

NAME

djvused - Multi-purpose DjVu document editor.

SYNOPSIS

djvused [*options*] *djvufile*

DESCRIPTION

Program **djvused** is a powerful command line tool for manipulating multi-page documents, creating or editing annotation chunks, creating or editing hidden text layers, pre-computing thumbnail images, and more. The program first reads the DjVu document *djvufile* and executes a number of djvused commands.

Djvused commands can be read from a specific file (when option **-f** is specified), read from the command line (when option **-e** is specified), or read from the standard input (the default).

OPTIONS

- v** Cause **djvused** to print a command line prompt before reading commands and a brief message describing how each command was executed. This option is very useful for debugging djvused scripts and also for interactively entering djvused commands on the standard input.

- f** *scriptfile*
Cause **djvused** to read commands from file *scriptfile*.

- e** *command*
Cause **djvused** to execute the commands specified by the option argument *commands*. It is advisable to surround the djvused commands by single quotes in order to prevent unwanted shell expansion.

- s** Cause **djvused** to save the file *djvufile* after executing the specified commands. This is similar to executing command **save** immediately before terminating the program.

- u** Cause **djvused** to print hidden text and annotations as UTF-8 instead of encoding non-ASCII characters with octal escape sequences for maximal portability. This option is convenient for manually editing or viewing the djvused output. This option also causes the emission of an UTF-8 BOM under Windows.

- n Cause **djvused** to disregard save commands. This is useful for debugging djvused scripts without overwriting files on your disk.

DJVUSED EXAMPLES

There are many ways to use program **djvused**. The following examples illustrate some common uses of this program.

Obtaining the size of a page

Command **size** outputs the width and height of the selected pages using a HTML friendly syntax. For instance, the following command prints the size of page 3 of document *myfile.djvu*.

```
djvused myfile.djvu -e 'select 3; size'
```

Extracting the hidden text

Command **print-pure-txt** outputs the text associated with a page or a document. For instance, the following shell command outputs the text for the entire document. Lines and pages are delimited by the usual control characters.

```
djvused myfile.djvu -e 'print-pure-txt'
```

Command **print-txt** produces a more extensive output describing the structure and the location of the text components. The syntax of this output is described later in this man page. For instance, the following shell command outputs extended text information for page 3 of document *myfile.djvu*.

```
djvused myfile.djvu -e 'select 3; print-txt'
```

Extracting the annotations

Annotation data can be extracted using command **print-ant**. The syntax of the annotation data is described later in this man page. For instance, the following shell command outputs the annotation data for the first page of document **myfile.djvu**.

```
djvused myfile.djvu -e 'select 1; print-ant'
```

Command **print-ant** only prints the annotations stored in the selected component file. Command

print-merged-ant also retrieves annotations from all the component files referenced by the current page (using INCL chunks) and prints the merged information.

Dumping/restoring annotations and text

Three commands, **output-txt**, **output-ant**, and **output-all**, produce djvused scripts. For instance, the following shell command produces a djvused script, *myfile.dsed*, that recreates all the text and annotation data in document *myfile.djvu*.

```
djvused myfile.djvu -e 'output-all' > myfile.dsed
```

Script *myfile.dsed* is a text file that can be easily edited. The following shell command then recreates the text and annotation information in file *myfile.djvu*.

```
djvused myfile.djvu -f myfile.dsed -s
```

Extracting a page

Both commands **save-page** and **save-page-with** create a DjVu file representing the selected component file of a document. The following shell command, for instance, creates a file *p05.djvu* containing page 5 of document *myfile.djvu*.

```
djvused myfile.djvu -e 'select 5; save-page p05.djvu'
```

Each page of a document might import data from another component file using the so-called inclusion (INCL) chunks. Command **save-page** then produces a file with unresolved references to imported data. Such a file should then be made part of a multi-page document containing the required data in other component files. On the other hand, command **save-page-with** copies all the imported data into the output file. This file is directly usable. Yet collecting several such files into a multi-page document might lead to useless data replication.

Pre-computing thumbnails

Commands **set-thumbnails** constructs thumbnails that can be later displayed by DjVu viewers. The following shell command, for instance, computes thumbnails of size *64x64* pixels for all pages of file *myfile.djvu*.

djvused *myfile.djvu* -e 'set-thumbnails 64' -s

DJVUSED COMMANDS

Command lines might contain zero, one, or more djvused commands and an optional comment.

