

## An AutoFS executable map to automount device nodes

For my company's hard disk-based backup system I needed the ability to automount disk drives by their device name into a standard directory structure.

One possible approach would be to add some lines like these to fstab:

```
/dev/sda1 /mnt/sda1 ext3 defaults,noauto 0 0
```

This may be good enough in some cases, though it wasn't sufficient for me, when there were dozens of device nodes which could get mounted eventually.

So I basically wanted something that would allow me to just access a directory, while the underlying disk was mounted automatically, then having it unmounted automatically if not in use, but still being dynamic in it's nature so it would auto-adjust.

Now there's a simple trick using an AutoFS feature called "executable maps", which would allow me to achieve this all.

The idea is, that all devices (let's say /dev/sda1, /dev/sda2, /dev/sdb1, /dev/sdc1 as an example) will get mounted to /mnt/disks/[devicename].

First make sure, that AutoFS is installed. On Debian for example, it is installed like this:

```
apt-get install autofs
```

Then create a file called /etc/auto.disks with the following lines therein:

```
#!/bin/bash
```

```
# $1 is passed-over from automount
```

```
# key refers to the mount point we are looking for  
key="$1"
```

```
# default mount options
```

```
opts="-fstype=ext3,rw"
```

```
# if a block device exists at /dev/[key]
```

```
# pass it back to automount
```

```
[ -b /dev/${key} ] && { echo "$opts "; echo -e "t:/dev/${key}"; }
```

```
Don't forget to chmod 755 /etc/auto.disks.
```

This script will create an automounter map dynamically as soon as it passed a device node. If it finds it (e.g. while looking up /dev/sda1, which exists), it'll return the map to automount, which will cause the device node to be mounted.

In my case, the script didn't need to be very sophisticated as I only have ext3-formatted disks, but it's easy to script it for automatic file system recognition.

Btw, the script can be tested like this to see if it's actually working:

```
satyr:~# bash /etc/auto.disks sda1  
-fstype=ext3,rw  
:/dev/sda1
```

```
satyr:~# bash /etc/auto.disks sdx1
```

The first command returns the map for an existing device node `/dev/sda1`, while the second command returns nothing as `/dev/sdx1` doesn't exist on the system.

Now set AutoFS to use the executable map for `/mnt/disks` directory. Add this line to `/etc/auto.master`:

```
/mnt/disks /etc/auto.disks --timeout=360
```

This will cause AutoFS to examine the executable map on all requested sub directories beneath `/mnt/disks`. So if you're going to access `/mnt/disks/sda1`, `/mnt/disks/sda2`, `/mnt/disks/sdb1`, `/mnt/disks/sdc1`, the block devices corresponding to the directories are mounted automatically -- as long as the devices exist of course.

The timeout value designates after how much time (of inactivity) an automounted file system expires and get's unmounted.