

This topic from the old SourceForge.net forum was donated from a personal archive and has been edited to improve readability. If you do not want your posts reproduced herein, please notify one of the maintainers.

[USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by danmero » Mon May 04, 2009 12:02 am

The main idea is to auto-backup your important data on USB(encrypted).

1. Prepare the disk

- o Create a key file

Code: [Select all](#)

```
$ dd if=/dev/urandom of=/temp/ukey.key count=1024
```

- o Encrypt the disk from CLI(shell)

Code: [Select all](#)

```
$ geli init -P -s 4096 -K /temp/ukey.key /dev/da0
```

- o For safety backup meta data

Code: [Select all](#)

```
$ geli backup da0 /temp/usb.key.backup
```

- o Attach the disk to create the filesystem

Code: [Select all](#)

```
$ geli attach -p -k /temp/ukey.key /dev/da0
```

- o Format the disk(UFS soft-updates)

Code: [Select all](#)

```
$ newfs -U -L Travel -b 4096 -o space /dev/da0.eli
Reduced frags per cylinder group from 13697 to 13696 to enlarge last cyl group
/dev/da0.eli: 1926.9MB (3946360 sectors) block size 4096, fragment size 4096
        using 37 cylinder groups of 53.50MB, 13696 blks, 3424 inodes.          with soft updates
super-block backups (for fsck -b #) at:
144, 109712, 219280, 328848, 438416, 547984, 657552, 767120, 876688, 986256, 1095824, 1205392, 1314960, 1424528, 1534096, 1643664,
1753232, 1862800, 1972368, 2081936, 2191504, 2301072, 2410640, 2520208, 2629776, 2739344, 2848912, 2958480, 3068048, 3177616, 3287184,
3396752, 3506320, 3615888, 3725456, 3835024, 3944592
```

2. Modify /etc/devd.conf and reload

- o Add new directive to devd.conf just to find information's about your device(USB disk).

Code: [Select all](#)

```
attach 0 {
    device-name "umass[0-9]+";
    action "sleep 2; logger vendor:$vendor,product:$product,release:$release,device-name:$device-name";
};
```

Plug the USB and check your system log for information's

- o Next add a second directive(you can keep the first one) to act on match, call the script /temp/secret.sh in my example(see below).

Code: [Select all](#)

```
attach 5 {
    match "vendor" "0x0930";
    match "product" "0x6544";
    match "release" "0x0100";
    action "sleep 1; /temp/secret.sh $device-name";
};
```

- o Save the configuration file (devd.conf) and restart devd service

Code: [Select all](#)

```
/etc/rc.d/devd restart
```

3. Create the main script (here is my sample script).

Code: [Select all](#)

```
#!/bin/bash
key_file=/temp/ukey.key
device=da${1//[a-z]/}
origin=/temp/scripts
target=/mnt/secret

_start() {
  if [ ! -f /var/run/${0##*/}.pid ];then
    echo $$ > /var/run/${0##*/}.pid
  else
    /usr/bin/logger "Error syncing $origin on $device";
    echo "A16P16A16P16A16P16CP8CP8CP8A16P16A16P16A16" > /dev/speaker
    _finish;
  fi
}
_finish() {
  if [ -f /var/run/${0##*/}.pid ];then
    rm /var/run/${0##*/}.pid;
  fi
  exit
}
_mount() {
  /sbin/geli attach -p -k ${key_file} /dev/$device
  /bin/mkdir -p ${target} &&
  /sbin/mount /dev/${device}.eli ${target}/ &&
  /sbin/fsck_ufs -pB /mnt/${target};
}
_umount() {
  /sbin/umount /dev/${device}.eli;
  /bin/rm -r /mnt/${target};
  /bin/sync /dev/${device}.eli
  /sbin/geli detach /dev/${device};
  /bin/sync /dev/${device}
}
_sync_func() {
  /usr/local/bin/rsync -a --log-file=${origin}/backup.log ${origin}/ ${target}/
  /usr/bin/logger "Finish syncing $origin on $device";
}
_start; _mount; _sync_func; _umount;
while [ -e /dev/${device} ];do echo "A8P8A8P8A8" > /dev/speaker;sleep 10;done
_finish;
```

4. Insert again the USB disk and the content of origin directory will be back-up(rsync) on target directory on USB stick/disk.

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

[by nadsinoz](#) » Thu Aug 20, 2009 8:38 am

I am running through the steps to backup to portable usb drive.

Should the second from last line of the script read:

Code: [Select all](#)

```
while [ -e /dev/${device} ];do echo "A8P8A8P8A8" > /dev/speaker;sleep 10;done
```

instead of:

Code: [Select all](#)

```
while [ -e /dev/da0 ];do echo "A8P8A8P8A8" > /dev/speaker;sleep 10;done
```

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

[by danmero](#) » Thu Aug 20, 2009 10:47 am

*nadsinoz wrote:*Should the second from last line of the script read:

Code: [Select all](#)

```
while [ -e /dev/${device} ];do echo "A8P8A8P8A8" > /dev/speaker;sleep 10;done
```

That's correct 😊 , I was working on da0.

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

[by smartfish65](#) » Sat Jan 16, 2010 6:37 pm

Freenas - Version 0.7.1 Shere (revision 4944) - Platform i386-embedded on Intel(R) Pentium(R) M processor 1.80GHz

This is my first post to this forum. So a friendly Hello to everyone.

I would like to use the script from danmero because what it offers is exactly what I need. I adjusted the script to my system and it's located on a shared Raid 5 drive now. To be sure there will be no access problem the whole path to the script is xr for everyone. The script itself too.

When I execute the script as user root from the shell it works perfect!

