

NAME

MAKEDEV - create devices

SYNOPSIS

```
cd /dev; ./MAKEDEV [ -n ] [ -v ] [ update ]
cd /dev; ./MAKEDEV [ -n ] [ -v ] [ generic ] [ local ] [ group-keyword ... device ... ]
cd /dev; ./MAKEDEV [ -n ] [ -v ] [ -d ] device ...
```

DESCRIPTION

MAKEDEV is a script that will create the devices in `/dev` used to interface with drivers in the kernel.

Note that programs giving the error “ENOENT: No such file or directory” normally means that the device file is missing, whereas “ENODEV: No such device” normally means the kernel does not have the driver configured or loaded.

OPTIONS

- n** Do not actually update the devices, just print the actions that would be performed.
- d** Delete the devices. The main use for this flag is by **MAKEDEV** itself.
- v** Be verbose. Print out the actions as they are performed. This is the same output as produced by **-n**.

CUSTOMISATION

Since there is currently no standardisation in what names are used for system users and groups, it is possible that you may need to modify **MAKEDEV** to reflect your site’s settings. Near the top of the file is a mapping from device type to user, group and permissions (e.g. all CD-ROM devices are set from the `$cdrom` variable). If you wish to change the defaults, this is the section to edit.

GENERAL OPTIONS**update**

This only works on kernels which have `/proc/interrupts` (introduced during 1.1.x). This file is scanned to see what devices are currently configured into the kernel, and this is compared with the previous settings stored in the file called **DEVICES**. Devices which are new since then or have a different major number are created, and those which are no longer configured are deleted.

generic

Create a generic subset of devices. This subset consists of the standard devices, plus floppy drives, various hard drives, CD-ROM drives, pseudo-terminals, console devices, basic serial devices, busmice, audio devices, video framebuffers, printer ports, and some specialized devices. The generic subset varies somewhat according to architecture; see the `/dev/MAKEDEV` script itself for details.

local This simply runs **MAKEDEV.local**. This is a script that can create any local devices.

DEVICE GROUPS

MAKEDEV creates groups of devices when passed keywords for the group. Each listing below shows the **MAKEDEV** keyword and then lists the devices which will be created. Many devices can also be specified individually.

STANDARD DEVICES

std Creates this group of standard devices: **mem** for access to physical memory, **kmem** for access to kernel virtual memory, **null** the null device (infinite sink), **port** for access to I/O ports, **zero** the null byte source (infinite source), **core**, a symlink to `/proc/kcore` (for kernel debugging), **full** which always returns ENOSPACE on write, **random** and **urandom** random byte generators, and **tty** to access the controlling tty of a process. The loopback disk devices **loop0** through **loop7** are also created in the **std** group. These allow you to use a regular file as a block device. A filesystem image can be mounted, and used as though it were a filesystem on a partition or other block device. **loop** may also be used as a separate keyword to create the 8 loop devices. Finally, the **ram** group of

memory devices is also created by the **std** keyword (see below).

MEMORY DEVICES

ram This is the keyword used to generate the ramdisk devices **ram**{0..16} and the **ram** symlink. This group does not include **initrd**.

initrd Ramdisk which has been pre-initialized by a bootloader. **initrd** is not created in the **ram** group; it must be specifically included in the command line if you want it to be created.

cpu or microcode

Creates the CPU microcode update interface in the **cpu/** folder, with devices **microcode**, and subfolders {0..3} each containing devices **msr** and **cpuid**.

rom Creates the **rom**{0..7} **rrom**{0..7} **flaxh**{0..7} and **rflash**{0..7} flash memory card devices. **rrom** and **rflash** devices are read-only.

CONSOLE DEVICES

console

This keyword creates virtual consoles; **tty**{0..63} devices, the corresponding **vcs** and **vcsa** devices which are used to generate screen-dumps, and the **console** device itself plus appropriate symlinks. To create the console device alone, use **consoleonly**. The device **tty0** is the currently active virtual console. The **console** device serves the same function.

PSEUDO TERMINALS

pty This keyword creates the Pseudo-TTY masters **pty**{a..e,p..z} and corresponding **tty**{a..e,p..z} devices, along with **ptmx**. Each possible argument will create a bank of 16 master and slave pairs. The master pseudo-terminals are **pty**{p..s}{0..9a..f}, and the slaves are **tty**{p..s}{0..9a..f}.

