LaTeX in proper ConTeXt

Berend de Boer

2000-feb-25

Contents

1 ConTEXt for LATEX users

1.1 For who is this document

If you are a LATEX user, switching to an entirely different macro–package is a very big step. Everything you put so much effort in to learn, doesn't work anymore.

I can't help but quote from that famous book "The psychology of computer programming" (chapter 11):

> "In making our adjustments to our particular programming language, we can easily become attached to it simply because we now have to much invested in it. We often listen to a man complaining about his nagging, slovenly, and prodigal wife, only to find that when asked why he doesn't leave her, he replies that he cannot live without her. Most people would prefer almost any amount of pain to giving up the familiarity of some constant companion for an unknown quantity. We see this effect when we try to teach a programmer his *second* language. Teaching the first is no great problem, for he has no investment in any other. By the time he has learned two or more, he is aware that more things exist in this world than he has dreamed of. But letting go of the first is, to him, just a promise of pain with no promise of compensating pleasure."

To help lessen the pain for users make the switch, this document shows short LATEX code snippets and how you do the same in ConTEXt.

ConTEXt is a macro package that's far more advanced than LATEX. You can enhance LATEX with third party packages, but not all macro packages work together with each other. ConTEXt is an integrated, powerful and flexible macro package for which you seldom need third party packages. ConTEXt also has been used to create large and complex on–screen documents, including hyperlinks, on screen buttons, forms, cross–document links, and so on.

This document is not a reference to the ConTEXt manual. It only shows you the ConTEXt macro's for the familiar LATEX macro's. It does not explain the ConTEXt macro's in detail nor shows you the numerous options almost every command has. You are referred to the `http://www.pragma-ade.nl/general/manuals/beta/cont-nli.pdf` instead.

1.2 References

If you take up ConTEXt, you probably need help. You can find it at the following locations:

- The ConTEXt support site is http://www.pragma-ade.nl.
- There is a beginner's manual in english at http://www.pragma-ade.nl/zipped/bman-en.zip.
- ConTEXt main reference however is still in dutch and can be found at http://www.pragma-ade.nl/general/manuals/beta/cont-nli.pdf.
- The 4TEX manual also contains much material on ConTEXt, see http://4tex.ntg.nl/4tex5.
- There is a ConTEXt mailling list. Subscribe by sending a message to mailto:majordomo@ntg.nl. In the body of the message type:

  ```
  subscribe ntg-context yourname@yourserver.yourdomain
  ```

1.3 Acknowledgements

Thanks to **David Arnold**, **Wybo Dekker** and **Hans Hagen** for offering suggestions and sending corrections to this document.

2 ConTEXt basics

This chapter tells you ConTEXt's basics.

2.1 A basic document

A basic 'hello world' kind of document in LaTeX looks like:

```
\documentclass{article}

\begin{document}
Hello world.
\end{document}
```

In ConTEXt it looks like:

```
\starttext
Hello world.
\stoptext
```

ConTEXt does not have a special command to load the style for a certain document. If you have them, you can load style files and such with the TEX macro `\input`.

2.2 Compiling your document

If your LaTeX 'Hello world' document is called `text.tex`, you probably compile your document with:

```
latex test.tex
```

In ConTEXt you use a wrapper around the TEX compiler to compile your document. Your ConTEXt document is compiled with:

```
texexec test.tex
```

`texexec` also takes care of the table of contents, indexes, references and sorted lists (see also **section 11.1**. It recompiles your document as many times as necessary to make sure references in the document are ok.

While you could use something as:

```
context test.tex
```

to compile your document, this is not advised. Learn to use `texexec`:

```
context test.tex
```

The first output of `texexec` should look like:

```
TeXExec 1.9 - ConTeXt / PRAGMA ADE 1997-2000

          executable : pdfetex
              format : cont-en
           inputfile : testc
              output : dvips
           interface : en
        current mode : all
            TeX run : 1
```

and then it starts compiling your document.

ConTEXt supports various language specific interfaces. This document presents the english language interface but ConTEXt also has a dutch and german based interface. If the default interface on your system is not correct, you can either change the initialization file `texexec.ini` or give a parameter to `texexec`:

```
texexec -interface=en test.tex
```

ConTEXt supports also various output formats like `.dvi` and `.pdf`. To generate a `.pdf` document instead of a `.dvi` document, change `texexec.ini` or call `texexec` with the following parameter:

```
texexec -output=pdftex test.tex
```

Or even shorter with:

```
texexec --pdf test.tex
```

Probably the best option is to put the `texexec` parameters as the first line in your document like:

```
% interface=en output=pdftex

\starttext
This example compiles with the english interface and
generates pdf output.
\stoptext
```

2.3 Chapters and sections

Chapters and sections in ConTEXt are very much like LATEX. In LATEX you write:

```
\documentclass{report}

\begin{document}

\chapter{One}
My first chapter.

\chapter{Two}
My second chapter.

\end{document}
```

In ConTEXt this looks like:

```
\starttext

\chapter{One}
My first chapter.

\chapter{Two}
My second chapter.

\stoptext
```

Not much difference here. However, referencing to chapters or sections is done differently in ConTEXt, see **section 8.1**.

