



subscribed mailing lists

I am currently archiving and making [freebsd-questions](#), [freebsd-hackers](#), [freebsd-stable](#), and [cvs-all](#) available on my website with [mhonarc](#). The lists have been indexed and made searchable with [namazu](#). I have found that [freebsd-questions](#) is usefull to people of any *nix platform... [You can learn more about FreeBSD at their website.](#)

Matrox PowerDesk

Matrox PowerDesk makes it easy to configure my MatroxG550 graphics card for X11. I love how it makes dual display so simple to set up. [You can get Matrox PowerDesk for Linux off of the matrox website.](#)

A.I.N.A.B.

Adam.KungFooHamster.Com Is Not A Blog. I don't like journals and you will never find anything out about my life here. (update: it has been pointed out to me that this has become a blog... but.. i still won't say anything personal on here... dangit)

Adaptec 2400A or Western Digital 120G Hard Drive?

I have been experiencing some strange issues with my machine during high load times... I [posted to freebsd-questions](#) and got a [very interesting responce](#). The whole time I had assumed that the problem was with my raid controller card when in actuality it may be my Western Digital Hard Disks.

Pong

I know I said that I wouldn't put journal sorta stuff on here, but I just couldn't pass this up. [I was playing Atari's Pong and I found a sweet spot!](#) I didn't have to move the paddle!!! It just scored over and over and over and over and over... I found the sweet spot right around when I scored 5 points and I just left it there till it got to 21 and the level stopped. -- time passes -- WOO HOO!!! It left the paddle where I left it! I was able to do it again from the start and get [60 seconds of proof!](#) [The madness!!](#) WOO HOO!!

Hi

I will eat your soul.

Why is portupgrade so cool?

I personally use portupgrade on my FreeBSD system and have found it to be invaluable. Another [slashdot post](#) pointing to onlamp.com has a great run-thru on the [power of portupgrade](#).

CVS Tags - Which should my FreeBSD system use?

The FreeBSD handbook has a [great page](#) that explains the differences between all of the different branch tags so that one can decide which version is best for them.

FreeBSD CUPS howto

The [FreeBSD Diary](#) has a [simple howto](#) on setting up the [CUPS \(Common UNIX Printing System\)](#) server and client in FreeBSD. [The message](#) this person posted to freebsd-questions seems to agree. Another [thread beginning](#) here helps a newbie get a better grasp on what files to look at, etc for setting up CUPS.

KVim: freebsd configure: error: could not find Qt directory

When compiling on FreeBSD along with instructions on [KVim's site](#), try --with-qt-dir=/usr/X11R6

What is Virtual FreeBSD?

[This message](#) explains that 'virtual FreeBSD' is simply jails on a FreeBSD system. You can find out more about jails by going to [the page](#) on onlamp.com that was discovered on a [slashdot post](#). Of course, [the FreeBSD man page on jail](#) is a good resource too!