SERIAL DEVICES

ttyS{0..63}

Standard serial ports. There is no group keyword, you must specify these individually. However **ttyS**{0..3} are created under the **generic** option for most architectures.

cyclades or **ttyC**

Creates Cyclades ports **ttyC**{0..31}.

digi or **ttyD**

Creates Digiboard serial card ports **ttyD**{0..15}.

stallion or **ttyE**

Creates Stallion devices **ttyE**{0..255} and **staliomem**{0..3}.

computone or **ttyF**

Creates CompuTone serial card ports **ttyH**{0..255} and special devices **ip2ipl**{0,4,8,12} and **ip2stat**{1,5,9,13}.

chase or **ttyH**

Creates Chase serial card ports **ttyH**{0..15}.

isdnmodem or **ttyI**

Creates isdn4linux virtual modem ports **ttyI**{0..63}.

isdn-tty

Also creates isdn4linux virtual modem ports **ttyI**{0..7}.

isdnbri

Creates ISDN BRI driver devices **isdn**{0..63} **isdnctrl**{0..63} **ippp**{0..63} and **isdninfo**.

isdn-io

Also creates ISDN BRI driver devices **isdn**{0..7} **isdnctrl**{0..7} **ippp**{0..7} and **isdninfo**. The **isdn-ippp** keyword can be used separately to create only the **ippp**{0..7}

devices.

- ppp** Creates a device independent PPP interface.
- dcbri** Creates Spellcaster DataComm/BRI ISDN card devices **dcbri{0..3}**.
- riscom** or **ttyL**
Creates Riscom serial card ports **ttyL{0..15}**.
- PAM** or **ttyM**
Creates PAM multimodem (or ISI serial card) ports **ttyM{0..15}**.
- ESP** or **ttyP**
Creates ESP ports **ttyP{0..4}**.
- rocketport** or **ttyR**
Creates Rocketport devices **ttyR{0..63}**.
- ttyV** Creates Control VS-1000 serial controller ports **ttyV{0..15}**.
- specialixIO8** or **ttyW**
Creates Specialix IO8+ ports **ttyW{0..15}**.
- specialix** or **ttyX**
Creates Specialix ports **ttyX{0..15}**.
- i2c** Creates **i2c{0..7}** devices for the I2C bus interface.
- tlk** Philips SAA5249 Teletext signal decoder {2.6} ports **tlk{0..3}**.

PARALLEL PORTS

- lp* Creates the standard parallel ports **lp0**, **lp1**, and **lp2** normally used for printers. These correspond to ports at 0x3bc, 0x378 and 0x278. Hence, on some machines, the first printer port may actually be **lp1**.
- par* Alternative to *lp*. The same ports are created, but are named **par{0..2}** instead of **lp{0..2}**.
- parport*
Creates raw parallel ports **parport0**, **parport1**, and **parport2**.
- slm** Creates the Atari SLM ACSI laser printer (68k/Atari) ports **slm{0..3}**.
- pg** Parallel port generic ATAPI interface (devices **pg{0..3}**).
- paride**
Parallel port IDE disk devices **pd{a..d}** with 15 partitions on each. Also creates **pcd{0..3}** and **pf{0..3}**.

OTHER BUS PORTS

- netlink** or **tap**
Creates NetLink devices **route skip fwmonitor** and **tap{0..15}** Ethertap devices. The **tapx** virtual ethernet device was designed as low level kernel support for Ethernet tunneling. Userland application can write Ethernet frame to **/dev/tapX** and the kernel will receive this frame from tapX interface. Every frame the kernel writes to a **tapX** interface can be read by a userland application from the corresponding **/dev/tapX** device.
- enskip**
ENskip kernel encryption package.
- qng** ComScire Quantum Noise Generator.
- ipsec** The Free S/WAN implementation of IPSEC.
- adb** On powerpc, creates **adb** for the Apple Data Bus and **adbmouse**. On m68k, **adb** creates the ACSI disk device **adb** and partitions **adb1** through **adb15**.

hamradio

Creates the **scc**{0..7} and **bc**{0..3} device groups.

comx Creates COMX devices **comx**{0..4}.

irda Creates IrCOMM devices (IrDA serial/parallel emulation) **ircomm0** **ircomm1** **irlpt0** and **irlpt1**.

comedi

Control and Measurement devices **comedi**{0..3}.

