

Data conversion



- [Text data conversion tools](#)
 - [Converting a text file with iconv](#)
 - [Checking file to be UTF-8 with iconv](#)
 - [Converting file names with iconv](#)
 - [EOL conversion](#)
 - [TAB conversion](#)
 - [Editors with auto-conversion](#)
 - [Plain text extraction](#)
 - [Highlighting and formatting plain text data](#)
- [XML data](#)
 - [Basic hints for XML](#)
 - [XML processing](#)
 - [The XML data extraction](#)
 - [The XML data lint](#)
- [Type setting](#)
 - [roff typesetting](#)
 - [TeX/LaTeX](#)
 - [Pretty print a manual page](#)
 - [Creating a manual page](#)
- [Printable data](#)
 - [Ghostscript](#)
 - [Merge two PS or PDF files](#)
 - [Printable data utilities](#)
 - [Printing with CUPS](#)
- [The mail data conversion](#)
 - [Mail data basics](#)
- [Graphic data tools](#)
 - [Graphic data tools \(metapackage\)](#)
 - [Graphic data tools \(GUI\)](#)

- [Graphic data tools \(CLI\)](#)
- [Miscellaneous data conversion](#)

Tools and tips for converting data formats on the Debian system are described.

Standard based tools are in very good shape but support for proprietary data formats are limited.

Text data conversion tools

Following packages for the text data conversion caught my eyes.

package	popcon	size	keyword	description
<code>libc6</code>	@-@popcon1@-@	@-@psize1@-@	charset	text encoding converter between locales by <code>iconv</code> (1) (fundamental)
<code>recode</code>	@-@popcon1@-@	@-@psize1@-@	charset+eol	text encoding converter between locales (versatile, more aliases and features)
<code>konwert</code>	@-@popcon1@-@	@-@psize1@-@	charset	text encoding converter between locales (fancy)
<code>nkf</code>	@-@popcon1@-@	@-@psize1@-@	charset	character set translator for Japanese
<code>tcs</code>	@-@popcon1@-@	@-@psize1@-@	charset	character set translator
<code>unaccent</code>	@-@popcon1@-@	@-@psize1@-@	charset	replace accented letters by their unaccented equivalent
<code>tofromdos</code>	@-@popcon1@-@	@-@psize1@-@	eol	text format converter between DOS and Unix: <code>fromdos</code> (1) and <code>todos</code> (1)
<code>macutils</code>	@-@popcon1@-@	@-@psize1@-@	eol	text format converter between Macintosh and Unix: <code>frommac</code> (1) and <code>tomac</code> (1)

List of text data conversion tools

Converting a text file with iconv

“ [!TIP] `iconv` (1) is provided as a part of the `libc6` package and it is always available on practically all Unix-like systems to convert the encoding of characters.

You can convert encodings of a text file with `iconv` (1) by the following.

```
$ iconv -f encoding1 -t encoding2 input.txt >output.txt
```

Encoding values are case insensitive and ignore "-" and "_" for matching. Supported encodings can be checked by the "`iconv -l`" command.

encoding value	usage
ASCII	American Standard Code for Information Interchange , 7 bit code w/o accented characters
UTF-8	current multilingual standard for all modern OSs
ISO-8859-1	old standard for western European languages, ASCII + accented characters
ISO-8859-2	old standard for eastern European languages, ASCII + accented characters
ISO-8859-15	old standard for western European languages, ISO-8859-1 with euro sign
CP850	code page 850, Microsoft DOS characters with graphics for western European languages, ISO-8859-1 variant
CP932	code page 932, Microsoft Windows style Shift-JIS variant for Japanese
CP936	code page 936, Microsoft Windows style GB2312 , GBK or GB18030 variant for Simplified Chinese
CP949	code page 949, Microsoft Windows style EUC-KR or Unified Hangul Code variant for Korean
CP950	code page 950, Microsoft Windows style Big5 variant for Traditional Chinese
CP1251	code page 1251, Microsoft Windows style encoding for the Cyrillic alphabet
CP1252	code page 1252, Microsoft Windows style ISO-8859-15 variant for western European languages

encoding value	usage
KOI8-R	old Russian UNIX standard for the Cyrillic alphabet
ISO-2022-JP	standard encoding for Japanese email which uses only 7 bit codes
eucJP	old Japanese UNIX standard 8 bit code and completely different from Shift-JIS
Shift-JIS	JIS X 0208 Appendix 1 standard for Japanese (see CP932)

List of encoding values and their usage {#list-of-encoding-values}

“ [!NOTE] Some encodings are only supported for the data conversion and are not used as locale values (???).

For character sets which fit in single byte such as [ASCII](#) and [ISO-8859](#) character sets, the [character encoding](#) means almost the same thing as the character set.

For character sets with many characters such as [JIS X 0213](#) for Japanese or [Universal Character Set \(UCS, Unicode, ISO-10646-1\)](#) for practically all languages, there are many encoding schemes to fit them into the sequence of the byte data.

- [EUC](#) and [ISO/IEC 2022 \(also known as JIS X 0202\)](#) for Japanese
- [UTF-8](#), [UTF-16/UCS-2](#) and [UTF-32/UCS-4](#) for Unicode

For these, there are clear differentiations between the character set and the character encoding.

The [code page](#) is used as the synonym to the character encoding tables for some vendor specific ones.

