

ConTeXt an excursion

English
Version

Ton Otten & Hans Hagen
PRAGMA ADE

Version 990527 of this document is produced using CON_TE_XT, T_EXEDIT, T_EXUTIL and Web2C PDF_TE_X. This document is typeset in Lucida Bright Typefaces. For testing we used GHOSTSCRIPT, Acrobat Exchange, DVIVIEW and DVIWINDO

T_EX and $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX are trademarks of the American Mathematical Society; METAFONT is a trademark of Addison-Wesley Publishing Company; PostScript, Portable Document Format and Acrobat are trademarks of Adobe Systems Incorporated; DVIPSONE and DVIWINDO are trademarks of Y&Y Incorporated; IBM is a trademark of International Business Machines Corporation; MSDOS is a trademark of MicroSoft Corporation; all other product names are trademarks of their producers.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission of the publisher.

© 1991-1999 PRAGMA, Ridderstraat 27, 8061GH Hasselt, The Netherlands, pragma@pi.net

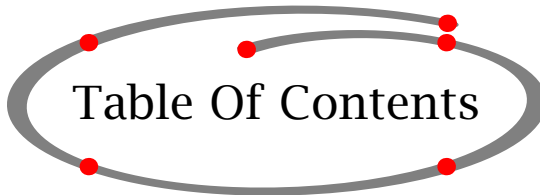
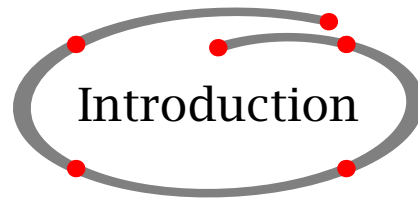


Table Of Contents

1	How to create a document	3	24	Synonyms	55
2	How to process a file / Run <code>CONTEXT</code>	5	25	Referring to text elements	56
3	Special characters	6	26	Color	58
4	Defining a document	7	27	Background text	59
5	Headers	8	28	Page backgrounds	60
6	Itemize	10	29	Alignment	61
7	Formulae	15	30	Interactive mode in electronic documents	62
8	Legends	17	31	Fonts and font switches	66
9	Units	18	32	Composite characters	70
10	Figures	19	33	Page layout	71
11	Tables	25	34	Paragraph spacing	75
12	Paragraph formatting	35	35	Setup commands	79
13	Columns	38	36	Defining commands / macros	80
14	Footnotes	40	37	Miscellaneous	82
15	Definitions	41	38	Using modules	92
16	Numbered definitions	43			
17	Outlined text	44	A	User specifications	93
18	Outlined paragraphs	45	B	Processing steps	94
19	Margin texts	46	C	Auxilliary files	95
20	Page breaking and page numbering	48	D	Command definitions	96
21	Page headers and footers	49	E	Command Index	122
22	Table of contents (lists)	51	F	Subject Index	125
23	Registers	54			



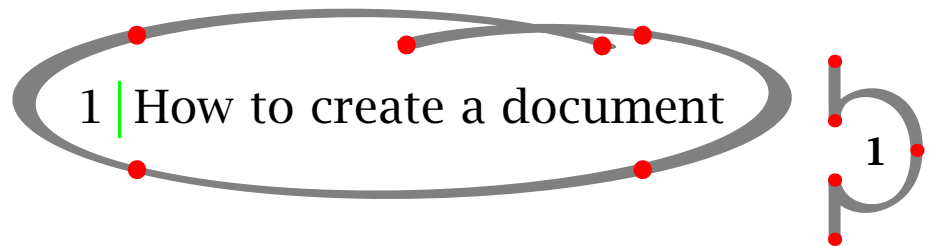
CON_TE_XT is a document engineering system based on T_EX. T_EX is a typesetting system and a program to typeset and produce documents. CON_TE_XT is easy to use and enables you to make complex paper and electronic documents.

This manual describes the capabilities of CON_TE_XT and the available commands and their functionality.¹

CON_TE_XT is developed for practical applications: the typesetting and production of documents ranging from simple straight forward books up to very complex and advanced technical manuals and studybooks in a paper or electronic version. This introductory manual describes the CON_TE_XT functionality necessary to apply standard text elements in a manual or studybook. CON_TE_XT, however, is capable of much more and for users who want more there are other CON_TE_XT manuals and sources available.

CON_TE_XT has a multilingual interface to enable users to work with CON_TE_XT in their own language. CON_TE_XT and this manual are available in Dutch, German and English.

¹ All paper and electronic products around CON_TE_XT are produced with CON_TE_XT. All sources of these products are or will be made available electronically to give you insight in the way these products are made up.



Let's assume you want to create a simple document. It has some structure and contains a title page, a few chapters, sections and sub sections. Of course there is a table of contents and an index.

CONTEXT can create such a document automatically if you offer the right input by means of a file. So first you have to create an input file. An input file consists of a name and an extension. You can choose any name you want but the extension has to be `tex`. If you create a file with the name `myfile.tex` you will find no difficulties in running CONTEXT.

An input file could look like this:

```
\starttext
\startstandardmakeup
  \midaligned{How to make a document.}
  \midaligned{by}
  \midaligned{The Author}
\stopstandardmakeup
\completecontent
\chapter{Introduction}
... your text\index{indexentry} ...
\chapter{One Chapter}
\section[firstsection]{The first section}
... your text ...
\section{The second section}
\subsection{the first sub section}
... your text\index{another indexentry} ...
\subsection{the second sub section}
... your text ...
\section{The third section}
... your text ...
\chapter{Another Chapter}
... your text ...
\chapter[lastchapter]{The Last Chapter}
```

... your text ...

\completeindex

\stoptext

CONTEX expects a plain ASCII input file. Of course you can use any texteditor or wordprocessor you want, but you should not forget that CONTEX can only read ASCII input. Most texteditors or wordprocessors can export your file as plain ASCII.

The input file should contain the text you want to be processed by CONTEX and the CONTEX commands. A CONTEX command begins with a backslash \. With the command \starttext you indicate the beginning of your text. The area before \starttext is called the set up area and is used for defining new commands and setting up the layout of your document.

A command is usually followed by a left and right bracket pair [] and/or a left and right brace {}. In \chapter[*lastchapter*]{The Last Chapter} the command \chapter for example tells CONTEX to perform a few actions concerning design, typography and structure. These actions might be:

1. start a new page
2. increase chapter number by one
3. place chapter number in front of chapter title
4. reserve some vertical space
5. use a big font
6. put chapter title (and page number) in table of contents

These actions will be performed on the argument that is given between the left and right braces: *The Last Chapter*.

