

<http://arobaseinformatique.eklablog.com/afficher-l-espace-disque-avec-pydf-a117873342>

```
# Configuration file for pydf
#
#
# colours can be:
# 'none' - no change from previous colour
# 'default' - default system colour
#
# special attributes:
# 'bold'
# 'underline'
# 'blink'
# 'reverse'
# 'concealed'
#
# foreground:
# 'black'
# 'red'
# 'green'
# 'yellow'
# 'blue'
# 'magenta'
# 'cyan'
# 'white'
#
# background:
# 'on_black'
# 'on_red'
# 'on_green'
# 'on_yellow'
# 'on_blue'
# 'on_magenta'
# 'on_cyan'
# 'on_white'
#
# beep:
# 'beep'
#
#
# or any combination of these, separated with commas

# normal text colour - used to switch to after one row is displayed
```

```
normal_colour = 'default'

# colour of the header
header_colour = 'green'

# colour for local filesystems
local_fs_colour = 'cyan'

# colour for remote filesystems (such as nfs, samba, afs....)
remote_fs_colour = 'green'

# colour for special filesystems (such as proc, pty)
special_fs_colour = 'blue'

# colour for readonly mounted filesystems
readonly_fs_colour = 'cyan'

# colour for filesystems with usage > FILL_THRESH
filled_fs_colour = 'red'

# colour for filesystems with usage > FULL_THRESH
full_fs_colour = 'on_red', 'green', 'blink'

# default format for displaying sizes "-h" or "-H" or "-m" or "-k" or "--blocks"
sizeformat = "-m"

# string used to separate columns in the table
column_separator = ''

# colour of the string
column_separator_colour = 'none'

# if the screen is wider than necessary, stretch the bar:
# 0 - do not stretch
# 1 - stretch to fill the whole screen
# real number in between - stretch by this ratio of free space
stretch_screen = 0.3

# filesystem filled up over this limit (in percents) is displayed
# with filled_fs_colour (to show it is dangerously filled up)
FILL_THRESH = 95.0

# filesystem filled up over this limit is displayed with
# full_fs_colour (to show it is FULL)
FULL_THRESH = 99.0
```

```
# Format used to display information: (keyword, size, justify).
# keyword - one of 'fs' 'fstype' 'size' 'used' 'avail' 'perc' 'bar' 'on'.
# size - if 'size' is integer, it is a minimal possible column width of the entry
#       if 'size' is float, it is a minimal column width in percent of screen width
#
# justify is either "l" for left justify, "r" for right justify or "c" for
# center.
# You can use any order and any combination of keywords, but
# be careful not to exceed the size of your screen
```

```
#format = [
#     ('fs', 15, "l"), ('size', 9, "r"),
#     ('used', 9, "r"), ('avail', 9, "r"), ('perc', 5, "r"),
#     ('bar', 8, "l"), ('on', 16, "l")
# ]
```

```
# this is somewhat conservative
# use fixed width for everything, since you want it readable
# only the bar is specified by percentage, because you want it dynamic
```

```
format = [
    ('fs', 10, "l"), ('size', 5, "r"),
    ('used', 5, "r"), ('avail', 5, "r"), ('perc', 5, "r"),
    ('bar', 0.1, "l"), ('on', 11, "l")
]
```

```
# character to display filesystem size bar
barchar = '#'
```

```
# fill the empty space of the size bar with this character
bar_fillchar = '.'
```

```
# hide 'mount --bind' binds?
hidebinds = True
```

```
# list of files to try to get mount information from
# on normal linux systems only /etc/mstab or /proc/mounts make sense
mountfile = ['/etc/mstab', '/etc/mnttab', '/proc/mounts']
```