NAME

config - kernel configuration file format

DESCRIPTION

A kernel configuration file specifies the configuration of a FreeBSD kernel. It is processed by config(8) to create a build environment where a kernel may be built using make(1).

Lexical Structure

A kernel configuration file comprises a sequence of specification directives.

A specification directive starts with a keyword at the beginning of the line and is followed by additional parameters.

A specification directive may be terminated by a semicolon ‘;’ or by a newline. Long input lines may be broken into shorter lines by starting the second and subsequent lines with a white space character.

Case is significant, "machine" and "MACHINE" are different tokens.

A double quote character ‘"’ starts a quoted string. All characters up to the next quote character form the value of the quoted string. A ‘"’ character may be inserted into a quoted string by using the sequence ‘\"’.

Numbers are specified using C-style syntax.

A ‘#’ character starts a comment; all characters from the ‘#’ character till the end of the current line are ignored.

Whitespace between tokens is ignored, except inside quoted strings. Whitespace following a comment line is ignored.

Configuration Directives

Kernel configuration directives may appear in any order in a kernel configuration file. Directives are processed in order of appearance with subsequent directive lines overriding the effect of prior ones.

The list of keywords and their meanings are as follows:

cpu *cputype*

> Specify the CPU this kernel will run on. There can be more than one **cpu** directive in a configuration file. The allowed list of CPU names is architecture specific and is defined in the file *sys/conf/options.<arch>*.

device *name* [, *name* [...]]
devices *name* [, *name* [...]]

Configures the specified devices for inclusion into the kernel image. Devices that are common to all architectures are defined in the file *sys/conf/files*. Devices that are specific to architecture *arch* are defined in the file *sys/conf/files.<arch>*.

env *filename*

Specifies a filename containing a kernel environment definition.

The kernel will augment this compiled-in environment with the environment prepared for it at boot time by loader(8). Environment variables specified in the loader(8) environment will take precedence over environment variables specified in *filename*, and environment variables specified in the dynamic environment take precedence over both of these.

loader_env.disabled=1 may be specified in the static environment to disable the loader(8) environment. Disabling the loader(8) should be done with caution and due consideration for whether or not it supplies environment variables needed for properly booting the system.

static_env.disabled=1 may be specified in the loader(8) environment to disable use of the static environment. This option has no effect if specified in any environment after the loader(8) environment is processed. This option is not usable in conjunction with *loader_env.disabled*.

This directive is useful for setting kernel tunables in embedded environments that do not start from loader(8).

All **env** and **envvar** directives will be processed and added to the static environment in reversed order of appearance so that later specified variables properly override earlier specified variables. Note that within *filename*, the first appearance of a given variable will be the first one seen by the kernel, effectively shadowing any later appearances of the same variable within *filename*.

envvar *setting*

Specifies an individual environment setting to be added to the kernel's compiled-in environment. *setting* must be of the form "*name=value*". Optional quotes are supported in both name and value.

All **env** and **envvar** directives will be processed and added to the static environment in reversed order of appearance so that later specified variables properly override earlier specified variables.

files *filename*

Specifies a file containing a list of files specific to that kernel configuration file (a la *files*.<*arch*>).

hints *filename*

Specifies a file to load a static device configuration specification from. From FreeBSD 5.0 onwards, the kernel reads the system's device configuration at boot time (see device.hints(5)). This directive configures the kernel to use the static device configuration listed in *filename*.

Hints provided in this static device configuration will be overwritten in the order in which they're encountered. Hints in the compiled-in environment takes precedence over compiled-in hints, and hints in the environment prepared for the kernel by loader(8) takes precedence over hints in the compiled-in environment.

Once the dynamic environment becomes available, all compiled-in hints will be added to the dynamic environment if they do not already have an override in the dynamic environment. The dynamic environment will then be used for all searches of hints.

static_hints.disabled=1 may be specified in either a compiled-in environment or the loader(8) environment to disable use of these hints files. This option has no effect if specified in any environment after the loader(8) environment is processed.