The [*lastchapter*] between brackets has not been mentioned yet. This is a label with a logical name that can be used for referring to that specific chapter. This can be done with yet some other CONTEX commands: \in{chapter}[*lastchapter*] typesets the chapter number, while \about[*lastchapter*] returns the title.

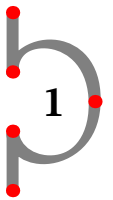
So now the list of actions can be extended with:

7. let label *lastchapter* be chapter number (and store this for later use)

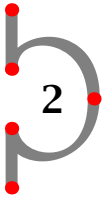
Other actions concerning running heads, number resetting and interactivity are disregarded at this moment.

If you have CONTEX process this file, you would obtain a very simple document with a few numbered chapter and section headers.

While processing the file CONTEX takes care of many things. One of these things is for example page numbering. But in order to make a table of contents CONTEX needs page numbers that are not yet known to CONTEX at the first run. So you have to process this file twice (a two pass job). CONTEX will produce a few auxilliary files to store this kind of



information. These are to be processed by $\text{T}_{\text{E}}\text{X}$ UTIL. In some instances you have to process an input file thrice (a three pass job). One can use $\text{T}_{\text{E}}\text{X}$ EXEC to setup CONTEXT from the command line. This PERL script also takes care of the multiple passes.



2 | How to process a file / Run CONTEXT

If you want to process a CONTEXT input file, you might type at the command line prompt:

```
context filename
```

The availability of the batch command `context` depends on the system you're using. Ask your system administrator the command you use to start CONTEXT. If your filename is `myfile.tex` this can be:

```
context myfile
```

the extension `.tex` is not needed.

After pressing Enter processing will be started. CONTEXT will show processing information on your screen. If processing is successful the command line prompt will return and CONTEXT will produce a `dvi` or `pdf` file.

If processing is not successful—for example because you typed `\stptext` instead of `\stoptext`—CONTEXT produces a `?` on your terminal and tells you it has just processed an error. It will give you some basic information on the type of error and the line number where the error becomes effective.

At the instant of `?` you can type:

H	for help information on your error
I	for inserting the correct CONTEXT command
Q	for quitting and entering batch mode
X	for exiting the running mode
Enter	for ignoring the error

Most of the time you will type Enter and processing will continue. Then you can edit the input file and fix the error.

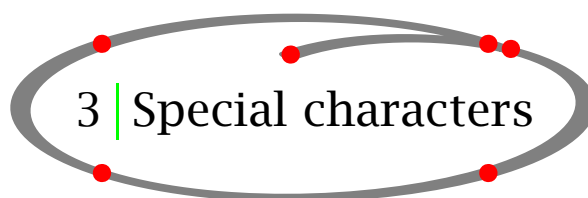
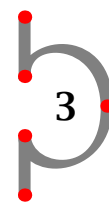
Some errors will produce a `*` on your screen and processing will stop. This error is due to a fatal error in your input file. You can't ignore this error and the only option you have is to type `\stop` or CTRL Z. The program will be halted and you can fix the error.



Special characters

During the processing of your input file `CONTEXT` will also inform you of what it is doing with your document. For example it will show page numbers and information about process steps. Further more it gives warnings. These are of a typographical order and tells you when line breaking is not successful. All information on processing is stored in a `log` file that can be used for reviewing warnings and errors and the respective line numbers where they occur in your file.

When processing is succesful `CONTEXT` produces a new file, with the extension `.dvi`. If you run `myfile.tex`, a file `myfile.dvi` is produced. The abbreviation `dvi` stands for Device Independent. This means that the file can be processed by a printer driver to make the file suitable for printing or viewing. The `dvi` file can be transformed to a `ps` file, `HP` file or whatever file you want.



You have seen that `CONTEXT` commands are preceded by a `\` (backslash). This means that `\` has a special meaning to `CONTEXT`. Aside from `\` there are other characters that need special attention when you want them to appear in verbatim mode or in text mode. Table 3.1 gives an overview of these special characters and what you have to type to produce them.

Special character	Verbatim		Text	
	Type	To produce	Type	To produce
#	<code>\type{#}</code>	#	<code>\#</code>	#
\$	<code>\type{\$}</code>	\$	<code>\\$</code>	\$
&	<code>\type{&}</code>	&	<code>\&</code>	&
%	<code>\type{%}</code>	%	<code>\%</code>	%

Table 3.1 Special characters (1).

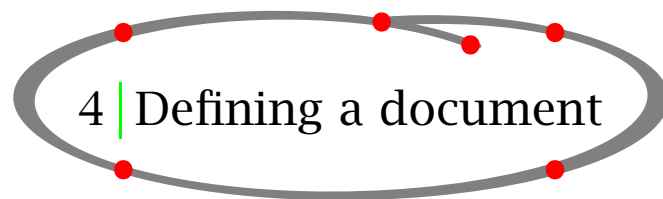
Other special characters have a meaning in typesetting mathematical expressions and some can be used in math mode only (see chapter 7).



Defining a document

Special character	Verbatim		Text	
	Type	To produce	Type	To produce
+	<code>\type{+}</code>	+	<code>\$+\$</code>	+
-	<code>\type{-}</code>	-	<code>\$-\$</code>	-
=	<code>\type{=}</code>	=	<code>\$=\$</code>	=
<	<code>\type{<}</code>	<	<code>\$<\$</code>	<
>	<code>\type{>}</code>	>	<code>\$>\$</code>	>

Table 3.2 Special characters (2).



Every document is started with `\starttext` and closed with `\stoptext`. All textual input is placed between these two commands and `CONTEXT` will only process that information.

Setup information is placed in the set up area just before `\starttext`.

```
\setupbodyfont[12pt]
\starttext
This is a one line document.
\stoptext
```

Within the `\starttext ... \stoptext` a document can be divided into four main divisions:

1. front matter
2. body matter
3. back matter
4. appendices

The divisions are defined with:

```
\startfrontmatter ... \stopfrontmatter
\startbodymatter ... \stopbodymatter
\startbackmatter ... \stopbackmatter
\startappendices ... \stopappendices
```

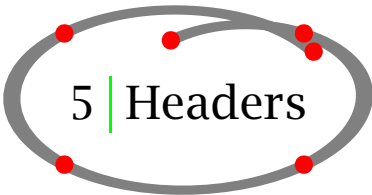
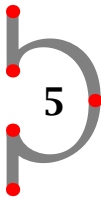
In the front matter as well as back matter section the command `\chapter` produces an un-numbered header in the table of contents. This section is mostly used for the table of contents, the list of figures and tables, the preface, the acknowledgements etc. This section often has a roman page numbering.



