

NAME

nanobsd.sh - utility used to create a FreeBSD system image suitable for embedded applications

SYNOPSIS

nanobsd.sh [-bfhiKknqvwx] [-c *config-file*]

DESCRIPTION

The **nanobsd.sh** utility is a script which produces a minimal implementation of FreeBSD (called **NanoBSD**), which typically fits on a small media such as a Compact Flash card, or other mass storage medium. It can be used to build specialized install images, designed for easy installation and maintenance.

The following options are available:

- b** Skip the build stages (both for kernel and world).
- c *config-file*** Specify the configuration file to use.
- f** Skip the code slice extraction.
- h** Display usage information.
- i** Skip the disk image build stage.
- K** Skip the **installkernel** stage of the build.
- k** Skip the **buildkernel** stage of the build.
- n** Do not cleanup before each build stage. This suppresses the normal cleanup work done before the **buildworld** stage and adds **-DNO_CLEAN** to the make command line used for each build stage (world and kernel).
- q** Make output more quiet.
- v** Make output more verbose.
- w** Skip the **buildworld** stage of the build.
- X** Make **native-xtools**.

The features of **NanoBSD** include:

- Ports and packages work as in FreeBSD. Every single application can be installed and used in a **NanoBSD** image, the same way as in FreeBSD.
- No missing functionality. If it is possible to do something with FreeBSD, it is possible to do the same thing with **NanoBSD**, unless the specific feature or features were explicitly removed from the **NanoBSD** image when it was created.
- Everything is read-only at run-time. It is safe to pull the power-plug. There is no necessity to run `fsck(8)` after a non-graceful shutdown of the system.
- Easy to build and customize. Making use of just one shell script and one configuration file it is possible to build reduced and customized images satisfying any arbitrary set of requirements.

NanoBSD Media Layout

The mass storage medium is divided into three parts by default (which are normally mounted read-only):

- Two image partitions: `code#1` and `code#2`.
- The configuration file partition, which can be mounted under the `/cfg` directory at run time.

The `/etc` and `/var` directories are `md(4)` (malloc backed) disks.

The configuration file partition persists under the `/cfg` directory. It contains files for `/etc` directory and is briefly mounted read-only right after the system boot, therefore it is required to copy modified files from `/etc` back to the `/cfg` directory if changes are expected to persist after the system restarts.

BUILDING NanoBSD

A **NanoBSD** image is built using a simple `nanobsd.sh` shell script, which can be found in the `src/tools/tools/nanobsd` directory. This script creates a bootable image, which can be copied on the storage medium using the `dd(1)` utility.

The necessary commands to build and install a **NanoBSD** image are:

```
cd /usr/src/tools/tools/nanobsd
sh nanobsd.sh
cd /usr/obj/nanobsd.full
dd if=_disk.full of=/dev/da0 bs=64k
```

CUSTOMIZING NanoBSD

This is probably the most important and most interesting feature of **NanoBSD**. This is also where you will be spending most of the time when developing with **NanoBSD**.

Customization is done in two ways:

- Configuration options.
- Custom functions.

With configuration settings, it is possible to configure options passed to both the **buildworld** and **installworld** stages of the **NanoBSD** build process, as well as internal options passed to the main build process of **NanoBSD**. Through these options it is possible to cut the system down, so it will fit on as little as 64MB. You can use the configuration options to trim down the system even more, until it will consist of just the kernel and two or three files in the userland.

The configuration file consists of configuration options, which override the default values. The most important directives are:

NANO_NAME Build name (used to construct the working directory names).

NANO_SRC Path to the source tree used to build the image.

NANO_KERNEL Name of the kernel configuration file used to build the kernel.

NANO_ARCH Machine processor architecture to build. Defaults to output of **uname -p**.

NANO_BOOT0CFG

Controls the options passed to `boot0cfg(8)`; these dictate **boot0**'s behaviour.

NANO_BOOTLOADER

The **boot0** loader to use relative to the *NANO_WORLDDIR* variable. This defaults to *boot/boot0sio* and should be overridden to *boot/boot0* to provide a VGA console.

CONF_BUILD Options passed to the **buildworld** stage of the build.

CONF_INSTALL Options passed to the **installworld** stage of the build.

CONF_WORLD Options passed to both the **buildworld** and **installworld** stages of the build.

FlashDevice Defines the type of media to use. Check the *FlashDevice.sub* file for more details.

For more configuration options, please check the **nanobsd.sh** script.

To build **NanoBSD** image using the *nanobsd.conf* configuration file, use the following command:

```
sh nanobsd.sh -c nanobsd.conf
```

It is possible to fine-tune **NanoBSD** using shell functions in the configuration file. The following example illustrates the basic model of custom functions:

```
cust_foo () (  
    echo "bar=topless" > \  
        ${NANO_WORLDDIR}/etc/foo  
)  
customize_cmd cust_foo
```

There are a few pre-defined customization functions ready for use:

cust_comconsole Disables `getty(8)` on the virtual `syscons(4)` or `vt(4)` terminals (*/dev/ttyv**) and enables the use of the first serial port as the system console.

cust_allow_ssh_root Allow root to log in via `sshd(8)`.

cust_install_files Installs files from the *nanobsd/Files* directory, which contains some useful scripts for system administration.

FILES

src/tools/tools/nanobsd Base directory of the **NanoBSD** build script.

EXAMPLES

Making persistent changes to */etc/resolv.conf*:

```
vi /etc/resolv.conf  
...  
mount /cfg  
cp /etc/resolv.conf /cfg  
umount /cfg
```

A more useful example of a customization function is the following, which changes the default size of the */etc* directory from 5MB to 30MB:

```
cust_etc_size () (  
    cd ${NANO_WORLDDIR}/conf
```

```
        echo 30000 > default/etc/md_size
    )
    customize_cmd cust_etc_size
```

SEE ALSO

make.conf(5), boot(8), boot0cfg(8)

HISTORY

The **nanobsd.sh** utility first appeared in FreeBSD 6.0.

AUTHORS

NanoBSD was developed by Poul-Henning Kamp <phk@FreeBSD.org>. This manual page was written by Daniel Gerzo <danger@FreeBSD.org>.