

\starttext

Steve Peter

1 Introduction

Welcome to the first installment of the \starttext column. Together we'll explore the vast world that ConT_EXt offers. If you don't already have ConT_EXt installed on your system, head over to the Pragma website at www.pragma-ade.com, and grab it. You can get just ConT_EXt, or a complete system with the underlying T_EX distribution. If you have no version of T_EX whatsoever, you might find Doug Waud and Tim Null's [notes on installing T_EX](#) to be useful.

Michael Guravage offers a good definition of what ConT_EXt is (over in the [Ask Nelly](#) section), so I'll concentrate here on how to use ConT_EXt. We'll start out with a traditional first document, and then we'll have some fun tweaking it.

To get the most out of this column, you should have ConT_EXt running on your system, and you should type in the examples as we go. If you don't have T_EX, don't have a computer,¹ or are just curious about ConT_EXt, and don't want to take the plunge yet, I've also supplied some illustrations.

Let's get started!

2 Hello, World!

Since ConT_EXt is a T_EX macro package, we'll follow the standard workflow by first entering the text of our document into a plain text file, interspersed with commands that tell T_EX to do something with the text (like make it bold, or format it like a footnote).

¹ Hey, then how are you reading this?

Then we run T_EX on the file, and finally we look at the beautiful output.

So fire up your favorite text editor² and enter the following:

```
\starttext  
  
Hello, World! This is \ConTeXt.  
  
\stoptext
```

The body of your document is enclosed in a `\start`—`\stop` pair. `\starttext` handles various setup details for you. Save the file as `document.tex`.

If you've used any variety of T_EX before, the next step is slightly different, so watch out. (And be amazed!) To run this document through T_EX, we'll use `texexec`, a perl script that greatly simplifies life. More about that a bit later. For now, just type the following in a shell window (if you're not using one of the editing environments discussed above):

```
texexec document
```

You should now have a new file, `document.dvi`, in your directory. You can view the file with `xdvi` on Unix, T_EXnscope on Mac, or `yap` on Windows. You can convert the dvi (DeVice Independent) file to pdf with the `dvipdfm` utility. However, your machine may be configured to run pdfT_EX automatically (as my machine is). In that case, simply open the resulting `document.pdf` file. Whether dvi or pdf, the result should look something like this:

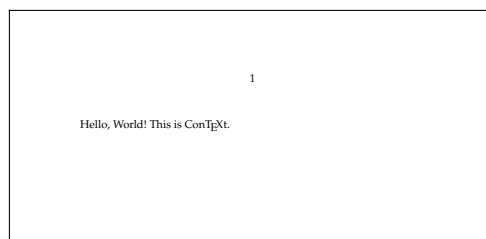


Figure 1 (The top of) your first ConT_EXt document!

² You can use any editor that outputs plain text, such as `emacs`, `vi`, Text Edit, or Notepad. Don't use a word processor like Word or OpenOffice Writer. There are also complete T_EX editing environments like TeXShop for Mac OS X or T_EXnicCenter on Windows that allow you to edit your files, run T_EX, and view the output from within a single application.

The page number at the top tells us that this is a default ConT_EXt document, and not simply a Plain T_EX one. Let's put the number into the footer.

Setting up something like the location of the page number is done with a `\setup` command in ConT_EXt. Don't worry right now about the exact form of the command. We'll go over them in much greater detail in a later column. So, to put the number in the footer, add the following line to the top of your document, before the `\starttext`. Run it through `texexec` and look at the file produced.

```
\setuppagenumbering [location=footer]
```

Now the folio is in the footer, like in this article.

Text of any length is usually subdivided. Let's put in some sections. This time *after* `\starttext`, put the line

```
\section{First section}
```

Add a few more `\sections` and some text.³ We'll need them for the next section.

3 texexec

I mentioned before that `texexec` greatly simplifies life. Why is that? Well, typesetting is a complicated business, and T_EX frequently has to collect information on one pass to use in a later pass. For example, let's add a table of contents. Just after `\starttext`, add

```
\completecontent
```

But how does T_EX know what page `\section two` is on until after it has typeset the document? The answer, of course, is that it doesn't. T_EX gathers up information from all the `\sections` you have in the document and writes that information to an auxiliary file. Normally, you have to then run T_EX a second time so that T_EX can read that information in and set the table of contents. (And if the TOC is long, it will push everything down, meaning that you have to rerun T_EX again!)

Sometimes you find yourself rerunning T_EX needlessly just to make certain there aren't any unresolved references. But `texexec` changes that. It automatically reruns T_EX as many times as necessary, so you can go refill your coffee.