The appendices section is used for (indeed) appendices. Headers may be typeset in a different way; for example, `\chapter` may be numbered alphabetically.

Section style can be set up with:

`\setupsectionblock[...] [.,.,.=.,.,.]`



The structure of a document is determined by its headers. Headers (heads) are created with the commands shown in table 5.1:

Numbered header	Un-numbered header
<code>\chapter</code>	<code>\title</code>
<code>\section</code>	<code>\subject</code>
<code>\subsection</code>	<code>\subsubject</code>
<code>\subsubsection</code>	<code>\subsubsubject</code>
<code>...</code>	<code>...</code>

Table 5.1 Headers.

`\chapter[ref,...]{...}`

`\section[ref,...]{...}`

`\subsection[ref,...]{...}`

```
\title[ref,...]{...}
```

```
\subject[ref,...]{...}
```

```
\subsubject[ref,...]{...}
```

These commands will produce a header in a predefined fontsize and fonttype with some vertical spacing before and after the header.

The heading commands can take several arguments, like in:

```
\title[hasselt-by-night]{Hasselt by night}
```

and

```
\title{Hasselt by night}
```

The bracket pair is optional and used for internal references. If you want to refer to this header you type for example `\at{page}[hasselt-by-night]`.

Of course these headers can be set to your own preferences and you can even define your own headers. This is done by the command `\setuphead` and `\definehead`.

```
\definehead[.1.][.2.]
```

```
\setuphead[...][...,...=...,...]
```

```
\definehead
  [myheader]
  [section]
```

```
\setuphead
  [myheader]
  [numberstyle=bold,
   textstyle=cap,
   before=\hairline\blank,
   after=\nowhitespace\hairline]
```

```
\myheader[myhead]{Hasselt makes headlines}
```

A new header `\myheader` is defined and it inherits the properties of `\section`. It would look something like this:

5.1 HASSELT MAKES HEADLINES

There is one other command you should know now, and that is `\setupheads`. You can use this command to set up the numbering of the numbered headers. If you type:

```
\setupheads
  [alternative=inmargin,
   separator=--]
```

all numbers will appear in the margin. Section 1.1 would look like 1-1.

Commands like `\setupheads` are typed in the set up area of your input file.

`\setupheads[...]=...]`

6 | Itemize

One way of structuring your information is by way of enumeration or summing up items. The `itemize` command looks like this:

`\startitemize[...][...]=...]` ... `\stopitemize`

For example:

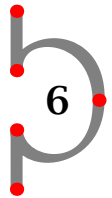
```
\startitemize[R,packed,broad]
\item Hasselt was founded in the 14th century.
\item Hasselt is known as a so called Hanze town.
\item Hasselt's name stems from a tree.
\stopitemize
```

Within the `\startitemize ... \stopitemize` pair you start a new item with `\item`. The space after `\item` is required. In the example above R specifies Roman numbering and

packed keeps line spacing to a minimum. The parameter broad takes care of vertical spacing after the item separator. The example would produce:

- I. Hasselt was founded in the 14th century.
- II. Hasselt is known as a so called Hanze town.
- III. Hasselt's name stems from a tree.

Itemization is a two pass job. This means that you have to run your file twice for an optimal layout. The bracket pair contains information on item separators and local set up variables.



Argument	Item separator
1	•
2	—
3	★
:	:
n	1 2 3 4 ...
a	a b c d ...
A	A B C D ...
r	i ii iii iv ...
R	I II III IV ...

Table 6.1 Item separators in itemize.

You can also define your own item separator by means of `\definesymbol`. For example if you try this:

```
\definesymbol[5][\clubsuit$]
\startitemize[5,packed]
\item Hasselt was built on a riverdune.
\item Hasselt lies at the crossing of two rivers.
\stopitemize
```

You will get:

- ♣ Hasselt was built on a riverdune.
- ♣ Hasselt lies at the crossing of two rivers.

Sometimes you would like to have headers within an enumeration. In that case you should use `\head` instead of `\item`.

```
Hasselt lies in the province of Overijssel and there are a number of
customs that are typical of this province.
\startitemize
```

`\head kraamschudden \hfill (child welcoming)`

When a child is born the neighbours come to visit the new parents. The women come to admire the baby and the men come to judge the baby (if it is a boy) if he will become a strong man. The neighbours will bring a *krentenwegge* along. A *krentenwegge* is a loaf of currant bread of about 1 \Meter\ long. Of course the birth is celebrated with *jenever*.

`\head nabuurschap (naberschap) \hfill (neighbourship)`

Smaller communities used to be very dependent on the cooperation among the members for their well being. Members of the *nabuurschap* helped each other in difficult times during harvest times, funerals or any hardship that fell upon the community.

`\head Abraham \& Sarah \hfill (identical)`

When people turn 50 in Hasselt it is said that they see Abraham or Sarah. The custom is to give these people a *speculaas* Abraham or a Sarah. *Speculaas* is a kind of hard spiced biscuit.

`\stopitemize`

The `\head` can be set up with `\setupitemize`. In case of a page breaking a `\head` will appear on a new page. (The `\Meter` command is explained chapter 9.)

The example of old customs will look like this:

Hasselt lies in the province of Overijssel and there are a number of customs that are typical of this province.

- kraamschudden (child welcoming)
When a child is born the neighbours come to visit the new parents. The women come to admire the baby and the men come to judge the baby (if it is a boy) if he will become a strong man. The neighbours will bring a *krentenwegge* along. A *krentenwegge* is a loaf of currant bread of about 1 m long. Of course the birth is celebrated with *jenever*.
- nabuurschap (naberschap) (neighbourship)
Smaller communities used to be very dependent on the cooperation among the members for their well being. Members of the *nabuurschap* helped each other in difficult times during harvest times, funerals or any hardship that fell upon the community.
- Abraham & Sarah (identical)
When people turn 50 in Hasselt it is said that they see Abraham or Sarah. The custom is to give these people a *speculaas* Abraham or a Sarah. *Speculaas* is a kind of hard spiced biscuit.

The set up parameters are described in table 6.2.

Set up	Meaning
standard	standard (global) set up
packed	no vertical spacing between items
serried	no horizontal spacing between separator and text
joinedup	no vertical spacing before and after itemize
broad	horizontal spacing between separator and text
inmargin	place separator in margin
atmargin	place separator on margin
stopper	place full stop after separator
columns	put items in columns
intro	prevent page breaking after introduction line
continue	continue numbering or lettering

Table 6.2 Set up parameters in itemize.

