# how to install ConTEXt

## Introduction

Nowadays most TEX users will use one of the many distributions that are available for free or commercially. This means that much of the hard work of installation is already taken care of. When installed properly, the user will quite certainly have the binaries, hyphenation patterns, fonts and some macro packages on the system.

This manual is therefore not targeted at installing TEX, but focusses on how to get CONTEXT running within an existing distribution. When, after reading this text, you still cannot get CONTEXT working properly, we advice you to contact the maintainer of your distribution, or to send your questions to the CONTEXT mailing list.

## Unpacking the archives

The CONTEXT distribution consists of several zipped archives. You need to unpack these to the appropriate directory of you file system. The archives are zipped using the free `zip` program, and can be unzipped using its counterpart.[1] Just type:

```
unzip -a <archive>.zip
```

Everything you need to get started can be found in the main CONTEXT archive:

`cont-tmf.zip` the CONTEXT sources and programs

There are some more archives, like the one with METAPOST tools and the still experimental TEXWORK editing environment. We just mention them here, since you *don't* need them when you fetched the main archive.

`cont-mpd.zip` a couple of METAPOST tools
`cont-wrk.zip` the TEXWORK editing environment

There are two more somewhat redundant archives. These can be useful for users of packages other than CONTEXT. If you are using the files from `cont-tmf.zip`, it is not necessary to include the files from these archives.

`cont-ppc.zip` the PPCHTEX only files
`cont-uti.zip` the TEXUTIL program

To prevent duplication in files, we strongly advise to obey the path as coded in the archives. The files in these archives will be unpacked into the appropriate directories

[1] Shareware programs like WinZip for MSWindows also work well.

of the official texmf tree. Some files are stored in the directories for generic TEX files or the LATEX package, but those are not important for CONTEXT users.

| | |
|---|---|
| texmf/tex/context/config | some configuration files |
| texmf/tex/context/base | all CONTEXT core files, modules |
| texmf/tex/context/extra | some goodies |
| texmf/tex/context/sample | a couple of sample files |
| texmf/tex/context/user | user specific files |
| texmf/tex/context/third | third party contributions |
| texmf/metapost/context | the METAPOST modules |
| texmf/context/config | some configuratiom files |
| texmf/context/data | data filed used by scripts |
| texmf/context/perltk | the PERL scripts |
| texmf/doc/context/base | CONTEXT documentation |

Users can best not put files in the base, extra and sample directories. That way they can conveniently be removed and reinstalled. The files in third are to be organized by author.[2]

On UNIX, the PERL scripts should be installed without the .pl extension, because these scripts and possibly other programs rely on these names. They should be moved to the search path for binaries and scripts. On MS WINDOWS running scripts is not supported by the operating system. In the perltk path you can find a zipped file with the executable runperl.exe, written by Fabrice Popineau. When copied to <scriptname>.exe, this program launches the script with the same name.

The initialization files for TEXEXEC go into texmf/context (TETEX) or into the same directory as the binaries. When setting up TETEX, make sure you enable generation of the format files, by uncommenting the lines that specify the CONTEXT formats. You can edit the TETEX (and FPTEX) configuration file using:

```
fmtutil
```

In TETEX you should pass the switch --edit. When you pass --all you get all formats. When in need for patterns other than the default ones, texconfig can be used to enable more hyphenation patterns in the file cont-usr.tex. You can also edit this file directly.

CONTEXT comes with a font that contains navigational symbols. These fonts go to the corresponding places in the fonts tree, in our case:

[2] You need to create this (yet empty) directory yourself.

| | |
|---|---|
| `texmf/fonts/tfm/hoekwater/context` | the files with suffix `tfm` |
| `texmf/fonts/afm/hoekwater/context` | the files with suffix `afm` |
| `texmf/fonts/type1/hoekwater/context` | the files with suffix `pfb` |

There is also a ConTeXt specific encoding/map file. This one is stored in the following path:

`texmf/dvips/config` ConTeXt fontmap file

Don't forget to add a reference to this file `context.map` to the `pdftex.cfg` file that resides in the pdfTeX configuration directory.

```
map +context.map
```

## Setting up TeXEXEC

TeXEXEC is the command line interface to ConTeXt. There is nothing wrong with running ConTeXt in the traditional way, like

```
pdfetex  &cont-en filename
```

or for a web2c bases TeX

```
pdfetex  &cont-en --progname=context filename
```

but, and this will be more clear when we provide more options, the next call is more convenient:

```
texexec  filename
```

This not only hides the implementation specific switches, but also removes the need for specifying the format.

