

`\starttext` ConT_EXt Text Editors

Steve Peter

1 Introduction

Last time we saw a bit about the capabilities ConT_EXt has, while writing a short source file and running `texexec` from the command line. This time, we'll look at some text editors and T_EX editing environments to make your life a bit easier. Many editors have some sort of T_EX support built in, and with a bit of coaxing, they can be made to play nicely with ConT_EXt. This column is a (partial) answer to frequent questions on the ConT_EXt mailing list on how to make various L^AT_EX-oriented text editors usable for ConT_EXt.

For those of us who make a living from writing and typesetting, the choice of a text editor is a deeply personal matter, and people frequently adopt almost religious fervor in defending their choice. My advice is to try as many editors as you can and pick the one you feel best meets your needs. What's right for me may not be right for you, and only you can decide that. In fact, I use several different editors, depending on what platform I'm on, what my mood is, whether I've had my 6 cups of coffee yet, and so on. All of the editors feature things you absolutely must have to do T_EXnical work, like syntax highlighting, bracket matching, etc.

In the rest of this column, I'll simply take up each major operating system and take a look at a few text editors. In fact, I only look at free products, because I've generally been happy with the free alternatives, and I've not felt a need to spend any money on a commercial product. I will, however, point out a couple of popular commercial offerings in the appropriate place.

2 Windows

Windows has no shortage of fine editors. Not all of them, however, are ConT_EXt-friendly. With a little bit of tweaking, though, most of them will run it just fine. Let us start with an editor that *does* play well with ConT_EXt.

2.1 SciTE

The Scintilla Text Editor, or **SciTE**, is a general-purpose configurable text editor for Windows and X. It has support for T_EX, and Hans Hagen has added lots of very nice support for ConT_EXt, complete with a **manual** to orient you. One very nice feature is command completion, whereby when you type the first couple of letters of a command, a completion box pops up; just hit tab to complete the command.

Basically, you'll want to move the various `.properties` files included in the ConT_EXt distribution to the SciTE folder, and simply add the line

```
import context
```

to one of the Options files (accessible via the Options menu). You should probably add it to the User Options file, although I have had much better luck adding it to my Global Options file.

Once you've done that, and assuming you have ruby¹ installed and configured, you can type in your source code and hit F7 to transform it into PDF. Hit F5 and your PDF viewer opens up and loads the file you just ran. If you don't like the defaults, it is fairly straightforward to edit the `context.properties` file to configure everything just as you like it.

2.2 TeXnicCenter

TeXnicCenter is a widely-used free T_EX integrated development environment (IDE), whose interface closely resembles the Microsoft development tools. As with SciTE,

¹ An interpreted scripting language similar to perl or python. See the [web site](#) for more information.

within the single environment, you can edit your text, hit a button to compile the source, and hit another button to launch your DVI or PDF viewer and load your document. TeXnicCenter also has various palettes and wizards to fill in parts of T_EX syntax. Out of the box, it is strongly oriented towards L^AT_EX, but you can add ConT_EXt support. Let's do that now.

- Go to the BUILD menu and select DEFINE OUTPUT PROFILES
- On the left, click the Add button, and give the new profile a name, such as "ConT_EXt => PDF"
- Check the box that says "Run (La)TeX in this profile", and give the path, not to L^AT_EX, but to texexec or texmfstart. (This will depend on your distribution. For MikT_EX, look in C:\texmf\miktex\bin\.)
- For the command-line arguments, you can add any switches you use for texexec or texmfstart, such as --pdf. Use the arrow button to the right of the text box to select a placeholder for the file (for example %pm will pass the entire path of the file, along with the extension).

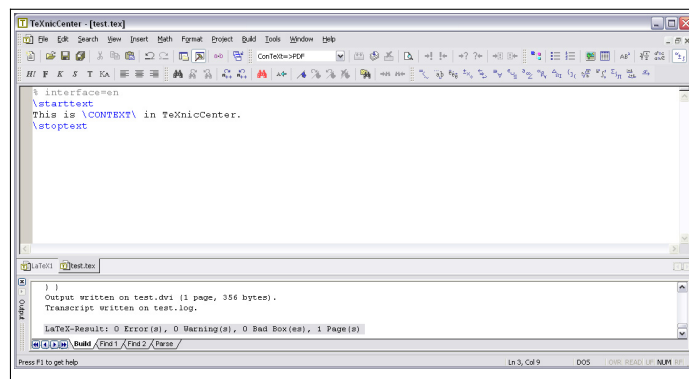


Figure 1 TeXnicCenter doing ConT_EXt!