2.4 Table of contents

A table of contents in LATEX is done with the `\tableofcontents` command:

```
\documentclass{report}

\begin{document}

\tableofcontents

\chapter{One}
My first chapter.

\chapter{TwoOne}
My second chapter.

\end{document}
```

In ConTEXt you get a table of contents with the `\completecontent` command:

```
\starttext

\completecontent

\chapter{One}
My first chapter.

\chapter{Two}
My second chapter.

\stoptext
```

Remember to compile with `texexec`, or else you will not get the table of contents (see **section 11.1**). Besides `\completecontent`, ConTEXt also has `\placecontent`. `\completecontent` starts a new page, while `\placecontent` doesn't.

3 Common LaTeX Environments

In the following sections the most common LaTeX environments are discussed and ConTeXt alternatives are given. Environments are discussed in alphabetic order.

3.1 The abstract environment

With LaTeX's abstract environment you can mark one or more paragraphs in some special way. It also depends on the documentclass you have loaded, either article or report. Its layout is equal to the quotation environment.

You use abstract as follows:

```
\documentclass{article}
\begin{document}

\begin{abstract}
This is an abstract of this article.
\end{abstract}

\end{document}
```

There is no abstract environment in ConTeXt, because it really is a style option. However, you can get the same effect as LaTeX's article abstract environment with:

```
\starttext

\startnarrower\switchtobodyfont[small]
\midaligned{\bf Abstract}\par
This is an abstract of this article.
\stopnarrower

\stoptext
```

With `\startnarrower` you get a paragraph, left and right indented by some white space. With `\switchtobodyfont` you get a font somewhat smaller than the current body font. With `\midaligned` you get a centered line.

3.2 The bibliography environment

In LaTeX you can create a bibliography within the `bibliography` environment environment:

```
\documentclass{article}

\begin{document}

\section{My life}

I've read only two books in my life \cite{Bekke92}, \cite{Brodie84}.

\begin{thebibliography}{Brodie84}

\bibitem[Bekke92]{Bekke92}
  J.H.~ter~Bekke, \emph{Semantic datamodeling}, Prentice Hall, Hemel
  Hempstead, ISBN~0-13-806050-9, 1992.

\bibitem[Brodie84]{Brodie84}
  L.~Brodie, \emph{Thinking Forth, a language and philosophy for solving
  problems}, Prentice Hall, ISBN 0-13-917568-7, 1984.

\end{thebibliography}

\end{document}
```

Entries are started with `\bibitem` and appear in the `bibliography` environment environment. Entries are referred to with the `\cite` command.

In ConTEXt, bibliographies are supported by the `bib` module. This module is not yet in the standard release, you have to download it separately at `http://www.cybercomm.nl/~bittext/temp/m-bib.zip`. See **section 11.2** on how to generate the documentation for this module. It seems the documentation is a bit ahead of the implementation, but expect the module to improve over the course of the next few months.

In ConTEXt, references have to be in a separate file. For this example it is easiest to have them in a file with the same name, but with the `.bbl` extension.

An example of such a file is:

```
\startpublication[k=Brodie84,
                  t=article,
                  a=L.~Brodie,
                  y=1984,
                  s=LB84]
\artauthor[]{Leo}[L.]{}{Brodie}
\arttitle{Thinking Forth, a language and philosophy for solving problems}
\journal{Prentice Hall}
\pubyear{1984}
\stoppublication

\startpublication[k=Bekke92,
                  t=article,
                  a=J.H.~ter~Bekke,
```

```
                y=1992,
                s=JB92]
\artauthor[]{Johan}[J.H.]{ter}{Bekke}
\arttitle{Semantic datamodeling}
\journal{Prentice Hall}
\pubyear{1992}
\stoppublication
```

Entries are started with the `\startpublication` command.

Having this file, it is now possible to use them as follows:

```
\usemodule[bib]

\setuppublications
  [numbering=yes,
  sort=author]

\starttext

\section{My life}

I've read only two books in my life \cite[Bekke92,Brodie84]. But I've
admit that I no longer think Forth \cite[Brodie84].

\completepublications

\stoptext
```

The command `\setuppublications` is optional. It defines such things as which entries to include, how to sort them or to include every entry or only the referenced ones. It currently does not seem possible to mimick the way LaTeX references things.