You can use the set up parameters in `\startitemize`, but for reasons of consistency you can make them valid for the complete document with `\setupitemize`.

The parameter `columns` is used in conjunction with a number. If you type this:

```
\startitemize[n,columns,four]
\item Achter 't Werk
.
.
.
\item Justitiebastion
\stopitemize
```

You will get:

- | | | | |
|-------------------|-------------------|-----------------|---------------------|
| 1. Achter 't Werk | 4. Eikenlaan | 7. Heerengracht | 10. Julianakade |
| 2. Baangracht | 5. Eiland | 8. Hofstraat | 11. Justitiebastion |
| 3. Brouwersgracht | 6. Gasthuisstraat | 9. Hoogstraat | |

Sometimes you want to continue the enumeration after a short intermezzo. Then you type for example `\startitemize[continue,columns,three,broad]` and numbering will continue in a three column format.

- | | | |
|------------------|--------------------|----------------------|
| 12. Kaai | 18. Prinsengracht | 24. Vicariehof |
| 13. Kalverstraat | 19. Raamstraat | 25. Vissteeg |
| 14. Kastanjelaan | 20. Ridderstraat | 26. Watersteeg |
| 15. Keppelstraat | 21. Rosmolenstraat | 27. Wilhelminalaan |
| 16. Markt | 22. Royenplein | 28. Ziekenhuisstraat |
| 17. Meestersteeg | 23. Van Nahuijsweg | |

The parameter `broad` enlarges the horizontal space between item separator and itemtext.

```
\setupitemize[.1.][...,2,...][...,...=,...]
```

An itemize within an itemize is automatically typeset in a correct way. For example if you type:

```
In The Netherlands the cities can determine the size of a number of
taxes. So the cost of living can differ from town to town. There are
differences of up to 50\% in taxes such as:
```

```
\setupitemize[2][width=5em]
```

```
\startitemize[n]
```

```
\item real estate tax
```

The real estate tax is divided into two components:

```
\startitemize[a,packed]
```

```
\item the ownership tax
```

```
\item the tenant tax
```

```
\stopitemize
```

If the real estate has no tenant the owner pays both components.

```
\item dog licence fee
```

The owner of one or more dogs pays a fee. When a dog has died or been sold the owner has to inform city hall.

```
\stopitemize
```

then the horizontal space between item separator and text at the second level is set with `\setupitemize[2][width=5em]`.

The example will look like this:

In The Netherlands the cities can determine the size of a number of taxes. So the cost of living can differ from town to town. There are differences of up to 50% in taxes such as:

1. real estate tax

The real estate tax is divided into two components:

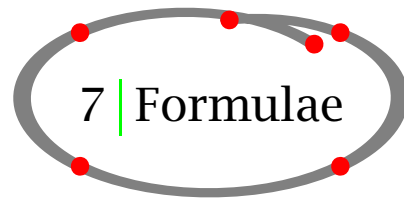
- a. the ownership tax

- b. the tenant tax

If the real estate has no tenant the owner pays both components.

2. dog licence fee

The owner of one or more dogs pays a fee. When a dog has died or been sold the owner has to inform city hall.



You can typeset numbered formulas with:

```
\placeformula[ref,...]{.1.}$$$.2.$$
```

```
\startformula ... \stopformula
```

Two examples:

```
\placeformula[formula:aformula]
```

```
\startformula
```

```
y=x^2
```

```
\stopformula
```

```
\placeformula
```

```
\startformula
```

```
\int_0^1 x^2 dx
```

```
\stopformula
```

$$y = x^2 \tag{7.1}$$

$$\int_0^1 x^2 dx \tag{7.2}$$

Between `\startformula` and `\stopformula` you are in math mode so you can define any formula you want by using \TeX commands. We advise you to do some further reading on typesetting formulae in \TeX . See for example:

- *The \TeX Book* by D.E. Knuth
- *The Beginners Book of \TeX* by S. Levy and R. Seroul

Math mode can occur in two modes: in text mode and display mode. Mathematic expressions in text mode are placed between `$` and `$`.

The Hasselt community covers an area of 42,05 \Square \Kilo \Meter.
Now if you consider a circular area of this size with the market place of Hasselt as the center point `M` you can calculate its diameter with `$$\frac{1}{4} \pi r^2$`.

This will become:

The Hasselt community covers an area of 42,05 km². Now if you consider a circular area of this size with the market place of Hasselt as the center point M you can calculate its diameter with $\frac{1}{4}\pi r^2$.

The many $\{\}$ in $\frac{1}{4}\pi r^2$ are essential for separating operations in the expression. If you omit the outer curly braces like this: $\frac{1}{4}\pi r^2$, you would get a non desired result: $\frac{1}{4\pi r^2}$.

Although display mode can also be entered and exit with $\$$, but only `\startformula` and friend get the spacing right. So if you type:

```
$$
\int_0^1 x^2 dx
$$
```

you will get an expression that is *displayed* in the middle of a page, but not well as aligned as the previous examples.

$$\int_0^1 x^2 dx$$

The command `\placeformula` handles spacing around the formulae and the numbering of the formula. The bracket pair is optional and is used for cross-references and switching numbering off.

$$y = x^2 \tag{7.3}$$

$$y = x^3 \tag{7.4}$$

$$y = x^4 \tag{7.5}$$

Formula 7.4 was typed like this:

```
\placeformula[middle one]
\startformula
y=x^3
\stopformula
```

The label `[middle one]` is used for refering to this formula. Such a reference is made with `\in{formula}[middle one]`.

If no numbering is required you type:

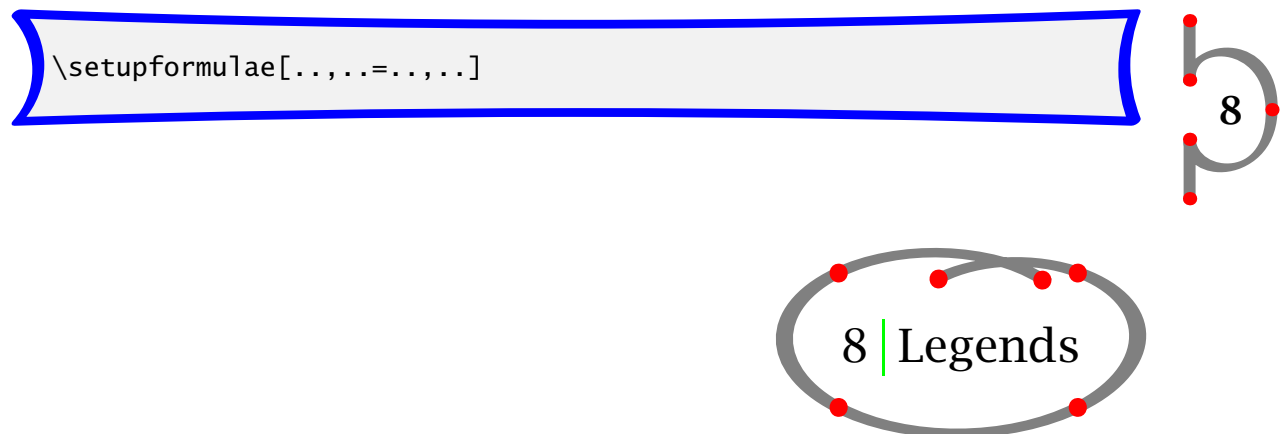
```
\placeformula[-]
```

Numbering of formulae is set up with `\setupnumbering`. In this manual numbering is set up with `\setupnumbering[way=bychapter]`. This means that the chapter number

Legends

preceeds the formula number and numbering is reset with each new chapter. For reasons of consistency the tables, figures, intermezzi etc are numbered in the same way. Therefore you use `\setupnumbering` in the set up area of your input file.

Formulae can be set up with:



The commands `\startlegend ... \stoplegend` are used to explain the meaning of symbols in formulae. For example, if you want to display a formula and its legend:

$$s = vt + \frac{1}{2}at^2 \quad (8.1)$$

s = displacement m
 v = velocity m/s
 t = time s
 a = acceleration m/s²

You will have to type something like this:

```
\placeformula$$ s = v t + {{1}\over{2}} a t^2 $$
\startlegend
\leg s \\ displacement \\ m      \\
\leg v \\ velocity      \\ m/s   \\
\leg t \\ time          \\ s      \\
\leg a \\ acceleration  \\ m/s^2  \\
\stoplegend
```

The command `\leg` starts a new row and `\\` is a column separator. The last `\\` in each row is essential. The spaces are optional and have no effect on the outcome. However, these commands are easily mistyped and misread while checking. So take some time to make a readable input file.

```
\startlegend[.1.]\leg.2.\.3.\.4.\\leg ... \stoplegend
```

The first column is used for symbols and is typeset in mathematical mode. The second column is used for explanations of the symbols and the third one for units and dimensions. Because of the necessity of sub- and superscript the third column is in mathematical mode, but the text is displayed in the normal font.

Complementary to legends is a command that displays facts in a consistent way when formulae are used for calculations.

Such a command looks like this:

```
\startfact
\fact velocity    \v v \ 10~m/s \
\fact acceleration \a a \ - 3~m/s^2 \
\fact time        \t t \ 3~s \
\stopfact
```

The two last columns are in mathematical mode, the last column displays text in the normal font. The ~ is obligatory, because spaces are ignored in mathematical mode.

```
velocity    v = 10 m/s
acceleration a = -3 m/s^2
time        t = 3 s
```

9 | Units

To force yourself to use dimensions and units consistently throughout the document you can make your own list with units. These are specified in the set up area of your input file.

In `CONTEXT` there is an external module available that contains almost all SI-units. When this module is loaded with `\usemodule[units]` you can call units with:

```
\Meter \Per \Square \Meter
\Cubic \Meter \Per \Sec
\Square \Milli \Meter \Per \Inch
\Centi \Liter \Per \Sec
```

Figures

```
\Meter \Inverse \Sec  
\Newton \Per \Square \Inch  
\Newton \Times \Meter \Per \Square \Sec
```

It looks like a lot of typing but it does guarantee a consistent use of units. The command `\unit` also prevents the separation of value and unit at line breaks, because a number typeset at the end of a line and the unit at the beginning of the next one, is far from perfect. These examples come out as:

```
m/m2  
m3/s  
mm2/inch  
cl/s  
ms-1  
N/inch2  
N·m/s2
```

You can define your own units with:

```
\unit[Ounce]{oz}{}
```

Later on in the document you can type `15.6 \Ounce` that will be displayed as 15.6 oz.

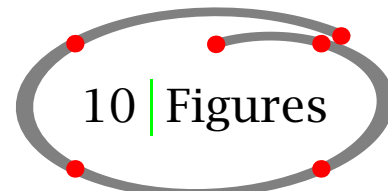
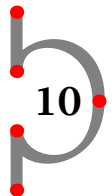
The legend on page 17 would be more consistent if you typed it like this:

```
\startlegend  
\leg s \ place          \ \ \Meter                \\  
\leg v \ velocity      \ \ \Meter \Per \Sec        \\  
\leg t \ time          \ \ \Sec                    \\  
\leg a \ acceleration  \ \ \Meter \Per \Square \Sec \\  
\stoplegen
```

The command `\unit` is an application of synonyms. See chapter 24 for more information on synonyms.

In order to write % and ‰ in a consistent way there are two specific commands:

```
\percent  
\permille
```



Photographs and pictures can be inserted in your document with the following command:



Figures

```
\placefigure  
  [][fig:church]  
  {Stephanus Church.}  
  {\externalfigure[hasselt church]}
```

After processing this will come out as figure 10.1 at the first available place.



Figure 10.1 Stephanus Church.

The command `\placefigure` handles numbering and vertical spacing before and after your figure. Furthermore this command initializes a float mechanism, which means that `CONTEXT` looks whether there is enough space for your figure on the page. If not the figure will be placed at another location and the text carries on, while the figure floats in your document until the optimal location is found. You can influence this mechanism within the first bracket pair.

The command `\placefigure` is a predefined example of:

```
\placeblock[.1.][ref,...]{.2.}{.3.}
```

The options are described in table 10.1.

The second bracket pair is used for cross-referencing. You can refer to this particular figure by typing:

```
\in{figure}[fig:church]
```

The first brace pair is used for the caption. You can type any text you want. If you want no caption and no number, you can type `{none}`. The figure labels are set up with `\setupcaptions` and the numbering is (re)set by `\setupnumbering` (see paragraph 37.1).

Option	Meaning
here	put figure at this location if possible
force	force figure placement here
page	put figure on its own page
top	put the figure at the top of the page
bottom	put the figure at the bottom of the page
left	place figure at the left margin
right	place figure at the right margin

Table 10.1 Options in `\placefigure`.

The second brace pair is used for defining the figure and addressing the file names of external figures.