The only problem is that the "geli attach" command isn't executed when invoked by the devd. No error message, the following line simply does nothing:

Code: [Select all](#)
`/sbin/geli attach -p -k ${key_file} /dev/${device} 2>&1 >> ${origin}/backup.log`

I echoed the line into the log and executed it to check all variables and this work perfectly. So the only problem I can imagine is some missing rights.

But reading the first post of danmero I think the above script is designed to be called from devd so I hope there is a solution ... ? I'm a little bit helpless now.

Martin

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

[by danmero](#) » Sun Jan 17, 2010 5:03 am

*smartfish65 wrote:*When I execute the script as user root from the shell it works perfect!

Maybe you have a cron/path problem 😞

smartfish65 wrote:

Code: [Select all](#)
`/sbin/geli attach -p -k ${key_file} /dev/${device} 2>&1 >> ${origin}/backup.log`

Add -x after shebang and redirect the output to a "log" file.

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

[by smartfish65](#) » Sun Jan 17, 2010 9:19 pm

Good hint! I've never used that -x option before 😊

I got the message:

Code: [Select all](#)
`geli: Cannot open /dev/dal: No such file or directory.`

Wow, it was a timing problem! My USB drive needs some time to spin up.

When I start the script manually I have to walk to my laptop in another room. Enough time for the usb drive so spin up. That's why the script did the job without errors when started manually.

I removed the sleep 2 in devd.conf and put a sleep 10 in the first line of

the `_mount()` function. So I get an immediate beep and have a longer wait period. Great, everything is perfect!

Thank you very much for the script and your help 😊

Martin

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

[by danmero](#) » Sun Jan 17, 2010 10:27 pm

smartfish65 wrote:

Code: [Select all](#)

```
geli: Cannot open /dev/dal: No such file or directory.
```

Wow, it was a timing problem! My USB drive needs some time to spin up.

Try to sleep more here 😊

Code: [Select all](#)

```
attach 5 {
  match "vendor" "0x0930";
  match "product" "0x6544";
  match "release" "0x0100";
  action "sleep 1; /temp/secret.sh $device-name";
};
```

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

[by smartfish65](#) » Sun Jan 17, 2010 10:56 pm

Is there any advantage having the sleep in `devd.conf`?

I put a "sleep 10" in the `_mount()` function of the `secret.sh` script because I like the immediate sound feedback when I plugin the drive.

Works perfect now. Many thanks!

Martin

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

[by danmero](#) » Sun Jan 17, 2010 11:56 pm

smartfish65 wrote: Is there any advantage having the sleep in `devd.conf`?

I put a "sleep 10" in the `_mount()` function of the `secret.sh` script because I like the immediate sound feedback when I plugin the drive.

I don't know how fast your system detect the new device, however if works for you, why not 😊

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

[by c0mputerking](#) » Sun Feb 28, 2010 2:05 am

Hello I am having issues getting all this to work first the strange thing is I already had the USB drive in use as a freenas disk I followed the instruction including the `newfs` part, but i still have stuff on the USB drive. Do i need to clear it somehow first i have removed the drive and the mount point from the GUI, but it still seems to get mounted to `/mnt/usbackup` directory with contents intact. It is not in `/etc/fstab`, it is not listed with the mount command, where is it coming from? why did the `newfs` command not erase this data? is this why the script is not working?

```
classicgiga:/mnt/usbackup# ls
.snap Bo Didly King Todd's ~JACKASS
```

1-Backup - CDs Classic Files RV LOGOS WHAJUSHAE

```
classicgiga:/mnt/usbackup# mount
/dev/ad0s1a on / (ufs, local, soft-updates)
devfs on /dev (devfs, local)
procfs on /proc (procfs, local)
/dev/ad0s2 on /mnt/ospart (ufs, local, soft-updates, acls)
/dev/mirror/ClassicMirrorsp1 on /mnt/ClassicStorage (ufs, local, soft-updates, acls)
```

Anyways i have added the info to the /etc/devd.conf as instructed but the second part confuses me as mine looks like this? i gathered this info from dmesg not as nothing seemed to show up in the system log as explained in your intructions

```
# For exact USB drive and backup
attach 5 {
match "vendor" "Sunplus Technology Inc. USB to Serial-ATA bridge";
match "product" "class 0/0";
match "release" "rev 2.00/1.03";
action "sleep 1; /temp/secret.sh $device-name";
};
```

Apon running the script i got various errors made some changes to the variables like this and the script seems to run but hangs forever and i have to hit ctr C see below

```
#!/bin/bash
key_file=/temp/ukey.key
#device=da${1//[a-z]/}
device=da0
origin=/temp/scripts
target=secret
```

```
classicgiga:/mnt# /temp/scripts/back2usb.sh
/mnt/secret: FILE SYSTEM CLEAN; SKIPPING CHECKS
/mnt/secret: clean, 240303827 free (0 frags, 240303827 blocks, 0.0% fragmentation)
```

I get this in the backup.log

```
2010/02/28 08:57:20 [2344] building file list
2010/02/28 08:57:20 [2344] >f.st..... backup.log
2010/02/28 08:57:20 [2344] sent 3518 bytes received 33 bytes total size 4507
2010/02/28 08:57:56 [2366] building file list
2010/02/28 08:57:56 [2366] >f.st..... backup.log
2010/02/28 08:57:56 [2366] sent 3693 bytes received 33 bytes total size 4682
```

dmesg gives this

```
GEOM_ELI: Device da0.eli created.
GEOM_ELI: Encryption: AES-CBC 128
GEOM_ELI: Crypto: software
GEOM_LABEL: Label for provider da0.eli is ufsid/4b89b44f43007141.
GEOM_LABEL: Label for provider da0.eli is ufs/Travel.
GEOM_LABEL: Label ufs/Travel removed.
GEOM_LABEL: Label ufsid/4b89b44f43007141 removed.
GEOM_LABEL: Label for provider da0.eli is ufsid/4b89b44f43007141.
GEOM_LABEL: Label for provider da0.eli is ufs/Travel.
GEOM_ELI: Device da0.eli destroyed.
GEOM_LABEL: Label ufs/Travel removed.
GEOM_LABEL: Label ufsid/4b89b44f43007141 removed.
GEOM: da0: the secondary GPT table is corrupt or invalid.
GEOM: da0: using the primary only -- recovery suggested.
```

PS if i run the script again it gives no error but i still cannot seem to find my backup data anywhere nothing in the system.log for any of this

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [niaxilin](#) » Thu Mar 25, 2010 2:12 pm

danmero Thanks for your mini tutorial! You really helped us get our FreeNAS backups going.