With `\completepublications` the list of publications is given. And of course there is also a `\placepublications` command which does not add something to the table of contents.

For bibTeX information, see **section 8.4**.

3.3 The description environment

LaTeX's `description` environment produces a list where each label is a keyword instead of a bullet or a number.

```
\documentclass{article}
\begin{document}
```

```
\begin{description}
\item[gnat] A small animal, found in the North Woods, that causes no
  end of trouble.
\item[gnu] A large animal, found in crossword puzzles, that causes no
  end of trouble.
\item[armadillo] A medium-sized animal.
\end{description}

\end{document}
```

In ConTEXt you use the `\definedescription` command to setup an environment. A description is not an enumeration like it is in LATEX. In the following code we first define the environment, called `animal`. It's also possible to define the environment `description`, but that's quite confusing. In the main text we show how this new environment is used:

```
\definedescription
  [animal]
  [location=hanging,
   margin=standard,
   headstyle=bold]

\definestartstop
  [animals]
  [before=\blank\startpacked,
   after=\stoppacked\blank]

\starttext

\startanimals
\animal{gnat} A small animal, found in the North Woods, that
  causes no end of trouble.

\animal{gnu} A large animal, found in crossword puzzles, that
  causes no end of trouble.

\startanimal{armadillo}
A medium-sized animal.
\stopanimal

\stopanimals

\stoptext
```

Within `\startanimals`, descriptions (animals) can be given. You can use both `\animal` or `\startanimal` to define a description. The start/stop pair is of course more robust. The type `\animal` expects a `\par` (or empty line) to work.

The use of `\definestartstop` is optional, but this helps to clearly mark the definition of animals and to have a common point of settings like white space before and after. With `\blank` we get the default blank space before (the space between paragraphs). With `\startpacked` we get paragraphs that do not have white space between them.

A different solution, however without the hanging indent feature, is:

```
\starttext

\startitemize[4*broad,packed]
\sym{gnat} A small animal, found in the North Woods, that causes no end
of trouble.
\sym{gnu} A large animal, found in crossword puzzles, that cuases no end
 of trouble.
\sym{armadillo} A medium-sized animal.
\stopitemize
\stoptext
```

In this example, the text `armadillo` overlaps with its definition if the margin is not defined sufficiently large. So the first solution is better.

This example also shows ConTEXt's way of setting options. With `4*broad` we set the width of the symbol to 4 times the `broad` setting. With `packed` we specify that we don't want white space between paragraphs.

3.4 The enumerate environment

LATEX's `enumerate` environment is used to produce a numbered list:

```
\documentclass{article}
\begin{document}

\begin{enumerate}
\item First item.
\item Second item.
\end{enumerate}

\end{document}
```

In ConTEXt lists, both numbered and unnumbered, are started with `\startitemize`. You use the `n` option to produce a numbered list:

```
\starttext

\startitemize[n]
\item First item.
```

```
\item Second item.
\stopitemize

\stoptext
```

See also the example in **section 3.5**. See **section 7.9** for how to influence the layout of lists.

3.5 The itemize environment

LaTeX's `itemize` environment is used to produce an unnumbered list:

```
\documentclass{article}
\begin{document}

\begin{itemize}
\item First item.
\item Second item.
\end{itemize}

\end{document}
```

Every line starts with a bullet.

In ConTeXt lists, both numbered and unnumbered, are started with `\startitemize`. If you don't use an option, you get an unnumbered list:

```
\starttext

\startitemize
\item First item.
\item Second item.
\stopitemize

\stoptext
```

See also the example in **section 3.4**. See **section 7.9** for how to influence the layout of lists.

3.6 The quotation environment

You can quote someone in LaTeX with the `quote` environment:

```
\documentclass{article}
\begin{document}
```

```
\begin{quote}
  But letting go of the first is, to him, just a promise
  of pain with no promise of compensating pleasure.
\end{quote}

\end{document}
```

In ConTEXt you achieve the same effect with the `\startquotation` command:

```
\starttext

\startquotation
  But letting go of the first is, to him, just a promise
  of pain with no promise of compensating pleasure.
\stopquotation

\stoptext
```

LATEX also has a `quotation` environment used for quotations of more than one paragraph. In ConTEXt you can always use `\startquotation`.

ConTEXt also has ‘inline’ quotes. Use either its `\quote` or `\quotation` command:

```
\starttext

This is a \quote{quote} and this is a \quotation{quotation}.

\stoptext
```

`\quote` surrounds your quote with single quote characters, `\quotation` surrounds your quote with double quote characters.