MOUSE DEVICES*busmice*

This keyword creates the following devices: **logibm** (Logitech bus mouse), **psaux** (PS/2-style mouse), **inportbm** (Microsoft Inport bus mouse) and **atibm** (ATI XL bus mouse) and **jbm** (J-mouse).

m68k-mice

Creates mouse devices for the m68k architecture, including: **amigamouse**, **amigamouse1**, **atarimouse** and **adbmouse**.

input

On powerpc, this keyword creates the *input* folder which groups input devices **mice**, **mouse**{0..3}, **event**{0..3}, and **js**{0..3} (joystick), and creates these devices inside.

JOYSTICK DEVICES

js Joystick. Creates **js0** and **js1**.

djs Digital joystick. Creates **djs0** and **djs1**.

USB DEVICES

usb USB is a general purpose I/O bus which can serve many purposes. The **usb** keyword creates a **usb** folder, and devices in the folder: **lp**{0..15} (printer), **mouse**{0..15} (USB connected mice), **ez**{0..15} (firmware loaders) **scanner**{0..15} (scanner interfaces), **tty-ACM**{0..15} and **ttyUSB**{0..15} (dialout devices), and **rio500** the Diamond Rio 500 device.

DISK DEVICES**fd**{0..7}

Floppy disk devices. The device **fdx** is the device which autodetects the format, and the additional devices are fixed format (whose size is indicated in the name). The other devices are named as **fdx**{*dghu*}*n*. The single letter (*d*, *q*, *h* or *u*) signifies the type of drive: 5.25 Double Density (*d*), 5.25 Quad Density (*q*), 5.25 High Density (*h*) or 3.5 (any model, *u*). The number *n* represents the capacity of that format in K. Thus the standard formats are **fdxd360**, **fdxh1200**, **fdx720**, **fdx1440**, and **fdx2880**.

For more information see Alain Knaff's *fdutils* package.

Devices **fd0*** through **fd3*** are floppy disks on the first controller, and devices **fd4*** through **fd7*** are floppy disks on the second controller.

fd{0..7}-bare

Creates just the autodetecting floppy device specified, without the fixed format devices.

hd{*a..l*}

AT (ide) hard disks. The device **hd***x* provides access to the whole disk, with the partitions being **hd***x*{**1..63**}. For i386, the four primary partitions are **hd***x***1** through **hd***x***4**, with the logical partitions being numbered from **hd***x***5** though **hd***x***20**. (A primary partition can be made into an extended partition, which can hold 4 logical partitions). Other architectures may not differentiate partition types. By default, devices for 20 logical partitions are made. The kernel supports up to 63 partitions per device.

Drives **hda** and **hdb** are the two on the primary controller **hdc** and **hdd** are the two drives on the secondary controller. These devices can also be used to access IDE

CDROMs. Additional devices **hd**{*e..l*} can be created.

xd{*a..d*}

XT hard disks. Partitions are the same as IDE disks, except only 8 partitions are created.

sd{*a..h*}

SCSI hard disks. The partitions are similar to the IDE disks, but there is a limit of 11 logical partitions **sd***x***5** through **sd***x***15**, to allow there to be 8 SCSI disks on a system (addresses 0 through 7).

sd{*i..z*}

and **sd**{*a..d*}{*a..z*} The kernel (and MAKEDEV) can handle up to 128 SCSI disks (up to **sddx**). 15 partition devices are created for each.

eda edb

MCA ESDI hard disk. Partitions are handled the same as hd.

dasd{*a..z*}

Direct Access Storage Devices for the s390 architecture. Currently only one device partition is created (for example, **dasda1**).

ada{*a..p*}

ACSI disk (68k/Atari). 15 partitions are created for each.

dac960.{*0..7*}

Mylex DAC960 PCI RAID controller. For this device, an **rd** directory is created. 32 logical devices **cx***d*{*0..31*} are created for each unit *x* specified, each with 7 partitions **cx***d*{*0..31*}**p**{*1..7*}. The **dac960** keyword will create all 7 units at once.

dpti

Adaptec I2O RAID and DPT SmartRAID V I2O controllers. Creates 7 devices for handling up to 7 controllers.

ataraid.{*0..7*}

Obsolete, device not in current devices.txt. For this device, an **ataraid** directory is created. **dx** is created for each unit *x* specified, and 15 partitions **dx***p*{*1..15*}. The **ataraid** keyword will create all 7 units at once.

