

ConTeXt for beginners

Willi Egger

Abstract

In 2017 we had a joint meeting of BachoTeX and TUG. During this conference a workshop for ConTeXt beginners was requested. The following article comes from this workshop.

As with any typesetting system offering possibilities to handle virtually any project, ConTeXt is a huge system. As with the workshop, this article can only lift the veil a little. The workshop was a hands-on session for playing with basic elements to create a document. Towards the end there was a chance to work on a small project — a single-sided document containing all the elements to build an invoice.

Keywords: ConTeXt, beginner, workshop

1 About ConTeXt

TeX, developed by Donald Knuth in the 1970s and 1980s, is still widely used. There are three principal flavours: Plain TeX, L^ATeX and ConTeXt.

ConTeXt is an advanced macro package which uses TeX as an engine. ConTeXt is a development of PRAGMA ADE in Hasselt, The Netherlands. The code base was written and continues to be actively maintained by Hans Hagen. It was first developed for typesetting schoolbooks and school math.

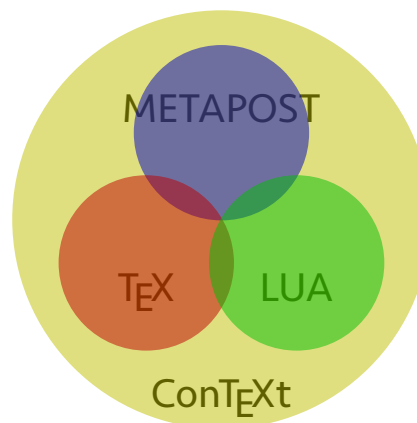
At present two versions of ConTeXt are in use. ConTeXt MkII makes use of the pdfTeX engine. This version is not developed any further and the code is frozen, but bug-fixes are still applied. The current version ConTeXt MkIV makes use of LuaTeX and is under continuous development. For new users of ConTeXt we advise starting with ConTeXt MkIV with LuaTeX.

In contrast to other macro packages ConTeXt is a primarily monolithic system. Essentially all facilities are provided out of the box. Nevertheless, some modules for add-on features, written by third parties, can be installed and invoked as necessary.

ConTeXt makes heavy use of MetaPost. The latter is therefore used as a library which makes it possible to generate graphics at runtime very efficiently.

ConTeXt can be used for processing TeX-coded documents but it also provides a fully developed environment to process XML-coded data. ConTeXt has interface capabilities with SQL databases.

Graphically the ConTeXt-typesetting system can be presented as follows:



2 Availability

ConTeXt is included in TeX Live, so you may well already have it. However, for users who wish to use ConTeXt extensively or exclusively, we recommend the standalone release, which can be obtained from the ConTeXt garden wiki: wiki.contextgarden.net. The garden also has information on installation of the system for current operating systems, notably Windows, Mac OS X, and Linux.

3 Documentation

ConTeXt is a highly complex typesetting system. It comes with many detailed manuals, many of which are included in the distribution, and all can be downloaded from the Pragma website (pragma-ade.com). Along with this, there are a couple of printed books; these are available from H₂O-books (see also section 8).

4 Basic elements

A basic difference compared to other macro packages is that in ConTeXt almost all commands are defined as structured elements, enclosed by `start ... stop` commands. This is a prerequisite for working with XML and provides much control over the beginning and end of an element.

Each document starts with `\starttext` and ends with `\stoptext`. Other examples of such structured commands are:

- `\startchapter ... \stopchapter`
- `\startsection ... \stopsection`
- `\startplacefigure ... \stopplacefigure`

The system comes with a reasonable set of presets for many constructs that are needed while building a new document. ConTeXt provides maximum flexibility to adjust almost anything according to the user's wishes. For this purpose most commands

come with a `\setup...` command, where appropriate variables can be customized. A couple of examples, taking options in square brackets (common in ConTeXt):

- `\setupframedtext[...] [...]`
- `\setuplayer[...] [...]`

It is important to note that, in case it's needed, ConTeXt still understands most of Plain TeX.

5 Export formats

When running a document with ConTeXt the output is by default PDF. Correctly coded documents can also be exported as XML, HTML, XHTML and ePub.

6 A first ConTeXt document

As with other TeX environments, it is best to use a text editor for coding.

