

# Table of Contents

0. Goal.....	1
1. Tools and Configuration.....	1
2. Installation.....	2
3. Fresh Build.....	2
4. Make a TinyBSD image.....	3
5. Maintenance on the image.....	4
6. Virtualization.....	5
7. Distribution.....	5
8. License.....	6

For current information look at <http://martenvijn.nl/trac/wiki/TinyBSD>

## 0. Goal

Building and Maintenain images of FreeBSD based on TinyBSD. For this it is nesseciary to mount images to the filesystem to alter the content of the image. For testing virtualization is used to speed up to development process.

### NanoBSD versus TinyBSD

NanoBSD	TinyBSD
full freebsd	stripped freebsd
180-200MB	16-28 MB
makes world and kernel	makes kernel and takes files from the installation on the host

I choose for TinyBSD for:

- smaller images for download (gzip)
- systems with limit flash onboard (soekris net4826)

### **Disclaimer**

Read 8. License.

## 1. Tools and Configuration

1.1 Storage to put the image on:

- compact-flash
- usb-drive
- sd-card

1.2 embedded hardware (option)

- soekris
- wrap / alix

- i386 based stuff
- power supply

### 1.3 connectivity (in case of 1.2)

- utp (cross and straight) cables
- null-modem cable
- usb2serial (check)

```
man ucom
man uplcom (or other)
echo 'ucom1|ucom1:dv=/dev/cuaU0"br#9600:pa=none' >>/etc/remote
tip ucom1
```

### 1.4 a fast machine as a buildhost

- i368
- 2 Gb Mem
- fast disks
- read for optimisations

```
man make
man make.conf
```

## 2. Installation

Install [FreeBSD 7.0 ISO](#)

Just follow the chapter [installation](#) from the FreeBSD handbook.

## 3. Fresh Build

Since new source it available I suggest to build the lasted version.

Update src [handbook](#).

```
# csup -h cvsup.nl5.freebsd.org /usr/share/examples/cvsup/standard-supfile
```

My mirror is cvsup.nl5.freebsd.org since I live in the Netherlands. Please select a local [mirror](#) site.

```
cvsup /root/sup
# cd /usr/src
# make buildworld
```

Optional strip your kernel. [handbook](#)

```
%cd /sys/i386/conf
# cp GENERIC MY_KERNEL
# vi MY_KERNEL
```

Build from src [handbook](#)

```
# cd /usr/src
# make buildkernel KERNCONF=MY_KERNEL
# make install KERNCONF=MY_KERNEL
```

Reboot in single user mode.

```
reboot
###choose single user mode in the bootmenu
# mergemaster
# cd /usr/src
# make installworld
# mergemaster -p
# reboot
```

Update ports. [handbook](#)

```
# portsnap fetch
# portsnap extract
```

## 4. Make a TinyBSD image

[TinyBSD](#) strips FreeBSD to smaller image.

```
# cd /usr/src/tools/tools/tools/tinybsd
```

for configuration options:

```
# ls conf/
apachecon      default      minimal      wireless
bridge         firewall    vpn          wrap
```

to see the configuration files `# ls conf/minimal/ TINYBSD tinybsd.basefiles etc tinybsd.ports }}`

- TINYBSD is the kernel config file [handbook](#)
- etc is a directory where you can put additional config files, they will be copied to the image
- tinybsd.basefiles files from base
- tinybsd.ports here you can configure packages to be added from ports

Normally I copy one of these directories to start me own setup.

```
# cp -rv conf/wrap conf/project
conf/wrap -> conf/project
conf/wrap/etc -> conf/project/etc
conf/wrap/etc/fstab -> conf/project/etc/fstab
conf/wrap/etc/rc.conf -> conf/project/etc/rc.conf
conf/wrap/etc/ttys -> conf/project/etc/ttys
conf/wrap/TINYBSD -> conf/project/TINYBSD
conf/wrap/tinybsd.basefiles -> conf/project/tinybsd.basefiles
conf/wrap/tinybsd.ports -> conf/project/tinybsd.ports
```

and add the files to svn (or something similar)

```
cd conf/project/
svn ci http://martenvijn.nl/svn/project .
svn add *
svn ci
```