I want to post our changes for reference. Instead of mounting/copying on USB connection, our drive stays connected (weekly rotation) and mounts/copies at night.

We placed our keys and scripts in /mnt/WD1TB/admin/ so they would be accessible after a reboot.

Instead of modifying /etc/devd.conf, we added /bin/sh /mnt/WD1TB/admin/usb-backup-da0.sh into FreeNAS's System > Advanced > Cron scripts.

Code: [Select all](#)

```
[admin@nas /mnt/WD1TB/admin]$ ls -l
-rwx----- 1 admin admin 131072 Mar 23 13:03 da0.key
-rwx----- 1 admin admin 512 Mar 23 13:04 da0.key.backup
-rwx----- 1 admin admin 1329 Mar 24 16:20 usb-backup-da0.sh
```

The backup script below attaches the encrypted drive, mounts and fschk (like *danmero's* script). The sync function does two things: 1) rsync's all /mnt/WD1TB/backup/ files. 2) Inserts a new rsync of /mnt/WD1TB/share/ into the beginning of five incremental (daily) "share" backups.

Using cp -al and then rsync below, we first make hard links to yesterday's files in /share.2/, then we overwrite--breaking the hard link--any changed or deleted files in /share.1/. I've used this system a lot and it works great. You get days of incremental backups, but only use the space equivalent to one share + the files that were changed. Hope this helps!

Code: [Select all](#)

```
#!/bin/bash
device=da0
origin=/mnt/WD1TB
scriptdir=/mnt/WD1TB/admin

dev=/dev/${device}
mnt=/mnt/${device}
key_file=${scriptdir}/${device}.key
log=${scriptdir}/usb-backup-${device}.log
lock=${scriptdir}/usb-${device}.pid

_start() {
    if [ ! -f ${lock} ];then
        echo $$ > ${lock}
    else
        /bin/echo "Error: USB already being processed?";
        /usr/bin/logger "Error backing up $device.";
        exit;
    fi
}

_mount() {
    /sbin/geli attach -p -k ${key_file} ${dev}
    /bin/mkdir -p ${mnt} &&
    /sbin/mount ${dev}.eli ${mnt} &&
    /sbin/fsck_ufs -pB ${mnt};
}

_sync_func() {
    /usr/local/bin/rsync -a --log-file=${log} ${origin}/backup/ ${mnt}/backup/
    touch ${mnt}/backup/

    rm -rf ${mnt}/share.5
    mv ${mnt}/share.4 ${mnt}/share.5
    mv ${mnt}/share.3 ${mnt}/share.4
    mv ${mnt}/share.2 ${mnt}/share.3
    cp -al ${mnt}/share.1 ${mnt}/share.2
    rsync -rtv --delete ${origin}/share/ ${mnt}/share.1/
    touch ${mnt}/share.1/

    /usr/bin/logger "Finish syncing $origin to $device.";
}

_umount() {
    /sbin/umount ${dev}.eli;
    /bin/rmdir ${mnt};
    /bin/sync ${dev}.eli
    /sbin/geli detach ${dev};
    /bin/sync ${dev}
}

_finish() {
    if [ -f ${lock} ];then
        rm ${lock};
    fi
    exit
}

_start;
```

```
_mount;  
_sync_func;  
_umount;  
_finish;
```

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [bubulein](#) » Sat Apr 17, 2010 3:54 am

Hi,
im running 0.7.1 Shere (revision 5127)
Just a little note related to:
/etc/devd.conf

Code: [Select all](#)

```
attach 0 {  
    device-name    "umass[0-9]+";  
    action "sleep 2; logger vendor:$vendor,product:$product,release:$release,device-name:$device-name";  
};
```

In my version of FreeNAS attach 0 is already in use, so i have to replace attach 0 with attach 1. Just as a note if some has the same issue.

Also it should look like

Code: [Select all](#)

```
attach 1 {  
    device-name    "umass[0-9]+";  
    action "sleep 2; logger vendor:$vendor,product:$product,release:$release,device-name:$device-name";  
};
```

Very good script - THX
bubulein

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [bigalnz](#) » Tue Dec 14, 2010 6:33 am

I wanted, as a newbie to FreeNAS to make a few comments about this script, what I have done and what issues I still have.

1. I plugged in my USB Disk before step one. Not sure if this was right or not?

*danmero wrote:*The main idea is to auto-backup your important data on USB(encrypted).

1. Prepare the disk
 - o Create a key file

Code: [Select all](#)

```
$ dd if=/dev/urandom of=/temp/ukey.key count=1024
```

I didnt have a /temp/ directory only a /tmp directory, so I had to mkdir temp from root directory.