³ To get a bunch of text quickly, try `\dorecurse{20}{\input knuth \par}`.

4 Fun and fancy

Just to whet your appetite, let's take a quick look at a couple of fancier things ConTeXt can do. We'll go into details in future columns, both here and in *TUGboat*. I realize these are a bit of a jump from the basic formatting considered in the other sections, but since we're just setting out, I thought I'd give you a glimpse of some really fancy stuff.

To maintain high typographic standards,⁴ you often have to align text, graphics, etc., to a grid, and your text should maintain a consistent position on the baseline grid. Add this to the top of your document and process it with `texexec`.

```
\setuplayout[width=middle,location=middle,grid=yes,marking=color]
\moveongrid[both]
\showgrid
```

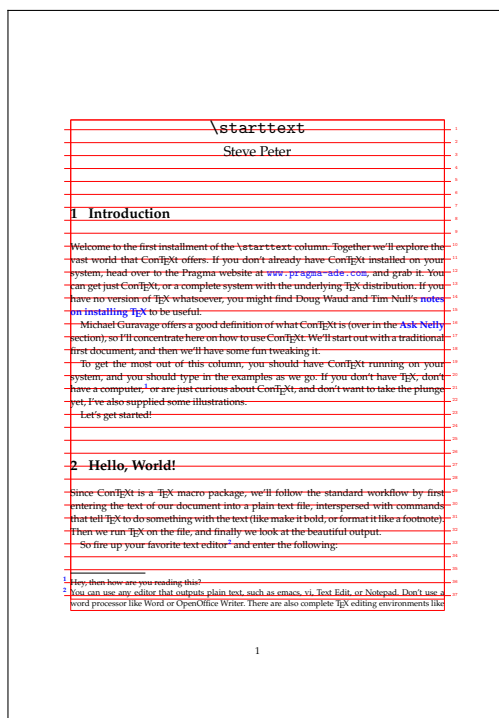


Figure 2 The matrix?
No, it's the grid.

⁴ See the discussion, for example, in Robert Bringhurst, *The Elements of Typographic Style*.

If you've ever tried to do that with Plain $\text{T}_{\text{E}}\text{X}$, or even $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, you know what a pain it can be. However, $\text{ConT}_{\text{E}}\text{Xt}$ does it easily, and even shows you where the grid is, so you can debug troublesome documents.

After all $\text{T}_{\text{E}}\text{X}$ is, when you get down to it, a programming language. That means at some point you'll need to debug your documents. The grid feature is but one of several nice visual debugging tools provided with $\text{ConT}_{\text{E}}\text{Xt}$. Add this to the beginning of your document to gain a view of how $\text{T}_{\text{E}}\text{X}$ puts boxes and glue together.

```
\showmakeup
```

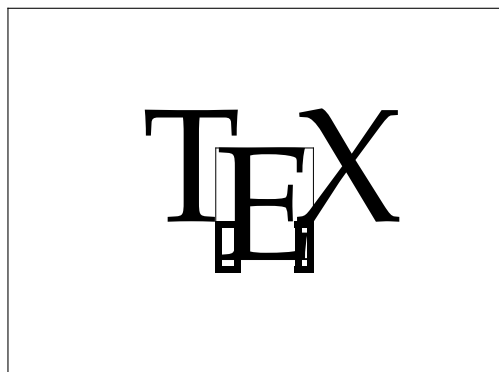


Figure 3 What $\text{T}_{\text{E}}\text{X}$ might look like to $\text{T}_{\text{E}}\text{X}$

Here we can see the bounding box for the E in $\text{T}_{\text{E}}\text{X}$, along with the negative kerns, shown as the thicker boxes near the base of the E. One more useful visualization command shows you the layout on the page of your text block, margins, headers and footers.⁵

```
\showframe
```

5 Links

I hope you've enjoyed this first look at $\text{ConT}_{\text{E}}\text{Xt}$. There are numerous topics we haven't addressed yet, such as cross references, hyperlinks, indexes, figures and other graphics, and $\text{ConT}_{\text{E}}\text{Xt}$'s incredible support for PDF trickery.

⁵ For more on this aspect of visual debugging, see Hans Hagen's paper in *TUGboat* [Volume 19, Number 3](#).

There's a lot of information out there, and there's plenty to explore. Start with the documentation on the Pragma website (www.pragma-ade.com). For examples, check the ConTEXt wiki at contextgarden.net and [Bill McClain's excellent beginner's help page](#). Last, but certainly not least, you can jump into the never-ending discussion on the official mailing list at <http://www.ntg.nl/mailman/listinfo/ntg-context>.

Join us here and in the pages of *TUGboat* for more on the practical use of ConTEXt.

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.