Multiple djvused commands must be separated by a semicolon character ';'. Comments are introduced by the '#' character and extend until the end of the command line.

Selection commands

Multi-page DjVu documents are composed of a number of component files. Most component files describe a specific page of a document. Some component files contain information shared by several pages such as shared image data, shared annotations or thumbnails. Many djvused commands operate on selected component files. All component files are initially selected. The following commands are useful for changing the selection.

n Print the total number of pages in the document.

ls List all component files in the document. Each line contains an optional page number, a letter describing the component file type, the size of the component file, and identifier of the component file. Component file type letters **P**, **I**, **A**, and **T** respectively stand for page data, shared image data, shared annotation data, and thumbnail data. Page numbers are only listed for component files containing page data. When it is set, the optional page title (see command **set-page-title** below) is displayed after the component file identifier.

select [*fileid*]

Select the component file identified by argument *fileid*. Argument *fileid* must be either a page number or a component file identifier. The **select** command selects all component files when the argument *fileid* is omitted.

select-shared-ant

Select a component file containing shared annotations. Only one such component file is supported by the current DjVu software. This component file usually contains annotations pertaining to the whole document as opposed to specific pages. An error message is displayed if there is no such component file.

create-shared-ant

Create and select a component file containing shared annotations. This command only selects the shared annotation component file if such a component file already exists. Otherwise it creates a new shared annotation component file and makes sure that it is imported by all pages in the

document.

showsel

Shows the currently selected component files with the same format as command **ls**.

Text and annotation commands**print-pure-txt**

Print the text stored in the hidden text layer of the selected pages. A similar capability is offered by program **djvutxt**. Structural information is sometimes represented by control characters. Text from different pages is delimited by form feed characters ("**\f**"). Lines are delimited by newline characters ("**\n**"). Columns, regions, and paragraphs are sometimes delimited by vertical tab ("**\013**"), group separators ("**\035**") and unit separators ("**\037**") respectively.

print-txt

Prints extensive hidden text information for the selected pages. This information describes the structure of the text on the document page and locates the structural elements in the page image. The syntax of this output is described later in this man page.

remove-txt

Remove the hidden text information from the selected component files. For instance, executing commands **select** and **remove-txt** removes all hidden text information from the DjVu document.

set-txt [*djvusedtxtfile*]

Insert hidden text information into the selected pages. The optional argument *djvusedtxtfile* names a file containing the hidden text information. This file must contain data similar to what is produced by command **print-txt**. When the optional argument is omitted, the program reads the hidden text information from the djvused script until reaching an end-of-file or a line containing a single period.

output-txt

Prints a djvused script that reconstructs the hidden text information for the selected pages. This script can later be edited and executed by invoking program **djvused** with option **-f**.

print-ant

Prints the annotations of the selected component file. The annotation data is represented using a simple syntax described later in this document.

print-merged-ant

Merge the annotations stored in the selected component files with the annotations imported from

other component files such as the shared annotation component file.. The annotation data is represented using a simple syntax described later in this document.

remove-ant

Remove the annotation information from the selected component files. For instance, executing commands **select** and **remove-ant** removes all annotation information from the DjVu document.

set-ant [*djvusedantfile*]

Insert annotations into the selected component file. The optional argument *djvusedantfile* names a file containing the annotation data. This file must contain data similar to what is produced by command **print-ant**. When the optional argument is omitted, the program reads the annotation data from the djvused script itself until reaching an end-of-file or a line containing a single period.

output-ant

Print a djvused script that reconstructs the annotation information for the selected pages. This script can later be edited and executed by invoking program **djvused** with option **-f**.

print-meta

Print the metadata part of the annotations for the selected component file. This command displays a subset of the information printed by command **print-ant** using a different syntax. metadata are organized as key-value pairs. Each printed line contains the key name such as **author**, **title**, etc., followed by a tab character ("\t") and a double-quoted string representing the UTF-8 encoded metadata value.

remove-meta

Remove the metadata part of the annotations of the selected component files.

set-meta [*djvusedmetafile*]