“ [!NOTE] Please note most encoding systems share the same code with ASCII for the 7 bit characters. But there are some exceptions. If you are converting old Japanese C programs and URLs data from the casually-called shift-JIS encoding format to UTF-8 format, use "CP932" as the encoding name instead of "shift-JIS" to get the expected results: 0x5C → "\" and 0x7E → "~". Otherwise, these are converted to wrong characters.

[!TIP] `recode`(1) may be used too and offers more than the combined functionality of `iconv`(1), `fromdos`(1), `todos`(1), `frommac`(1), and `tomac`(1). For more, see "`info recode`".

Checking file to be UTF-8 with iconv

You can check if a text file is encoded in UTF-8 with `iconv`(1) by the following.

```
$ iconv -f utf8 -t utf8 input.txt >/dev/null || echo "non-UTF-8 found"
```

“ [!TIP] Use "`--verbose`" option in the above example to find the first non-UTF-8 character.

Converting file names with iconv

Here is an example script to convert encoding of file names from ones created under older OS to modern UTF-8 ones in a single directory.

```
#!/bin/sh
ENCDN=iso-8859-1
for x in *;
do
  mv "$x" "$(echo "$x" | iconv -f $ENCDN -t utf-8)"
done
```

The "`$ENCDN`" variable specifies the original encoding used for file names under older OS as in [List of encoding values and their usage](#).

For more complicated case, please mount a filesystem (e.g. a partition on a disk drive) containing such file names with proper encoding as the `mount`(8) option (see [???](#)) and copy its entire contents to another filesystem mounted as UTF-8 with "`cp -a`" command.

EOL conversion

The text file format, specifically the end-of-line (EOL) code, is dependent on the platform.

platform	EOL code	control	decimal	hexadecimal
Debian (unix)	LF	<code>^J</code>	10	0A
MSDOS and Windows	CR-LF	<code>^M^J</code>	13 10	0D 0A
Apple's Macintosh	CR	<code>^M</code>	13	0D

List of EOL styles for different platforms

The EOL format conversion programs, `fromdos` (1), `todos` (1), `frommac` (1), and `tomac` (1), are quite handy. `recode` (1) is also useful.

“ [!NOTE] Some data on the Debian system, such as the wiki page data for the `python-moinmoin` package, use MSDOS style CR-LF as the EOL code. So the above rule is just a general rule.

“ [!NOTE] Most editors (eg. `vim`, `emacs`, `gedit`, ...) can handle files in MSDOS style EOL transparently.

“ [!TIP] The use of "`sed -e '/\r$/!s$/\r/'`" instead of `todos` (1) is better when you want to unify the EOL style to the MSDOS style from the mixed MSDOS and Unix style. (e.g., after merging 2 MSDOS style files with `diff3` (1).) This is because `todos` adds CR to all lines.

TAB conversion

There are few popular specialized programs to convert the tab codes.

function	<code>bsdmainutils</code>	<code>coreutils</code>
expand tab to spaces	<code>"col -x"</code>	<code>expand</code>
unexpand tab from spaces	<code>"col -h"</code>	<code>unexpand</code>

List of TAB conversion commands from `bsdmainutils` and `coreutils` packages

`indent` (1) from the `indent` package completely reformats whitespaces in the C program.

Editor programs such as `vim` and `emacs` can be used for TAB conversion, too. For example with `vim`, you can expand TAB with "`:set expandtab`" and "`:%retab`" command sequence. You can

revert this with `":set noexpandtab"` and `":%retab!"` command sequence.

Editors with auto-conversion

Intelligent modern editors such as the `vim` program are quite smart and copes well with any encoding systems and any file formats. You should use these editors under the UTF-8 locale in the UTF-8 capable console for the best compatibility.

An old western European Unix text file, "`u-file.txt`", stored in the latin1 (iso-8859-1) encoding can be edited simply with `vim` by the following.

```
$ vim u-file.txt
```

This is possible since the auto detection mechanism of the file encoding in `vim` assumes the UTF-8 encoding first and, if it fails, assumes it to be latin1.

An old Polish Unix text file, "`pu-file.txt`", stored in the latin2 (iso-8859-2) encoding can be edited with `vim` by the following.

```
$ vim '+e ++enc=latin2 pu-file.txt'
```

An old Japanese unix text file, "`ju-file.txt`", stored in the eucJP encoding can be edited with `vim` by the following.

```
$ vim '+e ++enc=eucJP ju-file.txt'
```

An old Japanese MS-Windows text file, "`jw-file.txt`", stored in the so called shift-JIS encoding (more precisely: CP932) can be edited with `vim` by the following.

```
$ vim '+e ++enc=CP932 ++ff=dos jw-file.txt'
```

When a file is opened with "`++enc`" and "`++ff`" options, "`:w`" in the Vim command line stores it in the original format and overwrite the original file. You can also specify the saving format and the file name in the Vim command line, e.g., "`:w ++enc=utf8 new.txt`".

Please refer to the `mbyte.txt` "multi-byte text support" in `vim` on-line help and [List of encoding values and their usage](#) for locale values used with "`++enc`".

The `emacs` family of programs can perform the equivalent functions.