i2o.hd{**a..d**}{**a..z**}

I2O based harddisk drives. Device nodes are located in the **i2o** directory. The filename is followed by a number that specifies the partition on each disk. The numbers are handled the same as hd.

ida.{*0..7*}

Compaq Intelligent Drive Array. For this device, an **ida** directory is created. 16 logical devices **cx***d*{*0..15*} are created for each unit *x* specified, each with 15 partitions **cx***d*{*0..15*}**p**{*1..15*}. The **ida** keyword will create the first three units.

cciss.{*0..7*}

Compaq Next Generation Drive Array. For this device, a **cciss** directory is created. 16 logical devices **cx***d*{*0..15*} are created for each unit *x* specified, each with 15 partitions **cx***d*{*0..15*}**p**{*1..15*}. The **cciss** keyword will create the first three units.

md

Creates Metadisk (RAID) disk array with 16 devices.

TAPE DEVICES

st{*0..7*}

SCSI tape devices. This creates the rewinding tape device **st***x* and the non-rewinding tape device **nst***x*, for each of modes 0 through 3.

qic

QIC-11, -24, -120, and -150 tapes. The devices created are **ntp****qic****11** **tp****qic****11** **ntp****qic****24** **tp****qic****24** **ntp****qic****120** **tp****qic****120** **ntp****qic****150** and **tp****qic****150** tape devices, along with **rmt****8**, **rmt****16**, **tape-d**, and **tape-reset**.

ftape Floppy driver tapes (QIC-117). There are 4 methods of access depending on the floppy tape drive. For each of access methods 0, 1, 2 and 3, the devices **qftx** **zqftx** and **rawqftx** (rewinding) and **nqftx** **nzqftx** **nrawqdtx** (non-rewinding) are created. For compatibility, devices **ftape** and **nftape** are symlinks to **qft0** and **nqft0** respectively.

ht0 Creates IDE tape devices **ht0** and **nht0**.

pt{0..3}

Creates parallel port ATAPI tape devices **pt{0..3}** and **npt{0..3}**.

CDROM DEVICES

sr or **scd** or **scd-all**

Creates **scd{0..16}** SCSI CD players and **sr{0..16}** symlinks for these devices. **cdrom** is a symlink which can be created by the user to the active CD device. It is not created by **MAKEDEV**.

pktdvd

Provides packet writing devices **pktdvd{0..3}** for CD/DVD.

pcd{0..3}

Parallel port ATAPI CD-ROM devices

sonycd Sony CDU-31a CD-ROM

mcd Mitsumi CD-ROM

mcdx Obsolete, device not in current devices.txt.

cd535 Sony CDU-535 CD-ROM

lmscd Philips LMS CM-205 CD-ROM. The newer name for this device is *cm205*, but **MAKEDEV** creates only *lmscd* at this time.

cm206cd

Philips LMS CM-206 CD-ROM

bpcd MicroSolutions BackPack parallel port CD-ROM (Obsolete - use *pcd*)

sbpcd{0..15}

Matsushita (Panasonic/SoundBlaster) CD-ROM. Units {0..3} are created with the keyword **sbpcd**.

aztcd Aztech/Orchid/Okano/Wearnes CD-ROM

gscd GoldStar CD-ROM

optcd Optics Storage CD-ROM

sjcd Sanyo CD-ROM

hitcd Hitachi CD-ROM

SCANNERS

logiscan

Logitech ScanMan32 & ScanMan 256.

m105scan

Mustek M105 Handscanner.

ac4096 A4Tek Color Handscanner.

AUDIO DEVICES

audio This creates the audio devices used by the sound driver. These include **mixer** **mixer{1..3}** (Mixer controls), **sequencer** (Audio sequencer), **dsp** **dsp{1..3}** (Digital audio), **sndstat** (Sound card status information), **audioctrl** (SPARC audio control device) and **audio** **audio{1..3}** (Sun-compatible digital audio). MIDI devices are **midi00** through **midi03**, **midi{0..3}**, **rmidi{0..3}**, **smpte{0..3}**. In addition, devices **mpu401data** and **mpu401stat** are created.

Network Devices

Linux used to have devices in /dev for controlling network devices, but that is no longer the case. To see what network devices are known by the kernel, look at /proc/net/dev.

