NAME

texexec − front end to the ConTeXt typesetting system

SYNOPSIS

texexec [*OPTION ...*] *FILE* [*...*]

DESCRIPTION

texexec, a **ruby**(1) script, is the command-line front end to the ConTeXt typesetting system, an extensive macro package built on the **tex**(1) family of typesetting programs. **texexec** provides several facilties:

- Process a ConTeXt source file, performing as many runs as necessary of **tex**(1), **texutil**(1), and MetaPost (**mpost**(1)). Options control the output format, the pages to process, paper size, and so forth.
- Create new ConTeXt formats, useful during installation.
- Post-process existing PDF files, including merging multiple files, and extracting and rearranging pages within a file, similar to the facilities provided for PostScript files by **psnup**(1) or for PDF files by **pdftk**(1).
- Extract or strip documentation from ConTeXt source files using **texutil**(1).
- Run MetaPost (**mpost**(1)) to generate figures from MetaPost source.
- Produce proof sheets of figures used in a ConTeXt source file.

OPTIONS

All switches are specified in full here but can be abbreviated to the shortest unique prefix. Thus, **--env** works the same as **--environment**. With no options, **texexec** assumes that the file on the command line is a ConTeXt source file, i.e. a TeX file in the ConTeXt dialect, and produces a PDF file using the source file.

General Options

--alone Avoid calling other programs when possible. For example, **--alone** will prevent **texexec** from using **fmtutil**(1) to generate formats (this prevention can be handy when tracing installation problems).

--environments=*file[,file[,...]]*

Specify ConTeXt environment file(s), separated by commas, to use when processing the source file. This option is useful when converting from non-ConTeXt file formats without environment or layout settings.

--help Produce a summary of switches and arguments. A more detailed help is produced by including **--all**.

--interface=*language*

Specify the language ConTeXt should use to communicate with you. Options are

en	US English
nl	Dutch
de	German
uk	British English
cz	Czech
it	Italian

--keep Preserve a few of the temporary files generated while processing by appending *.keep* to their name. For example, after

```
texexec --keep document.tex
```

you will find (besides *document.pdf*) *document.log.keep* and *document.top.keep*. The *document.top* file is where **texexec** wraps *document.tex* with the proper ConTeXt macro commands to set the output format, etc.

--once Process a file exactly once. (By default, **texexec** processes the file as many times as necessary to sort out all references, typeset **MetaPost** code, and so forth.)

--purge
Get rid of most temporary files generated while processing the source file. For example,

```
texexec --purge somefile.tex
```

will generate *somefile.pdf*, cleaning up after itself and leaving only one extra file, *somefile.tuo*. See also the **--purge** option of **ctxtools**(1)

--purgeall
Get rid of all temporary files generated while processing the source file, including the *filename.tuo* file. See also the **--purge --all** option combination of **ctxtools**(1)

--randomseed=*NNNN*
Set the random seed.

--result=*FILENAME*
Allows you to change the basename of the output file. See **--mode** for an example.

--runs=*NUMBER*
Specify the number of runs to perform on a file. Overrides **texexec**'s calculations.

--separation
Perform color separations.

--silent Suppress a few diagnostic and progress messages.

--timeout=*NNN*
Abort the run if a subprocess waits for more than *NNN* seconds; e.g. while waiting for user input when **tex** reports an undefined control sequence. Useful for automated testing scripts, to make sure the runs finish.

--usemodules=*module1[,module2,[...]]*
Use the comma-separated list of modules. For example, to typeset *document.tex* using the *bib* and *units* modules:

```
texexec --usemodules=bib,units document.tex
```

--verbose
Output extra diagnostic information.

--version
Print the version number.

Processing ConTeXt Source Files

Including specifying paper sizes, formats, and so forth.

--arrange
Perform page rearrangements, which are used to produce booklets. This option tells ConTeXt to the first *n*-1 runs without taking account of arrangements, then on the last run to pay attention to the arrangement commands in the source file.

--batchmode
Process the file in batchmode, which means to typeset the whole document even if there are errors. More imformation about `batchmode` can be found in Donald E. Knuth's *TeXbook*.

--nonstopmode
Process the file in nonstopmode, which means to typeset the document and report errors, but not to stop at any error. It is similar to batchmode but more verbose. More imformation about `nonstopmode` can be found in Donald E. Knuth's *TeXbook*.

--bodyfont=*font*
The name of a font to preload for use in setting the body of the text (OBSOLETE).

--centerpage
Center the document on the page.

--color Turn on color mode. Color mode can also be set by commands embedded in the document. These commands override the **--color** option.