Plain text extraction

The following reads a web page into a text file. This is very useful when copying configurations off the Web or applying basic Unix text tools such as `grep` (1) on the web page.

```
$ w3m -dump https://www.remote-site.com/help-info.html >textfile
```

Similarly, you can extract plain text data from other formats using the following.

package	popcon	size	keyword	function
w3m	@-@popcon1@-@	@-@psize1@-@	html → text	HTML to text converter with the " <code>w3m -dump</code> " command
html2text	@-@popcon1@-@	@-@psize1@-@	html → text	advanced HTML to text converter (ISO 8859-1)
lynx	@-@popcon1@-@	@-@psize1@-@	html → text	HTML to text converter with the " <code>lynx -dump</code> " command
elinks	@-@popcon1@-@	@-@psize1@-@	html → text	HTML to text converter with the " <code>elinks -dump</code> " command
links	@-@popcon1@-@	@-@psize1@-@	html → text	HTML to text converter with the " <code>links -dump</code> " command
links2	@-@popcon1@-@	@-@psize1@-@	html → text	HTML to text converter with the " <code>links2 -dump</code> " command
catdoc	@-@popcon1@-@	@-@psize1@-@	MSWord → text,TeX	convert MSWord files to plain text or TeX
antiword	@-@popcon1@-@	@-@psize1@-@	MSWord → text,ps	convert MSWord files to plain text or ps
unhtml	@-@popcon1@-@	@-@psize1@-@	html → text	remove the markup tags from an HTML file
odt2txt	@-@popcon1@-@	@-@psize1@-@	odt → text	converter from OpenDocument Text to text

List of tools to extract plain text data

Highlighting and formatting plain text data

You can highlight and format plain text data by the following.

package	popcon	size	keyword	description
<code>vim-runtime</code>	@-@popcon1@-@	@-@psize1@-@	highlight	Vim MACRO to convert source code to HTML with " <code>:source \$VIMRUNTIME/syntax/html.vim</code> "
<code>cxref</code>	@-@popcon1@-@	@-@psize1@-@	<code>c</code> → <code>html</code>	converter for the C program to latex and HTML (C language)
<code>src2tex</code>	@-@popcon1@-@	@-@psize1@-@	highlight	convert many source codes to TeX (C language)
<code>source-highlight</code>	@-@popcon1@-@	@-@psize1@-@	highlight	convert many source codes to HTML, XHTML, LaTeX, Texinfo, ANSI color escape sequences and DocBook files with highlight (C++)
<code>highlight</code>	@-@popcon1@-@	@-@psize1@-@	highlight	convert many source codes to HTML, XHTML, RTF, LaTeX, TeX or XSL-FO files with highlight (C++)
<code>grc</code>	@-@popcon1@-@	@-@psize1@-@	<code>text</code> → <code>color</code>	generic colouriser for everything (Python)
<code>pandoc</code>	@-@popcon1@-@	@-@psize1@-@	<code>text</code> → <code>any</code>	general markup converter (Haskell)
<code>python3-docutils</code>	@-@popcon1@-@	@-@psize1@-@	<code>text</code> → <code>any</code>	ReStructured Text document formatter to XML (Python)
<code>markdown</code>	@-@popcon1@-@	@-@psize1@-@	<code>text</code> → <code>html</code>	Markdown text document formatter to (X)HTML (Perl)
<code>asciidocctor</code>	@-@popcon1@-@	@-@psize1@-@	<code>text</code> → <code>any</code>	AsciiDoc text document formatter to XML/HTML (Ruby)
<code>python3-sphinx</code>	@-@popcon1@-@	@-@psize1@-@	<code>text</code> → <code>any</code>	ReStructured Text based document publication system (Python)
<code>hugo</code>	@-@popcon1@-@	@-@psize1@-@	<code>text</code> → <code>html</code>	Markdown based static site publication system (Go)

List of tools to highlight plain text data

XML data

[The Extensible Markup Language \(XML\)](#) is a markup language for documents containing structured information.

See introductory information at [XML.COM](#).

- ["What is XML?"](#)
- ["What Is XSLT?"](#)
- ["What Is XSL-FO?"](#)
- ["What Is XLink?"](#)

Basic hints for XML

XML text looks somewhat like [HTML](#). It enables us to manage multiple formats of output for a document. One easy XML system is the `docbook-xsl` package, which is used here.

Each XML file starts with standard XML declaration as the following.

```
<?xml version="1.0" encoding="UTF-8"?>
```

The basic syntax for one XML element is marked up as the following.

```
<name attribute="value">content</name>
```

XML element with empty content is marked up in the following short form.

```
<name attribute="value" />
```

The "`attribute="value"`" in the above examples are optional.

The comment section in XML is marked up as the following.

```
<!-- comment -->
```

Other than adding markups, XML requires minor conversion to the content using predefined entities for following characters.

predefined entity	character to be converted into
"	" : quote
'	' : apostrophe
<	< : less-than
>	> : greater-than
&	& : ampersand

List of predefined entities for XML

- “ [!CAUTION] "<" or "&" can not be used in attributes or elements.
- “ [!NOTE] When SGML style user defined entities, e.g. "&some-tag;", are used, the first definition wins over others. The entity definition is expressed in "<!ENTITY some-tag "entity value">".
- “ [!NOTE] As long as the XML markup are done consistently with certain set of the tag name (either some data as content or attribute value), conversion to another XML is trivial task using [Extensible Stylesheet Language Transformations \(XSLT\)](#).

XML processing

There are many tools available to process XML files such as [the Extensible Stylesheet Language \(XSL\)](#).

