

NAME

config - build system configuration files

SYNOPSIS

config [-CVgp] [-I *path*] [-d *destdir*] [-s *srcdir*] *SYSTEM_NAME*
config [-x *kernel*]

DESCRIPTION

The **config** utility builds a set of system configuration files from the file *SYSTEM_NAME* which describes the system to configure. A second file tells **config** what files are needed to generate a system and can be augmented by configuration specific set of files that give alternate files for a specific machine (see the *FILES* section below).

Available options and operands:

- V** Print the **config** version number.
- C** If the `INCLUDE_CONFIG_FILE` is present in a configuration file, kernel image will contain full configuration files included literally (preserving comments). This flag is kept for backward compatibility.
- I path** Search in *path* for any file included by the **include** directive. This option may be specified more than once.
- d destdir** Use *destdir* as the output directory, instead of the default one. Note that **config** does not append *SYSTEM_NAME* to the directory given.
- s srcdir** Use *srcdir* as the source directory, instead of the default one.
- m** Print the `MACHINE` and `MACHINE_ARCH` values for this kernel and exit.
- g** Configure a system for debugging.
- x kernel** Print kernel configuration file embedded into a kernel file. This option makes sense only if **options INCLUDE_CONFIG_FILE** entry was present in your configuration file.

SYSTEM_NAME Specify the name of the system configuration file containing device specifications, configuration options and other system parameters for one system configuration.

The **config** utility should be run from the *conf* subdirectory of the system source (usually */sys/ARCH/conf*), where *ARCH* represents one of the architectures supported by FreeBSD. The **config** utility creates the directory *./compile/SYSTEM_NAME* or the one given with the **-d** option as necessary and places all output files there. The output of **config** consists of a number of files; for the i386, they are: *Makefile*, used by *make(1)* in building the system; header files, definitions of the number of various devices that will be compiled into the system.

The **config** utility looks for kernel sources in the directory *../.* or the one given with the **-s** option.

After running **config**, it is necessary to run "make depend" in the directory where the new makefile was created. The **config** utility prints a reminder of this when it completes.

If any other error messages are produced by **config**, the problems in the configuration file should be corrected and **config** should be run again. Attempts to compile a system that had configuration errors are likely to fail.

DEBUG KERNELS

Traditional BSD kernels are compiled without symbols due to the heavy load on the system when compiling a "debug" kernel. A debug kernel contains complete symbols for all the source files, and enables an experienced kernel programmer to analyse the cause of a problem. The debuggers available prior to 4.4BSD-Lite were able to find some information from a normal kernel; *gdb(1)* (*ports/devel/gdb*) provides very little support for normal kernels, and a debug kernel is needed for any meaningful analysis.

For reasons of history, time and space, building a debug kernel is not the default with FreeBSD: a debug kernel takes up to 30% longer to build and requires about 30 MB of disk storage in the build directory, compared to about 6 MB for a non-debug kernel. A debug kernel is about 11 MB in size, compared to about 2 MB for a non-debug kernel. This space is used both in the root file system and at run time in memory. Use the **-g** option to build a debug kernel. With this option, **config** causes two kernel files to be built in the kernel build directory:

- *kernel.debug* is the complete debug kernel.
- *kernel* is a copy of the kernel with the debug symbols stripped off. This is equivalent to the normal non-debug kernel.

There is currently little sense in installing and booting from a debug kernel, since the only tools available which use the symbols do not run on-line. There are therefore two options for installing a debug kernel:

- "make install" installs *kernel* in the root file system.
- "make install.debug" installs *kernel.debug* in the root file system.

FILES

<code>/sys/conf/files</code>	list of common files system is built from
<code>/sys/conf/Makefile.ARCH</code>	generic makefile for the <i>ARCH</i>
<code>/sys/conf/files.ARCH</code>	list of <i>ARCH</i> specific files
<code>/sys/ARCH/compile/SYSTEM_NAME</code>	default kernel build directory for system <i>SYSTEM_NAME</i> on <i>ARCH</i> .

SEE ALSO

`config(5)`

The *SYNOPSIS* portion of each device in section 4.

S. J. Leffler and M. J. Karels, "Building 4.3 BSD UNIX System with Config", *4.4BSD System Manager's Manual (SMM)*.

HISTORY

The **config** utility appeared in 4.1BSD.

Before support for **-x** was introduced, **options INCLUDE_CONFIG_FILE** included entire configuration file that used to be embedded in the new kernel. This meant that `strings(1)` could be used to extract it from a kernel: to extract the configuration information, you had to use the command:

```
strings -n 3 kernel | sed -n 's/^___//p'
```

BUGS

The line numbers reported in error messages are usually off by one.