Set the metadata part of the annotations of the selected component file. The remaining part of the annotations is left unchanged. The optional argument *djvusedmetafile* names a file containing the metadata. This file must contain data similar to what is produced by command **print-meta**. When the optional argument is omitted, the program reads the annotation data from the djvused script itself until reaching an end-of-file or a line containing a single period.

print-xmp

Print the XMP metadata string contained in the annotation chunk of the selected component file. This command displays in fact a subset of the information printed by command **print-ant**.

remove-xmp

Removes the XMP tag from the annotation chunk of the selected component file.

set-xmp [*xmpfile*]

Set the XMP metadata part of the annotations of the selected component file. The remaining part of the annotations is left unchanged. The optional argument *xmpfile* names a file containing the XMP metadata in a format similar to that produced by command **print-xmp**. When the optional argument is omitted, the program reads the XMP annotation data from the djvused script itself until reaching an end-of-file or a line containing a single period.

output-all

Print a djvused script that reconstructs both the hidden text and the annotation information for the selected pages. This script can later be edited and executed by invoking program **djvused** with option **-f**.

Outline/bookmarks commands**print-outline**

Print the outline of the document. Nothing is printed if the document contains no outline.

remove-outline

Removes the outline from the document.

set-outline [*djvusedoutlinefile*]

Insert outline information into the document. The optional argument *djvusedoutlinefile* names a file containing the outline information. This file must contain data similar to what is produced by command **print-outline**. When the optional argument is omitted, the program reads the hidden text information from the djvused script until reaching an end-of-file or a line containing a single period.

Thumbnail commands**set-thumbnails** *sz*

Compute thumbnails of size *sz*x*sz* pixels and insert them into the document. DjVu viewers can later display these thumbnails very efficiently without need to download the data for each page. Typical thumbnail size range from 48 to 128 pixels.

remove-thumbnails

Remove the pre-computed thumbnails from the DjVu document. New thumbnails can then be computed using command **set-thumbnails**.

Save commands

The above commands only modify the memory image of the DjVu document. The following commands provide means to save the modified data into the file system.

save

Save the modified DjVu document back into the input file *dvufile* specified by the arguments of the program **djvused**. Nothing is done if the DjVu file was not modified. Passing option **-s** program **djvused** is equivalent to executing command **save** before exiting the program.

save-bundled *filename*

Save the current DjVu document as a bundled multi-page DjVu document named *filename*. A similar capability is offered by program **djvmcvt**.

save-indirect *filename*

Save the current DjVu document as an indirect multi-page DjVu document. The index file of the indirect document will be named **filename**. All other files composing the indirect document will be saved into the same directory as the index file. A similar capability is offered by program **djvmcvt**.

save-page *filename*

Save the selected component file into DjVu file *filename*. The selected component file might import data from another component file using the so-called inclusion (INCL) chunks. This command then produces a file with unresolved references to imported data. Such a file should then be made part of a multi-page document containing the required data in other component files.

save-page-with *filename*

Save the selected component file into DjVu file *filename*. All data imported from other component files is copied into the output file as well. This command always produces a usable DjVu file. On the other hand, collecting several such files into a multi-page document might lead to useless data replication.

Miscellaneous commands**help**

Display a help message listing all commands supported by **djvused**.

dump

Display the EA IFF 85 structure of the document or of the selected component file. A similar capability is offered by program **djvudump**.

size

Display the width and the height of the selected pages. The dimensions of each page are displayed using a syntax suitable for direct insertion into the <EMBED...></EMBED> tags. This command also displays the default page orientation when it is different from zero.

set-rotation [*+-*]*rot*

Changes the default orientation of the selected pages. The orientation is expressed as an integer in range 0..3 representing a number of 90 degree counter-clockwise rotations. When the argument is preceded by a sign + or -, argument *rot* counts how many additional 90 degree counter-clockwise rotations should be applied to the page. Otherwise, argument *rot* represents the desired absolute page orientation. Only DjVu pages can be rotated. Pages represented as a raw IW44 image cannot be rotated.

set-dpi *dpi*

Sets the resolution of the page image in dots per inch. Argument *dpi* should be in range 25..6000.

set-page-title *title*

Sets a page title for the selected page. When page titles are available, recent versions of the DjVuLibre viewers display these page titles instead of page numbers and also accept them in page selection options. Command **ls** can be used to see both the page titles and page identifiers. To unset a page title, simply make it equal to the page identifier.