TeXEXEC is written in Perl, a scripting language that is available on most leading software platforms. In order to operate well, we need to set up TeXEXEC. Of course you must have Perl running on your system. On UNIX systems, this is always the case, but for MS Windows you have to install Perl yourself.[3].

First you have to move TeXEXEC and its relative TeXUTIL to a location in the binaries path. When issuing the command `texexec` you should get some response. Even better, when saying:

[3] A good choice is the Active State distribution that you can fetch from `www.activestate.com`

```
texexec  --verbose
```

you should get some information on how TEXEXEC is set up. When generating formats and processing files fail, you need to set up the initialization file `texexec.ini`. This file comes disguised as `texexec.rme`, so when not present, you need to copy this file. The initialization file should be present in the `config` path, or in the same path as the script. Next you need to edit this file. The simplest way is to comment and uncomment one of the following lines:

```
set  TeXShell  to  tetex
%set TeXShell  to  fptex
%set TeXShell  to  miktex
%set TeXShell  to  private
```

When this is done, you should check to what extent the rest of the variables in this file match the local settings. We hope that the names of the variables used are clear.

When it is not set up properly, TEXEXEC tries very hard to locate the files it needs. Normally TEXEXEC should start up rather quickly. When you are under the impression that you are waiting too long, there is probably an error in the setup.

## Using TETEX

When you are using teTEX or derived distributions, you can usually stick to the regular updates, unless you want to use the latest version of CONTEXT. In any event, you should make sure that only one copy is present on your system, because otherwise files can get mixed, due to the often aggressive file searching algorithms. If you want to update anyway, you can unzip `cont-tmf.zip` from within the `texmf` directory and regenerate the format files.

The PPCHTEX only archive is for users who maintain their own files and only want to install this package. The TEXUTIL archive is for those who want to use this script, but don't want to install CONTEXT. Neither archive is needed when you install CONTEXT from `cont-tmf.zip`.

## Using FPTEX

The first step in installing CONTEXT under FPTEX, is to unzip the file `cont-tmf.zip` within the `texmf` directory. Afterwards the `texexec` binaries and PERL script should

be copied to the directories that contain the other TEX binaries. The TEX binaries path, that should also be part of your PATH variable, often looks like:

```
../tex/bin/win32
```

When no file texexec.exe is found, you have to unzip texexec.zip and copy runperl.exe to texexec.exe. Don't forget to update the ls-R file database by running mktexlsr.

Next you need to locate the file texmf/web2c/fmtutil.cnf. In this file, which contains documentation, you need to activate the CONTEXT formats. Now you can run:

```
texexec --make
```

When everything is installed properly, TEX generates one or more formats. When TEXEXEC fails, you have to check the file texexec.ini.

## Using MIKTEX

This section is provided by Grzegorz Sapijaszko and Ed L. Cashin and concerns the installation of CONTEXT under MIKTEX. Installation in the MIKTEX environment isn't much different from the TETEX installation. Nevertheless, you should take a few steps to achieve good results. The first one is copying a texexec.rme file to texexec.ini and uncommenting the lines for MIKTEX as follows:

```
%set TeXShell  to  tetex
%set TeXShell  to  fptex
set TeXShell  to  miktex
%set TeXShell  to  private
```

In next step you should add a string \context\perltk to your environment variable PATH (in autoexec.bat under WIN9X, or in Control Panel in WINNT), for example:

```
PATH=c:\miktex\context\perltk;
```

If you are using languages other than Dutch, German or English, you should uncomment the lines in the cont-usr.tex file for the hyphenation patterns of the languages you need. After that you can refresh the filename databases, for example, by using "Refresh Filename Databases" from the "Start | Programs | MikTeX | Maintenance" menu or "MIKTEX | Options" menu, depending on the version you run. You should also generate a format file. This is described in the next section. Notice that you should have PERL installed on your system.

After generating the formats you should copy the format file `cont-xx.fmt` from `\context\perltk` to the directory where MiKTeX is storing formats (usually `\localtexmf\miktex\fmt`). Another way is to add a `\context\perltk` directory to `miktex.ini` file:

```
[MiKTeX]
...
...
;; Where MiKTeX searches for .fmt files.
FMTPath=.;%R\miktex\fmt;C:\localtexmf\context\perltk//
```

As a convenience, you can copy the `runperl.exe` file from the `cont-wrk.zip` archive to `texexec.exe`. You should make sure sure that those binaries are in the same directory as the Perl scripts.