--convert=*FORMAT*
Convert the input file to ConTeXt format from *FORMAT* before processing. In most cases, this conversion will result in a TeX file. Currently supported input *FORMAT*s are **xml** and **sgml**.

--dvipdfmx, --dvipdfm, --dpx, --dpm
Use the TeX engine (e.g. **pdftex** or **pdfetex**) to make a DVI file and **dvipdfmx**(1) to turn it into PDF.

--dvi, --ps, --dvips
Use the TeX engine (e.g. **pdftex** or **pdfetex**) to make a DVI file and **dvips**(1) to turn it into PostScript. It's counterintuitive that **--dvi** produces a PostScript file in addition to the DVI file. But that's because **--dvi** is shorthand for **--dvips**; adding the **--nobackend** option prevents **texexec**'s running **dvips**(1). See also the **--engine** option.

--fast Typeset the document(s) as fast as possible without causing problems.

--final Perform a final run without skipping anything. This option is typically used with **--fast**.

--language=*LANGUAGE*
Set the language for hyphenation. Can be specified in your source file. Options are the same as those for **--interface**.

--mode=*MODELIST*, **--modes=***MODELIST*
Allows you to change the mode used while typesetting the source file. The *MODELIST* is a comma separated list of modes. Modes are a conditional-compilation facility like `#ifdef` in C. So one source file can be used to produce several typeset documents: one for A4 paper, one for screen display in full color, one for letter paper, etc. For example:

```
texexec --pdf --mode=A4     --result=manual-a manual-t.tex
texexec --pdf --mode=letter --result=manual-l manual-t.tex
texexec --pdf --mode=screen --result=manual-s manual-t.tex
```

Here the **--mode** tells ConTeXt which mode directives to use when typesetting the source file. The **--result** option tells ConTeXt where to put the output file.

--modefile=*file*
Load this file before most of the usual processing; usually used for mode-related material.

--noarrange
Ignore arrangement commands in the source file.

--nobackend
Do not run the backend, e.g. **dvips**(1) or **dvipdfmx**(1). See the **--dvips** or **--dvipdfmx** options. Why would you give one of those options to choose a backend, yet tell **texexec** not to run the backend? Because each backend has its own syntax for `\special` calls. Specifying the backend allows the ConTeXt macros to use the correct syntax so that when you later run the backend to produce PostScript or PDF, the specials will be interpreted correctly.

--pages=*PAGENUMBERLIST*
Specify the pages or page range to appear in the output file. *PAGENUMBERLIST* may be the keyword **odd** or **even**; or one or more pages or page ranges separated by commas. For example,

```
texexec --pages=1,7,8-11,14 somefile.tex
```

--paperformat=*KEY*
For typesetting multiple pages on a single piece of paper. *KEY* has the form **a4a3** (for printing A4 pages on A3 paper), **a5a4** (for printing A5 pages on A4 paper), or in general **aMaN**. The actual layout of the pages is specified with the **--printformat** option.

--pdf, --pdftex
Use **pdftex**(1) to produce a pdf document (the default).

--printformat=*KEY*
Specify the layout of the final output. *KEY* can be **up**, resulting in 2 pages per sheet, double sided; or **down**, resulting in 2 rotated pages per sheet, double sided. Use the **--paperformat** option to specify the original page and sheet size.

--utfbom
Turn on UTF-8 encoding.

--xetex, --xtx
Use **xetex**(1) to produce a pdf document.

Creating ConTeXt Format Files

--make Generate a ConTeXt format file. For example, to make *cont-en.fmt* and have it placed in a default format directory:

```
texexec --make de
```

The most common invocation, which is used by scripts that install a new version of ConTeXt (see **ctxtools**(1)), uses **--all** so that **texexec** makes the usual formats:

```
texexec --make --all
```

--local When searching for TeX or MetaPost formats, look in the current directory rather than in the location set by the kpse library. See **kpathsea**(1) for more information on path searching.

--check
Check and report information about the ConTeXt version, the distribution, the TeX engine, and the language interfaces/formats.

Expert options

You should know what you're doing if you use these options!

--alpha Use the TEXMFALPHA environment variable to find and run an alpha release of ConTeXt.

--beta Use the TEXMFBETA environment variable to find and run a beta release of ConTeXt.

--distribution=*dist*
Usually one of **standard**, **web2c**, or **miktex**. **texexec** should figure it out automatically, and you shouldn't need to use this option.