3.7 The verbatim environment

Verbatim text (text not subject to macro expansion) in LATEX is done with the verbatim environment:

```
\documentclass{article}
\begin{document}

\begin{verbatim}
This is
verbatim \LaTeX.
\end{verbatim}

\end{document}
```

In ConTEXt you write this as:

```
\starttext

\starttyping
This is
verbatim \ConTeXt.
\stoptyping

\stoptext
```

4 Floats

Floats are pieces of text that do not follow the main flow, but can go on the same page or elsewhere. ConTEXt has very extensive support for them.

4.1 Figures

In LATEX you can include a bitmap if you use the `graphics` or `graphicx` package. With the `\includegraphics` command you can include a bitmap.

If you use pdftex, you can now define a figure as simply as:

```
\documentclass{article}

\usepackage[pdftex]{graphicx}

\begin{document}

\begin{figure}
\includegraphics{test.png}
\caption{Test picture}
\label{fig:test}
\end{figure}

\end{document}
```

This figure also has a caption and it has a label so you can refer to it. In ConTEXt you do this with:

```
\starttext

\placefigure
  []
  [fig:test]
  {Test picture}
  {\externalfigure[test.png]}

\stoptext
```

Probably the size of the figure differs between LATEX and ConTEXt. In ConTEXt figures have their natural size by default. You can influence the scaling of a figure with the scale option:

```
\starttext
```

```
\placefigure
  []
  [fig:test]
  {Test picture}
  {\externalfigure[test.png][scale=2000]}

\stoptext
```

The default scale is 1000 (100%). Scale 2000 gives you a figure twice as large. You can also specify the width of the figure in dimensions, for example half the text width (see example below).

In ConTEXt you usually define your figures at the top of your file, above the `\starttext` command. You can recall them when needed:

```
\useexternalfigure
  [testone]
  [test.png]
  [scale=2000]

\useexternalfigure
  [testtwo]
  [test.png]
  [width=.5\textwidth]

\starttext

\placefigure
  []
  [fig:testone]
  {Test picture: twice as large as it natural size}
  {\externalfigure[testone]}

\placefigure
  []
  [fig:testtwo]
  {Test picture: half as large as the text}
  {\externalfigure[testtwo]}

\stoptext
```

With `\useexternalfigure` the first command is the name of the macro you want to define, the next is the name of the file. The third parameter are scaling and sizing options.

In http://www.pragma-ade.nl/zipped/bman-en.zip you can find more things you can do with figures like placing two figures together or placing text left or right of a figure.

4.2 List of figures

A list of figures in LaTeX can be given with its `\listoffigures` command:

```
\documentclass{article}

\usepackage[pdftex]{graphicx}

\begin{document}

\listoffigures

\begin{figure}
\includegraphics{test.png}
\caption{Test picture}
\label{figure:test}
\end{figure}

\end{document}
```

In ConTeXt you use the `\completelistoffigures` command:

```
\starttext

\completelistoffigures

\placefigure
  [fig:test]
  {Test picture}
  {\externalfigure[test.png]}

\stoptext
```

Just like `\completecontent`, besides `\completelistoffigures` their also is a `\placelistoffigures` command which doesn't start a new page.

Be aware that `\completelistoffigures` only works at the beginning of a document, not at the end. There probably is a workaround, and as soon as I know it, I'll say so here.

5 Tables

5.1 The tabular environment

In LaTeX tables are defined with the `tabular` environment environment. A famous LaTeX example demonstrating many of its features is:

```
\documentclass{article}

\begin{document}

\begin{tabular}{|r||r@{--}l|p{1.25in}|}
\hline
\multicolumn{4}{|c|}{GG\&A Hoofed Stock}
  \\ \hline\hline
&\multicolumn{2}{c|}{Price}& \\ \cline{2-3}
\multicolumn{1}{|c||}{Year}
& \multicolumn{1}{r@{\,\vline\,}}{low}
& high & \multicolumn{1}{c|}{Comments}
  \\ \hline
1971 & 97 & 245 & Bad year for
          farmers in the west.   \\ \hline
  72 & 245 & 245 & Light trading due to a
                 heavy winter. \\ \hline
  73 & 245 & 2001 & No gnus was very
         good gnus this year.    \\ \hline
\end{tabular}

\end{document}
```

In ConTeXt tables are created within the `\starttable` environment:

```
\starttext

\starttable[|r|r|l|p(1.25in)|]
\HL
\VL \FOUR{GG\&A Hoofed Stock} \VL\SR
\HL
\VL \LOW{Year} \VL \TWO{Price}  \VL \LOW{Comments} \VL\SR
\DC          \DL[2]          \DC              \DR
\VL          \VL low \VL high \VL              \VL\SR
\HL
\VL 1971 \VL \TWO{97--245} \VL
  Bad year for farmers in the west. \VL\SR
\HL
\VL 72 \VL \TWO{245--245} \VL
  Light trading due to a heavy winter. \VL\SR
\HL
```

```
\VL 73 \VL \TWO{245--2001} \VL
  No gnus was very good gnus this year. \VL\SR
\HL
\stoptable

\stoptext
```

In ConTEXt you can use `\starttables` to create a table that can be split accross pages.. In LATEX this is provided by the `longtable` environment, provided by an external package.