## Generating formats

From its name you can deduce that ConTeXt is written in the typographic language TeX. ConTeXt is parameter driven, which means that users change its behaviour by setting variables and changing keys. ConTeXt comes with a multi–lingual interface. Currently there are three such interfaces: Dutch, English and German.

Users who want complete control can edit the file `cont-usr.tex` and generate a format using the main file `context.tex`. Users who want an Dutch, English or German format, can stick to the files named `cont-nl`, `cont-en`, and `cont-de`. Again, by editing the file `cont-usr.tex`, you can influence the outcome.

In the early years of TeX, generating a format was common practice and users were pretty well aware of format files, hyphenation patterns and fonts. Nowadays, disributions take care of the more complicated issues, so users can comfortably skip many nasty installation steps. To make life even more comfortable, ConTeXt comes with TeXexec, a command line interface to TeX. When properly set up, this Perl script can save you much time.

For instance, generating the three formats mentioned is accomplished by:

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

When TeXexec is set up properly, this command should work. Before you read on, you should try to generate at least the English and Dutch format. The Dutch format is needed by some of the postprocessing features built into TeXexec.

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

Additionally, you need to make the MetaFun format. This is an extension to the standard METAPOST format which offers not only more features, but also interfaces quite well to ConTeXt.

```
texexec  --make metafun
```

The formats associated with the interfaces default to the language of the interface. This is all right for Dutch, English or German users, but Polish and Czech users are worse off. For them a format file that defaults to their own language makes more sense. Poles will like to say:

```
texexec  --make  --language=pl  --bodyfont=plr  en
```

while Czech people will go for:

```
texexec  --make  --language=cz  --bodyfont=csr  en
```

Both produce a format called cont-en with an English interface, but the first one defaults to Polish hyphenation patterns and fonts, and the second one to Czech ones. If wanted, you may pass a comma separated list of languages.

```
texexec  --make  --language=pl,it,uk  --bodyfont=plr  en
```

or, to generate a english interface format with Czech and Slovak patterns and Czech–Slovak Computer Modern Roman fonts:

```
texexec  --make  --language=cz,sk,en  --bodyfont=csr  en
```

Unfortunately, the hyphenation patterns are of hard coded in a format file and cannot be loaded at run time. When patterns are needed other than the ones loaded by default, you can consider adapting the file cont-usr.tex. This file is loaded at format generation time. When for instance Italian patterns are to be used, given that these are available either in the file lang-it.pat, or in a file onto which this filename is mapped, you should uncomment the line:

```
\installlanguage [\s!it] [\c!status=\v!start] % italian
```

The strange looking \s! and \c! things are needed in order to support multiple interfaces. Don't touch these!

When using web2c, in texmf.cnf some ConTeXt specific memory settings take place. When directly generating a format —i.e. when you're not using the teTeX initialization script or TeXexec— you should supply the program name: -progname=context

Make sure you read the manual to TEXEXEC. Apart from the normal processing of files, there are quite a few useful options: mode dependant processing, output selection, generating booklets, typesetting contact sheets of figures, manipulating PDF files, and more.

## Changing defaults

The somewhat more run-time specific settings, like certain special drivers, can be added to `cont-sys.tex`. This file is loaded at run time. For instance, this file can can contain the line:

```
\setupoutput[pdftex]
```

This commands tells CONTEXT to produce PDF output by default. For Y&Y and Acrobat support, you just say:

```
\setupoutput[dviwindo,acrobat]
```

Of course you can also load location specific layout settings in this file. The next few lines tell CONTEXT to default to the CONTEXT navigational symbols, instead of the ones composed from other glyphs.

```
\usesymbols [nav]
\setupsymbolset [navigation 1]
```

At PRAGMA ADE we want to process METAPOST files at run-time, so there we also have entries like:

```
\recycleMPslotstrue
\runMPgraphicstrue
\runMPTEXgraphicstrue
\useMETAFUNformattrue
```

The first setting leads to less intermediate files. The next two settings let CONTEXT process the graphics real-time. For that you need to enable `write18` in the TEX configuration file (for WEB2C this is `texmf.cnf`). The last setting only makes sense if you have also generated a `MetaFun` format. If you want to know more about `MetaFun`, take a look at its manual.