--engine=*texengine*
Specify the program to do the hard work of typesetting. Currently either **pdftex** (the default), **xetex**, or **aleph**. The **luatex** value is experimental. The **--engine** option is not usually needed. Instead, let **texexec** figure out the setting based on other command-line information. See for example the **--xetex** or **--pdf** switches.

Postprocess PDF Files

--combination=*ROWS*******COLS*
Specify the number of pages to show on a single page. Use with **--pdfcombine**.

--pdfarrange
For rearranging pages in PDF files.

```
texexec --pdfarrange --paperformat=a5a4 --printformat=up foo.pdf
```

This command creates an A5 booklet from a PDF file *foo.pdf*. **--pdfarrange** is used in conjunction with the following options.

--pdfcopy

Copy and perhaps process pages from the pdf file. The resulting file is *texexec.pdf* by default, but you can change that using **--result**. Use the **--scale** option to magnify or demagnify the original pages and the **--pages** option to select the pages to copy. Here is an example using all these options:

```
texexec --pages=4-7 --pdfcopy --scale=750 --result=one images.pdf
```

It takes pages 4-7 from *images.pdf*, scales them by 75%, and copies them to *one.pdf*.

--scale=*integer*

If the integer is less than 10, then it is taken as an (integer) magnification factor. Otherwise, it is taken as a magnification factor in TeX terms, i.e. with 1000 meaning full scale.

--paperoffset=*dimen*

Specify the space between the edge of the pages and the beginning of the text block.

--backspace=*dimen*

Specify the inside (gutter) margins.

--topspace=*dimen*

Specify the top and bottom margin.

--markings

Add crop marks.

--addempty=*PAGES*

Add empty pages after the pages specified in *PAGES*. (Useful for, among other things, adding blank pages after a table of contents.)

--textwidth=*WIDTH*

Set the width of the original text. Specifying this parameter with a single-sided original will allow ConTeXt to adjust the page layout for double-sided output, producing much more attractive results.

With the **--pdfarrange** flag, specifying more than one file will result in all of the files being combined in the final result, allowing you to add title pages, decorated part separators, and so forth.

You can also do more complex manipulations, such as adding additional text to the page by setting up a small file with layout definitions and a simple figure insertion loop.

--pdfcombine

Combine multiple pages. Requires the **--combination** option.

--pdfselect

Extract pages from a file. Use in combination with the **--selection** switch, as in

```
texexec --pdfselect --paperformat=S6
--selection=1,9,14 file-1
```

which extracts pages 1, 9, and 14 from *file-1.pdf*, and places them in *texexec.pdf* (the default output filename if an output file isn't specified).

See **--pdfarrange** for other options.

--selection=*PAGES*

Specify pages to be affected by another option. See **--pdfarrange** and **--pdfselect** for examples.

XML handling

--filters=*filter1[,filter2[,...]]*

Specify XML filters to use.

Extract or Strip Out Documentation

--listing

Produce a typeset version of the source code in *FILE*. You can specify the format of the output file. For example, use

```
texexec --ps --listing readme.now
```

to produce a PostScript file called *texexec.ps*.

See also **--backspace**, **--topspace**, and **--result**.

--module

Create documentation for ConTeXt, MetaPost (see **mpost**(1)), **perl**(1), and **ruby**(1) modules. Converts the documentation to ConTeXt format and then typesets a documentated version of the source file.

Documentation lines in ConTeXt source files are specified by beginning lines with these strings:

`%C` : Copyright information

`%D` : Documentation lines

`%I` : TeXEdit information lines (mostly in Dutch)

`%M` : Macro code needed to processs the documentation

`%S` : Suppressed lines

The same forms can be used for Perl or ruby scripts, except that the `%` character (the TeX comment character) is replaced by `#` (the Perl comment character).

See also the **--documentation** option to **ctxtools**(1).

Process MetaPost Figures

--mpsformats=*name*

The name of a MetaPost format file, e.g. **metafun** (the default).

--mptex

Strips out and typesets TeX code embedded in a MetaPost file.

--nomp

Do not run **mpost**(1), even if needed.

--nomprun

Do not run **mpost**(1) on embedded MetaPost code.

Producing Proof Sheets of Figures

Generate information and proof sheets of one or more (non-EPS) graphics files. For example,

```
texexec --figures *.png *.jpg
```

scans the current directory for PNG and JPG files and extracts useful information about their sizes and types. By default, this information is stored in *rlxtools.rli*. Then the given figures are made into a proof sheet (by default *texexec.pdf*) according to the method specified by the **--method** option. Note that newer versions of **pdftex**(1) do not support TIFF inclusion.

