# 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. Just type:

```
unzip  -L - 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 |

When you want more, there are two additional archives, one with documentation styles, and another one with the TeXwork editing environment. We just name them. The Perl/Tk implementation of TeXwork is still rather experimenal.

| | |
|---|---|
| `cont-doc.zip` | the documentation styles |
| `cont-wrk.zip` | the TeXwork editing environment |

There are also two sort of redundant archives. These can be useful for users of packages other than ConTeXt.

| | |
|---|---|
| `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 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/local` | user specific files |
| `texmf/tex/context/<third>` | third party extensions to ConTEXt |
| `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 |

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 MSWindows running scripts is not supported by the operating system. In the `perltk` path one 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, using:

```
fmtutil --edit
```

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

| | |
|---|---|
| `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/fonts/dvips/config` | ConTEXt fontmap file |

Don't forget to add a reference to this file `context.map` to the `pdftex.cfg` file that resides somewhere on your system.

## 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  --progname=context  filename
```

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

```
texexec  filename
```

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. First one has 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:

```
texexec  --verbose
```

you should get some information on how TEXEXEC is set up. When generating formats and processing files fail, one needs to set up the initialization file `texexec.ini`. This file comes disguised as `texexec.rme`, so when not present, one needs to copy this file. The initialization file should be present in the `config` path, or in the same path as the script. Next one needs to edit this file. The most simple 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, one should check to what extend the rest of the variables in this file match the local settings. We hope that the names of the variables used are clear.

When not set up properly, TEXEXEC tries very hard to locate the files it needs. Normally TEXEXEC should start up rather fast. 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 occasion, 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. Both archives are *not* needed in teTEX!

## 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. 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 needs 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 now generates one or more formats. When TEXEXEC fails, you have to check the file `exexec.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:\localtexmf\context\perltk;
```

If you are using other languages 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. 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 one 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 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`, one 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 live 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 format.

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, one may pass a comma separated list of languages.

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

Unfortunately, the hyphenation patterns are sort of hard coded in a format file and cannot be loaded at run time. When always more patterns are needed than the ones loaded by default, one 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, one 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 some 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. This file for instance 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 one 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:

```
\runMPgraphicstrue
\recycleMPslotstrue
```

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

```
\newprettytrue
```

When ConTEXt cannot determine the dimensions of an external figure, and no `texutil.tuf` file is present, one can let ConTEXt call TEXutil directly:

```
\runutilityfiletrue
```

Running more instances of TEX on one 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 switched instead of more readable commands.

## The WEB2C configuration

Although not stricktly needed, ConTeXt will operate more smoothly when in the file texmf.cmf the next switches are set:

```
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 running METAPOST directly. The last line makes sure that when opening files like `somefile.tuo`, TeX will not try to open `somefile.tuo.tex` first.

When one embeds TeX code in a METAPOST definition, using `btex ... etex`, the next line will use TeXEXEC to process that fragment.

```
TEX = texexec --once --batch
```

## Processing files

When ConTeXt is set up, one can make a simple TeX file, say

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

This file is processed by saying:

```
texexec  filename
```

Normally this will produce a file `filename.dvi`. Unless a file was already processed before, one 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. When the run was 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: ε-TeX permits ConTeXt to run more efficient, while PDFTeX provides PDF output. 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=pdf translate=cp1250pl
```

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, one can set some defaults in the file `texexec.ini`. The `translate` key is only needed when one uses 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: ntg-context@ntg.nl
the PPCHTEX mailing list: ntg-ppchtex@ntg.nl

These lists are so called *majordomo* ones, therefore one 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 one 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   e-mail: pragma@wxs.nl   ConTeXt: www.pragma-ade.nl