In the next example you see how `\Hasselt` is defined within the last brace pair to show you the function of `\placefigure{}{}{}`.

```
\placefigure
  {A framed Hasselt.}
  {\framed{\tfd Hasselt}}
```

This will produce:

Hasselt

Figure 10.2 A framed Hasselt.

However, your pictures are often created using programs like Corel Draw or Illustrator and photos are — after scanning — improved in packages like PhotoShop. Then the figures are available as files. `CONTEXT`, in conjunction with `TEXUTIL`, supports EPS, TIF, JPG, MPS, PDF, and PNG files, although inclusion depends on the DVI drivers used. Currently `PDFTEX` supports the most formats. Users normally can trust `CONTEXT` to find the best possible file type.

In figure 25.1 you see a photo and a graphic combined into one figure.

You can produce this figure by typing something like:

```
\useexternalfigure
  [Photo] [hass03g]
  [width=.4\textwidth]
\useexternalfigure
  [Graphic] [gracht]
  [width=.4\textwidth]
```

Figures



a bitmap picture



a vector graphic

Figure 10.3 The Hasselt Canals.

```
\placefigure
[here,force]
[fig:canal]
{The Hasselt Canals.}
\startcombination[2*1]
{\externalfigure[Photo]} {a bitmap picture}
{\externalfigure[Graphic]} {a vector graphic}
\stopcombination
```

```
\useexternalfigure[.1.][.2.][...,...=...,...]
```

The command `\useexternalfigure` has three bracket pairs. The first pair contains a logical name of the figure. This is optional and used when one figure appears more than once in your document.

The second pair contains the file name (without extension). The third pair leaves room for filetype definition and some set up parameters. In well structured documents you would type `\useexternalfigure` in the set up area of your input file.

```
\startcombination[...] ... \stopcombination
```

The `\startcombination... \stopcombination` pair is used for combining two pictures in one figure. You can type the number of pictures within the bracket pair. If you want to display one picture below the other you would have typed `[1*2]`. You can imagine what happens when you combine 6 pictures as `[3*2]` (`[rows*columns]`).

The examples shown above are enough for creating illustrated documents. Sometimes however you want a more integrated layout of the picture and the text. For that purpose you can use:


```
\startblocktext[.1.][ref]{.2.}{.3.} ... \stopblock
```

Figure and table texts are already predefined:

```
\startfiguretext
[left]
[fig:citizens]
{none}
{\externalfigure[hass07g]
[width=.5\makeupwidth]}
Hasselt has always had a varying number of citizens due to
economic events. For example the Dedemsvaart was dug around 1810.
This canal runs through Hasselt and therefore trade flourished.
This led to a population growth of almost 40\% within 10~years.
Nowadays the Dedemsvaart has no commercial value anymore and the
canals have become a tourist attraction.
\stopfiguretext
```

is shown in the figure below.



Hasselt has always had a varying number of citizens due to economic events. For example the Dedemsvaart was dug around 1810. This canal runs through Hasselt and therefore trade flourished. This led to a population growth of almost 40% within 10 years. Nowadays the Dedemsvaart has no commercial value anymore and the canals have become a tourist attraction.

```
\externalfigure[...][...,...=...,...]
```

The last curly brace pair encloses the command `\externalfigure`. This command gives you the freedom to do anything you want with a figure. `\externalfigure` has two bracket pairs. The first is used for the exact file name without extension, the second for file formats and dimensions. It is not difficult to guess what happens if you type:²

² See page 47

Figures

```
\inmarge
  {\externalfigure
    [hass23g]
    [width=\marginwidth]}
```

You can set up the layout of figures with:

```
\setupfloats[...]=...]
```

You can set up the numbering and the labels with:

```
\setupcaptions[...]=...]
```

These commands are typed in the set up area of your input file and have a global effect on all floating blocks.

```
\setupfloats
  [location=right]
\setupcaptions
  [location=top,
   style=boldslanted]
\placefigure
  {Just a picture.}
  {\externalfigure[hass18g][width=4cm]}
```

Figure 10.4 *Just a picture.*



11 | Tables

In general, a table consists of columns which may be independently left adjusted, centered, right adjusted, or aligned on decimal points. Headings may be placed over single columns or groups of columns. Table entries may contain equations or several rows of text. Horizontal and vertical lines may be drawn wholly or partially across the table.

This is what Michael J. Wichura wrote in the preface of the manual of `TABLE` (`TABLE` manual, 1988.). Michael Wichura is also the author of the `TABLE` macros `CONTEXT` is relying on when processing tables. A few `CONTEXT` macros were added to take care of consistent line spacing and to make the interface a little less cryptic.³

For placing a table the command `\placetable` is used which is a predefined example of:

```
\placeblock[.1.][ref,...]{.2.}{.3.}
```

For defining the table you use:

```
\starttable[...] ... \stoptable
```

The definition of a table could look something like this:

```
\placetable[here][tab:ships]{Ships that moored at Hasselt.}
\starttable[|c|c|]
\HL
\NC \bf Year \NC \bf Number of ships \NC\SR
\HL
\NC 1645 \NC 450 \NC\FR
\NC 1671 \NC 480 \NC\MR
\NC 1676 \NC 500 \NC\MR
\NC 1695 \NC 930 \NC\LR
\HL
\stoptable
```

This table is typeset as table 11.1.

³ `CONTEXT` was developed for non-technical users in the WYSIWYG era. Therefore a user friendly interface and easy file and command handling were needed, and cryptic commands, programming and logical reasoning had to be avoided.

Tables

Year	Number of ships
1645	450
1671	480
1676	500
1695	930

Table 11.1 Ships that moored at Hasselt.

The first command `\placetable` has the same function as `\placefigure`. It takes care of spacing before and after the table and numbering. Furthermore the floating mechanism is initialized so the table will be placed at the most optimal location of the page.

The table entries are placed between the `\starttable` \cdots `\stoptable` pair. Between the bracket pair you can specify the table format with the column separators `|` and the format keys (see table 11.2).

Key	Meaning
<code> </code>	column separator
<code>c</code>	center
<code>l</code>	flush left
<code>r</code>	flush right
<code>s<n></code>	set intercolumn space at value $n = 0, 1, 2$
<code>w<></code>	set minimum column width at specified value

Table 11.2 Table format keys.

In addition to the format keys there are format commands. Table 11.3 shows a few of the essential commands.

Command	Meaning
<code>\JustLeft</code>	flush left and suppress column format
<code>\JustRight</code>	flush right and suppress column format
<code>\JustCenter</code>	center and suppress column format
<code>\SetTableToWidth{}</code>	specify exact table width
<code>\use{n}</code>	use the space of the next n columns

Table 11.3 Table format commands.