Basically, once you create well formed XML file, you can convert it to any format using [Extensible Stylesheet Language Transformations \(XSLT\)](#).

The [Extensible Stylesheet Language for Formatting Objects \(XSL-FO\)](#) is supposed to be solution for formatting. The `fop` package is new to the Debian `main` archive due to its dependence to the [Java programming language](#). So the LaTeX code is usually generated from XML using XSLT and the LaTeX system is used to create printable file such as DVI, PostScript, and PDF.

package	popcon	size	keyword	description
---------	--------	------	---------	-------------

<code>docbook-xml</code>	@-@popcon1@-@	@-@psize1@-@	xml	XML document type definition (DTD) for DocBook
<code>docbook-xsl</code>	@-@popcon1@-@	@-@psize1@-@	xml/xslt	XSL stylesheets for processing DocBook XML to various output formats with XSLT
<code>xsltproc</code>	@-@popcon1@-@	@-@psize1@-@	xslt	XSLT command line processor (XML→XML, HTML, plain text, etc.)
<code>xmlto</code>	@-@popcon1@-@	@-@psize1@-@	xml/xslt	XML-to-any converter with XSLT
<code>fop</code>	@-@popcon1@-@	@-@psize1@-@	xml/xsl-fo	convert Docbook XML files to PDF
<code>dblatex</code>	@-@popcon1@-@	@-@psize1@-@	xml/xslt	convert Docbook files to DVI, PostScript, PDF documents with XSLT
<code>dbtoepub</code>	@-@popcon1@-@	@-@psize1@-@	xml/xslt	DocBook XML to .epub converter

List of XML tools

Since XML is subset of [Standard Generalized Markup Language \(SGML\)](#), it can be processed by the extensive tools available for SGML, such as [Document Style Semantics and Specification Language \(DSSSL\)](#).

package	popcon	size	keyword	description
<code>openjade</code>	@-@popcon1@-@	@-@psize1@-@	<code>dsssl</code>	ISO/IEC 10179:1996 standard DSSSL processor (latest)
<code>docbook-dsssl</code>	@-@popcon1@-@	@-@psize1@-@	<code>xml/dsssl</code>	DSSSL stylesheets for processing DocBook XML to various output formats with DSSSL
<code>docbook-utils</code>	@-@popcon1@-@	@-@psize1@-@	<code>xml/dsssl</code>	utilities for DocBook files including conversion to other formats (HTML, RTF, PS, man, PDF) with <code>docbook2*</code> commands with DSSSL

“ [!TIP] [GNOME](#)'s `yelp` is sometimes handy to read [DocBook](#) XML files directly since it renders decently on X.

The XML data extraction

You can extract HTML or XML data from other formats using followings.

package	popcon	size	keyword	description
<code>man2html</code>	@-@popcon1@-@	@-@psize1@-@	<code>manpage</code> → <code>html</code>	converter from manpage to HTML (CGI support)
<code>doclifter</code>	@-@popcon1@-@	@-@psize1@-@	<code>troff</code> → <code>xml</code>	converter from troff to DocBook XML
<code>texi2html</code>	@-@popcon1@-@	@-@psize1@-@	<code>texi</code> → <code>html</code>	converter from Texinfo to HTML
<code>info2www</code>	@-@popcon1@-@	@-@psize1@-@	<code>info</code> → <code>html</code>	converter from GNU info to HTML (CGI support)
<code>wv</code>	@-@popcon1@-@	@-@psize1@-@	<code>MSWord</code> → <code>any</code>	document converter from Microsoft Word to HTML, LaTeX, etc.
<code>unrtf</code>	@-@popcon1@-@	@-@psize1@-@	<code>rtf</code> → <code>html</code>	document converter from RTF to HTML, etc
<code>wp2x</code>	@-@popcon1@-@	@-@psize1@-@	<code>WordPerfect</code> → <code>any</code>	WordPerfect 5.0 and 5.1 files to TeX, LaTeX, troff, GML and HTML

The XML data lint

For non-XML HTML files, you can convert them to XHTML which is an instance of well formed XML. XHTML can be processed by XML tools.

Syntax of XML files and goodness of URLs found in them may be checked.

package	popcon	size	function	description
---------	--------	------	----------	-------------

libxml2-utils	@-@popcon1@-@	@-@psize1@-@	xml ↔ html ↔ xhtml	command line XML tool with xmllint(1) (syntax check, reformat, lint, ...)
tidy	@-@popcon1@-@	@-@psize1@-@	xml ↔ html ↔ xhtml	HTML syntax checker and reformatter
weblint-perl	@-@popcon1@-@	@-@psize1@-@	lint	syntax and minimal style checker for HTML
linklint	@-@popcon1@-@	@-@psize1@-@	link check	fast link checker and web site maintenance tool

List of XML pretty print tools

Once proper XML is generated, you can use XSLT technology to extract data based on the mark-up context etc.

Type setting

The Unix troff program originally developed by AT&T can be used for simple typesetting. It is usually used to create manpages.

