NAME

textdump - textdump kernel dumping facility

SYNOPSIS

options DDB
options KDB

options TEXTDUMP_PREFERRED
options TEXTDUMP_VERBOSE

DESCRIPTION

The **textdump** facility allows the capture of kernel debugging information to disk in a human-readable rather than the machine-readable form normally used with kernel memory dumps and minidumps. This representation, while less complete in that it does not capture full kernel state, can provide debugging information in a more compact, portable, and persistent form than a traditional dump. By combining **textdump** with other ddb(4) facilities, such as scripting and output capture, detailed bug information can be captured in a fully automated manner.

FORMAT

textdump data is stored in a dump partition in the same style as a regular memory dump, and will be automatically extracted by savecore(8) if present on boot.

textdump files are stored in the tar(5) format, and consist of one or more text files, each storing a particular type of debugging output. The following parts may be present:

ddb.txt Captured ddb(4) output, if the capture facility has been used. May be disabled by clearing the *debug.ddb.textdump.do_ddb* sysctl.

config.txt Kernel configuration, if **options INCLUDE_CONFIG_FILE** has been compiled into the kernel. May be disabled by clearing the *debug.ddb.textdump.do_config* sysctl.

msgbuf.txt
Kernel message buffer, including recent console output if the capture facility has been used. May be disabled by clearing the *debug.ddb.textdump.do_msgbuf* sysctl.

panic.txt Kernel panic string, if the kernel panicked before the dump was generated. May be disabled by clearing the *debug.ddb.textdump.do_panic* sysctl.

version.txt Kernel version string. My be disabled by clearing the *debug.ddb.textdump.do_version* sysctl.

Kernel textdumps may be extracted using tar(1).

CONFIGURATION

The **textdump** facility is enabled as part of the kernel debugger using **options KDB** and **options DDB**. By default, kernel dumps generated on panic or via explicit requests for a dump will be regular memory dumps; however, by using the **textdump set** command in ddb(4), or by setting the *debug.ddb.textdump.pending* sysctl to 1 using sysctl(8), it is possible to request that the next dump be a textdump. One may also directly trigger a textdump in ddb(4) by running the command **textdump dump**.

If at the ddb(4) command line, the commands **textdump set**, **textdump status**, and **textdump unset** may be used to set, query, and clear the textdump pending flag.

As with regular kernel dumps, a dump partition must be automatically or manually configured using dumpon(8).

Additional kernel config(8) options:

TEXTDUMP_PREFERRED sets textdumps to be the default manner of doing dumps. This means there will be no need to sysctl(8) or use the **textdump set** ddb(8) commands.

TEXTDUMP_VERBOSE will have the textdump facility be more verbose about each file it is emitting as well as other diagnostics useful to debug the textdump facility itself.

EXAMPLES

In the following example, the script *kdb.enter.panic* will run when the kernel debugger is entered as a result of a panic, enable output capture, dump several useful pieces of debugging information, and then invoke panic in order to force a kernel dump to be written out followed by a reboot:

```
script kdb.enter.panic=textdump set; capture on; show allpcpu; bt;
  ps; alltrace; show alllocks; textdump dump; reset
```

In the following example, the script *kdb.enter.witness* will run when the kernel debugger is entered as a result of a witness violation, printing lock-related information for the user:

```
script kdb.enter.witness=show locks
```

These scripts may also be configured using the ddb(8) utility.

SEE ALSO

tar(1), ddb(4), tar(5), ddb(8), dumpon(8), savecore(8), sysctl(8)

HISTORY

The **textdump** facility first appeared in FreeBSD 7.1.

AUTHORS

The **textdump** facility was created by Robert N. M. Watson.