Now, you can select your new profile from the popup menu to the left of the build button. Click the button (or hit Ctrl+F7) and away you go!

2.3 LaTeX Editor

LaTeX Editor, as its name implies, is mainly an editor for L^AT_EX, but it can be changed into a ConT_EXt Editor. The editor itself has borrowed ideas and code from both the

Scintilla project (and thus SciTE) and TeXnicCenter, but the editor still has its own feel. Unlike TeXnicCenter, there are no build profiles so the editor has loyalty to one format only. Let's make it our partisan.

- Go to OPTIONS ▶ CONFIGURE TOOLS
- In the "(La)TeX" box, change the current entry to "C:\texmf\MikTeX\bin\texexec.exe" --pdf "\$(FileTitle).tex" (or any variants thereof you might wish)

Now, your ConTeXt files will be happy!

2.4 Others

[WinEdt](#) is a very popular commercial Windows T_EX editor which has built-in support for ConT_EXt. TUG and Dante members get a discount on the registration fee. [WinShell](#) is another editor that is geared toward L^AT_EX, but can be modified in the same way as above to work with ConT_EXt.

And of course the Unix workhorses Emacs and Vi(m) have been ported to Windows as well. See below in the Unix/Linux section for further discussion.

3 Macintosh

Since the advent of Mac OS X, T_EX has made huge strides in the Mac community. Gone is Textures of OS 9 days, but OzT_EX has made the transition, and Gerben Wierda's i-Installer has raised the bar for T_EX installers on any platform. However, all of these OS X offerings require an outside editor. Fortunately, we have several excellent choices.

3.1 TeXShop

Without a doubt, my favorite integrated development environment (IDE) for T_EX on the Mac is [TeXShop](#) by Richard Koch and others. The built-in editor, which may be replaced by an external editor if you wish, features many of the things that make editing

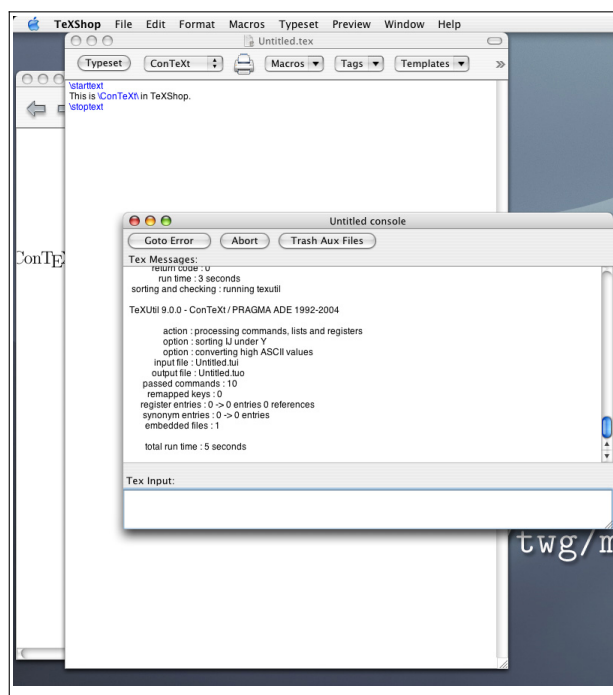


Figure 2 ConT_EXt in TeXShop

text more pleasant that we've talked about elsewhere in this column.

Auto completion will set up common snippets for you. So if you type `_` for a subscript, TeXShop will automatically add braces `_{}` and put your cursor in the middle. Command completion allows you to type in the first couple of letters of a command and hit the Escape key to complete the command. If there are multiple matches, hit the Escape key again to show other completions. So, typing `\be` and hitting Escape will produce `\beta`; hit Escape again to get `\begin`.

The Find panel features searching based on regular expressions. While regular expressions are not trivial to deal with, they offer superior search and replace capabilities. TeXShop is almost endlessly configurable, and its developer is very responsive and open to feature requests.

