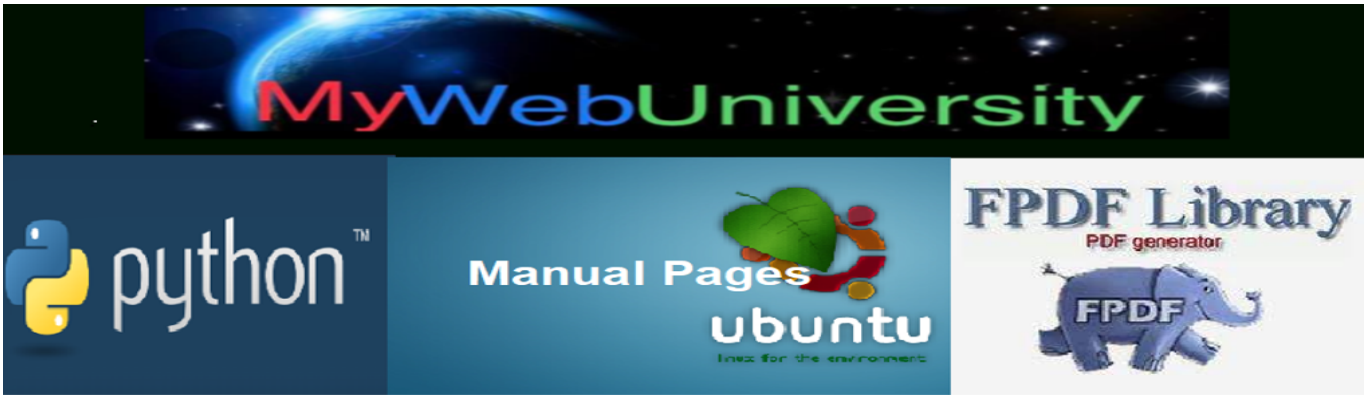

Rocky Enterprise Linux 9.2 Manual Pages on command 'sfdisk.8'

$ man sfdisk.8

SFDISK(8) System Administration SFDISK(8)

NAME

sfdisk - display or manipulate a disk partition table

SYNOPSIS

sfdisk [options] device [-N partition-number]

sfdisk [options] command

DESCRIPTION

sfdisk is a script-oriented tool for partitioning any block device. It runs in interactive mode if executed on a terminal (stdin refers to a terminal).

Since version 2.26 sfdisk supports MBR (DOS), GPT, SUN and SGI disk labels, but no longer provides any functionality for CHS (Cylinder-Head-Sector) addressing. CHS has never been important for Linux, and this addressing concept does not make any sense for new devices.

sfdisk protects the first disk sector when create a new disk label. The option --wipe always disables this protection. Note that fdisk(8) and cfdisk(8) completely erase this area by default.

sfdisk (since version 2.26) aligns the start and end of partitions to block-device I/O limits when relative sizes are specified, when the default values are used or when multiplicative suffixes (e.g., MiB) are used for sizes. It is possible that partition size will be optimized (reduced or enlarged) due to alignment if the start offset is specified exactly in sectors and partition size relative or by multiplicative suffixes.

The recommended way is not to specify start offsets at all and specify partition size in MiB, GiB (or so). In this case sfdisk aligns all partitions to block-device I/O limits (or when I/O limits are too small then to megabyte boundary to keep disk layout portable). If

this default behaviour is unwanted (usually for very small partitions) then specify offsets and sizes in sectors. In this case sfdisk entirely follows specified numbers without any optimization.

sfdisk does not create the standard system partitions for SGI and SUN disk labels like fdisk(8) does. It is necessary to explicitly create all partitions including whole-disk system partitions.

sfdisk uses BLKRRPART (reread partition table) ioctl to make sure that the device is not used by system or other tools (see also --no-reread). It?s possible that this feature or another sfdisk activity races with udevd. The recommended way how to avoid possible collisions is to use --lock option. The exclusive lock will cause udevd to skip the event handling on the device.

The sfdisk prompt is only a hint for users and a displayed partition number does not mean that the same partition table entry will be created (if -N not specified), especially for tables with gaps.

COMMANDS

The commands are mutually exclusive.

[-N partition-number] device

The default sfdisk command is to read the specification for the desired partitioning of device from standard input, and then create a partition table according to the specification. See below for the description of the input format. If standard input is a terminal, then sfdisk starts an interactive session.