For the first document we can open a TeX file, say `myfirstfile.tex` and type e.g.

```
\starttext
Dzień dobry at the \CONTEXT
beginners tutorial!
\stoptext
```

After saving this file, we can compile it from the terminal with `context myfirstfile`. Provided that the installation was successful it will result in a document typeset as A4 with the name `myfirstfile.pdf`.

It is worth mentioning that ConTeXt is UTF-8 aware out of the box, so typesetting accented glyphs, as above, is no problem.

7 Elements of a document

Now that we know that the installation is fine, we can start describing the common elements of our documents. In a second step these elements will be used in a small project for creating a simple invoice.

We will deal in this first step with the paper size, the general layout of the page, fonts, two types of table environments, headers and footers, and the layer mechanism and the basic placement of graphical elements.

7.1 Paper size

By default ConTeXt produces A4 pages, but of course one is in no way bound to this format. The system comes with many predefined paper sizes including the DIN-A series, American paper sizes and oversized paper sizes for the print industry.

Because ConTeXt produces A4 by default we do not have to set up the paper size for that case. However if you are using letter size you will have to tell this to the system in the preamble of each document:

```
\setuppapersize[letter][letter]
```

To set a custom paper size the following two commands are needed:

```
\definepapersize[Mypsize][width=80mm,height=95mm]
\setuppapersize[Mypsize][A4]
```

The `\setuppapersize` command accepts two arguments, used to make up the page and place it on paper. The first argument tells ConTeXt the size of the page, and the second argument tells the system how to put the page on a certain paper size for printing. To both arguments one can add the attributes `portrait` or `landscape`. So the above defines a small layout sized, to be placed on A4 paper.

Of course one often wants the layout size and printing size to be the same, so both arguments are the same:

```
\definepapersize[Mypsize][width=80mm,height=95mm]
\setuppapersize[Mypsize][Mypsize]
```

Finally, we do not need to define our own paper sizes when using any of the predefined ones. For instance, to place an A4 page on an A3 in landscape:

```
\setuppapersize[A4][A3,landscape]
```

7.2 Page layout

Although ConTeXt comes out of the box with a decent set of presets one has all sort of possibilities to adapt a layout of a page to the needs at hand. In order to be able to work with this rather extensive command it is necessary to have a look at the division of the space of a page.

The following drawing shows all regions which we can manipulate individually: the outermost areas indicated by `top` and `bottom` as well as `leftmargin` and `leftedge` and such. The edges are normally only used in interactive documents like presentations. We will not discuss those here.

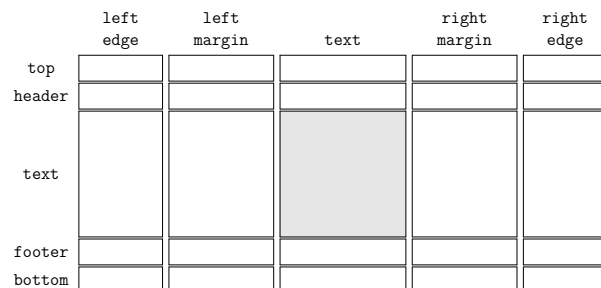


Figure 1: Page layout areas

The one command to set up these different regions of the page is `\setuplayout`. An example:

```
\setuplayout
[topspace=15mm,
backspace=15mm,
header=0pt,
footer=1.2\bodyfontsize,
height=middle,
width=middle]
```

This command has many arguments and needs to be studied in the documentation. Designing a page layout is done always on a right-hand (odd) page. When we need a double-sided document we tell this to ConTeXt and the system will automatically switch the backspace and eventually-determined margins to the right position on the even page.

- The key `topspace` gives the white space above the content, and key `backspace` denotes the white space left of the content.
- Normally ConTeXt has an active header. If we do not want a header we switch the key `header` to zero, using any of the usual TeX dimensions, e.g. pt, mm, cm, in. When the header is zero, automatically the header-distance (not shown above; it's the space below the header and above the text) is set to zero too. The same applies to the footer.
- The key `height` specifies the total height in which any content can occur, i.e. it includes the header, header-distance, text body, footer-distance and footer. If we give `height=middle`, the system will calculate the height of the typesetting area such that there is an even distribution of white on top and bottom of the page.
- The key `width` allows us to set the width of the typesetting area for the header, text-body and footer. `width=middle` causes ConTeXt to calculate the width such that at the right side of the typesetting area is the same amount of white space as the `backspace`.
- If the document has content in the margins, we can set up margins for the left and right, as well as the distance they should get from the text body. Margin content is placed into the backspace and adjacent to the text body at the right side. When using margins we need to adjust the backspace and the text body width in order to keep this content on the paper.