TeX created by Donald Knuth is a very powerful type setting tool and is the de facto standard. LaTeX originally written by Leslie Lamport enables a high-level access to the power of TeX.

package	popcon	size	keyword	description
texlive	@-@popcon1@-@	@-@psize1@-@	(La)TeX	TeX system for typesetting, previewing and printing
lilypond	@-@popcon1@-@	@-@psize1@-@	(La)TeX	music typesetter
groff	@-@popcon1@-@	@-@psize1@-@	troff	GNU troff text-formatting system

List of type setting tools

roff typesetting

Traditionally, roff is the main Unix text processing system. See roff(7), groff(7), groff(1), grotty(1), troff(1), groff_mdoc(7), groff_man(7), groff_ms(7), groff_me(7), groff_mm(7), and "

```
info groff".
```

You can read or print a good tutorial and reference on "-me" [macro](#) in `/usr/share/doc/groff/` by installing the `groff` package.

“ [!TIP] `"groff -Tascii -me -"` produces plain text output with [ANSI escape code](#). If you wish to get manpage like output with many `"^H"` and `"_"`, use `"GROFF_NO_SGR=1 groff -Tascii -me -"` instead.

“ [!TIP] To remove `"^H"` and `"_"` from a text file generated by `groff`, filter it by `"col -b -x"`.

TeX/LaTeX

The [TeX Live](#) software distribution offers a complete TeX system. The `texlive` metapackage provides a decent selection of the [TeX Live](#) packages which should suffice for the most common tasks.

There are many references available for [TeX](#) and [LaTeX](#).

- [The teTeX HOWTO: The Linux-teTeX Local Guide](#)
- `tex`(1)
- `latex`(1)
- `texdoc`(1)
- `texdoctk`(1)
- "The TeXbook", by Donald E. Knuth, (Addison-Wesley)
- "LaTeX - A Document Preparation System", by Leslie Lamport, (Addison-Wesley)
- "The LaTeX Companion", by Goossens, Mittelbach, Samarin, (Addison-Wesley)

This is the most powerful typesetting environment. Many [SGML](#) processors use this as their back end text processor. [Lyx](#) provided by the `lyx` package and [GNU TeXmacs](#) provided by the `texmacs` package offer nice [WYSIWYG](#) editing environment for [LaTeX](#) while many use [Emacs](#) and [Vim](#) as the choice for the source editor.

There are many online resources available.

- The TEX Live Guide - TEX Live 2007 (`"/usr/share/doc/texlive-doc-base/english/texlive-en/live.html"`) (`texlive-doc-base` package)

- [A Simple Guide to Latex/Lyx](#)
- [Word Processing Using LaTeX](#)

When documents become bigger, sometimes TeX may cause errors. You must increase pool size in `" /etc/texmf/texmf.cnf "` (or more appropriately edit `" /etc/texmf/texmf.d/95NonPath "` and run `update-texmf (8)`) to fix this.

“ [!NOTE] The TeX source of "The TeXbook" is available at [www.ctan.org tex-archive site for texbook.tex](http://www.ctan.org/tex-archive/site/texbook.tex). This file contains most of the required macros. I heard that you can process this document with `tex (1)` after commenting lines 7 to 10 and adding `" \input manmac \proofmodefalse "`. It's strongly recommended to buy this book (and all other books from Donald E. Knuth) instead of using the online version but the source is a great example of TeX input!

Pretty print a manual page

You can print a manual page in PostScript nicely by one of the following commands.

```
$ man -Tps some_manpage | lpr
```

Creating a manual page

Although writing a manual page (manpage) in the plain [troff](#) format is possible, there are few helper packages to create it.

package	popcon	size	keyword	description
<code>docbook-to-man</code>	@-@popcon1@-@	@-@psize1@-@	<code>SGML</code> → <code>manpage</code>	converter from DocBook SGML into roff man macros
<code>help2man</code>	@-@popcon1@-@	@-@psize1@-@	<code>text</code> → <code>manpage</code>	automatic manpage generator from --help
<code>info2man</code>	@-@popcon1@-@	@-@psize1@-@	<code>info</code> → <code>manpage</code>	converter from GNU info to POD or man pages
<code>txt2man</code>	@-@popcon1@-@	@-@psize1@-@	<code>text</code> → <code>manpage</code>	convert flat ASCII text to man page format

List of packages to help creating the manpage

Printable data

Printable data is expressed in the [PostScript](#) format on the Debian system. [Common Unix Printing System \(CUPS\)](#) uses Ghostscript as its rasterizer backend program for non-PostScript printers.

Printable data may also be expressed in the [PDF](#) format on the recent Debian system.

PDF files can displayed and its form entries may be filled using GUI viewer tools such as [Evince](#) and [Okular](#) (see [???](#)); and modern browsers such as [Chromium](#).

PDF files can be edited using some graphics tools such as [LibreOffice](#), [Scribus](#), and [Inkscape](#) (see [Graphic data tools](#)).

“ [!TIP] You can read a PDF file with [GIMP](#) and convert it into [PNG](#) format using higher than 300 dpi resolution. This may be used as a background image for [LibreOffice](#) to produce a desirable altered printout with minimum efforts.

Ghostscript

The core of printable data manipulation is the [Ghostscript PostScript \(PS\)](#) interpreter which generates raster image.

package	popcon	size	description
<code>ghostscript</code>	@-@popcon1@-@	@-@psize1@-@	The GPL Ghostscript PostScript/PDF interpreter
<code>ghostscript-x</code>	@-@popcon1@-@	@-@psize1@-@	GPL Ghostscript PostScript/PDF interpreter - X display support
<code>@libpoppler@</code>	@-@popcon1@-@	@-@psize1@-@	PDF rendering library forked from the xpdf PDF viewer
<code>libpoppler-glib8t64</code>	@-@popcon1@-@	@-@psize1@-@	PDF rendering library (GLib-based shared library)
<code>poppler-data</code>	@-@popcon1@-@	@-@psize1@-@	CMaps for PDF rendering library (for CJK support: Adobe-*)

“ [!TIP] "`gs -h`" can display the configuration of Ghostscript.