To make ConT_EXt the default T_EX format in TeXShop, go to the `TEXSHOP ▸ PREFERENCES` menu. Go to the "Typesetting" tab and click "ConTeXt" as the Default Command. Now, when you open a file or project and hit `Cmd-T`, TeXShop will automatically call `texexec`.

3.2 iTeXMac

iTeXMac was originally Jérôme Laurens' fork of the TeXShop project that has developed in its own direction. To enable ConT_EXt support, go to the TeX menu and select ACTIVATE PROJECT ▸ CONTEXT. If you need to customize the parameters, the selection is nearly the same, but choose SHOW PROJECT instead of ACTIVATE PROJECT.

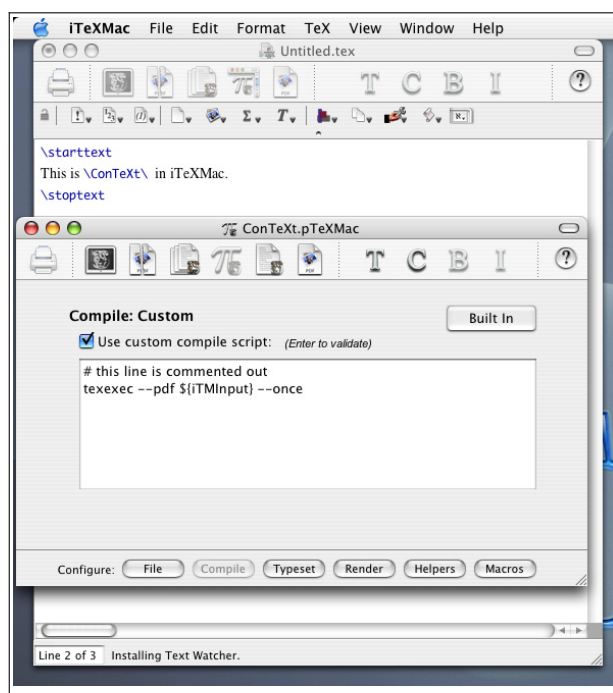


Figure 3 iTeXMac in action

Like TeXShop, iTeXMac is more than just a text editor. It is a complete T_EX IDE, and as such, you can choose a different editor if you so desire.

3.3 Others

Adam Lindsay has done a port of SciTE (see above) for the Mac X11 environment. For users coming from Windows who enjoy SciTE, this is a good choice.

Another popular editor for T_EX on Mac is **Alpha**, which is the default editor for

Oz \TeX . It has good support for Plain \TeX and \LaTeX , but I think you would have to roll your own Con \TeX t support.

In the commercial realm, a perennial favorite is **BEdit** by Bare Bones Software. Recently, they made a lite version, called TextWrangler, freely-available on their website.

Finally, providing a good segue into the Unix/Linux section, Emacs and Vi(m) both come with the Mac OS, and windowed versions are also available. See below for further discussion.

4 Unix/Linux

Unix is the birthplace of \TeX , and the editors here are powerful, mature, and cultish. Emacs vs. Vi(m) is easily as volatile as red state vs. blue state.

4.1 Emacs

Emacs is an editor, an ersatz GUI, and a way of life. It is my editor of choice on Unix. Together with **Auc \TeX** , it provides to my mind the complete editing experience. The learning curve is a bit challenging, but once you form “finger habits”, you can move around source files, edit, \TeX , check spelling, incrementally search for words, check into CVS, etc., all at lightning speed. If only my brain composed sentences as quickly! As of Auc \TeX v. 11.50, Con \TeX t support is built in.

4.2 Vi(m)

I use Vi(m), but to edit various system files. There are, however, people who swear by it to do all of their text editing. Vi(m) has \LaTeX support, but Con \TeX t support is limited, and at this point, basically hand-rolled.

4.3 Others

There is no shortage of text editors for Unix/Linux, and it seems as if new projects spring up daily. Aside from SciTE (described in the Windows section above), most are geared toward editing programming source code, although some (like [Kile](#)) do offer L^AT_EX support and can probably be coaxed into doing ConT_EXt as well.

Steve Peter is president and typographer at Beech Stave Press, where he has the joy and frustration of making multilingual text look good. Send him compliments at speter@dandy.net.