In the examples you have seen so far a number of `CONTEXT` formatting commands were used. These commands are somewhat longer than the original and less cryptic but they

Tables

also handle a lot of table typography. In table 11.4 an overview of these commands is given.

Command		Meaning
\NR	next row	make row with no vertical space adjustment
\FR	first row	make row, adjust upper spacing
\LR	last row	make row, adjust lower spacing
\MR	mid row	make row, adjust upper and lower spacing
\SR	separate row	make row, adjust upper and lower spacing
\VL	vertical line	draw a vertical line, go to next column
\NC	next column	go to next column
\HL	horizontal line	draw a horizontal
\DL	division line*	draw a division line over the next column
\DL[n]	division line*	draw a division line over <i>n</i> columns
\DC	division column*	draw a space over the next column
\DR	division row*	make row, adjust upper and lower spacing
\LOW{text}	—	lower <i>text</i>
\TWO, \THREE etc.	—	use the space of the next <i>two</i> , <i>three</i> columns

* \DL, \DC and \DR are used in combination.

Table 11.4 `CONTEXT` table format commands.

The tables below are shown with their sources. You can always read the `TABLE` manual by M.J. Wichura for more sophisticated examples.

```
\placetable
[here,force]
[tab:effects of commands]
{Effect of formatting commands.}
\startcombination[2*1]
{\starttable[|c|c|]
\HL
\VL \bf Year \VL \bf Citizens \VL\SR
\HL
\VL 1675      \VL ~428      \VL\FR
\VL 1795      \VL 1124      \VL\MR
\VL 1880      \VL 2405      \VL\MR
\VL 1995      \VL 7408      \VL\LR
\HL
\stoptable}{standard}
{\starttable[|c|c|]
\HL
\VL \bf Year \VL \bf Citizens \VL\NR
\HL
```

Tables

```

\VL 1675      \VL ~428      \VL\NR
\VL 1795      \VL 1124      \VL\NR
\VL 1880      \VL 2405      \VL\NR
\VL 1995      \VL 7408      \VL\NR
\HL
\stoptable}{only \type{\NR}}
\stopcombination

```

In the example above the first table \SR, \FR, \MR and \LR are used. These commands take care of line spacing within a table. As you can see below the command \NR only starts a new row.

Year	Citizens
1675	428
1795	1124
1880	2405
1995	7408

Year	Citizens
1675	428
1795	1124
1880	2405
1995	7408

standard

only \NR

Table 11.5 Effect of formatting commands.

In the example below column interspacing with the s0 and s1 keys is shown.

```

\startbuffer[one]
\starttable[|c|c|]
\HL
\VL \bf Year \VL \bf Citizens \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer
\startbuffer[two]
\starttable[s0 | c | c |]
\HL
\VL \bf Year \VL \bf Citizens \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR

```

Tables

```

\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer
\startbuffer[three]
\starttable[| s0 c | c |]
\HL
\VL \bf Year \VL \bf Citizens \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer
\startbuffer[four]
\starttable[| c | s0 c |]
\HL
\VL \bf Year \VL \bf Citizens \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer
\startbuffer[five]
\starttable[s1 | c | c |]
\HL
\VL \bf Year \VL \bf Citizens \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer

```

Tables

```
\placetable
[here,force]
[tab:formatcommands]
{Effect of formatting commands.}
\startcombination[3*2]
{\getbuffer[one]}   {standard}
{\getbuffer[two]}   {\type{s0}}
{\getbuffer[three]} {\type{s0} in column~1}
{\getbuffer[four]}  {\type{s0} in column~2}
{\getbuffer[five]}  {\type{s1}}
{}                  {}
\stopcombination
```

After processing these tables come out as table 11.3. The default table has a column interspacing is s2.

Year	Citizens	YearCitizens	YearCitizens
1675	428	1675 428	1675 428
1795	1124	1795 1124	1795 1124
1880	2405	1880 2405	1880 2405
1995	7408	1995 7408	1995 7408

standard s0 s0 in column 1

Year	Citizens	YearCitizens
1675	428	1675 428
1795	1124	1795 1124
1880	2405	1880 2405
1995	7408	1995 7408

s0 in column 2 s1

Table 11.6 Effect of formatting commands.

Columns are often separated with a vertical line | and rows by a horizontal line.

```
\placetable
[here,force]
[tab:divisions]
{Effect of options.}
\starttable[|c|c|c|]
\NC Steenwijk \NC Zwartsluis \NC Hasselt \NC\SR
```


Tables

```
\DC          \DL          \DC          \DR
\NC Zwartsluis \VL Hasselt  \VL Steenwijk \NC\SR
\DC          \DL          \DC          \DR
\NC Hasselt   \NC Steenwijk \NC Zwartsluis \NC\SR
\stoptable
```

	Steenwijk	Zwartsluis	Hasselt
Zwartsluis		Hasselt	Steenwijk
Hasselt	Steenwijk		Zwartsluis

Table 11.7 Effect of options.

A more sensible example is given in the table below.

```
\placetable
[here,force]
[tab:examplecontextcommands]
{Effect of \CONTEXT\ formatting commands.}
\starttable[|l|c|c|c|c|]
\HL
\VL \FIVE \JustCenter{City council elections in 1994} \VL\SR
\HL
\VL \LOW{Party} \VL \THREE{Districts} \VL \LOW{Total} \VL\SR
\DC          \DL[3]          \DC          \DR
\VL          \VL 1   \VL 2   \VL 3   \VL          \VL\SR
\HL
\VL PvdA      \VL 351 \VL 433 \VL 459 \VL 1243      \VL\FR
\VL CDA       \VL 346 \VL 350 \VL 285 \VL ~981      \VL\MR
\VL VVD       \VL 140 \VL 113 \VL 132 \VL ~385      \VL\MR
\VL HKV/RPF/SGP \VL 348 \VL 261 \VL 158 \VL ~767      \VL\MR
\VL GPV       \VL 117 \VL 192 \VL 291 \VL ~600      \VL\LR
\HL
\stoptable
```

In the last column a ~ is used to simulate a four digit number. The ~ has the width of a digit.

Sometimes your tables get too big and you want to adjust, for example, the body font or the vertical and/or horizontal spacing around vertical and horizontal lines. This is done by:

Tables

City council elections in 1994				
Party	Districts			Total
	1	2	3	
PvdA	351	433	459	1243
CDA	346	350	285	981
VVD	140	113	132	385
HKV/RPF/SGP	348	261	158	767
GPV	117	192	291	600

Table 11.8 Effect of CONTEXT formatting commands.