7.2.1 Show me the layout

Specifying your page layout can be difficult in the beginning. In order to facilitate this there is a handy command to show the actual page layout graphically: you can issue `\showframe` inside `\starttext ... \stoptext` or outside at the top of the docu-

ment. For an example of how this looks, see the appendix.

If you want to know all the dimensions set for the page you can put `\showlayout` into the document. It will show all dimensions on one page. The default values of page parameters in ConTeXt are as follows.

<code>\paperheight</code>	29.7000cm
<code>\paperwidth</code>	21.0000cm
<code>\printpaperheight</code>	29.7000cm
<code>\printpaperwidth</code>	21.0000cm
<code>\topspace</code>	2.4998cm
<code>\backspace</code>	2.5000cm
<code>\makeupheight</code>	25.0000cm
<code>\makeupwidth</code>	14.9999cm
<code>\topheight</code>	0.0000cm
<code>\topdistance</code>	0.0000cm
<code>\headerheight</code>	1.9999cm
<code>\headerdistance</code>	0.0000cm
<code>\textheight</code>	21.0002cm
<code>\footerdistance</code>	0.0000cm
<code>\footerheight</code>	1.9999cm
<code>\bottomdistance</code>	0.0000cm
<code>\bottomheight</code>	0.0000cm
<code>\leftedgewidth</code>	0.0000cm
<code>\leftedgedistance</code>	0.0000cm
<code>\leftmarginwidth</code>	2.6564cm
<code>\leftmargindistance</code>	0.4217cm
<code>\textwidth</code>	14.9999cm
<code>\rightmargindistance</code>	0.4217cm
<code>\rightmarginwidth</code>	2.6564cm
<code>\rightedgedistance</code>	0.0000cm
<code>\rightedgewidth</code>	0.0000pt
<code>\bodyfontsize</code>	10.0000pt
<code>\lineheight</code>	12.5720pt
<code>\strutheightfactor</code>	.72
<code>\strutdepthfactor</code>	.28
<code>\topskipfactor</code>	1.0
<code>\maxdepthfactor</code>	0.4

7.2.2 Header and footer

If we have set up the page to carry a header and/or a footer with `\setuplayout`, we can fill those areas with content. The commands to put content into the header and footer are:

```
\setupheadertexts[lo][ro][re][le]
\setupfootertexts[lo][ro][re][le]
```

In double sided documents we can have four different pieces of content, two for the left (odd) and two for the right (even) page. For a single sided document we need only two arguments. The above letters `lo .. re` are placeholders; to explain their meaning:

lo text left on odd pages
 ro text right on odd pages
 re text right on even pages
 le text left on even pages

We can give normal text into the fields or add commands, e.g. insert the page number or the current section title. Assuming we have a single sided document, we might do

```
\setupheadertexts
  [ {\getmarking[chapter]} ]
  [ \userpagenumber/\totalnumberofpages ]
\setupfootertexts
  [ \jobname ]
  [ \currentdate ]
```

We could also put information in the header-margin as shown in figure 1. In a double-sided document it could be

```
\setupheadertexts
  [margin] [ \userpagenumber ] [Document A]
```

In a single-sided document it would be

```
\setupheadertexts [margin] [ \userpagenumber ]
```

7.2.3 Marginal text

For a document with content in the margin we have to set this up with the `\setuplayout` command. For instance we could set the parameters to the following values

```
topspace          1.5cm
backspace         2.5cm
leftmargin        2.0cm
leftmargindistance 0.2cm
textwidth         15.0cm
rightmargindistance 0.2cm
rightmargin       2.0cm
```

In the text, we can then say

```
\inrightmargin { \tfxx text in the margin. }
```

to get marginal text as shown here. (`\tfxx` selects a small typewriter font; we'll briefly discuss fonts later, but see wiki.contextgarden.net/Font_Switching for more.)

7.3 Tables

7.3.1 Tabulation

The tabulation environment is specially suited to typeset tables in the text flow. The environment provides many facilities to customize a table, but there is no support for vertical rules. This environment uses a template at the beginning of the table, as shown here:

```
\starttabulate[|l|c|r|p|]
\NC left \NC center \NC right \NC para \NC\NR
\stoptabulate
```

The `|` characters in the template here merely delimit column specifications, and do not indicate

rules to be typeset. Additional option characters can be added in a column as follows.