jail start-up script

This assumes you have all the jail stuff(Jailer, Jail Utilities) from this dude(N. Nielsen)'s [jail page](#).

```
PATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin"
JAIL_NAME=jail-name
JAIL_HOSTNAME=jail-name.hostname.com
JAIL_IP=ip.ad.rde.ss
JAIL_DRIVE=vn0

jail_mount_drive () {
    vnconfig -s labels -c ${JAIL_DRIVE} /jail/ufs/${JAIL_NAME}.ufs && \
    disklabel -r -w ${JAIL_DRIVE} auto && \
    tunefs -n enable /dev/${JAIL_DRIVE} && \
    mount -o noatime /dev/${JAIL_DRIVE} /jail/home/${JAIL_NAME}
}

jail_unmount_drive () {
    umount /jail/home/${JAIL_NAME} && \
    vnconfig -u ${JAIL_DRIVE} && \
    vnconfig -r labels ${JAIL_DRIVE}
}

case "$1" in
start)
    ;;
nstart)
    jails | grep -q "${JAIL_HOSTNAME}"
    if [ $? = 0 ]; then
        echo " ${JAIL_NAME} already running"
    else
        if [ -x /jail/home/${JAIL_NAME}/usr/local/sbin/jailer ]; then
            /usr/bin/true
        fi
    fi
fi
```

```

else
    jail_mount_drive && echo " mounted ${JAIL_NAME}"
fi

if [ -x /jail/home/${JAIL_NAME}/usr/local/sbin/jailer ]; then
    /usr/sbin/jail /jail/home/${JAIL_NAME} ${JAIL_HOSTNAME} ${JAIL_IP} \
    /usr/local/sbin/jailer && echo " started ${JAIL_NAME}"
else
    echo " error: ${JAIL_NAME}: could not mount"
fi
fi
;;
stop)
    jails | grep -q "${JAIL_HOSTNAME}"
    if [ $? = 0 ]; then
        killjail ${JAIL_HOSTNAME} && echo " stopped ${JAIL_NAME}"
    else
        echo " ${JAIL_NAME} already stopped"
    fi
    ;;
restart)
    jails | grep -q "${JAIL_HOSTNAME}"
    if [ $? = 1 ]; then
        echo " ${JAIL_NAME} not running - try start"
    else
        killjail -r ${JAIL_HOSTNAME}
    fi
    ;;
mount)
    if [ -x /jail/home/${JAIL_NAME}/usr/local/sbin/jailer ]; then
        echo " ${JAIL_NAME} already mounted"
    else
        jail_mount_drive && echo " mounted ${JAIL_NAME}"
    fi
    ;;
unmount|umount)
    jails | grep -q "${JAIL_HOSTNAME}"
    if [ $? = 0 ]; then
        echo " error: ${JAIL_NAME}: could not unmount - still running"
    else
        if [ -x /jail/home/${JAIL_NAME}/usr/local/sbin/jailer ]; then
            jail_unmount_drive && echo " ${JAIL_NAME} unmounted"
        else
            echo " ${JAIL_NAME} already unmounted"
        fi
    fi
    ;;
status)
    jails | grep -q "${JAIL_HOSTNAME}"
    if [ $? = 0 ]; then
        echo -n " ${JAIL_NAME} running"
    else
        echo -n " ${JAIL_NAME} not running"
    fi

    if [ -x /jail/home/${JAIL_NAME}/usr/local/sbin/jailer ]; then
        echo -n '+mounted'
    else
        echo -n '+unmounted'
    fi
    echo
    ;;
*)
    echo ""
    echo "Usage: `basename $0` { start | stop | restart | mount | u(n)mount }"
    echo ""
    exit 64
    ;;
esac

```

Apache Module C API: Directory tidbits

I have come across some interesting guidelines if one is writing an Apache Module in C and wants to catch directories with their `SetHandler`. (Just imagine a directory as a file of type `httpd/unix-directory` or `DIR_MAGIC_TYPE`.)

`SetHandler` doesn't affect directories inside a `<Location>` or `<Directory>` directive. `SetHandler` does affect all files within those directories recursively.

[mod_mime.c](#)'s first few lines of its "type checker" immediately rule out directories as `r->content_type=DIR_MAGIC_TYPE` and returns OK. `mod_autoindex` and `mod_dir` will catch this mime type.

This is why the handler of the module I am working on isn't getting a chance. The way my module can get a chance is by implementing its own "type checker". I had it access `mod_mime`'s configuration data and check to see if its `cfg->handler` matched the name of my module's handler. If it checked out then I changed `r->handler` to equal the name of my module's handler. I also added another condition to check if the `r->uri`'s last character was a slash("/"). If it wasn't a slash then I returned `DECLINED` so that `mod_dir` would redirect it to the appropriate path.

The reason I had to do this is because every module's type checker gets a shot to change the attributes of a request object. It gets called every time there is a hit to apache and does not care if you defined a `SetHandler` inside a `<Location>` or `<Directory>` directive. I needed to know that the object being requested was within the bounds of a `SetHandler(cfg->handler)` with the same name as my handler handler. I found this out the hard way when every virtual host on my site ended up with a blank page because I assumed that my module would only be called within a `SetHandler` directive and that I wouldn't have to verify.

