

## NAME

dtach – simple program that emulates the detach feature of screen.

## SYNOPSIS

```
dtach -a <socket> <options>
dtach -A<socket> <options> <command...>
dtach -c<socket> <options> <command...>
dtach -n<socket> <options> <command...>
dtach -N<socket> <options> <command...>
dtach -p<socket>
```

## DESCRIPTION

**dtach** is a program that emulates the detach feature of screen. It is designed to be transparent and un-intrusive; it avoids interpreting the input and output between attached terminals and the program under its control. Consequently, it works best with full-screen applications such as emacs.

**dtach** is intended for users who want the detach feature of screen without the other overhead of screen. It is tiny, does not use many libraries, and stays out of the way as much as possible.

## SESSIONS

A session in **dtach** is a single instance in which a program is running under the control of **dtach**. The program is disassociated from the original terminal, and is thus protected from your original terminal being disconnected for some reason.

Other instances of **dtach** can attach themselves to a particular session. Input and output is copied between the program running in the **dtach** session, and the attached terminals.

**dtach** avoids interpreting the communication stream between the program and the attached terminals; it instead relies on the ability of the attached terminals to manage the screen.

Sessions are represented by Unix-domain sockets in the filesystem. No other permission checking other than the filesystem access checks is performed. **dtach** creates a master process that monitors the session socket, the program, and any attached terminals.

## MODES

**dtach** has several modes of operation. It can create a new session in which a program is executed, or it can attach to an existing session. The first argument specifies which mode **dtach** should operate in.

- a Attach to an existing session. **dtach** attaches itself to the session specified by <socket>. After the attach is completed, the window size of the current terminal is sent to the master process, and a redraw is also requested.
- A Attach to an existing session, or create a new one. **dtach** first tries to attach to the session specified by <socket> if possible. If the attempt to open the socket fails, **dtach** tries to create a new session before attaching to it.
- c Creates a new session. A new session is created in which the specified program is executed. **dtach** then tries to attach itself to the newly created session.
- n Creates a new session, without attaching to it. A new session is created in which the specified program is executed. **dtach** does not try to attach to the newly created session, however, and exits instead.
- N Creates a new session, without attaching to it or daemonizing. A new session is created in which the specified program is executed. **dtach** does not try to attach to the newly created session, however, and will stay in the foreground until the program exits.

- p** Copies the contents of standard input to a session. **dtach** connects to the session specified by *<socket>*, copies the contents of standard input to the session, and then exits. dtach will not scan the input for a detach character.

## OPTIONS

**dtach** has a few options that allow you to modify its behavior. Each attaching process can have separate settings for these options, which allows for some flexibility.

- e <char>**

Sets the detach character to *<char>*. When the detach character is pressed, **dtach** detaches itself from the current session and exits. The process running in the session is unaffected by the detach. By default, the detach character is set to `^` (Ctrl-`^`).

- E** Disables the detach character. **dtach** does not try to scan input from the terminal for a detach character. The only way to detach from the session is then by sending the attaching process an appropriate signal.

- r <method>**

Sets the redraw method to *<method>*. The valid methods are *none*, *ctrl\_l*, or *winch*.

*none* disables redrawing completely, *ctrl\_l* sends a Ctrl L character to the program if the terminal is in character-at-a-time and no-echo mode, and *winch* forces a WINCH signal to be sent to the program.

When creating a new session, the specified method is used as the default redraw method for the session. If not specified, the *ctrl\_l* method is used.

- z** Disables processing of the suspend key. Normally, **dtach** will suspend itself when the suspend key is pressed. With this option, the suspend character is sent to the session instead of being handled by **dtach**.

## EXAMPLES

The following example creates a new session that has the detach character and suspend processing disabled. A socket is created in the `/tmp` directory for the session.

```
$ dtach -c /tmp/foozle -Ez bash
```

The following example attaches to the `/tmp/foozle` session if it exists, and if not, creates a new session using `/tmp/foozle` as the socket for the session. Processing of the suspend character is also disabled for the attach instance.

```
$ dtach -A /tmp/foozle -z bash
```

The following example attaches to the `/tmp/foozle` session, using the *winch* redraw method to redraw the screen.

```
$ dtach -a /tmp/foozle -r winch
```

The following example creates a new session and sets the default redraw method for the session to the *winch* redraw method.

```
$ dtach -c /tmp/foozle -r winch bash
```

**AUTHOR**

Ned T. Crigler <crigler@users.sourceforge.net>.

**SEE ALSO**

**screen**(1)