Options for column width:

```
w(dim) fixed column width
p(dim) fixed paragraph width
p      maximum width paragraph
```

Options for style:

```
B,I,R,S,T bold, italic, roman, slanted, typewriter
m,M      inline math, display math
```

Then, within the tabulate body, `\NC` indicates the next column and `\NR` the next row, as shown above. Other commands can be included, for example, these can add vertical space between rows:

```
\TB[halfline] Vertical space of half a line
\TB[line]     Vertical space of a whole line
\TB[1cm]     Vertical space of 1 cm
```

7.3.2 Natural tables

Natural tables are an environment for large tables and provide a huge set of possibilities to customize cells, columns and or rows. The coding is generally similar to HTML. A basic table looks like this:

```
\bTABLE
  \bTR \bTD a \eTD \bTD x \eTD \bTD y \eTD \eTR
  \bTR \bTD b \eTD \bTD x \eTD \bTD y \eTD \eTR
  ...
\TABLE
```

Thus, the environment is started with the command `\bTABLE` and ended with `\TABLE`; each row starts with `\bTR` and ends with `\eTR`; and each cell of a row starts with `\bTD` and is closed with `\eTD`.

Default behaviour is tight cells, all frames on. For customizing we can use `\setupTABLE`:

```
\setupTABLE[row]
  [number,odd,even,first,last,each] [...]
\setupTABLE[column]
  [number,odd,even,first,last,each] [...]
```

Options are the same as in the `\framed` environment (wiki.contextgarden.net/Framed) and influence most aspects of how the table appears. So we have quite a number of [*option*]=*key*] possibilities for setting up the frame (each side), style, color, rule thickness etc.

If we want a consistent table design throughout the document we might place these setups at the beginning of the document. If we then need a local adaptation we place the `\setupTABLE` command inside the `\bTABLE ... \TABLE` construct. We can also attach to a given `\bTR` or `\bTD` options for this specific row or cell.

Cells can span multiple columns or rows, as in: `\bTD[nx=2]`, `\bTD[ny=3]`.

Natural tables can break over pages and we can define a table header for the first page and a table header for the following pages. It is also possible to define a table foot.

```
\bTABLEhead
  \bTR \bTD A \eTD \bTD B \eTD \bTD C \eTD \eTR
\eTABLEhead

\bTABLEnext
  \bTR \bTD X \eTD \bTD Y \eTD \bTD Z \eTD \eTR
\eTABLEnext

\bTABLEfoot
  \bTR[bottomframe=on] \eTR
\eTABLEfoot

\bTABLEbody
  \bTR \bTD 1 \bTD 10 \eTD \bTD 100 \eTD \eTR
  \bTR \bTD 1 \bTD 10 \eTD \bTD 100 \eTD \eTR
\eTABLEbody
```

7.4 Layers

In ConTeXt one can place content into specific locations. This is done with layers. The procedure is to define the layer, fill it and place it.

7.4.1 Defining a layer

```
\definelayer[Logo]
  [width=3cm, height=4cm]
```

The layer is by default attached to the left top corner of the typesetting area, if not stated otherwise.

7.4.2 Fill the layer

To fill a defined layer we use `\setlayer`, with the name of the layer as a first bracketed option, and x and y positioning dimensions relative to the anchor point as a second option. Finally, the actual content is given between braces (not brackets) . This can be any content like figures, tabulations, tables ...

```
\setlayer[Logo]
  [x=-15mm,y=-35mm]
  {\externalfigure[cow] [width=3cm]}
```

7.4.3 Placing the layer

The layer is placed with `\placelayer[<name>]`. A layer can only be placed (or flushed) if there is already content on the page. If the flushing is done before the page is started, the layer will not appear. If we have to place the layer as the first action we can use the command `\strut` or `\null`, which add the necessary anchor point.

```
\strut
\placelayer[Logo]
```

7.4.4 Dealing with floats

In TeX figures, pictures and tables often are floating objects, meaning that the typesetting system determines where such an object is to be placed.

ConTeXt supports the following image formats: JPEG, PNG, PDF, MetaPost; it will also honor EPS if Ghostscript is installed, as additional converters can kick in.