--method=*ALTERNATIVE*

Specify one of three options to produce the document containing the images used in the source file:

a : A proof sheet with additional information provided for each figure (the default)

b : A proof sheet with the graphics only

c : One figure per page, with the page clipped to the bounding box of the figure

See also **--paperoffset**, which allows you to specify an offset to be added to the page, as in

```
texexec --figures --method=c --paperoffset=.5cm *.pdf *.png *.jpg
```

USAGE

Each ConTeXt user interface (language) has its own format. The following command generates two formats, one using the English interface for typesetting in English, and one for Dutch:

```
texexec --make en nl
```

By default, the language used for typesetting matches the user-interface language (set with **--interface**. It is possible to use one language for typesetting and another for messages by changing the relevant settings in *cont-usr.tex*. These languages can also be changed on the command line with a command such as

```
texexec --make --language=pl,cz,sk en
```

That command generates a ConTeXt format file with an English user interface, and the main language set to Polish (**pl**). Czech and Slovak hyphenation patterns are also loaded so that Czech and Slovak text included in a source file will be typeset properly (**cz** and **sk**).

o When the appropriate formats are present, a file can be typeset by typing

```
texexec test
```

texexec tries to determine what interface it should use to typeset *test.tex* by looking for a line such as

```
% interface=en tex=pdftex output=pdftex
```

at the top of the file (i.e., on the very first line). This line is equivalent to **TeX**'s format line, ‘‘&*FORMAT*’’).

By default, **texexec** will produce a PDF file using **pdftex**(1). The **--dvips** flag tells **texexec** to produce a PostScript file instead.

After an error-free run, **texexec** will run **texutil**(1) to determine whether additional runs of **tex**(1) (or **pdftex**(1)) or any utility programs (e.g., **bibtex**(1), **makeindex**(1)) are necessary. You can suppress these additional runs by specifying the **--once** or **--runs** flags:

```
texexec --once test
texexec --runs=2 test
```

EXAMPLES

Produce PDF from ConTeXt source (the .tex extension is optional):

```
texexec file.tex
```

Same as the above but without rerunning for crossreferences, etc.:

```
texexec --once file.tex
```

Produce PostScript from ConTeXt source:

```
texexec --ps file.tex
```

Produce file-a4.pdf using conditional compilation (modes):

```
texexec --mode=a4 --result=file-a4 file.tex
```

Generate format (.fmt) files used by ConTeXt (used during installation):

```
texexec --make --all
```

INITIALIZATION

texexec requires ruby. On Unix and Unix-like systems, no special steps have to be taken to get **texexec** to work beyond installing ruby and having the **ruby**(1) binary in your path.

ENCODINGS

Some languages require specific character encodings to represent their alphabets (beyond the basic ASCII encoding). Although you can use TeX commands to represent these characters, such as "`\.z`", it's easier to use a text editor that includes direct support for these characters and let ConTeXt translate them to the necessary TeX commands. For some languages, this approach can also improve the performance of TeX's hyphenation algorithms.

ConTeXt supports several of the most commonly used encodings. Check the files beginning with *enco-*, *lang-*, and *font-* in the ConTeXt distribution for more information.

web2c distributions (such as **teTeX**) support a mechanism to map document encodings to ConTeXt's internal encoding, font encodings, and hyphenation patterns. **texexec** provides a document option and a command-line flag to pass the necessary information to **tex**(1) or **pdftex**(1). You can add lines such as

```
%& --translate-file=cp1250pl
```

or

```
% --translate=cp1250pl
```

to the beginning of your document, or you can specify the **--translate** flag on the command line, as

```
texexec --translate=il2pl somefile
```

Using language-specific encodings will make your file less portable than using ASCII. It may then not be possible for other people to typeset your documents on their systems.

SEE ALSO

- **bibtex**(1), **dvipdfmx**(1), **dvips**(1), **fmtutil**(1), **makeindex**(1), **mpost**(1), **pdftex**(1), **pdftk**(1), **xetex**(1), **ruby**(1), **psnup**(1), **tex**(1), **texshow**(1), **texutil**(1).
- The texexec manual *mtexexec.pdf*, available from PRAGMA ADE ⟨`http://www.pragma-ade.com/dir/general/manuals/`⟩.
- Donald E. Knuth's *The TeXbook*.
- ConTeXt wiki ⟨`http://www.contextgarden.net`⟩.

AUTHOR

This manpage was written by Tobias Burnus <burnus@gmx.de> and C.M. Connelly <c@eskimo.com> and updated by Sanjoy Mahajan <sanjoy@mit.edu>. It is based on the *mtexexec.pdf* manual written by Hans Hagen <pragma@wxs.nl>.