Merge two PS or PDF files

You can merge two [PostScript \(PS\)](#) or [Portable Document Format \(PDF\)](#) files using `gs` (1) of Ghostscript.

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite -sOutputFile=bla.ps -f foo1.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

“ [!NOTE] The [PDF](#), which is a widely used cross-platform printable data format, is essentially the compressed [PS](#) format with few additional features and extensions.

“ [!TIP] For command line, `psmerge` (1) and other commands from the `psutils` package are useful for manipulating PostScript documents. `pdftk` (1) from the `pdftk` package is useful for manipulating PDF documents, too.

Printable data utilities

The following packages for the printable data utilities caught my eyes.

package	popcon	size	keyword	description
<code>poppler-utils</code>	@-@popcon1@-@	@-@psize1@-@	<code>pdf</code> → <code>ps</code> , <code>text</code> , ...	PDF utilities: <code>pdftops</code> , <code>pdfinfo</code> , <code>pdfimages</code> , <code>pdftotext</code> , <code>pdffonts</code>
<code>psutils</code>	@-@popcon1@-@	@-@psize1@-@	<code>ps</code> → <code>ps</code>	PostScript document conversion tools
<code>poster</code>	@-@popcon1@-@	@-@psize1@-@	<code>ps</code> → <code>ps</code>	create large posters out of PostScript pages

package	popcon	size	keyword	description
enscript	@-@popcon1@-@	@-@psize1@-@	text→ps, html, rtf	convert ASCII text to PostScript, HTML, RTF or Pretty-Print
a2ps	@-@popcon1@-@	@-@psize1@-@	text → ps	'Anything to PostScript' converter and pretty-printer
pdftk	@-@popcon1@-@	@-@psize1@-@	pdf → pdf	PDF document conversion tool: pdftk
html2ps	@-@popcon1@-@	@-@psize1@-@	html → ps	converter from HTML to PostScript
gnuhtml2latex	@-@popcon1@-@	@-@psize1@-@	html → latex	converter from html to latex
latex2rtf	@-@popcon1@-@	@-@psize1@-@	latex → rtf	convert documents from LaTeX to RTF which can be read by MS Word
ps2eps	@-@popcon1@-@	@-@psize1@-@	ps → eps	converter from PostScript to EPS (Encapsulated PostScript)
e2ps	@-@popcon1@-@	@-@psize1@-@	text → ps	Text to PostScript converter with Japanese encoding support
impose+	@-@popcon1@-@	@-@psize1@-@	ps → ps	PostScript utilities
trueprint	@-@popcon1@-@	@-@psize1@-@	text → ps	pretty print many source codes (C, C++, Java, Pascal, Perl, Pike, Sh, and Verilog) to PostScript. (C language)
pdf2svg	@-@popcon1@-@	@-@psize1@-@	pdf → svg	converter from PDF to Scalable vector graphics format
pdftoipe	@-@popcon1@-@	@-@psize1@-@	pdf → ipe	converter from PDF to IPE's XML format

List of printable data utilities

Printing with CUPS

Both `lp(1)` and `lpr(1)` commands offered by [Common Unix Printing System \(CUPS\)](#) provides options for customized printing the printable data.

You can print 3 copies of a file collated using one of the following commands.

```
$ lp -n 3 -o Collate=True filename

$ lpr -#3 -o Collate=True filename
```

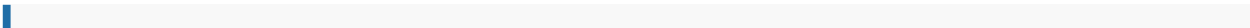
You can further customize printer operation by using printer option such as "`-o number-up=2`", "`-o page-set=even`", "`-o page-set=odd`", "`-o scaling=200`", "`-o natural-scaling=200`", etc., documented at [Command-Line Printing and Options](#).

The mail data conversion

The following packages for the mail data conversion caught my eyes.

package	popcon	size	keyword	description
<code>sharutils</code>	@-@popcon1@-@	@-@psize1@-@	mail	<code>shar(1)</code> , <code>unshar(1)</code> , <code>uuencode(1)</code> , <code>uudecode(1)</code>
<code>mpack</code>	@-@popcon1@-@	@-@psize1@-@	<code>MIME</code>	encoding and decoding of MIME messages: <code>mpack(1)</code> and <code>munpack(1)</code>
<code>tnef</code>	@-@popcon1@-@	@-@psize1@-@	<code>ms-tnef</code>	unpacking MIME attachments of type "application/ms-tnef" which is a Microsoft only format
<code>uudeview</code>	@-@popcon1@-@	@-@psize1@-@	<code>mail</code>	encoder and decoder for the following formats: uuencode , xxencode , BASE64 , quoted printable , and BinHex

List of packages to help mail data conversion



[!TIP] The [Internet Message Access Protocol](#) version 4 (IMAP4) server may be used to move mails out from proprietary mail systems if the mail client software can be configured to use IMAP4 server too.

