NAME

dpv - stream data from stdin or multiple paths with dialog progress view

SYNOPSIS

dpv [options] [*bytes***:**]*label*
dpv [options] **-m** [*bytes1*]:*label1 path1* [[*bytes2***:**]*label2 path2* ...]

DESCRIPTION

dpv provides a dialog progress view, allowing a user to see current throughput rate and total data transferred for one or more streams.

The **dpv** utility has two main modes for processing input.

The default input mode, without '**-m**', **dpv** reads bytes from standard input. A label for the data must be provided.

The secondary input mode, with '**-m**', **dpv** reads multiple paths (up to 2047 or "ARG_MAX/2-1"), sequentially.

Data read in either mode is either thrown away (default), sent to a spawned instance of the program specified via '**-x** *cmd*', or sent to a unique file specified by '**-o** *file*'.

With or without '**-m**', progress is displayed using one of dialog(3) (default), dialog(1) (see '**-D**'), or instead Xdialog(1) (*ports/x11/xdialog*) (see '**-X**').

The following options are available:

-a *text* Display *text* below the file progress indicator(s).

-b *backtitle* Display *backtitle* on the backdrop, at top-left, behind the dialog widget. When using Xdialog(1) (*ports/x11/xdialog*), this is displayed inside the window (at the top) followed by a separator line.

-D Do not use the default interface of dialog(3), but instead spawn an instance of dialog(1). The path to dialog(1) is taken from the DIALOG environment variable or simply "dialog" if unset or NULL.

-d Debug mode. Print dialog prompt data to standard out and provide additional debugging on standard error.

-h Produce a short syntax usage with brief option descriptions and exit. Output is produced on standard error.

-I *format* Customize the multi-file format string used to update the status line. Ignored when using either '**-D**' or '**-X**' which lack the ability to display the status line (containing bytes/rate/thread information). Default value is "%'10lli bytes read @ %'9.1f bytes/sec. [%i/%i busy/wait]". This format is used when handling more than one file.

-i *format* Customize the single-file format string used to update the status line. Ignored when using either '**-D**' or '**-X**' which lack the ability to display the status line (containing bytes/rate/thread information). Default value is "%'10lli bytes read @ %'9.1f bytes/sec.". This format is used when handling one file.

-k Keep tite. Prevent visually distracting initialization/exit routines for scripts running dialog(1) several times.

-L *size* Label size. If negative, shrink to longest label width.

-l Line mode. Read lines from input instead of bytes.

-m Multi-input mode. Instead of reading bytes from standard input, read from a set of paths (one for each label). By default, each path is processed sequentially in the order given.

-N No overrun. If enabled, stop reading known-length inputs when input reaches stated length.

-n *num* Display at-most *num* progress indicators per screen. If zero, display as many as possible. If negative, only display the main progress indicator. Default is 0. Maximum value is 10.

-o *file* Output data to *file*. The first occurrence of '%s' (if any) in '*file*' will be replaced with the *label* text.

-P *size* Mini-progressbar size. If negative, don't display mini-progressbars (only the large overall progress indicator is shown). If zero, auto-adjust based on number of files to read. When zero and only one file to read, defaults to -1. When zero and more than one file to read, defaults to 17.

-p *text* Display *text* above the file progress indicator(s).

-T Test mode. Simulate reading a number of bytes, divided evenly across the number of files, while stepping through each percent value of each file to process. Appends "[TEST

MODE]" to the status line (to override, use '**-u** *format*'). No data is actually read.

-t *title* Display *title* atop the dialog box. Note that if you use this option at the same time as '**-X**' and '**-b** *backtitle*', the *backtitle* and *title* are effectively switched (see BUGS section below).

-U *num* Update status line *num* times per-second. Default value is '2'. A value of '0' disables status line updates. If negative, update the status line as fast as possible. Ignored when using either '**-D**' or '**-X**' which lack the ability to display the status line (containing bytes/rate/thread information).

-w Wide mode. Allows long *text* arguments used with '**-p**' and '**-a**' to bump the dialog width. Prompts wider than the maximum width will wrap unless using Xdialog(1) (*ports/x11/xdialog*); see BUGS section below.