5.2 The tabbing environment

To align data vertically, one can use the `tabbing` environment in LATEX. It's usage is quite complex, only a simple example is given:

```
\documentclass{article}

\begin{document}
\begin{tabbing}
  Armadillo: \= \kill
  Gnat: \> not edible \\
  Armadillo: \> not edible\\
\end{tabbing}
\end{document}
```

The same example in ConTEXt using its `\starttabulate` environment:

```
\starttext
\starttabulate[|l|p|]
\NC Gnat: \NC not edible \NC\NR
\NC Armadillo: \NC not edible \NC\NR
\stoptabulate
\stoptext
```

In LATEX the first line more or less determines the format (but can be changed in latter lines). In ConTEXt the `\starttabulate` environment works more or less like its `\starttable` environment. The header specifies the format of the columns. The width of the columns will be equal to the row with the largest contents. You can also explicitly specify the width.

Every row starts with `\NC` (next column). Every row ends with a `\NC` and `\NR` (next row).

The difference between ConTEXt's `\starttabulate` and `\starttable` environments are that the former splits across pages. And the `p` column type is a bit smarter in

the `\starttabulate` environment. The drawback of `\starttabulate` is, that it does not support vertical lines.

5.3 List of tables

In LaTeX one gets the list of tables with the `\listoftables` command. Only tables placed within the `table` environment environment are put in the table of contents.

In ConTeXt one uses the `\placelistofoftables` command. And as in LaTeX, only tables given as argument to `\placetable` are pu in the table of contents.

6 Math

6.1 General

Math in ConTEXt and LATEX is not that different. Both depend mostly on TEX. LATEX adds some environments for doing math. ConTEXt is mostly very close to TEX. In ConTEXt probably every TEX math command just works.

6.2 In-text formulas

In ConTEXt in-text formula's are produced by surrounding the formula by the $ character.

6.3 Display style formulas

In ConTEXt display style formula's are produced by surrounding the formula by two $ characters. You can also use the `\startformula` environment for exactly the same effect.

6.4 Numbered formulas

With LATEX numbered formula's are made within the `equation` environment:

```
\documentclass{article}

\begin{document}
\begin{equation}
E = mc^2
\end{equation}
\end{document}
```

In ConTEXt numbered formulas are produced to prefix the display formula command with `\placeformula`.

```
\starttext
\placeformula $$ E = mc^2 $$
\stoptext
```

6.5 Multiline formulas

In LATEX multiline formula's are produced with the `eqnarray` environment. A new line is started after the `\\` command.

In ConTEXt you use TEX's displaylines command. A new line is started after the `\cr` command.

```
\starttext

$$\displaylines{ZeroOrOne = ( ( Amount\ mod\ Currency_RoundingFactor )\cr
  +\ Currency_Boundary )\cr
  div\ Currency_RoundingFactor}$$

\stoptext
```

6.6 Theorems and such

LATEX has the `\newtheorem` command to define environments for theorems-like structures. You would use it as follows:

```
\documentclass{article}

\newtheorem{guess}{Conjecture}

\begin{document}
This is the first one:
\begin{guess}
  All conjectures are interesting.
\end{guess}
\end{document}
```

With ConTEXt you would use the `\defineenumeration` command. This command has many options, the following settings make it more or less equal to LATEX's `\newtheorem`:

```
\defineenumeration
  [guess]
  [text=Conjecture,
  location=left,
  letter=it]

\starttext
This is the first one:
\guess All conjectures are interesting.

This is the second one:
\startguess Except this one. \stopguess
\stoptext
```

As can be seen both `\guess` and `\startguess` can be used. Also `\subguess`, `\subsubguess` and so on are now available.

ConTEXt's `\defineenumeration` can be used for any kind of thing you want to enumerate.

7 Changing the layout

As soon as you are able to write basic documents, you probably want to change their appearance. This chapter documents the differences between LaTeX and ConTeXt. As ConTeXt is a very, very flexible macro package, this is one of the largest chapters. Almost everything can be customized, you just call `\setup`*`something`*.

