

# Use Apple's USB SuperDrive with Linux



I'm really surprised and disappointed that Apple prevents us from using their USB SuperDrive with non Apple devices.

## How to outsmart Apple's firmware

Fortunately, with a little hack, we can awake the drive from its deep slumber. It's required to send a "magic" byte sequence after the drive was connected. I got this byte sequence from a source I no longer can find on the web. So, I don't take full credits for this.

You have several options for making this work. In this post I'd like to unveil two of them.

## Unlock with SCSI Generic (sg) driver

For communicating with the SCSI device directly we need the [Linux SCSI Generic \(sg\) driver](#) packages.

```
# Debian
sudo apt-get install sg3-utils
```

Lookup the device, it should be **sr0** or **sr1** by default depending on how many USB disc drives are currently attached. Check the output of following command to get a list off all device paths:

```
ls /dev
```

After you've the SuperDrive identified, we'll send the magic sequence to the device.

```
# Do magic...
sg_raw /dev/sr0 EA 00 00 00 00 01
```

Try to insert a disc, the drive should be awake now and start initialising the disc. For now the last step is necessary each time the drive is unplugged, so let's automate it!

## Custom udev rule

We'll make us of the udev device manager. It runs as a daemon and receives events each time a device is initialised or removed. Furthermore, it features an extensible rule set for easy customising. Please check out this very good [guide](#) for further instructions.

Let's write such a custom rule.

```
# Debian
sudo nano /etc/udev/rules.d/99-local.rules
```

Add following rule definition.

```
# Initialise Apple SuperDrive
ACTION=="add", ATTRS{idProduct}=="1500", ATTRS{idVendor}=="05ac", DRIVERS=="usb", RUN+="/usr/bin/sg_raw /dev/$kernel EA 00 00 00 00 01"
```

This will do the “magic” each time a SuperDrive device is connected. To test the rule, disconnect the drive and connect it again, the drive should be unlocked, already.

## Unlock with Superdrive-Enabler

[Superdrive-Enabler](#) is a little app that sends the magic byte sequence to a device.

### Superdrive-Enabler for Raspberry Pi

I precompiled a binary for the Raspberry Pi in cases where you can't or don't want to install “sg3-utils”. Make sure that SuperDrive is connected via an active USB hub to the Pi. Copy the executable binary to your Raspberry with SMB or WGET.

```
# download binary
wget https://github.com/onmomo/superdrive-enabler/raw/master/dist/rpi/superdriveEnabler_rpi
# make binary executable
chmod +x superdriveEnabler_rpi
```

### Other distributions

Easily compile the superdrive-enabler source.

```
# Download the source from my github repository
wget https://raw.githubusercontent.com/onmomo/superdrive-enabler/master/src/superdriveEnabler.c
# compile it
gcc -o superdriveEnabler superdriveEnabler.c
# Make it executable
chmod +x superdriveEnabler
```

### Custom udev rule

Let's write a custom rule in a new \*.rules file or separated by a line break in an existing rules file.

```
# Debian
sudo nano /etc/udev/rules.d/99-local.rules
```

Add following rule definition.

```
# Initialise Apple SuperDrive with superdrive-enabler
# Make sure that you adjust the path to the superdriveEnabler on your system
ACTION=="add", ATTRS{idProduct}=="1500", ATTRS{idVendor}=="05ac", DRIVERS=="usb", RUN+="/home/pi/superdrive-enabler/superdriveEnabler /dev/$kernel"
```

This will trigger the “superdriveEnabler” app with the device path as parameter (e.g /dev/sr0) each time a SuperDrive device is connected. Reconnect the drive again and enjoy your CD/DVD collection with XBMC or any other media player!