*danmero wrote:*Encrypt the disk from CLI(shell)

Code: [Select all](#)

```
$ geli init -P -s 4096 -K /temp/ukey.key /dev/da0
```

- For safety backup meta data

Code: [Select all](#)

```
$ geli backup da0 /temp/usb.key.backup
```

- Attach the disk to create the filesystem

Code: [Select all](#)

```
$ geli attach -p -k /temp/ukey.key /dev/da0
```

- Format the disk(UFS soft-updates)

Code: [Select all](#)

```
$ newfs -U -L Travel -b 4096 -o space /dev/da0.eli
Reduced frags per cylinder group from 13697 to 13696 to enlarge last cyl group
/dev/da0.eli: 1926.9MB (3946360 sectors) block size 4096, fragment size 4096
      using 37 cylinder groups of 53.50MB, 13696 blks, 3424 inodes.          with soft updates
super-block backups (for fsck -b #) at:
144, 109712, 219280, 328848, 438416, 547984, 657552, 767120, 876688, 986256, 1095824, 1205392, 1314960, 1424528, 1534096, 164366
1753232, 1862800, 1972368, 2081936, 2191504, 2301072, 2410640, 2520208, 2629776, 2739344, 2848912, 2958480, 3068048, 3177616, 32
3396752, 3506320, 3615888, 3725456, 3835024, 3944592
```

I got a bunch of errors thrown up here to do with the superblocks. No longer on my screen or I would post.

danmero wrote:[list][*] Add new directive to devd.conf just to find information's about your device(USB disk).

Code: [Select all](#)

```
attach 0 {
    device-name    "umass[0-9]+";
    action "sleep 2; logger vendor:$vendor,product:$product,release:$release,device-name:$device-name";
};
```

Plug the USB and check your system log for information's

[*] Next add a second directive(you can keep the first one) to act on match, call the script `/temp/secret.sh` in my example(see below).

Code: [Select all](#)

```
attach 5 {
    match    "vendor"    "0x0930";
    match    "product"   "0x6544";
    match    "release"   "0x0100";
    action   "sleep 1; /temp/secret.sh $device-name";
};
```

[*] Save the configuration file (devd.conf) and restart `devd` service

I got in my log:

```
Dec 14 15:54:40 nas kernel: umass0: detached
Dec 14 15:54:56 nas kernel: umass0: <Western Digital External HDD, class 0/0, rev 2.00/1.75, addr 2> on uhub4
Dec 14 15:54:56 nas root: Unknown USB device: vendor 0x1058 product 0x1003 bus uhub4
Dec 14 15:54:56 nas kernel: da0 at umass-sim0 bus 0 target 0 lun 0
Dec 14 15:54:56 nas kernel: da0: <WD 6400AAV External 1.75> Fixed Direct Access SCSI-4 device
```

But no value for reference???

danmero wrote:

Code: [Select all](#)

```
attach 5 {
    match    "vendor"    "0x0930";
```

```

match "product" "0x6544";
match "release" "0x0100";
action "sleep 1; /temp/secret.sh $device-name";
};

```

I didnt realise that you need to change the above values (the hex? ones) depending on what you get from the log.

danmero wrote:[*] Create the main script (here is my sample script).

Code: [Select all](#)

```

#!/bin/bash
key_file=/temp/ukey.key
device=da${1//[a-z]/}
origin=/temp/scripts
target=/mnt/secret

_start() {
  if [ ! -f /var/run/${0##*/}.pid ];then
    echo $$ > /var/run/${0##*/}.pid
  else
    /usr/bin/logger "Error syncing $origin on $device";
    echo "A16P16A16P16A16P16CP8CP8A16P16A16P16A16" > /dev/speaker
    _finish;
  fi
}
_finish() {
  if [ -f /var/run/${0##*/}.pid ];then
    rm /var/run/${0##*/}.pid;
  fi
  exit
}
_mount() {
  /sbin/geli attach -p -k ${key_file} /dev/$device
  /bin/mkdir -p ${target} &&
  /sbin/mount /dev/${device}.eli ${target}/ &&
  /sbin/fsck_ufs -pB /mnt/${target};
}
_umount() {
  /sbin/umount /dev/${device}.eli;
  /bin/rm -r /mnt/${target};
  /bin/sync /dev/${device}.eli
  /sbin/geli detach /dev/${device};
  /bin/sync /dev/${device}
}
_sync_func() {
  /usr/local/bin/rsync -a --log-file=${origin}/backup.log ${origin}/ ${target}/
  /usr/bin/logger "Finish syncing $origin on $device";
}
_start; _mount; _sync_func; _umount;
while [ -e /dev/${device} ];do echo "A8P8A8P8A8" > /dev/speaker;sleep 10;done
_finish;

```

I dont know if it matters what you call the scrit?

I think it must be saved in /temp/scripts/ which you may need to create?

I think to call the script you type ./script_name <dev> ie ./scriptname ad0 ?

Change the 4th line "origin=" to whatever data you want backed up?

Finally after all that I keep getting errors:

```

nas:/temp/scripts# ./sync_ext.sh da0
Can't stat /mnt//mnt/secret: No such file or directory
nas:/temp/scripts# ./sync_ext.sh da0
Can't stat /mnt//mnt/secret: No such file or directory

```

```

rm: /mnt//mnt/secret: No such file or directory
^C
nas:/temp/scripts#

```

So something is still not quite right? Please help.

Cheers

-Al

Re: USB attach, auto-mount, rsync, detach & umount(auto-backup)

by [bigalnz](#) » Tue Dec 14, 2010 6:43 am

PS: Should there be a script called secret.sh? If so - which script is called this?

Thanks

-Al

Re: USB attach, auto-mount, rsync, detach & umount(auto-backup)

by [bigalnz](#) » Wed Dec 15, 2010 3:50 am

Ok - I changed the name of the main script to secret.sh and tried calling it manually with ./secret.sh da0 (putting it in /temp/) but still getting :

```
nas:/temp# ./secret.sh da0
Can't stat /mnt//mnt/secret: No such file or directory
rm: /mnt//mnt/secret: No such file or directory
^C
nas:/temp#
```

Why am I ending up with double /mnt//mnt ?

