After compiling you should look at the [Dedicated server](#) page.

The extract utility is a simple utility for extracting the game from an .apk file.

The utility is not provided as a binary, because it is stand-alone and really straightforward to compile.

5.1 Usage

```
mcpelauncher-extract <source .apk> <destination dir>
```

The client binary can use any directory as the game directory, but generally the preference is to use `~/local/share/mcpelauncher/versions/1.2.3.4` as the directory for a given version, as this allows you to easily multiversion the game. This means, that if you have a file called `mcpe.apk` in the current working directory and it is of the 1.5.3.0 version of Minecraft and would simply like to extract it, you can do it as follows:

```
mkdir -p ~/.local/share/mcpelauncher/versions/ && mcpelauncher extract  
mcpe.apk ~/.local/share/mcpelauncher/versions/1.5.3.0
```

To start a given version of Minecraft you can then run the `mcpelauncher-client` with the `-dg` argument, eg. `mcpelauncher-client -dg ~/.local/share/mcpelauncher/versions/1.5.3.0`

5.2 Build instructions

5.2.1 Prerequisites

- Ubuntu - `sudo apt-get install cmake libzip-dev`
- macOS - `brew install cmake libzip`

5.2.2 Compiling

```
git clone https://github.com/minecraft-linux/mcpelauncher-extract.git
cd mcpelauncher-extract && mkdir -p build && cd build
cmake ..
make -j12
```

Dedicated server

If you want to use a specific Minecraft version you'll need to start the `mcpelauncher-server` with the `-dg` argument, pointing to the directory in which Minecraft was extracted using *the extract utility* or using the Qt UI launcher.

Note: When using the Qt UI use the following command line option to select the game version (replace 1.2.3.4 with the version you want to use):

- On Linux: `-dg ~/.local/share/mcpelauncher/versions/1.2.3.4`
 - On MacOS: `-dg ~/Library/Application\ Support/versions/1.2.3.4`
-

6.1 Server Properties

The dedicated server is configurable using a `server.properties` file with a syntax similar to the one used in the desktop Minecraft (*key=value*).

Note: The server automatically saves a default `server.properties` file the first time it is started in the data directory (which is `~/.local/share/mcpelauncher/` by default; it can be overridden using the `-dd` command line option).

6.1.1 Supported properties

- `level-dir` (recommended to set) - name of the world in `games/com.mojang/minecraftWorlds`. An empty name will cause the world to be saved directly in the root worlds directory and can cause issues with saving.
- `level-name` - name of the world
- `level-generator` - the world generator (0 - old, 1 - infinite, 2 - flat)
- `level-seed` - world seed

- `difficulty` - difficulty level (default: 0 - peaceful)
- `gamemode` - default gamemode (default 0 - survival)
- `force-gamemode` - whether the gamemode should be enforced for existing players on join (default: false)
- `motd` - Message of the Day, displayed on the server list (has to be set for the server to be shown on the LAN world list, default: empty)
- `server-port` - port (default: 19132)
- `server-port-v6` - port for IPv6 connections (default: 19133)
- `max-players` - maximal player count on the server (default: 20)
- `online-mode` - specifies whether Xbox Live login is required to join the server (default: true)
- `view-distance` - maximal supported view distance (default: 22)
- `player-idle-timeout` - timeout in minutes after which players will be disconnected. Unlike the desktop edition, this value supports real numbers; (default: 0 - no timeout)

6.1.2 Example of a custom *server.properties* file

```
level-dir=world
level-name=My World
level-generator=1
level-seed=21389432043234
difficulty=0
gamemode=1
force-gamemode=false
motd=$eMy Minecraft server
server-port=19132
server-port-v6=19133
max-players=100
online-mode=false
view-distance=22
player-idle-timeout=0
```

CHAPTER 7

About

This project allows you to launch Minecraft: Bedrock Edition (as in the edition w/o the Edition suffix, previously known as Minecraft: Pocket Edition). The launcher supports Linux and OS X.