Then make your necessary changes (and commit these every now and then)

To build an image

```
# ./tinybsd
```

## 5. Maintenance on the image

creating md0 and mount the image

```
# mdconfig -a -t vnode -f tinybsd.bin  
# mount /dev/md0a /mnt
```

Unmount the image after altering (examples below).

```
# umount /mnt  
# mdconfig -d -u md0
```

**Examples** Fix current bugs in tinybsd for example.

```
# cp /etc/mtree/* /mnt/etc/mtree/
```

Or the make ssh keys.

```
# chroot /mnt  
# /etc/rc.d/sshd start  
# exit
```

Adding your current kernel and disable old kernel. The .gz kernel will boot first

```
# cp /boot/kernel/kernel /mnt/boot/kernel  
# cp /mnt/boot/kernel/kernel.gz /mnt/boot/kernel/kernel_old.gz
```

This can be done also (properly) in the /boot/loader.conf. [manpage](#)

Adding openvpn. Find the binary, it's libs and startup scripts

```
# cd /usr/ports/security/openvpn  
# make install  
# rehash  
# which openvpn  
# ldd `which openvpn`  
# find start upscripts  
# ls /usr/local/etc/rc.d/
```

Making a backup.

```
# cd /mnt  
# tar -cvzf ~/image_tree.tgz .
```

I use this tree to rsync already running hosts.

```
# mkdir image  
# cd image  
# tar -xvzf ~/image_tree.tgz  
# ssh root@example.com 'mount -uwo noatime /'  
# rsync -larv * root@example.com:/  
# ssh root@example.com 'mount -a'  
# ssh root@example.com 'reboot'
```

**This can screw up a running host, be warned!**

However often it works fine.

## 6. Virtualization

I use qemu to test my images. [how-to](#)

Installing qemu (option kqemu = yes)

```
# cd /usr/ports/emulators/qemu
# make install
```

Prepare network for qemu.

This only needed if your want to bridge, elsewise your end behind virtual NAT.

qemu startscript for networking

```
# vi /etc/qemu-ifup
add these lines
#!/bin/sh
ifconfig ${1} 0.0.0.0
```

```
sudo chmod 755 /etc/qemu-ifup
```

Setting up the bridge.

I previously used the **tun** interface. It stopped working (openvpn related?). I use the **tap** interface now

I have fxp0 as a nic.

```
# ifconfig tap create
# ifconfig bridge create
# ifconfig bridge0 addm tap0 addm fxp0 up
```

Load modules.

```
kldload aio kqemu
```

Booting the image in qemu

```
% sudo qemu -hda tinybsd.bin -net nic -net tap
```

## 7. Distribution

This make a smaller image for download and more important Window users can write the file on flash or usb drive (alter /etc/fstab). See Manual Kaspers site for [monowall](#) and specially [physdiskwrite](#).

```
# tar -cvjf tinybsd.bin.bz tinybsd.bin
```

Writing to disk. (see dmesg for the correct disk)

```
# see dmesg for the correct disk
# zcat tinybsd.bin.bz | dd of=/dev/<disk> bs=16k
```

or setup [<http://martenvijn.nl/trac/wiki/PXEboot> PXEboot]

## 8. License

- This document may be copied.
- This document may be modified.
- This document may be redistributed.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

I would like to thank the following people for their contributions:

Oliver Fromme, Ian Smith, "OutbackDingo", Sunry Chen, Remko Lodder, Olli Hauer, Mark Tinguely.

cheers,

Marten

Feedback welcome: [info at martenvijn.nl](mailto:info@martenvijn.nl)