Cheers

-Al

secret.sh

```
#!/bin/bash
key_file=/temp/ukey.key
device=da${1//[a-z]/}
origin=/mnt/DATA/
target=/mnt/secret

_start() {
if [ ! -f /var/run/${0##*/}.pid ];then
echo $$ > /var/run/${0##*/}.pid
else
/usr/bin/logger "Error syncing $origin on $device";
echo "A16P16A16P16A16P16CP8CP8CP8A16P16A16P16A16" > /dev/speaker
_finish;
fi
}
_finish() {
if [ -f /var/run/${0##*/}.pid ];then
rm /var/run/${0##*/}.pid;
fi
exit
}
_mount() {
/sbin/geli attach -p -k ${key_file} /dev/$device
/bin/mkdir -p ${target} &&
/sbin/mount /dev/${device}.eli ${target}/ &&
/sbin/fsck_ufs -pB /mnt/${target};
}
_umount() {
/sbin/umount /dev/${device}.eli;
```

```
/bin/rm -r /mnt/${target};
/bin/sync /dev/${device}.eli
/sbin/geli detach /dev/${device};
/bin/sync /dev/${device}
}
_sync_func() {
/usr/local/bin/rsync -a --log-file=${origin}/backup.log ${origin}/ ${target}/
/usr/bin/logger "Finish syncing $origin on $device";
}
_start; _mount; _sync_func; _umount;
while [ -e /dev/da0 ];do echo "A8P8A8P8A8" > /dev/speaker;sleep 10;done
_finish;
```

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [bigalnz](#) » Mon Dec 20, 2010 5:03 pm

I have modified Danmero's script and I think I have it working, without encryption:

Code: [Select all](#)

```
#!/bin/bash -x
#key_file=/temp/ukey.key
device=da${1//[a-z]/}
origin=/mnt/DATA/Test/
target=/mnt/WD

echo ${device}
echo ${target}

_mount() {
# /sbin/geli attach -p -k ${key_file} /dev/$device
/bin/mkdir -p ${target} &&
/sbin/mount /dev/${device} ${target}/ &&
/sbin/fsck_ufs -pB ${target};
umount /mnt/WD;
}
_umount() {
/sbin/umount /dev/${device};
/bin/rm -r ${target};
/bin/sync /dev/${device}
# /sbin/geli detach /dev/${device};
/bin/sync /dev/${device}
}
_sync_func() {
echo "START SYNC"
/usr/local/bin/rsync -av --progress --log-file=${origin}/backup.log ${origin}/ ${target}/
/usr/bin/logger "Finish syncing $origin on $device";
}
_mount; _sync_func; _umount;
#while [ -e /dev/da0 ];do echo "A8P8A8P8A8" > /dev/speaker;sleep 10;done
#_finish;
```

I still dont understand what all the "A8P8A8" is all about?

Additionally I am going to try to get a email sent when the script has finished run via ssmtp. Will post here if I get it to work.

-Al

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [danmero](#) » Tue Dec 21, 2010 1:35 pm

Code: [Select all](#)

```
while [ -e /dev/da0 ];do echo "A8P8A8P8A8" > /dev/speaker;sleep 10;done
```

This is a loop at the end of the script, will send different tone to PC internal speaker every ten seconds until you remove your USB device 😊

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [tbo69](#) » Wed Jan 26, 2011 9:48 pm

danmero,

This script is pretty much what I need/want, minus the encryption (which looks easy enough to remove)....

I want a cheap "off site" back-up. So I'm wanting to periodically backup to an external USB HDD and simply bring it to work with me. This way if something awful happened at home, I'd still have all our important data and pictures around. As long as all of Vancouver doesn't sink into the ocean, I'm all good then.

My FreeNAS is running on a PIII 3.2 GHz with 512 MB RAM. Running Embedded LiveCD 0.7.2 Rev. 5543

Some modifications I'd like to make:

- 1) create a new script to modify /etc/devd.conf upon bootup (since I'm running embedded! Changes won't carry over upon reboot) with my Ext USB HDD info for attachment
- 2) Update your existing script to record the date/time of the sync/backup
- 3) setup a cron job to check the last sync/backup date/time and email a reminder if it is not done after X number of days....similar to what backula can do

Some questions I have for these mods:

1) how can I modify the /etc/devd.conf dynamically (making sure the "attach N" number is right/unique etc.)? Should it be done Pre or Post Init? I'd like to avoid just saving a modified copy on a mount then simply copying it over the existing /etc/devd.conf in case something changed in the original /etc/devd.conf....or am I worrying about nothing?

2) got any suggestions on an easy way to record the last date/time? even a command just to get the current day of the year (0-364) would be good enough...I could simply write it to a log file on a mount.

EDIT: Answered this question, hopefully. It works on my Suse box anyway...I'll try it when I get home, use "date +%j" to get the current julian day. Awesome! Next problem now 🤖

3) do you know how to send mail with dynamic or variable message content via smtp and a script?

Thanks in advance for any help given!

I'll post my script(s) if/when I get them working! 😊

FreeNAS Rocks! 🤖

🗨 [Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [tbo69](#) » Sun Jan 30, 2011 7:22 am

Alright, I've written a script and modified danmero's. One is the backup script. The other checks the staleness of the backup and will email a reminder.

Use the devd.conf method for the backup script. Add a cron job through WebGUI to run the checker once a day.

I do have a problem though, maybe danmero, you know the answer....when I plug in my External USB 1.5TB backupdrive formatted as UFS, the device name passed in is umass1. It won't mount. But if I run ./backup.sh da1p1 it works fine! Why is it telling me umass1? And if your script simply took the 1 from umass1 and added da to it to give me da1, it still won't mount! mount wants: "mount /dev/da1p1 /mnt/tempBackup".....I must be missing something. 🤔

Any help would be appreciated!