7.1 Page size

In LaTeX you give the page size as an option to the `\documentclass` command:

```
\documentclass[a4paper]{article}

\begin{document}
\end{document}
```

In ConTeXt you use the `\setuppapersize` command.

```
\setuppapersize[A4]

\starttext
\stoptext
```

With ConTeXt it is possible to do far more advanced things. You can typeset in A5 and print on A4 for example.

7.2 Fonts

Changing the font size in LaTeX is usually done with a document class option:

```
\documentclass[12pt]{report}

\begin{document}
\end{document}
```

In ConTeXt you use the `\setupbodyfont` command:

```
\setupbodyfont[12pt]

\starttext
\stoptext
```

Use `\setupbodyfont` only to set the document font. To switch to another font during typesetting, you should use `\switchtobodyfont`.

To switch to a postscript font in LaTeX, you can use certain packages. To switch to a Helvetica body font you would type:

```
\documentclass{article}
\usepackage{helvet}

\begin{document}
\end{document}
```

In ConTeXt you accomplish the same with the `\setupbodyfont` command:

```
\setupbodyfont[ber,phv,ss]

\starttext
\stoptext
```

With the option `ber` you specify that you want to use Karl Berry fontnames. With `phv` you specify that you want to load the Helvetica font definitions, and with `ss` you specify that you want to use a sans-serif font as the body font.

To switch to the default postscript fonts times, helvetica and courier, you would say:

```
\setupbodyfont[ber,pos]

\starttext
\stoptext
```

7.3 Interline spacing

To change the interline spacing in LaTeX you change the `baselineskip` variable:

```
\documentclass{article}

\setlength\baselineskip{12pt}

\begin{document}
\end{document}
```

This changes the distance between lines to 12pt, a good value for a 10pt font. As in LaTeX, if you change the size of the body font in ConTeXt, the line skip is automatically recalculated. You can set it yourself with `\setupinterlinespace`:

```
\setupinterlinespace[line=1.2\bodyfontsize]

\starttext
\stoptext
```

Be careful to always define the interlineskip in terms of the current `\bodyfontsize`, else you get unexpected results when the `\bodyfontsize` changes, for example in chapter headings.

7.4 Spacing between paragraphs

To change the interparagraph spacing in LaTeX, you change the `\parskip` variable:

```
\documentclass{article}

\setlength{\parskip}{3pt}

\begin{document}
This is my first paragraph.

This is my second paragraph.
\end{document}
```

This gives you 3 extra points of white space between paragraphs. In ConTeXt you use the `\setupwhitespace` command.

```
\setupwhitespace[3pt]

\starttext
This is my first paragraph.

This is my second paragraph.
\stoptext
```

However, instead of a fixed size specification, it is much better to use current font size related specifications like `medium` or `big`:

```
\setupwhitespace[medium]

\starttext
This is my first paragraph.

This is my second paragraph.
\stoptext
```

7.5 Paragraph indentation

In LaTeX every paragraph has some white space at the beginning of its first line. You can disable this by setting the amount of white space to 0:

```
\documentclass{article}

\setlength{\parindent}{0pt}

\begin{document}
This is my first paragraph. This is my first paragraph. This is my
first paragraph. This is my first paragraph. This is my first
paragraph. This is my first paragraph.

This is my second paragraph.
\end{document}
```

ConTeXt doesn't start with white space. To start a paragraph with some white space say:

```
\setupindenting[medium]

\starttext
This is my first paragraph. This is my first paragraph. This is my
first paragraph. This is my first paragraph. This is my first
paragraph. This is my first paragraph.

This is my second paragraph.
\stoptext
```

7.6 Position of the page number

In LaTeX the position of the page number can be set with the `\pagestyle` command. In ConTeXt you use the `\setuppagenumbering` command.

There are four LaTeX page styles. The corresponding ConTeXt settings are given below:

1. `\pagestyle{plain}`:

```
    \setuppagenumbering[location={footer,middle}]
```

2. `\pagestyle{empty}`:

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

3. `\pagestyle{headings}`:

```
    \setuppagenumbering[location={header,middle}]
```

4. `\pagestyle{myheadings}`:

```
    \setuppagenumbering[location=header]
```

(see ....)

Note that in ConTEXt a page consists of much more than LATEX's three default units *head*, *body* and *foot*, so you have much more options.