DJVUSED FILE FORMATS

Djvused uses a simple parenthesized syntax to represent both annotations and hidden text.

- * This syntax is the native syntax used by DjVu for storing annotations. Program **djvused** simply compresses the annotation data using the **bzz(1)** algorithm.
- * This syntax differs from the native syntax used by DjVu for storing the hidden text. Program **djvused** performs the translations between the compact binary representation used by DjVu and the easily modifiable parenthesized syntax.

General syntax

Djvused files are ASCII text files. The legal characters in djvused files are the printable ASCII characters and the space, tab, cr, and nl characters. Using other characters has undefined results.

Djvused files are composed of a sequence of expressions separated by blank characters (space, tab, cr, or nl). There are four kind of expressions, namely integers, symbols, strings and lists.

Integers:

Integer numbers are represented by one or more digits, with the usual interpretation.

Symbols:

Symbols, or identifiers, are sequences of printable ascii characters representing a name or a

keyword. Acceptable characters are the alpha-numeric characters, the underscore "_", the minus character "-", and the hash character "#". Names should not begin with a digit or a minus character.

Strings:

Strings denote an arbitrary sequence of bytes, usually interpreted as a sequence of UTF-8 encoded characters. Strings in djvused files are similar to strings in the C language. They are surrounded by double quote characters. Certain sequences of characters starting with a backslash ("\") have a special meaning. A backslash followed by letter "a", "b", "t", "n", "v", "f", "r", "\", and stands for the ascii character BEL(007), BS(008), HT(009), LF(010), VT(011), FF(012), CR(013), BACKSLASH(134) and DOUBLEQUOTE(042) respectively. A backslash followed by one to three digits stands for the byte whose octal code is expressed by the digits. All other backslash sequences are illegal. All non printable ascii characters must be escaped.

Lists:

Lists are sequence of expressions separated by blanks and surrounded by parentheses. All expressions types are acceptable within a list, including sub-lists.

Hidden text syntax

The building blocks of the hidden text syntax are lists representing each structural component of the hidden text. Structural components have the following form:

(type xmin ymin xmax ymax ...)

The symbol *type* must be one of **page**, **column**, **region**, **para**, **line**, **word**, or **char**, listed here by decreasing order of importance. The integers *xmin*, *ymin*, *xmax*, and *ymax* represent the coordinates of a rectangle indicating the position of the structural component in the page. Coordinates are measured in pixels and have their origin at the bottom left corner of the page. The remaining expressions in the list either is a single string representing the encoded text associated with this structural component, or is a sequence of structural components with a lesser type.

The hidden text for each page is simply represented by a single structural element of type **page**.

Various level of structural information are acceptable. For instance, the page level component might only specify a page level string, or might only provide a list of lines, or might provide a full hierarchy down to the individual characters.

Outline/Bookmark syntax

The outline syntax is a single list of the form

(bookmarks ...)

The first element of the list is symbol **bookmarks**. The subsequent elements are lists representing the toplevel outline entries. Each outline entry is represented by a list with the following form:

(title url ...)

The string *title* is the title of the outline entry. The destination string *url* can be either an arbitrary percent encoded URL, or composed of the hash character ("#") followed by a page name or number, or composed of the question mark character ("?") followed by cgi-style arguments interpreted by the djvu viewer. The remaining expressions in the list describe subentries of this outline entry.

Annotation syntax

Annotations are represented by a sequence of annotation expressions. The following annotation expressions are recognized:

(background *color*)

Specify the color of the viewer area surrounding the DjVu image. Colors are represented with the X11 hexadecimal syntax **#RRGGBB**. For instance, **#000000** is black and **#FFFFFF** is white.

(zoom *zoomvalue*)

Specify the initial zoom factor of the image. Argument *zoomvalue* can be one of **stretch**, **one2one**, **width**, **page**, or composed of the letter **d** followed by a number in range 1 to 999 representing a zoom factor (such as in **d300** or **d150** for instance.)

(mode *modevalue*)

Specify the initial display mode of the image. Argument *modevalue* is one of **color**, **bw**, **fore**, or **back**.