-X Enable X11 mode by using Xdialog(1) (*ports/x11/xdialog*) instead of dialog(1) or dialog(3).

-x *cmd* Execute *cmd* (via sh(1)) and send it data that has been read. Data is available to *cmd* on standard input. With '**-m**', *cmd* is executed once for each *path* argument. The first occurrence of '%s' (if any) in '*cmd*' will be replaced with the *label* text.

ENVIRONMENT

The following environment variables are referenced by **dpv**:

DIALOG Override command string used to launch dialog(1) (requires '**-D**') or Xdialog(1) (*ports/x11/xdialog*) (requires '**-X**'); default is either 'dialog' (for '**-D**') or 'Xdialog' (for '**-X**').

DIALOGRC If set and non-NULL, path to '.dialogrc' file.

HOME If '$DIALOGRC' is either not set or NULL, used as a prefix to '.dialogrc' (i.e., '$HOME/.dialogrc').

USE_COLOR If set and NULL, disables the use of color when using dialog(1). Does not apply to Xdialog(1) (*ports/x11/xdialog*).

DEPENDENCIES

If using '**-D**', dialog(1) is required.

If using '**-X**', Xdialog(1) (*ports/x11/xdialog*) is required.

FILES

$HOME/.dialogrc

EXAMPLES

Simple example to show how fast yes(1) produces lines (usually about ten-million per-second; your results may vary):

```
yes | dpv -l yes
```

Display progress while timing how long it takes yes(1) to produce a half-billion lines (usually under one minute; your results may vary):

```
time yes | dpv -Nl 500000000:yes
```

An example to watch how quickly a file is transferred using nc(1):

```
dpv -x "nc -w 1 somewhere.com 3000" -m label file
```

A similar example, transferring a file from another process and passing the expected size to **dpv**:

```
cat file | dpv -x "nc -w 1 somewhere.com 3000" 12345:label
```

A more complicated example:

```
tar cf - . | dpv -x "gzip -9 > out.tgz" \
        $( du -s . | awk '{print $1 * 1024}' ):label
```

Taking an image of a disk:

```
dpv -o disk-image.img -m label /dev/ada0
```

Writing an image back to a disk:

```
dpv -o /dev/ada0 -m label disk-image.img
```

Zeroing a disk:

```
dpv -o /dev/md42 "Zeroing md42" < /dev/zero
```

SEE ALSO

dialog(1), sh(1), Xdialog(1) (*ports/x11/xdialog*), dialog(3)

HISTORY

A **dpv** utility first appeared in FreeBSD 10.2.

AUTHORS

Devin Teske <dteske@FreeBSD.org>

BUGS

Xdialog(1) (*ports/x11/xdialog*), when given both ‘**--title** *title*’ (see above ‘**-t** *title*’) and ‘**--backtitle** *backtitle*’ (see above ‘**-b** *backtitle*’), displays the backtitle in place of the title and vice-versa.

Xdialog(1) (*ports/x11/xdialog*) does not wrap long prompt texts received after initial launch. This is a known issue with the ‘--gauge’ widget in Xdialog(1) (*ports/x11/xdialog*).

dialog(1) does not display the first character after a series of escaped escape-sequences (e.g., ‘‘\\n’’ produces ‘‘\’’ instead of ‘‘\n’’). This is a known issue with dialog(1) and does not affect dialog(3) or Xdialog(1) (*ports/x11/xdialog*).

If your application ignores USE_COLOR when set and NULL before calling dpv(1) with color escape sequences anyway, dialog(3) and dialog(1) may not render properly. Workaround is to detect when USE_COLOR is set and NULL and either not use color escape sequences at that time or use unset(1) [sh(1)] or unsetenv(1) [csh(1)] to unset USE_COLOR, forcing interpretation of color sequences. This does not effect Xdialog(1) (*ports/x11/xdialog*), which renders the color escape sequences as plain text. See "embedded "\Z" sequences" in dialog(1) for additional information.