Normally pictures and other graphics are separate from the TeX sources, so we need to tell ConTeXt where to look for the required files:

```
\setupexternalfigures[location={local,default}]
```

In this case ConTeXt is looking for picture files locally and secondly in the TeX tree.

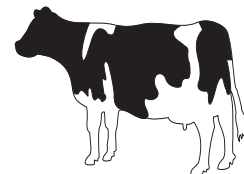
If we want ConTeXt to look for the files in a specific directory, we add the key `global` and give the path after the key `directory`:

```
\setupexternalfigures
  [location={local,default,global},
  directory={/MyDocs/Bachotex2017/tut/figures}]
```

7.4.5 Placing a figure

After the setup is done, we can insert a figure as follows, here to the right of the text:

```
\startplacefigure
  [title=Cow drawing.,
  location=right]
  {\externalfigure[cow]
  [height=.1\textheight]}
\stopplacefigure
```



To omit any caption and numbering we can add the keys `title=`, `number=`.

7.5 Fonts

In today's TeX environments, thanks especially to the great work by the Polish GUST e-foundry, we have an excellent collection of fonts available in modern font formats. The following are included in the ConTeXt standalone distribution: Latin Modern, the TeX Gyre collection, Antykwa Torńska, Iwona, Kurier, and XITS.¹

Of course a great number of additional fonts are available, included in TeX Live and/or separately downloadable, such as: Gentium (from SIL International), DejaVu (based on an original design from Bitstream), ...

7.5.1 Playing with built-in fonts

If no font is specified, ConTeXt will use Latin Modern by default. The traditional way of specifying a font is as follows:

```
\setupbodyfont [pagella,rm,10pt]
```

¹ Khaled Hosny adapted STIX to TeX needs to make XITS.

Here we give the name of the font (family), followed by the required style and the size.

With the \TeX Gyre fonts, we have a nice collection of fonts in the OpenType format, covering the so-called “base 35” PostScript fonts.

serif	rm	termes	(Times)
		pagella	(Palatino)
		schola	(New Century Schoolbook)
sans	ss	bonum	(Bookman)
		heros	(Helvetica)
mono	tt	adventor	(Avant Garde)
		cursor	(Courier)
calligraphic	cg	chorus	(Zapf Chancery)

In order to be able to use serif, sans serif, mono-space and math fonts mixed in one document, Con \TeX t has predefined sets (families), which are called “typefaces”. E.g. the `pagella` option offers:

```
serif pagella
sans latin modern
mono latin modern
math pagella
```

7.5.2 Another way of using fonts

As long as you do not use commercial fonts, for which you have to set up the typescripts/font loading instructions yourself, you can easily use the predefined typefaces as shown above.

So especially for the use of fonts, i.e. system fonts or commercial fonts, there is a font selection mechanism (written by Wolfgang Schuster) which is part of the Con \TeX t core. For a font that is not supported out of the box you can define a font family like this:

```
\definefontfamily [dejavu] [serif] [DejaVu Serif]
\definefontfamily [dejavu] [sans] [DejaVu Sans]
\definefontfamily [dejavu] [mono] [DejaVu Sans Mono]
\definefontfamily [dejavu] [math] [XITS Math]
                                [scale=1.1]

\definefontfamily [office] [serif] [Times New Roman]
\definefontfamily [office] [sans] [Arial]
                                [scale=0.9]

\definefontfamily [office] [mono] [Courier]
\definefontfamily [office] [math]
                                [TeX Gyre Termes Math]

\definefontfamily [linux] [serif] [Linux Libertine 0]
\definefontfamily [linux] [sans] [Linux Biolinum 0]
\definefontfamily [linux] [mono] [Latin Modern Mono]
\definefontfamily [linux] [math]
                                [TeX Gyre Pagella Math] [scale=0.9]

\definefontfamily [myfamily] [mono] [TeX Gyre Cursor]
                                [features=none]
```

When you want to combine fonts of which the design sizes are not directly compatible you can add

a scaling factor to the definitions, as shown above. [The line breaks above are for *TUGboat*’s narrow columns; normally such definitions are written all on one source line, although this is not required.]

7.5.3 Your own typescripts

The same approach is used for commercial fonts. Either you use Wolfgang’s core module as above, or you write your own typescripts, as briefly outlined here.