(align *horzalign vertalign*)

Specify how the image should be aligned on the viewer surface. By default the image is located in the center. Argument *horzalign* can be one of **left**, **center**, or **right**. Argument *vertalign* can be one of **top**, **center**, or **bottom**.

(maparea *url comment area ...*)

Define an hyper-link for the specified destination.

Argument *url* can have one of the following forms:

href
(**url href target**)

where *href* is a string representing the destination and *target* is a string representing the target frame for the hyper-link, as defined by the HTML anchor tag <A>. The destination string *href* can be either an arbitrary percent encoded URL, or composed of the hash character ("#") followed by a page name or number, or composed of the question mark character ("?") followed by cgi-style arguments interpreted by the djvu viewer. Page numbers may be prefixed with an optional sign to represent a page displacement. For instance the strings "#-1" and "#+1" can be used to access the previous page and the next page.

Argument *comment* is a string that might be displayed by the viewer when the user moves the mouse over the hyper-link.

Argument *area* defines the shape and the location of the hyperlink. The following forms are recognized:

(**rect** *xmin ymin width height*)
(**oval** *xmin ymin width height*)
(**poly** *x0 y0 x1 y1 ...*)
(**text** *xmin ymin width height*)
(**line** *x0 y0 x1 y1*)

All parameters are numbers representing coordinates. Coordinates are measured in pixels and have their origin at the bottom left corner of the page.

The remaining expressions in the **maparea** list represent the visual effect associated with the hyper-link.

A first set of options defines how borders are drawn for **rect**, **oval**, **polygon**, or **text** hyperlink areas.

(**none**)
(**xor**)

(border *color*)
(shadow_in [*thickness*])
(shadow_out [*thickness*])
(shadow_ein [*thickness*])
(shadow_eout [*thickness*])

where parameter *color* has syntax **#RRGGBB** as described above, and parameter *thickness* is an integer in range 1 to 32. The last four border options are only supported for **rect** hyperlink areas. Although the border mode defaults to (**xor**), it is wise to always specify the border mode. Border options do not apply to **line** areas.

When a border option is specified, the border becomes visible when the user moves the mouse over the hyperlink. The border may be made always visible by using the following option:

(border_avis)

The following two options may be used with **rect** hyperlink areas. The complete area will be highlighted using the specified color at the specified opacity (0-100, default 50). Some viewers (e.g., **djview4**) support opacities in range 0-200 with 200 representing a fully opaque color.

(hilite *color*)
(opacity *op*)

This is often used with an empty URL for simply emphasizing a specific segment of an image.

The following three options may be used with line areas to specify an optional ending arrow, the line width and color. The default is a black line with width 1 and without arrow.

(arrow)
(width *w*)
(lineclr *color*)

Finally the following three options can be used with text areas. The default background color is transparent. The default text color is black. The **pushpin** option indicates that the text is symbolized by a small pushpin icon. Clicking the icon reveals the text.

(backclr *bkcolor*)

(textclr *txtcolor*)

(pushpin)

(metadata ... (key value) ...)

Define metadata entries. Each entry is identified by a symbol *key* representing the nature of the meta data entry. The string *value* represents the value associated with the corresponding key. Two sets of keys are noteworthy: keys borrowed from the BibTeX bibliography system, and keys borrowed from the PDF DocInfo metadata. BibTeX keys are always expressed in lowercase, such as **year**, **booktitle**, **editor**, **author**, etc.. DocInfo keys start with an uppercase letter, such as **Title**, **Author**, **Subject**, **Creator**, **Produced**, **Trapped**, **CreationDate**, and **ModDate**. The values associated with the last two keys should be dates expressed according to RFC 3339.

LIMITATIONS

The current version of program **djvused** only supports selecting one component file or all component files. There is no way to select only a few component files.

CREDITS

This program was initially written by L on Bottou <leonb@users.sourceforge.net> and was improved by Yann Le Cun <profshadoko@users.sourceforge.net>, Florin Nicsa, Bill Riemers <docbill@sourceforge.net> and many others.

SEE ALSO

djvu(1), **djvutxt(1)**, **djvmcvt(1)**, **djvudump(1)**, **bzz(1)**, Emacs djvused front end **djvu.el** on GNU Elpa repository.