OTHER DEVICES

Many of these devices are architecture-specific.

scc Z8530 HDLC driver (HAM radio)
bc Baycom radio modem (HAM radio)
cfs0 or *cfs* or *coda*
Coda network file system
sunmouse
Sun mouse
smouse
Simple serial mouse driver
pc110pad
IBM PC-110 digitizer pad
vrtpanel
Vr41xx embedded touch panel
vpcmouse
Connectix Virtual PC Mouse
beep Fancy beep device
modreq Kernel module load request {2.6}
watchdog
Watchdog timer port
temperature
Machine internal temperature
hwtrap Hardware fault trap
exttrp External device trap
rtc Real Time Clock
efrtc Real Time Clock
openprom
SPARC OpenBoot PROM
relay8 Berkshire Products Octal relay card
relay16
Berkshire Products ISO-16 relay card
msr x86 model-specific registers {2.6}
pciconf PCI configuration space
nvrnm Non-volatile configuration RAM
hfmodem
Soundcard shortwave modem control {2.6}
graphics
Linux/SGI graphics device
opengl Linux/SGI OpenGL pipe
gfx Linux/SGI graphics effects device

lcd Front panel LCD display

led Front panel LEDs

mergemem
Memory merge device

pmu Macintosh PowerBook power manager

isictl MultiTech ISICom serial control

ac Applicom Intl Profibus card

nwbutton
Netwinder external button

nwdebug
Netwinder debug interface

nwflash
Netwinder flash memory

userdma
User-space DMA access

smbus System Management Bus

lik Logitech Internet Keyboard

ipmo Intel Intelligent Platform Management

vmmon
VMWare virtual machine monitor

tcldrv Technology Concepts serial control

specialix_sxctl
Specialix serial control

specialix_rioctl
Specialix RIO serial control

smapi or *thinkpad*
IBM Thinkpad **smapi** device, and a symlink **thinkpad**.

srripc QNX4 API IPC manager

usemaclone
Semaphore clone device

ipmi or *ipmikcs*
Intelligent Platform Management

uctrl SPARCbook 3 microcontroller

gtrsc Gorgy Timing radio clock

cbm Serial CBM bus

jsflash JavaStation OS flash SIMM

xsvc High-speed shared-mem/semaphore service

vrbuttons
Vr41xx button input device

toshiba Toshiba laptop SMM support

perfctr Performance-monitoring counters

intel_rng Intel i8x0 random number generator

atomicps Atomic shapshot of process state data

irnet IrNET device

smbusbios SMBus BIOS

ussp_ctl User space serial port control

crash Mission Critical Linux crash dump facility

nas_xbus NAS xbus LCD/buttons access

d7s SPARC 7-segment display

zkshim Zero-Knowledge network shim control

sexec Signed executable interface

kchuid Inter-process chuid control

mptctl Message passing technology (MPT) control

button/gulpb Transmeta GULP-B buttons

compaq/cpqphpc Compaq PCI Hot Plug Controller

compaq/cpqrid Compaq Remote InsightDriver

elographics/e2201 Elographics touchscreen E271-2201

fujitsu/apanel Fujitsu/Siemens application panel

i2o/ctl I2O configuration manager

impi/bt IMPI coprocessor block transfer

impi/smic IMPI coprocessor stream interface

input/mouse Linux/SGI Irix emulation mouse

input/keyboard Linux/SGI Irix emulation keyboard

modems/mwave MWave modem firmware upload

mvista/hssdsi Montavista PICMG hot swap system driver

mvista/hasi Montavista PICMG high availability

net/tun
TAP/TUN network device

ni/natmotn
National Instruments Motion

scanners/cuecat
:CueCat barcode scanner

touchscreen/ucb1x00
UCB 1x00 touchscreen

touchscreen/mk712
MK712 touchscreen

video/em8300
EM8300 DVD decoder control

video/em8300_mv
EM8300 DVD decoder video

video/em8300_ma
EM8300 DVD decoder audio

video/em8300_sp
EM8300 DVD decoder subpicture

watchdogs/{0..3}
Watchdog devices 0 through 3

SEE ALSO

Linux Allocated Devices, maintained by H. Peter Anvin, <Peter.Anvin@linux.org>, and devices.txt in the Linux kernel source.

AUTHOR

Nick Holloway, rewritten and updated by Chris Tillman