In any case, when adding a new font unknown to Con \TeX t the file database must be renewed. The way to do this is to open a terminal and issue the command `context --generate`.

Suppose you have the font Seravek. The set of typescripts would read as follows. First the font’s filenames are mapped on a symbolic name inside a typescript, which takes two arguments: a category name, such as `sans` or `serif`, and a symbolic name.

```
\starttypescript [sans] [seravek]
\definefontsynonym [Seravek-Regular]
                    [file:Seravek-Regular][features=default]
\definefontsynonym [Seravek-Bold]
                    [file:Seravek-Bold][features=default]
\definefontsynonym [Seravek-Italic]
                    [file:Seravek-RegularItalic][features=default]
\definefontsynonym [Seravek-Bold-Italic]
                    [file:Seravek-BoldItalic][features=default]
\stoptypescript
```

In a second step, the symbolic names of the font files are mapped to Con \TeX t’s internal names. As with the first typescript, this typescript has the same category name and the symbolic name.

```
\starttypescript [sans] [seravek]
\definefontsynonym [Sans]
                    [Seravek-Regular][features=default]
\definefontsynonym [SansItalic]
                    [Seravek-Italic][features=default]
\definefontsynonym [SansBold]
                    [Seravek-Bold][features=default]
\definefontsynonym [SansBoldItalic]
                    [Seravek-Bold-Italic][features=default]
\definefontsynonym [SansCaps]
                    [Seravek-Regular][features=smallcaps]
\definefontsynonym [SansBoldCaps]
                    [Seravek-Bold][features=smallcaps]
\definefontsynonym [SansItalicCaps]
                    [Seravek-Italic][features=smallcaps]
\definefontsynonym [SansBoldItalicCaps]
                    [Seravek-Bold-Italic][features=smallcaps]
\stoptypescript
```

Now that the definitions are ready we create a third typescript, which defines the font family as a typeface. This typescript has a symbolic name with which we will use the fonts inside the document.

```
\starttypescript [Seravek]
\definetypface[Seravek][ss][sans][seravek][default]
\definetypface[Seravek][mm][math][palatino][default]
\stoptypescript
```

The three typescript definitions can be saved in a file with the name `type-imp-seravek.tex`. This file is best placed in the \TeX tree, e.g. (in the standalone Con \TeX t distribution):

```
.../tex/texmf-context/tex/context/user.
```

To use this typeface in the document the font is set up with:

```
\usetypescriptfile[type-seravek]
\usetypescript[seravek]
\setupbodyfont[Seravek,ss,10pt]
```

And now the default text font will be Seravek, in the sans serif style that we've defined.

8 Documentation

What has been shown so far is only a glimpse of what Con \TeX t can do. Learning Con \TeX t calls for careful study of the documentation. Many manuals written by Hans Hagen are included in the distribution; they can be found in the \TeX tree, in (standalone Con \TeX t) `texmf-context/doc/context/documents` or (\TeX Live) `texmf-dist/doc/context`.

Also available is a large wiki containing very useful help, including command references with many explanations and examples. In addition, it has both shorter and longer articles contributed by many Con \TeX t users. You can find it at `wiki.contextgarden.net`. For convenience, a summary reference of the main commands in this tutorial are also available at `tug.org/TUGboat/tb38-3/tb120egger-cmds.pdf`. (They were prepared using the built-in command `\showsetup`, e.g. `\showsetup[setuplayout]`.)

A large set of test files is also available. These are used in development, but also contain very useful information for self-study. On the page `pragma-ade.com/download-1.htm` you can find a link to them as `context/latest/cont-tst.7z`.

Last but not least there is a mailing list to which you are invited to join. On this list practical issues with the system and questions are posted. The address is `ntg-context@ntg.nl` (with archives, etc., at `ntg.nl/mailman/listinfo/ntg-context`) — don't be shy or afraid; also simple questions are answered promptly!

9 Acknowledgements

\TeX is still alive after so many years. I would like to thank Hans Hagen cordially for his tremendous work given to us for free. With his system we are still competitive in the modern environment of typesetting from different sources, producing PDF from XML, linking to databases, exporting XML and other formats like ePub. The transition from pdf \TeX to Lua \TeX was and is supported by a number of people, notably Taco Hoekwater, Hans Hagen, Helmut

Henkel and Luigi Scarso. We owe them a big thank you! Another member of the community who spends a considerable amount of time and effort in supporting the system is Wolfgang Schuster. I would like to thank him for all that he does. Finally, behind the scenes a lot happens in order to make Con \TeX t available for anybody from the Con \TeX t garden. I would like to thank Mojca Miklavec for all the energy she puts into the preparation and testing of the Con \TeX t distribution.