I hope someone out there finds this information useful because I wasted a lot of time looking in the

wrong places to solve my problem.

Samba / smbpool / smbutil / mount_smbfs through a firewall hints

rule of thumb: know thy ip's - now you don't have to figure out what ports to keep open (destination to you).. (works best when dealing with static ip's)

printing - cups

```
$ smbpool smb://username:password@server/printer
```

```
ln -s /usr/local/bin/smbpool /usr/local/libexec/cups/backend/smb
```

```
(/usr/local)/etc/lmhosts
```

```
ip.ad.dre.ss server
```

mounting

```
$ mount_smbfs -l ip.ad.dre.ss //username@server/share /mnt/share
```

```
(/usr/local)/etc/nsmb.conf
```

```
[server:username:share]
password=password
```

```
(/usr/local)/etc/rc.d/samba_mount.sh
```

```
#!/bin/sh
case $1 in
start)
  /sbin/mount | grep -q '/mnt/share'
  if test $? = 0 ; then
    echo "alrady mounted"
  else
    /usr/sbin/mount_smbfs -I 192.168.1.2 //username@server/share /mnt/share
  fi
;;
```

```
stop)
  /sbin/mount | grep -q '/mnt/share'
  if test $? = 0 ; then
    /sbin/umount /mnt/share
  else
    echo "already unmounted"
  fi
  ;;
*)
  echo "Usage: `basename $0` (start|stop)"
esac
```

Client Side XSLT/XML Internet Explorer & Netscape/Mozilla/Firefox

HOW can I get mozilla and IE to render xml pages with xsl in similar fashions?

XML's `xml:space="preserve"` is an attribute that will tell a tag to preserve whitespace and linefeeds XSL's `<xsl:preserve-space elements="tag"/>` is a xsl tag that will tell the xsl processor to preserve whitespace and linefeeds for the specified elements (white seperated list) XML's second line! `<?xml-stylesheet type="text/xsl" href="template.xsl" alternate="no"?>` XSL's `<xsl:stylesheet ... > omit-xml-declaration="yes" -- strips the <?xml version="1.0"?>` from the top of the output. `indent="no" -- prevents IE from getting confuzed with whitespace if you have white-space: pre CSS usage being processed on the outputted HTML. method="html" -- tells mozilla to process this output like html (obviously it doesn't do anything intelligent with media-type, doctype-public, or doctype-system ... they're just output ... not instructions like method media-type="text/xml" -- tells IE to use this document like an XML document... so that it will respect the xml:space="preserve"'s doctype-public="-//W3C//DTD HTML 4.01 Transitional//EN" -- we can't use xhtml because method="html" doesn't make self ending tags like
 doctype-system="http://www.w3.org/TR/html4/loose.dtd" -- we can't use xhtml because method="html" doesn't make self ending tags like
`

[To see this page as XML processed against XSL on the fly with both IE and MOZ ... clicky ... clicky ...](#)

XSL-FO and FOP and white space (converting XML to PDF)

I never realized what an issue white space was until I started using, xml, xsl, and fo.

If you want the same effect as the HTML `<pre>` tag in FO then in your... XML make use of the `xml:space="preserve"` attribute XSL make use of `<xsl:preserve-space elements="tag"/>` instruction. FO [FOP](#) make use of the `white-space-collapse="false"`

and `linefeed-treatment="preserve"` attributes ([warning: Due to a bug in current versions of FOP, setting `white-space-collapse="false"` will also preserve line breaks in the text. Do not rely on this behavior, as it is non-conformant and will be changed.](#)) - meaning.. always set `linefeed-treatment` ... don't make assumptions Other FO engine consider FOP's primary way of preserving whitespace and maybe try out the `white-space-treatment="preserve"` attribute

Remember... WHEN IN DOUBT - W3C!! ([XSL-FO](#)) ([XSLT](#)) ([XML](#))

[To see this page as REALLY ROUGH BETA PDF OUTPUT \(processed by FOP\) ... clicky ... clicky ...](#) (and please remember ... if this output is ugly, its my fault ... not FOP's)