Oh, so would a code review/comments/refinements....I can code C in my sleep, but bash scripts, I'm a noob. 🤖

The backup script:

Code: [Select all](#)

```
#!/bin/bash
#-----
# Backup.sh
# Version: 1.0
# Date: Jan 27 2011
# Author: Trevor Skaling
# Target: FreeNAS 0.7.2 Rev. 5543
# Credit: Mostly copied from danmero's script.
# https://sourceforge.net/apps/phpbb/freenas/posting.php?mode=reply&f=12&t=2073&sid=f8aa95efea911460e50a4bdb2ceb5b4f
# USB attach, auto-mount, rsync, detach & umount(auto-backup) thread
#
```

```

# Overview:
#
#   This script:
#   -gets passed in one argument, the target device to use for backup
#   -Mounts the device
#   -backs up a list of directories to the backup device via "rsync"
#   -has optional parameters to allow deletion of no longer existing
#   files on the source directory
#   -Unmounts and cleans up temporary directories
#   -Stores the date of the backup into a file
#
# Backup List:
#
# This script is used to read a list of directories to backup.
# Each directory to be backed up should be on a separate line in the
# file. An optional "DEL" flag can appear after the directory to
# signal to the script to backup that directory and delete files
# no longer within the source directory.
#
# Example list:
#   /home/User/Data
#   /home/User/Scratch      DEL
#   /mnt/Data/Photos
#
# Backups are made to the target device in the exact full path specified
# in the backup list. For example, if we are using backup device da0 mounted to
# /mnt/tempBackup, the list above would create the following directory structure:
#   /mnt/tempBackup/home/User/
#                               Data/
#                               Scratch/
#   /mnt/tempBackup/mnt/Data/Photos/
#
# Backup timestamp:
#
# This script will store the "day" of the backup to file. This can be
# used to check the time since last backup (see checkBackup.sh script)
# and email reminders if needed. Currently limited to "day" resolution.
#
#-----
# VARIABLES
#-----

# reads the first commandline argument as the device to backup to
# for some reason, da0p1 gets parsed to da11?
#device=da${1//[a-z]/}
device=$1

# the directory to mount the above device to and to backup to
target=/mnt/tempBackup

#location of this script
scriptpath=/root/of/directories/below

#location to log to
logpath=${scriptpath}/log

#location of the file used to store the last backup timestamp
datefile=${logpath}/lastBackupDate.log

#list of directories to backup
synclist=${scriptpath}/backuplist.txt

#-----
# FUNCTIONS
#-----

#perform checks and actions needed to initialize this script
_start() {

    #check for proper usage, need one commandline argument
    if [ ! $device ]; then
        echo "Usage: backup device. Example: 'backup da0p1'"
        echo
        echo "No device given!"
        echo
        _finish;
    fi

    /usr/bin/logger "Starting Backup syncing on device $device!";

    # let the user know the device is connected and the backup started
    echo "A8P8A8P8A8" > /dev/speaker

```

```

#check if this script is already running!
if [ ! -f /var/run/${0##*/}.pid ];then
    echo $$ > /var/run/${0##*/}.pid
else
    /usr/bin/logger "Error syncing $origin on $device";

    #play tone to PC speaker to notify user of error
    echo "A16P16A16P16A16P16CP8CP8A16P16A16P16A16" > /dev/speaker
    _finish;
fi
}

#perform shutdown actions for this script
_finish() {
    if [ -f /var/run/${0##*/}.pid ];then
        rm /var/run/${0##*/}.pid;
    fi
    exit
}

#perform the needed steps to mount the backup disk
_mount() {
    #wait for the device to wake up
    sleep 5
    /bin/mkdir -p ${target} &&
    /sbin/mount /dev/${device} ${target} &&
    /sbin/fsck_ufs -pB ${target};
}

#perform the steps needed to unmount the backup disk
_umount() {
    /sbin/umount /dev/${device};
    /bin/rm -r ${target};
    /bin/sync /dev/${device};
}

#perform the steps needed to backup the data
_sync_func() {
    #for each line in the backuplist file, read the path and the optional "DEL" flag
    while read curPath del_option
    do
        /usr/bin/logger "Starting sync of $curPath to ${target}${curPath}..."

        #does the destination directory exist already, if not, make it
        if [ ! -e ${target}${curPath} ]; then
            /bin/mkdir -p ${target}${curPath}
        fi

        # was this source directory flagged to backup with "delete" enabled
        if [ "$del_option" == "DEL" ]; then
            /usr/bin/logger "Syncing $curPath on $device with Delete Files active!";
            /usr/local/bin/rsync -a --delete --log-file=${logpath}/backup.log ${curPath}/ ${target}${curPath}/
        else
            /usr/bin/logger "Syncing $curPath on $device";
            /usr/local/bin/rsync -a --log-file=${logpath}/backup.log ${curPath}/ ${target}${curPath}/
        fi
        /usr/bin/logger "Finish syncing $curPath on $device";
    done < $synclist
}

#perform the steps needed to timestamp the backup
# NOTE: If you change this logic, update the checkBackup.sh script (which reads this file)
#       to match the new logic.
_store_last_backup() {
    # remove the exiting timestamp file
    /bin/rm -f ${datefile};

    # create a new empty timestamp fiile
    /usr/bin/touch ${datefile};

    # store the current julian day into the file
    /bin/date +%j > ${datefile};
}

#-----
# SCRIPT MAIN LOOP
#-----

#start the script, mount the device, backup, unmount, then timestamp
_start; _mount; _sync_func; _umount; _store_last_backup;

# script is finished, sound a beep on the PC speaker every 10 seconds until the

```

```

# device is unplugged
while [ -e /dev/${device} ];do echo "A8P8A8P8A8" > /dev/speaker;sleep 10;done

#exit the script
_finish;

```

The backup checker/emailer script:

Code: [Select all](#)

```

#!/bin/bash
#-----
# checkBackup.sh
# Version: 1.0
# Date: Jan 27 2011
# Author: Trevor Skaling
# Target: FreeNAS 0.7.2 Rev. 5543
#
# Overview:
#
# A script to check the time since the last backup and email reminders
# if too much time has passed since the last backup. Use this script
# in conjunction with the backup.sh script.
#
# NOTE: This script currently is limited to "day" resolution. Cannot
# determine hours/minutes since last backup.
#
# This script:
# -checks the timestamp from a specified file and computes
# the elapsed "days" since that timestamp.
# -If the elapsed time is greater than a specified amount,
# an email is generated and sent.
#
#-----
# VARIABLES
#-----

#the location of the date timestamp file. This must match the location used in
#the backup.sh script
datefile=/same/location/as/other/script/lastBackupDate.log

#mail recipients, seperate multiple emails via semi-colon
mailto=email@email.com

#Maximum amount of days that can pass since the last backup before emails are sent
maxBackupSpanDays=31

#stores the days since the last backup. Global variable needed.
daysSinceLastBackup=0

#stores the first commandline argument
isTestMail=$1
numArgs=$#

#-----
# FUNCTIONS
#-----

#perform checks and actions needed to initialize this script
_start()
{
    # if we cannot read the timestamp file, there is nothing we can do
    if [ ! -r ${datefile} ]; then
        /usr/bin/logger "Cannot read last backup date!!!: ${datefile}";
        _finish;
    fi

    #check for proper usage, need one commandline argument
    if [ "$numArgs" -eq 0 ]; then
        echo "Normal Check Commencing..."
    elif [ "$isTestMail" != "-test" ]; then
        echo "Usage: checkBackup Optional: -test"
        echo
        echo "Bad argument!"
        echo
        _finish;
    else
        echo "Test email send Commencing..."
        _sendReminder;
        _finish;
    fi
}

```

```

#perform shutdown actions for this script
_finish() {
    exit
}

#perform steps needed to determine the days since the last update and send an email
#if required
_checkDate() {

    # assume the timestamp julian "day" is the number on the first line of the file. This
    # file is generated via the backup.sh script
    lastBackupDay=`head -n 1 ${datefile}`

    # get the current julian day
    curDay=`/bin/date +%j`

    # strip off leading zero, otherwise the number is treated a hex and the
    # math will be wrong
    while [[ $curDay = 0* ]]
    do
        curDay=${curDay#0}
    done

    # calculate the difference
    daysSinceLastBackup=$(( $curDay-$lastBackupDay))

    # check for next year. If the calculation above results in a negative number we
    # have moved to the next year (eg: last backup day was 345, current day is 2)
    if [ $daysSinceLastBackup -lt 0 ]; then
        # recalculate for the new year. Currently ignore leap years (no worries, backups
        # only get over-stale by one day!
        daysSinceLastBackup=$(( ($curDay+365)-$lastBackupDay))
    fi

    echo "Converting for year carry over: $daysSinceLastBackup"
    fi

    # check if the time since is over the threshold
    if [ $daysSinceLastBackup -gt $maxBackupSpanDays ]; then
        # send the email
        _sendReminder;
    else
        /usr/bin/logger "Backup check complete. No action needed."
    fi
}

#builds and sends the reminder email
_sendReminder() {

    /usr/bin/logger "Sending email reminder for NAS Backup!"

    # build the email
    MSG=$(printf "From:%s\nTo:%s\nSubject:NAS Backup\nDate:%s\n\n%s\n\nThanks!\n\n Your NAS" \
        "me" "${mailto}" "$(date "+%a, %d %b %Y %H:%M:%S %z")" \
        "It has been "${daysSinceLastBackup}" days since your last backup. Please Backup!" )

    # send the email
    /etc/rc.d/msmtp
    echo "${MSG}" | msmtp --file=${msmtp_config}"/var/etc/msmtp.conf" "${mailto}"
}

#-----
# SCRIPT MAIN LOOP
#-----

_start; _checkDate;
_finish;

```

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [danmero](#) » Sun Jan 30, 2011 12:57 pm

tbo69 wrote: Alright, I've written a script and modified danmero's. One is the backup script. The other checks the staleness of the backup and will email a reminder.

Use the devd.conf method for the backup script. Add a cron job through WebGUI to run the checker once a day.

I do have a problem though, maybe danmero, you know the answer...when I plug in my External USB 1.5TB backupdrive formatted as UFS, the device name passed in is umass1. It won't mount. But if I run ./backup.sh da1p1 it works fine! Why is it telling me umass1?

The most important part is the devd.conf , once you solve this problem you can do anything else. On the second section of this howto I try to explain howto play with devd.conf , this is not a *work-always* solution as suggested by [bubulein](#) on his post but will give you the general idea howto tweak the configuration for your situation. If *attach 0* exist , use *attach 1* or next available number. The following directive will send the info to *logger* and in the same time to */test_file*

Code: [Select all](#)

```
attach 1 {  
    device-name    "umass[0-9]+";  
    action "sleep 2; logger vendor:$vendor,product:$product,release:$release,device-name:$device-name; echo  
    vendor:$vendor,product:$product,release:$release,device-name:$device-name >> /test_file";  
};
```

Insert and remove the USB and check the */test_file* contents

Code: [Select all](#)

```
cat /test_file
```

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [tbo69](#) » Mon Jan 31, 2011 8:49 pm

The most important part is the devd.conf , once you solve this problem you can do anything else. On the second section of this howto I try to explain howto play with devd.conf , this is not a *work-always* solution as suggested by [bubulein](#) on his post but will give you the general idea howto tweak the configuration for your situation. If *attach 0* exist , use *attach 1* or next available number. The following directive will send the info to *logger* and in the same time to */test_file*

Hi danmero, I appreciate your response. I DO HAVE devd.conf running the script! It DOES recognize the drive and DOES run the backup.sh script.