◇ Willi Egger
Maasstraat 2
5836 BB SAMBEEK
The Netherlands
w dot egger (at) boede dot nl

A Bringing the elements together: An invoice skeleton

The following text shows a basic approach to setting up an invoice. It starts with a preamble setting up the body font and other definitions. After `\starttext` the invoice is built. The result is a one-page document containing all elements for the invoice. It's true that it is not yet the most beautiful document in the world, but it is the base for tuning ...

```
% Setup font to be used
\setupbodyfont[pagella,rm,10pt]

% Setup path to find graphics
\setupexternalfigures[location={local,default}]

% Switch off automatic page numbering
\setuppagenumbering[location=]

% Setup page layout
\setuplayout
[ topspace=15mm,
  backspace=15mm,
  header=40mm,
  footer=2.2cm,
  height=middle,
  width=middle,
  leftmarginwidth=10mm,
  rightmarginwidth=14mm]

% Setup headers and footers
\startsetups[Header]
  \framed
    [height=\headerheight,
     align=lohi,
     frame=off,
     foregroundcolor=green,
     foregroundstyle=\bfd]
    {Agricultural Services}
\stopsetups
%
\setupheadertexts[][\setups{Header}][][\setups{Header}]

\startsetups[Footer]
\switchtobodyfont[8pt]
\starttabulate[|p|p|p|p|]
```

```

\NC Bank account      \EQ P10123 23 3445 678      % Receiver address
      \NC Address \EQ Jackowskiego 12 m 3 \NC\NR
\NC Chamber of Commerce \EQ North-14 53 21      \startlines
      \NC      \NC 61-757 Pozna\'n      \NC\NR      Name
\NC VAT number      \EQ PL12653007      Company
      \NC      \NC      \NC\NR      Street
\NC Phone      \EQ \unknown      Postal code Town
      \NC e-Mail \NC info at boede.nl      \NC\NR      \stoptlines
\stoptabulate      \blank[1cm]
\stopsetups
\setupfootertexts[] [\setups{Footer}] [] [\setups{Footer}] % Invoice data
% Setup logo layer      \starttabulate{111lr}
\definelaye      \NC \bfc Invoice \NC      \NC\NR
      [Logo]      \NC Invoice date \EQ 26-04-2017 \NC\NR
      [width=3cm, height=4cm]      \NC Invoice number \EQ 01-2017 \NC\NR
\stoptabulate
% Standard text included on each invoice      \blank[2cm]
\startbuffer[Conditions]
All prices are in EUR unless stated otherwise.
\blank[small]
This invoice is payable within 14 days
after date of issuing.
\blank[small]
Our general terms and conditions apply to all quotes,
contracts and services unless stated otherwise.
The general terms and conditions are deposited
at the record office of the court of justice at
s'-Hertogenbosch, The Netherlands. On request
a copy in the Dutch language is available for free.
\stopbuffer
\startbuffer[Thanks]
Thank you for your order.
\stopbuffer
% for our example
\showframe
\showlayout
% Begin document
\starttext
% Logo
\setlayer
      [Logo]
      [x=-5mm,
      y=-42mm]
      {\externalfigure[cow] [width=3cm]}
%
\setlayer
      [Logo]
      [x=148mm,
      y=137mm]
      {\externalfigure[mill] [height=4cm]}
%
\placelayer[Logo]

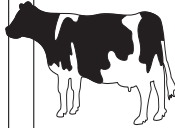
```

```

% Invoice content
\setupTABLE[each] [each] [frame=off]
\setupTABLE[r] [1] [style=bold]
\setupTABLE[c] [1] [width=2.5cm]
\setupTABLE[c] [2] [width=9cm]
\setupTABLE[c] [3] [width=2.5cm,align=flushright]
\setupTABLE[c] [4] [width=1.5cm,align=flushright]
\setupTABLE[c] [5] [width=2.5cm,align=flushright]
\bTABLE
  \bTR[bottomframe=on]
    \bTD Date \eTD\bTD Description \eTD
    \bTD Quantity \eTD\bTD Price \eTD\bTD Amount \eTD
  \eTR
  \bTR
    \bTD 26-04-2017 \eTD\bTD Delivery of goods \eTD
    \bTD 10 \eTD\bTD 25.00 \eTD\bTD 250.00 \eTD
  \eTR
  \bTR[topframe=on,rulethickness=1.5pt]
    \bTD \eTD\bTD \eTD\bTD Subtotal \eTD
    \bTD \eTD\bTD 250.00 \eTD
  \eTR
  \bTR
    \bTD \eTD\bTD \eTD\bTD VAT 20\% \eTD
    \bTD \eTD\bTD 50.00 \eTD
  \eTR
  \bTR
    \bTD \eTD\bTD \eTD\bTD Total \eTD\bTD \eTD
    \bTD[bottomframe=on,rulethickness=1.5pt]
    300.00\eTD
  \eTR
\eTABLE
\vfil
\getbuffer[Thanks]
\blank[2*big]
\getbuffer[Conditions]
\stoptext