7.7 Roman and arabic pagenumbers

In LATEX Roman and Arabic page numbers are specified with the `\pagenumbering` command.

```
\documentclass{article}

\begin{document}

\pagenumbering{roman}
\tableofcontents

\chapter{Introduction}
\pagenumbering{arabic}
This is a test.

\end{document}
```

In ConTEXt you use the `\setupnumbering` command.

```
\startfrontmatter

\setuppagenumbering[conversion=romannumerals]
\setuppagenumber[number=1]
\completecontent

\stopfrontmatter

\startbodymatter

\setuppagenumbering[conversion=numbers]
\setuppagenumber[number=1]
\chapter{Introduction}
This is a test.
```

```
\stopbodymatter
```

This example also shows a typical document structure by its use of `\startfontmatter` and `\startbodymatter`. The front matter has roman page numbers, the body matter arabic ones. We also use `\setuppagenumber` to start counting from 1 for each of the sections. The first `\setuppagenumber` could be omitted, but typically you have a front page and such, so page numbering starts a bit later than the first physical pages.

7.8 Setting up chapter and sections

In LaTeX changing the format of chapters and sections is not possible unless you change LaTeX internal commands. Therefore, no examples of doing this in LaTeX are given, but only the ConTeXt options are demonstrated.

Every section heading can be setup with ConTeXt's `\setuphead` command. Probably one of the first things you want to change is the font used for the chapters and sections.

```
\setuphead[chapter][letter={\switchtobodyfont[20pt,ss]\bf}]
\setuphead[section][letter={\switchtobodyfont[16pt,ss]\bf}]

\starttext

\chapter{This is a chapter}

First sentence.

\section{This is a section}

First sentence.

\stoptext
```

This gives you a 20 points, bold, sans serif font for chapter headings.

It is possible to define that you want to start a chapter on a right page:

```
\setuppagenumbering[alternative=doublesided]
\setuphead[chapter]
  [page=right]

\starttext

\chapter{First thought}

First sentence.

\chapter{Second thought}
```

```
Second sentence.

\stoptext
```

You also need to turn doublesided pagenumbering on with `\setuppagenumbering`. Now every chapter starts on an uneven numbered page.

The following example changes even more. The number is set in a different font, and before and after the chapter is set, we execute our own commands.

```
\setuphead
  [chapter]
  [numberstyle=bold,
  textstyle=cap,
  before=\hairline\blank,
  after={\nowhitespace\hairline\blank[line]}]

\starttext

\chapter{This is a chapter}

A sentence.

\stoptext
```

Note that almost any(!) setup command has before and after options, so really everything in ConTEXt can be changed easily.

It is also possible to format the entire section heading yourself. The following example formats the subsection heading. You need to write a macro which expects two parameters: the number of that section and the title of that section. And then you're on your own.

```
\setuphead
  [subsection]
  [command=\myhead]
\def\myhead#1#2{#2}

\starttext

\chapter{This is a chapter}

\section{Section}

\subsection{Yes}

A sentence.

\stoptext
```

Note that turning off the number is a standard option, so this setup can also be accomplished with:

```
\setuphead
  [subsection]
  [number=no]

\starttext

\chapter{This is a chapter}

\section{Section}

\subsection{Yes}

A sentence.

\stoptext
```

7.9 Setting up lists

Enumerations in ConTEXt have white space between each items. You can disable this by using the packed option.

```
\setupitemize[packed]

\starttext
\startitemize[n]
\item First line.
\item Second line, no white space above it.
\stopitemize
\stoptext
```

8 References

8.1 In document references

8.2 Cross document references

8.3 Index

In LaTeX producing indexes is a two step process. In the preamble you put the `\makeindex` command so an index file (`.idx`) is created. The contents of every `\index` command is put into that file. Next the `makeindex` command is used to produce the actual index which can be included in your document with the `\input` command.

```
\documentclass{article}

\makeindex

\begin{document}

Put \index{this} and \index{that} in the index.

\input{testlatex.ind}

\end{document}
```

In ConTeXt creating indexes is also a two step process, but this is transparant if you use `texexec` to compile your documents. With the `\index` command you put entries in the index, with `\completeindex` you get the entire index at that point.

```
\starttext

Put \index{this} and \index{that} in the index.

\completeindex
\stoptext
```

8.4 bibtex

You can use `bibtex` to produce `.bbl` files as usual. The new `bib` module (see **section 3.2**) can read `.bbl` files fine. You can specify on or more databases in `\setuppublications` after the database keyword.

```
\usemodule[bib]

\setuppublications
  [database={mybibs},
  numbering=yes,
  sort=author]

\starttext

\section{My life}

I’ve read only two books in my life \cite[laan95:_types_pascal]. But I’ve
admit that I no longer think Forth \cite[wogan:73a].

\completepublications

\stoptext
```

However, bib support is not yet foolproof.