If the option -N is specified, then the changes are applied to the partition addressed by partition-number. The unspecified fields of the partition are not modified.

Note that it?s possible to address an unused partition with -N. For example, an MBR always contains 4 partitions, but the number of used partitions may be smaller. In this case sfdisk follows the default values from the partition table and does not use built-in defaults for the unused partition given with -N. See also --append.

-A, --activate device [partition-number...]

Switch on the bootable flag for the specified partitions and switch off the bootable flag on all unspecified partitions. The special placeholder '-' may be used instead of the partition numbers to switch off the bootable flag on all partitions.

The activation command is supported for MBR and PMBR only. If a GPT label is detected, then sfdisk prints warning and automatically enters PMBR.

If no partition-number is specified, then list the partitions with an enabled flag.

--delete device [partition-number...]

Delete all or the specified partitions.

-d, --dump device

Dump the partitions of a device in a format that is usable as input to sfdisk. See the section BACKING UP THE PARTITION TABLE.

-g, --show-geometry [device...]

List the geometry of all or the specified devices. For backward compatibility the deprecated option --show-pt-geometry have the same meaning as this one.

-J, --json device

Dump the partitions of a device in JSON format. Note that sfdisk is not able to use JSON as input format.

-l, --list [device...]

List the partitions of all or the specified devices. This command can be used together with --verify.

-F, --list-free [device...]

List the free unpartitioned areas on all or the specified devices.

--part-attrs device partition-number [attributes]

Change the GPT partition attribute bits. If attributes is not specified, then print the current partition settings. The attributes argument is a comma- or space-delimited list of bits numbers or bit names. For example, the string "RequiredPartition,50,51" sets three bits. The currently supported attribute bits are:

Bit 0 (RequiredPartition)

If this bit is set, the partition is required for the platform to function. The creator of the partition indicates that deletion or modification of the contents can result in loss of platform features or failure for the platform to boot or operate. The system cannot function normally if this partition is removed, and it should be considered part of the hardware of the system.

Bit 1 (NoBlockIOProtocol)

EFI firmware should ignore the content of the partition and not try to read from it.

Bit 2 (LegacyBIOSBootable)

The partition may be bootable by legacy BIOS firmware.

Bits 3-47

Undefined and must be zero. Reserved for expansion by future versions of the UEFI specification.

Bits 48-63

Reserved for GUID specific use. The use of these bits will vary depending on the partition type. For example Microsoft uses bit 60 to indicate read-only, 61 for shadow copy of another partition, 62 for hidden partitions and 63 to disable automount.

--part-label device partition-number [label]

Change the GPT partition name (label). If label is not specified, then print the current partition label.

--part-type device partition-number [type]

Change the partition type. If type is not specified, then print the current partition type.

The type argument is hexadecimal for MBR, GUID for GPT, type alias (e.g. "linux") or type shortcut (e.g. 'L'). For backward compatibility the options -c and --id have the same meaning as this one.

--part-uuid device partition-number [uuid]

Change the GPT partition UUID. If uuid is not specified, then print the current partition UUID.

--disk-id device [id]

Change the disk identifier. If id is not specified, then print the current identifier.

The identifier is UUID for GPT or unsigned integer for MBR.

-r, --reorder device

Renumber the partitions, ordering them by their start offset.

-s, --show-size [device...]

List the sizes of all or the specified devices in units of 1024 byte size. This command is DEPRECATED in favour of blockdev(8).

-T, --list-types

Print all supported types for the current disk label or the label specified by --label.

-V, --verify [device...]

Test whether the partition table and partitions seem correct.

--relocate oper device

Relocate partition table header. This command is currently supported for GPT header only. The argument oper can be:

gpt-bak-std

Move GPT backup header to the standard location at the end of the device.

gpt-bak-mini

Move GPT backup header behind the last partition. Note that UEFI standard requires the backup header at the end of the device and partitioning tools can automatically relocate the header to follow the standard.

OPTIONS

-a, --append

Don?t create a new partition table, but only append the specified partitions. Note that unused partition maybe be re-used in this case although it is not the last partition in the partition table. See also -N to specify entry in the partition table.