The file *filename* must conform to the syntax specified by device.hints(5). Multiple hints lines are allowed. The resulting hints will be the files concatenated in reverse order of appearance so that hints in later files properly override hints in earlier files.

ident *name*

Set the kernel name to *name*. At least one **ident** directive is required.

include *filename*

Read subsequent text from file *filename* and return to the current file after *filename* is successfully processed.

machine *arch* [*cpuarch*]

Specifies the architecture of the machine the kernel is being compiled for. Legal values for *arch* include:

arm64	The 64-bit ARM application architecture.
arm	The ARM architecture
amd64	The AMD x86-64 architecture.
i386	The Intel x86 based PC architecture.
powerpc	The IBM PowerPC architecture.

riscv The RISC-V architecture.

If argument *cpuarch* is specified, it points config(8) to the cpu architecture of the machine. When *cpuarch* is not specified, it is assumed to be the same as *arch*. *arch* corresponds to MACHINE. *cpuarch* corresponds to MACHINE_ARCH.

A kernel configuration file may have only one **machine** directive, unless the second one matches the machine argument in the first one exactly.

makeoption *options*
makeoptions *options*

Add *options* to the generated makefile.

The *options* argument is a comma separated list of one or more option specifications. Each option specification has the form

MakeVariableName[=*Value*]
MakeVariableName+=*Value*

and results in the appropriate make(1) variable definition being inserted into the generated makefile. If only the name of the make(1) variable is specified, *value* is assumed to be the empty string.

Example:

```
makeoptions MYMAKEOPTION="foo"
makeoptions MYMAKEOPTION+="bar"
makeoptions MYNULLMAKEOPTION
```

maxusers *number*

This optional directive is used to configure the size of some kernel data structures. The parameter *number* can be 0 (the default) or an integer greater than or equal to 2. A value of 0 indicates that the kernel should configure its data structures according to the size of available physical memory. If auto configuration is requested, the kernel will set this tunable to a value between 32 and 384 for 32-bit systems, or scale the value higher based on available memory for 64-bit systems.

As explained in tuning(7), this tunable can also be set at boot time using loader(8).

nocpu *cputype*

Remove the specified CPU from the list of previously selected CPUs. This directive can be used to cancel the effect of **cpu** directives in files included using **include**.

nodevice *name* [, *name* [...]]
nodevices *name* [, *name* [...]]

Remove the specified devices from the list of previously selected devices. This directive can be used to cancel the effects of **device** or **devices** directives in files included using **include**.

nomakeoption *name*
nomakeoptions *name*

Removes previously defined make(1) option *name* from the kernel build. This directive can be used to cancel the effects of **makeoption** directives in files included using **include**.

nooption *name* [, *name* [...]]
nooptions *name* [, *name* [...]]

Remove the specified kernel options from the list of previously defined options. This directive can be used to cancel the effects of **option** or **options** directives in files included using **include**.

option *optionspec* [, *optionspec* [...]]
options *optionspec* [, *optionspec* [...]]

Add compile time kernel options to the kernel build. Each option specification has the form

name[=*value*]

If *value* is not specified, it is assumed to be NULL. Options common to all architectures are specified in the file *sys/conf/options*. Options specific to architecture *arch* are specified in the file *sys/conf/options*.<*arch*>.

FILES

sys/compile/NAME	Compile directory created from a kernel configuration.
sys/conf/Makefile.arch	*Makefile* fragments for architecture *arch*.
sys/conf/files	Devices common to all architectures.
sys/conf/files.arch	Devices for architecture *arch*.
sys/conf/options	Options common to all architectures.
sys/conf/options.arch	Options for architecture *arch*.

SEE ALSO

kenv(1), make(1), device.hints(5), loader.conf(5), config(8), kldload(8), loader(8)

Samuel J. Leffler and Michael J. Karels, *Building 4.4BSD Kernels with Config*.

HISTORY

The config(8) utility first appeared in 4.1BSD, and was subsequently revised in 4.4BSD.

The kernel configuration mechanism changed further in FreeBSD 4.0 and FreeBSD 5.0, moving toward an architecture supporting dynamic kernel configuration.