NAME

getconf - retrieve standard configuration variables

SYNOPSIS

getconf -a [*file*]
getconf [**-v** *environment*] *path_var file*
getconf [**-v** *environment*] *system_var*

DESCRIPTION

The **getconf** utility prints the values of POSIX or X/Open path or system configuration variables to the standard output. If a variable is undefined, the string "undefined" is output.

The first form of the command displays all of the path or system configuration variables to standard output. If *file* is provided, all path configuration variables are reported for *file* using pathconf(2). Otherwise, all system configuration variables are reported using confstr(3) and sysconf(3).

The second form of the command, with two mandatory arguments, retrieves file- and file system-specific configuration variables using pathconf(2). The third form, with a single argument, retrieves system configuration variables using confstr(3) and sysconf(3), depending on the type of variable. As an extension, the second form can also be used to query static limits from <*limits.h*>.

All sysconf(3) and pathconf(2) variables use the same name as the manifest constants defined in the relevant standard C-language bindings, including any leading underscore or prefix. That is to say, *system_var* might be ARG_MAX or _POSIX_VERSION, as opposed to the sysconf(3) names _SC_ARG_MAX or _SC_POSIX_VERSION. Variables retrieved from confstr(3) have the leading '_CS_' stripped off; thus, _CS_PATH is queried by a *system_var* of "PATH".

Programming Environments

The **-v** *environment* option specifies a IEEE Std 1003.1-2001 ("POSIX.1") programming environment under which the values are to be queried. This option currently does nothing, but may in the future be used to select between 32-bit and 64-bit execution environments on platforms which support both. Specifying an environment which is not supported on the current execution platform gives undefined results.

The standard programming environments are as follows:

POSIX_V6_ILP32_OFF32	Exactly 32-bit integer, long, pointer, and file offset. **Supported platforms**: None.
POSIX_V6_ILP32_OFFBIG	Exactly 32-bit integer, long, and pointer; at least 64-bit file offset.

Supported platforms: IA32, PowerPC.

POSIX_V6_LP64_OFF64 Exactly 32-bit integer; exactly 64-bit long, pointer, and file offset.
Supported platforms: AMD64, SPARC64.

POSIX_V6_LPBIG_OFFBIG At least 32-bit integer; at least 64-bit long, pointer, and file offset.
Supported platforms: None.

The command:

```
getconf POSIX_V6_WIDTH_RESTRICTED_ENVS
```

returns a newline-separated list of environments in which the width of certain fundamental types is no greater than the width of the native C type *long*. At present, all programming environments supported by FreeBSD have this property. Several of the confstr(3) variables provide information on the necessary compiler and linker flags to use the standard programming environments described above.

EXIT STATUS

The **getconf** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

The command:

```
getconf PATH
```

will display the system default setting for the PATH environment variable.

The command:

```
getconf NAME_MAX /tmp
```

will display the maximum length of a filename in the */tmp* directory.

The command:

```
getconf -v POSIX_V6_LPBIG_OFFBIG LONG_MAX
```

will display the maximum value of the C type *long* in the POSIX_V6_LPBIG_OFFBIG programming environment, if the system supports that environment.

DIAGNOSTICS

Use of a *system_var* or *path_var* which is completely unrecognized is considered an error, causing a diagnostic message to be written to standard error. One which is known but merely undefined does not result in an error indication. The **getconf** utility recognizes all of the variables defined for IEEE Std 1003.1-2001 ("POSIX.1"), including those which are not currently implemented.

SEE ALSO

pathconf(2), confstr(3), sysconf(3)

STANDARDS

The **getconf** utility is expected to be compliant with IEEE Std 1003.1-2001 ("POSIX.1").

HISTORY

The **getconf** utility first appeared in FreeBSD 5.0.

AUTHORS

Garrett A. Wollman <*wollman@lcs.mit.edu*>