9 Interactive documents

9.1 Defining an interactive document

```
\usepackage{article}
\usepackage
  [pdftitle={Test document},
   pdfauthor={Berend de Boer},
   pdfsubject={There is more than LaTeX},
   pdfkeywords={LaTeX ConTeXt TeX},
   colorlinks,
   linkcolor=blue]
  {hyperref}

\begin{document}
\end{document}
```

```
\setupinteraction
  [status=start,
   title={Test document},
   author={Berend de Boer},
   subtitle={There is more than LaTeX},
   keywords={LaTeX ConTeXt TeX},
   color=blue]

\starttext
\stoptext
```

9.2 References which are URLs

In LaTeX the hyperref package provides hyperlinking support. To generate a URL reference, you use the `\href` command:

```
\documentclass{article}

\usepackage{hyperref}

\begin{document}

The \href{http://www.freebsd.org}{greatest server OS} on this world.

\end{document}
```

```
\setupinteraction[state=start]

\useURL
  [contextsupport]
  [http://www.pragma-ade.nl]

\starttext

The \ConTeXt\ support site is \from[contextsupport].

\stoptext
```

10 Other

10.1 Title page

10.2 Text in margin

10.3 Color

You can enable color with the `\setupcolors` command.

```
\setupcolors[state=start]

\starttext
\stoptext
```

10.4 Babel

10.5 Interfaces

10.6 Breaking lines

10.7 Vertical white space

If you need vertical white space, LaTeX has the `\vspace` command. This command only works between paragraphs. At the beginning and end of a page white space ‘disappears’, use `\vspace*` for white space that does not disappear.

```
\documentclass{article}

\begin{document}
\vspace*{5cm}
This line comes first.

\vspace{2\baselineskip}

This line comes second.

\vspace{.5\baselineskip}
```

```
And this is the third line.

\end{document}
```

In ConTEXt the `\blank` command gives blank space. It also ends the preceding paragraph. Use the `force` option to force white space at the beginning and end of a page.

```
\starttext
\blank[5cm,force]
This line comes first.
\blank[2*line]
This line comes second.
\blank[medium]
And this is the third line.
\stoptext
```

11 Understanding ConTEXt

11.1 texexec

ConTEXt uses the external program `texexec` to do lots of things which are more difficult to do directly in TEX:

1. ConTEXt writes its table of contents entries to a `.tui` file. If the compilation went fine, `texexec` copies this to the corresponding `.tuo` file.

   This means that when a compilation does not run to completion (you cancel it for example), the table of contents entries and references have not disappeared.

2. `texexec` sorts indexes and sorted lists.

11.2 Module documentation

Generate ConTEXt module documentation with:

```
texexec --module m-bib
```

You need to have the documentation styles which can be downloaded from `http://www.pragma-ade.nl`, filename `cont-doc.zip`.

LaTeX commands

List of LaTeX commands used in this document.

ConTEXt commands

List of ConTEXt commands used in this document.

LaTeX to ConTeXt reference

Standard environments

A list of all LaTeX environments and the corresponding ConTeXt environment is provided below. Note that LaTeX environments are written as:

```
\begin{environment}
\end{environment}
```

In ConTeXt you write this as:

```
\startenvironment
\stopenvironment
```

LaTeX environment	ConTeXt environment	Remark
abstract	no equivalent.	
array	no equivalent.	
center	alignment	
description		See **section 3.3**
displaymath		
document	text	
enumerate	itemize	
equation	placeformula	
eqnarray	displaylines	Use the standard TeX commands.
figure		Use placefigure.
flushleft	alignment	
flushright	alignment	
itemize	itemize	
letter	no equivalent.	
list	itemize	
math		Use $ before and $ after the formula.
minipage	no equivalent.	
note	no equivalent.	
overlay	no equivalent.	
picture	MPgraphic	In ConTeXt you can include metapost code to achieve the same and better effects.
quote	citaat	
quotation	citaat	
slide	no equivalent.	
sloppypar		
tabbing	tabulate	
table	table	Use together with placetable
tabular	table	
titlepage	no equivalent.	
thebibliography	publication	Requires the `bib` module.
theindex		Use completeindex.
trivlist	itemize	

verbatim	typing
verse	

Standard commands

Many LaTeX commands come from third party packages. Most of the things they provide (or fix) are standard features in ConTeXt.

LaTeX command	ConTeXt command	Remark
setlanguage		Provided by the babel package.

Environments provided packages

Many LaTeX environments come from third party packages. Most of the things they provide (or fix) are standard features in ConTeXt.

LaTeX package	ConTeXt environment	Remark
comment	comment	Comment out certain parts of your document so it does not show up in the .dvi or .pdf file. LaTeX's `comment` environment is part of Rainer Schöpf's verbatim package.
longtable	tables	The longtable package provides tables that can be broken accross page.