-b, --backup

Back up the current partition table sectors before starting the partitioning. The default backup file name is ~/sfdisk-<device>-<offset>.bak; to use another name see option -O, --backup-file.

--color[=when]

Colorize the output. The optional argument when can be auto, never or always. If the when argument is omitted, it defaults to auto. The colors can be disabled; for the current built-in default see the --help output. See also the COLORS section.

-f, --force

Disable all consistency checking.

--Linux

Deprecated and ignored option. Partitioning that is compatible with Linux (and other modern operating systems) is the default.

--lock[=mode]

Use exclusive BSD lock for device or file it operates. The optional argument mode can be yes, no (or 1 and 0) or nonblock. If the mode argument is omitted, it defaults to "yes". This option overwrites environment variable $LOCK_BLOCK_DEVICE. The default is not to use any lock at all, but it?s recommended to avoid collisions with udevd or other tools.

-n, --no-act

Do everything except writing to the device.

--no-reread

Do not check through the re-read-partition-table ioctl whether the device is in use.

--no-tell-kernel

Don?t tell the kernel about partition changes. This option is recommended together with --no-reread to modify a partition on used disk. The modified partition should not be used (e.g., mounted).

-O, --backup-file path

Override the default backup file name. Note that the device name and offset are always appended to the file name.

--move-data[=path]

Move data after partition relocation, for example when moving the beginning of a partition to another place on the disk. The size of the partition has to remain the same, the new and old location may overlap. This option requires option -N in order to be processed on one specific partition only.

The optional path specifies log file name. The log file contains information about all read/write operations on the partition data. The word "@default" as a path forces sfdisk to use ~/sfdisk-<devname>.move for the log. The log is optional since v2.35.

Note that this operation is risky and not atomic. Don?t forget to backup your data!

See also --move-use-fsync.

In the example below, the first command creates a 100MiB free area before the first partition and moves the data it contains (e.g., a filesystem), the next command creates a new partition from the free space (at offset 2048), and the last command reorders partitions to match disk order (the original sdc1 will become sdc2).

echo '+100M,' | sfdisk --move-data /dev/sdc -N 1 echo '2048,' | sfdisk /dev/sdc --append sfdisk /dev/sdc --reorder

--move-use-fsync

Use the fsync(2) system call after each write when moving data to a new location by --move-data.

-o, --output list

Specify which output columns to print. Use --help to get a list of all supported columns.

The default list of columns may be extended if list is specified in the format +list (e.g., -o +UUID).

-q, --quiet

Suppress extra info messages.

-u, --unit S

Deprecated option. Only the sector unit is supported. This option is not supported when using the --show-size command.

-X, --label type

Specify the disk label type (e.g., dos, gpt, ...). If this option is not given, then sfdisk defaults to the existing label, but if there is no label on the device yet, then the type defaults to dos. The default or the current label may be overwritten by the "label: <name>" script header line. The option --label does not force sfdisk to create empty disk label (see the EMPTY DISK LABEL section below).

-Y, --label-nested type

Force editing of a nested disk label. The primary disk label has to exist already. This option allows editing for example a hybrid/protective MBR on devices with GPT.

-w, --wipe when

Wipe filesystem, RAID and partition-table signatures from the device, in order to avoid possible collisions. The argument when can be auto, never or always. When this option is not given, the default is auto, in which case signatures are wiped only when in interactive mode; except the old partition-table signatures which are always wiped before create a new partition-table if the argument when is not never. The auto mode also does not wipe the first sector (boot sector), it is necessary to use the always mode to wipe this area. In all cases detected signatures are reported by warning messages before a new partition table is created. See also the wipefs(8) command.

-W, --wipe-partitions when

Wipe filesystem, RAID and partition-table signatures from a newly created partitions, in order to avoid possible collisions. The argument when can be auto, never or always. When this option is not given, the default is auto, in which case signatures are wiped only when in interactive mode and after confirmation by user. In all cases detected signatures are reported by warning messages after a new partition is created. See also wipefs(8) command.

-v, --version

Display version information and exit.

-h, --help

Display help text and exit.

INPUT FORMATS

sfdisk supports two input formats and generic header lines.

Header lines

The optional header lines specify generic information that apply to the partition table.