Mail data basics

Mail ([SMTP](#)) data should be limited to series of 7 bit data. So binary data and 8 bit text data are encoded into 7 bit format with the [Multipurpose Internet Mail Extensions \(MIME\)](#) and the selection of the charset (see [List of encoding values and their usage](#)).

The standard mail storage format is mbox formatted according to [RFC2822 \(updated RFC822\)](#). See `mbox`(5) (provided by the `mutt` package).

For European languages, "`Content-Transfer-Encoding: quoted-printable`" with the ISO-8859-1 charset is usually used for mail since there are not much 8 bit characters. If European text is encoded in UTF-8, "`Content-Transfer-Encoding: quoted-printable`" is likely to be used since it is mostly 7 bit data.

For Japanese, traditionally "`Content-Type: text/plain; charset=ISO-2022-JP`" is usually used for mail to keep text in 7 bits. But older Microsoft systems may send mail data in Shift-JIS without proper declaration. If Japanese text is encoded in UTF-8, [Base64](#) is likely to be used since it contains many 8 bit data. The situation of other Asian languages is similar.

“ [!NOTE] If your non-Unix mail data is accessible by a non-Debian client software which can talk to the IMAP4 server, you may be able to move them out by running your own IMAP4 server.

“ [!NOTE] If you use other mail storage formats, moving them to mbox format is the good first step. The versatile client program such as `mutt`(1) may be handy for this.

You can split mailbox contents to each message using `procmail`(1) and `formail`(1).

Each mail message can be unpacked using `munpack`(1) from the `mpack` package (or other specialized tools) to obtain the MIME encoded contents.

Graphic data tools

Although GUI programs such as `gimp`(1) are very powerful, command line tools such as `imagemagick`(1) are quite useful for automating image manipulation via scripts.

The de facto image file format of the digital camera is the [Exchangeable Image File Format](#) (EXIF) which is the [JPEG](#) image file format with additional metadata tags. It can hold information such as date, time, and camera settings.

[The Lempel-Ziv-Welch \(LZW\) lossless data compression](#) patent has been expired. [Graphics Interchange Format \(GIF\)](#) utilities which use the LZW compression method are now freely available on the Debian system.

“ [!TIP] Any digital camera or scanner with removable recording media works with Linux through [USB storage](#) readers since it follows the [Design rule for Camera Filesystem](#) and uses [FAT](#) filesystem. See [???](#).

Graphic data tools (metapackage)

The following metapackages are good starting points for searching graphics data tools using `aptitude`(8). "[Packages overview for Debian PhotoTools Maintainers](#)" can be another starting point.

package	popcon	size	keyword	description
<code>education-graphics</code>	@-@popcon1@-@	@-@psize1@-@	<code>svg, jpeg, ...</code>	metapackage for teaching graphics and pictural art.
<code>open-font-design-toolkit</code>	@-@popcon1@-@	@-@psize1@-@	<code>ttf, ps, ...</code>	metapackage for open font design

List of graphics data tools (metapackage)

“ [!TIP] Search more image tools using regex "`~Gworks-with::image`" in `aptitude`(8) (see [???](#)).

Graphic data tools (GUI)

The following packages for the GUI graphics data conversion, editing, and organization tools caught my eyes.

package	popcon	size	keyword	description
<code>gimp</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(bitmap)</code>	GNU Image Manipulation Program
<code>xsane</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(bitmap)</code>	GTK-based X11 frontend for SANE (Scanner Access Now Easy)
<code>scribus</code>	@-@popcon1@-@	@-@psize1@-@	ps/pdf/SVG/...	Scribus DTP editor
<code>libreoffice-draw</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(vector)</code>	LibreOffice office suite - drawing
<code>inkscape</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(vector)</code>	SVG (Scalable Vector Graphics) editor
<code>dia</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(vector)</code>	diagram editor (Gtk)
<code>xfig</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(vector)</code>	Facility for Interactive Generation of figures under X11
<code>gocr</code>	@-@popcon1@-@	@-@psize1@-@	<code>image</code> → <code>text</code>	free OCR software
<code>eog</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	Eye of GNOME graphics viewer program
<code>gthumb</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	image viewer and browser (GNOME)
<code>geeqie</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	image viewer using GTK
<code>shotwell</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	digital photo organizer (GNOME)
<code>gwenview</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	image viewer (KDE)
<code>kamera</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	digital camera support for KDE applications
<code>digikam</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	digital photo management application for KDE
<code>darktable</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	virtual lighttable and darkroom for photographers
<code>hugin</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	panorama photo stitcher

package	popcon	size	keyword	description
gtkam	@-@popcon1@-@	@-@psize1@-@	image(Exif)	application for retrieving media from digital cameras (GTK)
fotoxx	@-@popcon1@-@	@-@psize1@-@	image(Exif)	Edit photos and manage a large collection
phototonic	@-@popcon1@-@	@-@psize1@-@	image(Exif)	image viewer and organizer
librecad	@-@popcon1@-@	@-@psize1@-@	DXF,...	2D CAD data editor
freecad	@-@popcon1@-@	@-@psize1@-@	DXF,...	3D CAD data editor
blender	@-@popcon1@-@	@-@psize1@-@	blend,TIFF,VRML,...	3D content editor for animation etc
mm3d	@-@popcon1@-@	@-@psize1@-@	ms3d,obj,dxf,...	OpenGL based 3D model editor
fontforge	@-@popcon1@-@	@-@psize1@-@	ttf,ps,...	font editor for PS, TrueType and OpenType fonts
xgridfit	@-@popcon1@-@	@-@psize1@-@	ttf	program for gridfitting and hinting TrueType fonts

