



how to install ConT_EXt

Introduction

Nowadays most T_EX 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 T_EX, but focusses on how to get CON_TE_XT running within an existing distribution. When, after reading this text, you still cannot get CON_TE_XT working properly, we advice you to contact the maintainer of your distribution, or to send your questions to the CON_TE_XT mailing list.

Unpacking the archives

The CON_TE_XT 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 -a <archive>.zip
```

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

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

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

`cont-doc.zip` the documentation styles

`cont-wrk.zip` the T_EXWORK editing environment

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

`cont-ppc.zip` the PPCH_TE_X only files

`cont-uti.zip` the T_EXUTIL 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 T_EX files or the L^AT_EX package, but those are not important for CON_TE_XT users.

<code>texmf/tex/context/config</code>	some configuration files
<code>texmf/tex/context/base</code>	all <code>CONTEXT</code> core files, modules
<code>texmf/tex/context/extra</code>	some goodies
<code>texmf/tex/context/sample</code>	a couple of sample files
<code>texmf/tex/context/user</code>	user specific files
<code>texmf/tex/context/third</code>	third party contributions
<code>texmf/metapost/context</code>	the <code>METAPOST</code> modules
<code>texmf/context/config</code>	some configuration files
<code>texmf/context/data</code>	data files used by scripts
<code>texmf/context/perlTk</code>	the PERL scripts
<code>texmf/doc/context/base</code>	<code>CONTEXT</code> 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.

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:

<code>texmf/fonts/tfm/hoekwater/context</code>	the files with suffix <code>tfm</code>
<code>texmf/fonts/afm/hoekwater/context</code>	the files with suffix <code>afm</code>
<code>texmf/fonts/type1/hoekwater/context</code>	the files with suffix <code>pfb</code>

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 --programe=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 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:

```
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 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, 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 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 T_ET_EX

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 F_PT_EX

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` now generates one or more formats. When `TEXEXEC` fails, you have to check the file `texexec.ini`.

Using M_IK_TE_X

This section is provided by Grzegorz Sapijaszko and Ed L. Cashin and concerns the installation of C_ON_TE_XT under M_IK_TE_X. Installation in the M_IK_TE_X environment isn't much different from the T_EX 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 M_IK_TE_X 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 M_IK_TE_X 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 $\text{CON}\text{T}_{\text{E}}\text{X}\text{T}$ is written in the typographic language $\text{T}_{\text{E}}\text{X}$. $\text{CON}\text{T}_{\text{E}}\text{X}\text{T}$ is parameter driven, which means that users change its behaviour by setting variables and changing keys. $\text{CON}\text{T}_{\text{E}}\text{X}\text{T}$ 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`, you can influence the outcome.

In the early years of $\text{T}_{\text{E}}\text{X}$, generating a format was common practice and users were pretty well aware of format files, hyphenation patterns and fonts. Nowadays, distributions take care of the more complicated issues, so users can comfortably skip many nasty installation steps. To make live even more comfortable, $\text{CON}\text{T}_{\text{E}}\text{X}\text{T}$ comes with $\text{T}_{\text{E}}\text{X}\text{EXEC}$, a command line interface to $\text{T}_{\text{E}}\text{X}$. 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 $\text{T}_{\text{E}}\text{X}\text{EXEC}$ 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, 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 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, 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 CON_TE_XT specific memory settings take place. When directly generating a format —i.e. when you're not using the T_ET_EX initialization script or T_EXE_XE_C— you should supply the program name: `-progrname=context`

Make sure you read the manual to T_EXE_XE_C. 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 CON_TE_XT to produce PDF output by default. For Y&Y and Acrobat support, you just say:

```
\setupoutput[dvwindo,acrobat]
```

Of course you can also load location specific layout settings in this file. The next few lines tell CON_TE_XT to default to the CON_TE_XT 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 you want even more fancy verbatim, for instance with in-between switching of a language interpreter, you should say:

```
\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:

```
\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 switches instead of more readable commands.

The WEB2C configuration

Although not stricktly needed, `CONTEXT` will operate more smoothly when in the file `texmf.cnf` 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 you embed `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 CON_TE_XT is set up, you can make a simple T_EX 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, you will notice that T_EXEXEC processes the file at least two times. During a T_EX run, CON_TE_XT 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. T_EXEXEC will reprocess the file until the `tuo` file is unchanged.

We *strongly recommend* to use PDF- ϵ -T_EX: ϵ -T_EX permits CON_TE_XT to run more efficient, while PDF_TE_X provides PDF output. By default, CON_TE_XT 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=cp1250p1
```

or

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

Now we can omit the `--pdf` switch when we launch T_EXEXEC. Normally T_EXEXEC 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 CON_TE_XT mailing list

the PPCH_TE_X mailing list

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

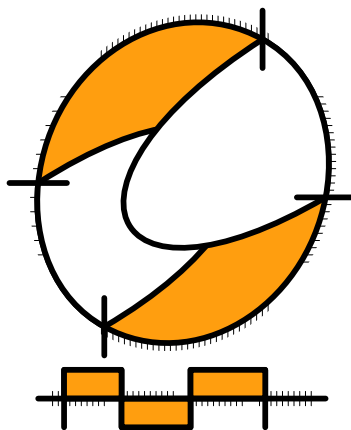
`majordomo@ntg.nl`

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