```

The output is on the next page [scaled for *TUGboat*, so the absolute dimensions given in the source will only match proportionally; sorry].

	 <h2 style="margin: 0;">Agricultural Services</h2>																										
	<p>Name Company Street Postal code Town</p> <p>Invoice Invoice date : 26-04-2017 Invoice number : 01-2017</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 20px;"> <thead> <tr> <th style="text-align: left;">Date</th> <th style="text-align: left;">Description</th> <th style="text-align: right;">Quantity</th> <th style="text-align: right;">Price</th> <th style="text-align: right;">Amount</th> </tr> </thead> <tbody> <tr> <td>26-04-2017</td> <td>Delivery of goods</td> <td style="text-align: right;">10</td> <td style="text-align: right;">25.00</td> <td style="text-align: right;">250,00</td> </tr> <tr> <td></td> <td style="text-align: right;">Subtotal</td> <td></td> <td></td> <td style="text-align: right;">250,00</td> </tr> <tr> <td></td> <td style="text-align: right;">VAT 20%</td> <td></td> <td></td> <td style="text-align: right;">50,00</td> </tr> <tr> <td></td> <td style="text-align: right;">Total</td> <td></td> <td></td> <td style="text-align: right; border-top: 1px solid black;">300,00</td> </tr> </tbody> </table> <p style="margin-top: 20px;">Thank you for your order.</p> <p>All prices are in EUR unless stated otherwise. This invoice is payable within 14 days after date of issuing. Our general terms and conditions apply to all quotes, contracts and services unless stated otherwise. The general terms and conditions are deposited at the record office of the court of justice at s'-Hertogenbosch, The Netherlands. On request a copy in the dutch language is available for free.</p> 	Date	Description	Quantity	Price	Amount	26-04-2017	Delivery of goods	10	25.00	250,00		Subtotal			250,00		VAT 20%			50,00		Total			300,00	
Date	Description	Quantity	Price	Amount																							
26-04-2017	Delivery of goods	10	25.00	250,00																							
	Subtotal			250,00																							
	VAT 20%			50,00																							
	Total			300,00																							
	<table style="width: 100%; border: none;"> <tr> <td style="width: 33%;">Bank account</td> <td style="width: 33%;">: PI0123 23 3445 678</td> <td style="width: 33%;">Address</td> <td style="width: 33%;">: Jackowskiego 12 m 3</td> </tr> <tr> <td>Chamber of Commerce</td> <td>: North-14 53 21</td> <td></td> <td>: 61-757 Poznań</td> </tr> <tr> <td>VAT number</td> <td>: PL12653007</td> <td></td> <td></td> </tr> <tr> <td>Phone</td> <td>: . . .</td> <td>e-Mail</td> <td>: info at boede.nl</td> </tr> </table>	Bank account	: PI0123 23 3445 678	Address	: Jackowskiego 12 m 3	Chamber of Commerce	: North-14 53 21		: 61-757 Poznań	VAT number	: PL12653007			Phone	: . . .	e-Mail	: info at boede.nl										
Bank account	: PI0123 23 3445 678	Address	: Jackowskiego 12 m 3																								
Chamber of Commerce	: North-14 53 21		: 61-757 Poznań																								
VAT number	: PL12653007																										
Phone	: . . .	e-Mail	: info at boede.nl																								