The header-line format is:

<name>: <value>

The currently recognized headers are:

unit

Specify the partitioning unit. The only supported unit is sectors.

label

Specify the partition table type. For example dos or gpt.

label-id

Specify the partition table identifier. It should be a hexadecimal number (with a 0x prefix) for MBR and a UUID for GPT.

first-lba

Specify the first usable sector for GPT partitions.

last-lba

Specify the last usable sector for GPT partitions.

table-length

Specify the maximal number of GPT partitions.

grain

Specify minimal size in bytes used to calculate partitions alignment. The default is 1MiB and it?s strongly recommended to use the default. Do not modify this variable if you?re not sure.

sector-size

Specify sector size. This header is informative only and it is not used when sfdisk creates a new partition table, in this case the real device specific value is always used and sector size from the dump is ignored.

Note that it is only possible to use header lines before the first partition is specified in the input.

Unnamed-fields format

start size type bootable

where each line fills one partition descriptor.

Fields are separated by whitespace, comma or semicolon possibly followed by whitespace; initial and trailing whitespace is ignored. Numbers can be octal, decimal or hexadecimal; decimal is the default. When a field is absent, empty or specified as '-' a default value is used. But when the -N option (change a single partition) is given, the default for each field is its previous value.

The default value of start is the first non-assigned sector aligned according to device I/O limits. The default start offset for the first partition is 1 MiB. The offset may be followed by the multiplicative suffixes (KiB, MiB, GiB, TiB, PiB, EiB, ZiB and YiB) then the number is interpreted as offset in bytes.

The default value of size indicates "as much as possible"; i.e., until the next partition or end-of-device. A numerical argument is by default interpreted as a number of sectors, however if the size is followed by one of the multiplicative suffixes (KiB, MiB, GiB, TiB, PiB, EiB, ZiB and YiB) then the number is interpreted as the size of the partition in bytes and it is then aligned according to the device I/O limits. A '+' can be used instead of a number to enlarge the partition as much as possible. Note '+' is equivalent to the default behaviour for a new partition; existing partitions will be resized as required.

The partition type is given in hex for MBR (DOS) where 0x prefix is optional; a GUID string for GPT; a shortcut or an alias. It?s recommended to use two letters for MBR hex codes to avoid collision between deprecated shortcut 'E' and '0E' MBR hex code. For backward compatibility sfdisk tries to interpret type as a shortcut as a first possibility in partitioning scripts although on other places (e.g. --part-type command) it tries shortcuts as the last possibility.

Since v2.36 libfdisk supports partition type aliases as extension to shortcuts. The alias is a simple human readable word (e.g. "linux").

Since v2.37 libfdisk supports partition type names on input, ignoring the case of the characters and all non-alphanumeric and non-digit characters in the name (e.g. "Linux /usr x86" is the same as "linux usr-x86").

Supported shortcuts and aliases:

L - alias 'linux'

Linux; means 83 for MBR and 0FC63DAF-8483-4772-8E79-3D69D8477DE4 for GPT.

S - alias 'swap'

swap area; means 82 for MBR and 0657FD6D-A4AB-43C4-84E5-0933C84B4F4F for GPT

Ex - alias 'extended'

MBR extended partition; means 05 for MBR. The original shortcut 'E' is deprecated due to collision with 0x0E MBR partition type.

H - alias 'home'

home partition; means 933AC7E1-2EB4-4F13-B844-0E14E2AEF915 for GPT

U - alias 'uefi'

EFI System partition, means EF for MBR and C12A7328-F81F-11D2-BA4B-00A0C93EC93B for GPT

R - alias 'raid'

Linux RAID; means FD for MBR and A19D880F-05FC-4D3B-A006-743F0F84911E for GPT

V - alias 'lvm'

LVM; means 8E for MBR and E6D6D379-F507-44C2-A23C-238F2A3DF928 for GPT

The default type value is linux.

The shortcut 'X' for Linux extended partition (85) is deprecated in favour of 'Ex'.

bootable is specified as [*|-], with as default not-bootable. The value of this field is irrelevant for Linux - when Linux runs it has been booted already - but it might play a role for certain boot loaders and for other operating systems.

Named-fields format

This format is more readable, robust, extensible and allows specifying additional information (e.g., a UUID). It is recommended to use this format to keep your scripts more readable.

[device :] name[=value], ...