The problem is it runs 'backup.sh umass1' (because \$device-name in devd.conf is "umass1").....

this results in the script trying to 'mount /dev/da1 /mnt/tempBackup'.....THIS WILL NOT WORK! I'm unsure how anyone gets the script to work!? I must be missing something.... ??

What needs to happen is 'mount /dev/da1p1 /mnt/tempBackup' (notice the 'p1', without the partition or slice, mount give "Invalid Argument").

So my question is, how do I get devd.conf to pass in the proper device info to your script?

I could simply hard code /dev/da1p1 to the script and it will work but I cannot guarantee that this external USB drive will be /dev/da1p1, it could be /dev/da0p1 (if I don't have another external USB drive plugged in already) or /dev/da2p1 if I already have 2 USB external devices plugged in!

I hope I am clear as to my problem. If not, please let me know and I will try to be even more specific..... 😊

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [danmero](#) » Tue Feb 01, 2011 12:15 am

From original post , section 3 (script)

Code: [Select all](#)

```
device=da${1//[a-z]/}
```

this will take the numerical value from first argument(in your case umass1 will be 1) and create the *device* variable, having the value of da1(your case).

since you want to mount a UFS filesystem from the first GPT partition(p1) from your device the line should be

Code: [Select all](#)

```
device=da${1//[a-z]/}p1
```

Let me know if this solve your problem

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [tbo69](#) » Tue Feb 01, 2011 12:29 am

this will take the numerical value from first argument(in your case umass1 will be 1) and create the device variable, having the value of da1(your case).
since you want to mount a UFS filesystem from the first GPT partition(p1) from your device the line should be
CODE: SELECT ALL
device=da\${1//[a-z]/}p1

Let me know if this solve your problem

It does answer my question 😊 Thanks!

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [jensahh](#) » Wed Feb 16, 2011 12:16 am

I'm trying to get this to work, but i'm unsure about the vendor product and so on, it seems like that's the problem.

When i check the /var/log/system.log i find this when i attach the device:

Code: [Select all](#)

```
Feb 16 00:11:32 freenas kernel: umass2: <Kingston DT Mini Lite, class 0/0, rev 2.00/1.00, addr 4> on uhub3
Feb 16 00:11:32 freenas kernel: umass2:2:2:-1: Attached to scbus2
Feb 16 00:11:32 freenas root: Unknown USB device: vendor 0x0951 product 0x1649 bus uhub3
Feb 16 00:11:33 freenas kernel: pass3 at umass-sim2 bus 2 target 0 lun 0
Feb 16 00:11:33 freenas kernel: pass3: <Kingston DT Mini Lite 1.00> Removable Direct Access SCSI-2 device
Feb 16 00:11:33 freenas kernel: pass3: 40.000MB/s transfers
Feb 16 00:11:33 freenas kernel: GEOM: new disk da3
Feb 16 00:11:33 freenas kernel: da3 at umass-sim2 bus 2 target 0 lun 0
Feb 16 00:11:33 freenas kernel: da3: <Kingston DT Mini Lite 1.00> Removable Direct Access SCSI-2 device
Feb 16 00:11:33 freenas kernel: da3: 40.000MB/s transfers
Feb 16 00:11:33 freenas kernel: da3: 3690MB (7557704 512 byte sectors: 255H 63S/T 470C)
```

The /etc/devd.conf has this:

Code: [Select all](#)

```
attach 5 {
    match "vendor" "0x0951";
    match "product" "0x1649";
    match "release" "0x100";
    action "sleep 1; /temp/secret.sh $device-name";
};
```

And can't find anything in the logs or on the drive after it has been inserted. (Don't have any speaker but quite sure that the melody isn't playing) and ofcourse i got the file at /temp/secret.sh and a folder with files at /temp/scripts

Any ideas?

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [tbo69](#) » Tue Feb 22, 2011 1:36 am

Are you sure you have the "release" correct?

thats why you need to have this in your file:

Code: [Select all](#)

```
attach 0 {
    device-name "umass[0-9]+";
    action "sleep 2; logger vendor:$vendor,product:$product,release:$release,device-name:$device-name";
};
```

```
};
```

So you can get the logger to log the values you need.....it must match all three before it will run the script on attach.....

I don't see this being logged in your logfile.....one problem I had was the sleep wasn't long enough....try 10 seconds....

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [tbo69](#) » Tue Oct 11, 2011 4:15 pm

Another problem I found while trying to get devd.conf attach actions is when using this method with a LiveCD install. I wrote a script to update /etc/devd.conf and to restart devd. But in my script I was not specifying the full path (/etc/rc.d/devd restart), I was just calling devd restart. This was causing m devd.conf changes to not be read in and never acted upon. Once I fixed my script, everything worked fine! 😊 My only problem left now is getting my pictures to show up properly over uPnp!

[Re: USB attach, auto-mount, rsync, detach & umount\(auto-backup\)](#)

by [diotta](#) » Wed Oct 19, 2011 9:10 pm

Hi,

I'm trying to run the script but i have a little big problem. My external HDD is NTFS formatted so if I run this script:

Code: [Select all](#)

```
#!/bin/bash
sleep 5
echo `date +%e-%m-%y`
sleep 5
mkdir -p /mnt/appoggio4/
sleep 5
ntfs-3g /dev/da0s1 /mnt/appoggio4
sleep 15
mkdir /mnt/appoggio4/backup-`date +%e-%m-%y`
```

ntfs-3g response is "no such file or directory"

but if I execute this "ntfs-3g /dev/da0s1 /mnt/appoggio4/" with / at the end, it mount my pen in this directory /mnt/appoggio4/?/ and when I create "backup-`date +%e-%m-%y`" directory it create "backup-`date +%e-%m-%y`?"

What does the question mark mean?

If I run the individual commands in the shall works without problems. Why??

Thanks