The verbatim environments provide pretty printing. When you want even more fancy verbatim, for instance with in-between switching of a language interpreter, you should

enable this feature. If you get no idea what we're talking about here, leave this variable untouched since tricky code is involved.

```
\newprettytrue
```

When ConTeXt cannot determine the dimensions of an external figure, and no `texutil.tuf` file is present, you can let ConTeXt call TeXutil directly. If you use pdfTeX, you can leave this switch off.

```
\runutilityfiletrue
```

Morethan one instance of TeX a single path, can lead to clashes in temporary files. The next switch enables a filename security feature:

```
\protectbufferstrue
```

For the moment, we use these low level boolean switches instead of more readable commands.

An important section in this file are the font defaults. You can set a default font encoding as well as enable map file loading for pdfTeX. Soem of these features are still experimental.

## The WEB2C configuration

Although not stricktly needed, ConTeXt will operate more smoothly when these features are set in the file texmf.cnf.

```
openout_any = a
shell_escape = t
allow_multiple_suffixes = f
```

The first line permits ConTeXt to open parent paths that can hold common styles. The second line enables enables users to embed METAPOST code in their document. As a result, METAPOST is called automatically.[4] The last line makes sure that when opening files like `somefile.tuo`, TeX will not try to open `somefile.tuo.tex` first.

When you embed TeX code in a METAPOST definition, using `btex ... etex`, the next line will use TeXexec to process that fragment in the case of using METAPOST directly.

[4] You can disable this feature by running TeXexec with the switch `--automp`.

```
MPXCOMMAND=texexec --mptex
```

## Processing files

You can test you installation with the following file.

```
\starttext
  \framed {Let's see if it works.}
\stoptext
```

If ConTeXt is set up properly, then the command

```
texexec  filename
```

will produce a file `filename.dvi`. Unless a file was already processed before, you will notice that TeXexec processes the file at least two times. During a TeX run, ConTeXt saves information in the files `filename.tui`: cross references, entries to the table of contents, data needed for optimization, etc. If the run is succesful, this file is converted to a file called `filename.tuo` and used in the next run. TeXexec will reprocess the file until the `tuo` file is unchanged.

We *strongly recommend* to use pdf-ε-TeX: Peter Breitenlohners ε-TeX permits ConTeXt to run more efficient, while Hàn Thế Thành's pdfTeX provides pdf output and pdf-ε-TeX combined both in one program. For those unfamiliar with ε-TeX: this is an extension of TeX that is not only more efficient and in some aspects faster, but that also offers a few features that come in handy when writing complicated code. For the moment, ConTeXt adapts its behaviour to the kind of TeX that is used, but future versions may rely on ε-TeX or other descendants completely.

By default, ConTeXt generates dvi output for dvips, unless the output is specified otherwise. We already mentioned the `\setupoutput` command. A second way of achieving this is:

```
texexec  --pdf  filename
```

And yet another way is adding a comment line in the document source, like:

```
% interface=en  output=pdftex  translate=cp1250pl
```

or

```
% interface=en  output=pdftex  translate=cp1250cz
```

Now we can omit the `--pdf` switch when we launch TEXEXEC. Normally TEXEXEC is able to sort out the interface itself, but in case of troubles, you can set some defaults in the file `texexec.ini`. The `translate` key is only needed when you use the reencoding-on-the-fly feature of WEB2C.

## Subscribing to the list

There are two dedicated mailing lists hosted by the NTG:

the CONTEXT mailing list
the PPCHTEX mailing list

These lists are so called *majordomo* ones, therefore you should send a subscription command to:

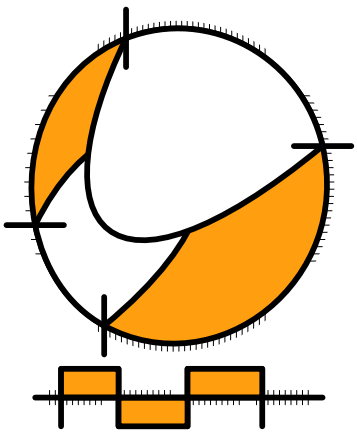
`majordomo@ntg.nl`

Just send a message to this address with the body text:

`subscribe ntg-context`

Of course you can also directly contact the authors at:

PRAGMA ADE: pragma@wxs.nl



PRAGMA

Advanced Document Engineering | Ridderstraat 27 | 8061GH Hasselt NL
tel: +31 (0)38 477 53 69 | email: pragma@wxs.nl | internet: www.pragma-ade.com