The device field is optional. sfdisk extracts the partition number from the device name. It allows specifying the partitions in random order. This functionality is mostly used by --dump. Don?t use it if you are not sure.

The value can be between quotation marks (e.g., name="This is partition name"). The currently supported fields are:

start=number

The first non-assigned sector aligned according to device I/O limits. The default start offset for the first partition is 1 MiB. The offset may be followed by the multiplicative suffixes (KiB, MiB, GiB, TiB, PiB, EiB, ZiB and YiB) then the number is

interpreted as offset in bytes.

size=number

Specify the partition size in sectors. The number may be followed by the multiplicative suffixes (KiB, MiB, GiB, TiB, PiB, EiB, ZiB and YiB), then it?s interpreted as size in bytes and the size is aligned according to device I/O limits.

bootable

Mark the partition as bootable.

attrs=string

Partition attributes, usually GPT partition attribute bits. See --part-attrs for more details about the GPT-bits string format.

uuid=string

GPT partition UUID.

name=string

GPT partition name.

type=code

A hexadecimal number (without 0x) for an MBR partition, a GUID for a GPT partition, a shortcut as for unnamed-fields format or a type name (e.g. type="Linux /usr (x86)"). See above the section about the unnamed-fields format for more details. For backward compatibility the Id= field has the same meaning.

EMPTY DISK LABEL

sfdisk does not create partition table without partitions by default. The lines with partitions are expected in the script by default. The empty partition table has to be explicitly requested by "label: <name>" script header line without any partitions lines. For example:

```
echo 'label: gpt' | sfdisk /dev/sdb
```

creates empty GPT partition table. Note that the --append disables this feature.

BACKING UP THE PARTITION TABLE

It is recommended to save the layout of your devices. sfdisk supports two ways.

Use the --dump option to save a description of the device layout to a text file. The dump format is suitable for later sfdisk input. For example:

```
sfdisk --dump /dev/sda > sda.dump
```

This can later be restored by:

```
sfdisk /dev/sda < sda.dump
```

If you want to do a full (binary) backup of all sectors where the partition table is stored, then use the --backup option. It writes the sectors to ~/sfdisk-<device>-<offset>.bak files. The default name of the backup file can be changed with the --backup-file option. The backup files contain only raw data from the device. Note that the same concept of backup files is used by wipefs(8). For example:

```
sfdisk --backup /dev/sda
```

The GPT header can later be restored by:

```
dd  if=~/sfdisk-sda-0x00000200.bak  of=/dev/sda  \ seek=$0x00000200  bs=1
conv=notrunc
```

Note that sfdisk since version 2.26 no longer provides the -I option to restore sectors. dd(1) provides all necessary functionality.

COLORS

Implicit coloring can be disabled by an empty file /etc/terminal-colors.d/sfdisk.disable. See terminal-colors.d(5) for more details about colorization configuration. The logical color names supported by sfdisk are:

header
: The header of the output tables.

warn
: The warning messages.

welcome
: The welcome message.

ENVIRONMENT

SFDISK_DEBUG=all
: enables sfdisk debug output.

LIBFDISK_DEBUG=all
: enables libfdisk debug output.

LIBBLKID_DEBUG=all
: enables libblkid debug output.

LIBSMARTCOLS_DEBUG=all
: enables libsmartcols debug output.

LOCK_BLOCK_DEVICE=<mode>
: use exclusive BSD lock. The mode is "1" or "0". See --lock for more details.

NOTES

Since version 2.26 sfdisk no longer provides the -R or --re-read option to force the kernel to reread the partition table. Use blockdev --rereadpt instead.

Since version 2.26 sfdisk does not provide the --DOS, --IBM, --DOS-extended, --unhide, --show-extended, --cylinders, --heads, --sectors, --inside-outer, --not-inside-outer options.

AUTHORS

Karel Zak <kzak@redhat.com>

The current sfdisk implementation is based on the original sfdisk from Andries E. Brouwer.

SEE ALSO

fdisk(8), cfdisk(8), parted(8), partprobe(8), partx(8)

REPORTING BUGS

For bug reports, use the issue tracker at https://github.com/karelzak/util-linux/issues.

AVAILABILITY

The sfdisk command is part of the util-linux package which can be downloaded from Linux Kernel Archive <https://www.kernel.org/pub/linux/utils/util-linux/>.