NAME

dconschat - user interface to dcons(4)

SYNOPSIS

dconschat [**-brvwRT1**] [**-e** *escape-char*] [**-h** *hz*] [**-C** *console_port*] [**-G** *gdb_port*] [**-M** *core*] [**-N** *system*]
dconschat [**-brvwR1**] [**-h** *hz*] [**-C** *console_port*] [**-G** *gdb_port*] [**-a** *address*] [**-u** *bus_num*] **-t** *target_eui64*

DESCRIPTION

The **dconschat** utility is designed to provide a way for users to access dcons(4) (dumb console device) on a local or remote system. The **dconschat** utility interacts with dcons(4) using kvm(3) or firewire(4), and interacts with the user over TTY or TCP/IP. To access remote dcons(4) using firewire(4), you have to specify target EUI64 address using the **-t** option. Physical DMA should be enabled on the target machine for access via FireWire.

The **dconschat** utility and the dcons(4) driver communicate using 2 ports, one for the console port and another for remote gdb(1) (*ports/devel/gdb*) port. Users are supposed to access **dconschat** using TTY, telnet(1) and gdb(1) (*ports/devel/gdb*). You can specify listen ports for console and gdb(1) (*ports/devel/gdb*) port using the **-C** and **-G** options respectively. The port number 0 has special meaning that current TTY (stdin/stdout) is used instead of TCP/IP. A negative port number will disable the port. By analogy with pty(4) device, the dcons(4) acts as a slave device and **dconschat** acts as a master device with **telnetd**.

Typed characters are normally transmitted directly to dcons(4). A escape character (the default is '~') appearing as the first character of a line is an escape signal; the following are recognized:

~. Drop the connection and exit.

~^G Invoke kgdb on the terminal on which dconschat is running.

~^R Reset the target over FireWire if a reset address is registered in Configuration ROM.

~^Z Suspend the dconschat process.

The following options are supported.

-b Translate Ctrl-C to ALT_BREAK (CR + '~' + Ctrl-B) on gdb(1) (*ports/devel/gdb*) port.

-r Replay old buffer on connection.

-v Verbose debug output. Multiple **-v** options increase verbosity.

-w Listen on a wildcard address rather than localhost.

-R Read-only. Do not write anything to the dcons(4) buffer.

-T Enable ad-hoc workaround for the TELNET protocol to remove unnecessary byte sequences. It should be set when you access **dconschat** using telnet(1).

-1 One-shot. Read available buffer, then exit. This implies the **-r** option.

-e *escape-char*
Specify escape character. The default is '~'.

-h *hz* Specify polling rate. The default value is 100.

-C *console_port*
Specify the console port. The default value is 0 (stdin/stdout).

-G *gdb_port*
Specify gdb(1) (*ports/devel/gdb*) port. The default value is -1 (disabled).

-M *core*
Specify core file.

-N *system*
Specify system file such as */boot/kernel/kernel*.

-t *target_eui64*
Specify the 64-bit extended unique identifier of the target, and use FireWire to access remote dcons(4).

-a *address*
Specify the physical I/O address of the dcons(4) buffer. See dcons(4) for details. If this option is not specified, **dconschat** tries to get the address from the Configuration ROM on the target. You are supposed to enable dcons_crom(4) on the target to omit this option.

-u *bus_num*
Specify FireWire bus number. The default is 0.

FILES

/dev/fwmem0.0

/dev/mem
/dev/kmem

EXAMPLES

To use **dconschat** with FireWire for remote dcons(4), you have to specify the EUI64 of the target. You can obtain EUI64 by running fwcontrol(8) without options. The first EUI64 is of the host running fwcontrol(8) and others on the bus follow.

```
# fwcontrol
2 devices (info_len=2)
node        EUI64         status
   1  77-66-55-44-33-22-11-00      0
   0  00-11-22-33-44-55-66-77      1
```

The EUI64 does not change unless you change the hardware as the ethernet address.

Now we can run **dconschat**.

```
# dconschat -br -G 12345 -t 00-11-22-33-44-55-66-77
```

You will get console output of the target and login prompt if a getty(8) is running on dcons(4). You can break to DDB with ALT_BREAK (CR + '~' + Ctrl-B) if DDB and ALT_BREAK_TO_DEBUGGER are enabled in the target kernel. To quit the session, type CR + '~' + '.' in the console port.

Using gdb(1) (*ports/devel/gdb*) port is almost the same as remote gdb(1) (*ports/devel/gdb*) over serial line except using TCP/IP instead of */dev/cu**. See *On-line Kernel Debugging Using Remote GDB* section of *The FreeBSD Developers Handbook* and gdb(4) for details.

```
% gdb -k kernel.debug
(kgdb) target remote :12345
```

Once gdb(1) (*ports/devel/gdb*) is attached and you specified the **-b** option to **dconschat**, typing Ctrl-C in gdb(1) (*ports/devel/gdb*) causes a break to debugger.

The following command gets the console log from the crash dump:

```
# dconschat -1 -M vmcore.0 -N kernel.0
```

If you want access to the console using telnet(1), try the following:

```
# dconschat -rTC 5555 &
# telnet localhost 5555
```

You may want to keep logging console output of several machines. **conserver-com** in the Ports collection may help you. Insert the following lines in *conserver.cf*:

```
console local {
        master localhost;
        type exec;
        exec /usr/sbin/dconschat -rh 25;
}
console remote {
        master localhost;
        type exec;
        exec /usr/sbin/dconschat -rh 25 -t 00-11-22-33-44-55-66-77;
}
```

SEE ALSO

gdb(1) (*ports/devel/gdb*), telnet(1), kvm(3), dcons(4), dcons_crom(4), ddb(4), firewire(4), fwohci(4), gdb(4), eui64(5), fwcontrol(8)

AUTHORS

Hidetoshi Shimokawa <*simokawa@FreeBSD.org*>

BUGS

This utility is currently under development.