List of graphics data tools (GUI)

Graphic data tools (CLI)

The following packages for the CLI graphics data conversion, editing, and organization tools caught my eyes.

package	popcon	size	keyword	description
imagemagick	@-@popcon1@-@	@-@psize1@-@	image(bitmap)	image manipulation programs
graphicsmagick	@-@popcon1@-@	@-@psize1@-@	image(bitmap)	image manipulation programs (fork of <code>imagemagick</code>)
netpbm	@-@popcon1@-@	@-@psize1@-@	image(bitmap)	graphics conversion tools

package	popcon	size	keyword	description
libheif-examples	@-@popcon1@-@	@-@psize1@-@	heif → jpeg(bitmap)	convert High Efficiency Image File Format (HEIF) to JPEG, PNG, or Y4M formats with <code>heif-convert</code> (1) command
icoutils	@-@popcon1@-@	@-@psize1@-@	png ↔ ico(bitmap)	convert MS Windows icons and cursors to and from PNG formats (favicon.ico)
pstoedit	@-@popcon1@-@	@-@psize1@-@	ps/pdf → image(vector)	PostScript and PDF files to editable vector graphics converter (SVG)
libwmf-bin	@-@popcon1@-@	@-@psize1@-@	Windows/image(vector)	Windows metafile (vector graphics data) conversion tools
fig2sxd	@-@popcon1@-@	@-@psize1@-@	fig → sxd(vector)	convert XFig files to OpenOffice.org Draw format
unpaper	@-@popcon1@-@	@-@psize1@-@	image → image	post-processing tool for scanned pages for OCR
tesseract-ocr	@-@popcon1@-@	@-@psize1@-@	image → text	free OCR software based on the HP's commercial OCR engine
tesseract-ocr-eng	@-@popcon1@-@	@-@psize1@-@	image → text	OCR engine data: tesseract-ocr language files for English text
ocrad	@-@popcon1@-@	@-@psize1@-@	image → text	free OCR software
exif	@-@popcon1@-@	@-@psize1@-@	image(Exif)	command-line utility to show EXIF information in JPEG files
exiv2	@-@popcon1@-@	@-@psize1@-@	image(Exif)	EXIF/IPTC metadata manipulation tool
exiftran	@-@popcon1@-@	@-@psize1@-@	image(Exif)	transform digital camera jpeg images

package	popcon	size	keyword	description
<code>gphoto2</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	The gphoto2 digital camera command-line client
<code>jhead</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	manipulate the non-image part of Exif compliant JPEG (digital camera photo) files
<code>exiftags</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	utility to read Exif tags from a digital camera JPEG file
<code>exifprobe</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code>	read metadata from digital pictures
<code>dcraw</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Raw)</code> → <code>ppm</code>	decode raw digital camera images
<code>findimagedupes</code>	@-@popcon1@-@	@-@psize1@-@	<code>image</code> → <code>fingerprint</code>	find visually similar or duplicate images
<code>ale</code>	@-@popcon1@-@	@-@psize1@-@	<code>image</code> → <code>image</code>	merge images to increase fidelity or create mosaics
<code>imageindex</code>	@-@popcon1@-@	@-@psize1@-@	<code>image(Exif)</code> → <code>html</code>	generate static HTML galleries from images
<code>outguess</code>	@-@popcon1@-@	@-@psize1@-@	jpeg,png	universal Steganographic tool
<code>jpegoptim</code>	@-@popcon1@-@	@-@psize1@-@	jpeg	optimize JPEG files
<code>optipng</code>	@-@popcon1@-@	@-@psize1@-@	png	optimize PNG files, lossless compression
<code>pngcrush</code>	@-@popcon1@-@	@-@psize1@-@	png	optimize PNG files, lossless compression
<code>pngquant</code>	@-@popcon1@-@	@-@psize1@-@	png	optimize PNG files, lossy compression
<code>pngnq</code>	@-@popcon1@-@	@-@psize1@-@	png	optimize PNG files, lossy compression
<code>apngopt</code>	@-@popcon1@-@	@-@psize1@-@	apng	optimize APNG (Animated Portable Network Graphics) animated images

List of graphics data tools (CLI)

Miscellaneous data conversion

There are many other programs for converting data. Following packages caught my eyes using regex "`~Guse::converting`" in `aptitude` (8) (see [???](#)).

package	popcon	size	keyword	description
<code>alien</code>	@-@popcon1@-@	@-@psize1@-@	<code>rpm/tgz</code> → <code>deb</code>	converter for the foreign package into the Debian package
<code>freepwing</code>	@-@popcon1@-@	@-@psize1@-@	<code>EB</code> → <code>EPWING</code>	converter from "Electric Book" (popular in Japan) to a single JIS X 4081 format (a subset of the EPWING V1)
<code>calibre</code>	@-@popcon1@-@	@-@psize1@-@	<code>any</code> → <code>EPUB</code>	e-book converter and library management

List of miscellaneous data conversion tools

You can also extract data from RPM format with the following.

```
$ rpm2cpio file.src.rpm | cpio --extract
```

	#2
2026-10-31 13:08:14 UTC	
2026-10-31 13:08:17 UTC	