`\setuptables[...]=...]`

```
\placetable
[here,force]
[tab:setuptable]
{Use of \type{\setuptables}.}
\startcombination[1*3]
{\setuptables[bodyfont=10pt]
\starttable[|c|c|c|c|c|c|]
\HL
\VL \use6 \JustCenter{Decline of wealth in
Dutch florine (Dfl)} \VL\SR
\HL
\VL Year \VL 1.000--2.000
\VL 2.000--3.000
\VL 3.000--5.000
\VL 5.000--10.000
\VL over 10.000 \VL\SR
\HL
\VL 1675 \VL 22 \VL 7 \VL 5 \VL 4 \VL 5 \VL\FR
\VL 1724 \VL ~4 \VL 4 \VL -- \VL 4 \VL 3 \VL\MR
\VL 1750 \VL 12 \VL 3 \VL 2 \VL 2 \VL -- \VL\MR
\VL 1808 \VL ~9 \VL 2 \VL -- \VL -- \VL -- \VL\LR
\HL
\stoptable}{\tt bodyfont=10pt}
{\setuptables[bodyfont=8pt]
\starttable[|c|c|c|c|c|c|]
```

Tables

```

\HL
\VL \use6 \JustCenter{Decline of wealth in
                        Dutch florine (Dfl)} \VL\SR
\HL
\VL Year \VL 1.000--2.000
        \VL 2.000--3.000
        \VL 3.000--5.000
        \VL 5.000--10.000
        \VL over 10.000 \VL\SR
\HL
\VL 1675 \VL 22 \VL 7 \VL 5 \VL 4 \VL 5 \VL\FR
\VL 1724 \VL ~4 \VL 4 \VL -- \VL 4 \VL 3 \VL\MR
\VL 1750 \VL 12 \VL 3 \VL 2 \VL 2 \VL -- \VL\MR
\VL 1808 \VL ~9 \VL 2 \VL -- \VL -- \VL -- \VL\LR
\HL
\stoptable}{\tt bodyfont=8pt}
{\setuptables[bodyfont=6pt,distance=small]}
\starttable[|c|c|c|c|c|c|]
\HL
\VL \use6 \JustCenter{Decline of wealth in
                        Dutch florine (Dfl)} \VL\SR
\HL
\VL Year \VL 1.000--2.000
        \VL 2.000--3.000
        \VL 3.000--5.000
        \VL 5.000--10.000
        \VL over 10.000 \VL\SR
\HL
\VL 1675 \VL 22 \VL 7 \VL 5 \VL 4 \VL 5 \VL\FR
\VL 1724 \VL ~4 \VL 4 \VL -- \VL 4 \VL 3 \VL\MR
\VL 1750 \VL 12 \VL 3 \VL 2 \VL 2 \VL -- \VL\MR
\VL 1808 \VL ~9 \VL 2 \VL -- \VL -- \VL -- \VL\LR
\HL
\stoptable}{\tt bodyfont=6pt,distance=small}
\stopcombination

```

You can also set up the layout of tables with:

```
\setupfloats[...]=...,...]
```

You can set up the numbering and the labels with:

Tables

Decline of wealth in Dutch florine (Dfl)					
Year	1.000-2.000	2.000-3.000	3.000-5.000	5.000-10.000	over 10.000
1675	22	7	5	4	5
1724	4	4	-	4	3
1750	12	3	2	2	-
1808	9	2	-	-	-

bodyfont=10pt

Decline of wealth in Dutch florine (Dfl)					
Year	1.000-2.000	2.000-3.000	3.000-5.000	5.000-10.000	over 10.000
1675	22	7	5	4	5
1724	4	4	-	4	3
1750	12	3	2	2	-
1808	9	2	-	-	-

bodyfont=8pt

Decline of wealth in Dutch florine (Dfl)					
Year	1.000-2.000	2.000-3.000	3.000-5.000	5.000-10.000	over 10.000
1675	22	7	5	4	5
1724	4	4	-	4	3
1750	12	3	2	2	-
1808	9	2	-	-	-

bodyfont=6pt,distance=small

Table 11.9 Use of \setuptables.

\setuptables[...,...,...]

These commands are typed in the set up area of your input file and have a global effect on all floating blocks.

\setupfloats[location=left]

\setupcaption[style=boldslanted]

\placetable{Library opening hours.}

\starttable[|l|c|c|]

\HL

\VL \bf Day \VL \use2 \bf Opening hours

\VL\SR

\HL

\VL Monday \VL 14.00 -- 17.30 \VL 18.30 -- 20.30 \VL\FR

\VL Tuesday \VL

\VL

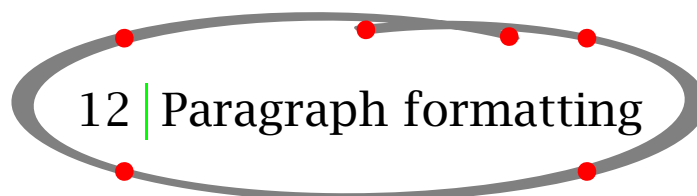
\VL\MR

Paragraph formatting

```
\VL Wednesday \VL 10.00 -- 12.00 \VL 14.00 -- 17.30 \VL\MR
\VL Thursday \VL 14.00 -- 17.30 \VL 18.30 -- 20.30 \VL\MR
\VL Friday \VL 14.00 -- 17.30 \VL \VL\MR
\VL Saturday \VL 10.00 -- 12.30 \VL \VL\LR
\HL
\stoptable
```

Day	Opening hours	
Monday	14.00 - 17.30	18.30 - 20.30
Tuesday		
Wednesday	10.00 - 12.00	14.00 - 17.30
Thursday	14.00 - 17.30	18.30 - 20.30
Friday	14.00 - 17.30	
Saturday	10.00 - 12.30	

Table 11.10 Library opening hours.



Sometimes you want to typeset paragraphs in a specially formatted way. This is done with:

```
\defineparagraphs[...] [..., ...=..., ...]
```

```
\setupparagraphs[.1.] [.2.] [..., ...=..., ...]
```

In the example below is shown how to typeset complete paragraphs columnwise.

```
\defineparagraphs
[mypar] [n=3, before={\blank}, after={\blank}]
```



Paragraph formatting

```
\setupparagraphs
  [mypar][1][width=.1\textwidth,style=bold]
\setupparagraphs
  [mypar][2][width=.4\textwidth]

\startmypar
1252
\mypar
Hasselt obtains its city charter from bishop Hendrik van Vianden.
\mypar
Hendrik van Vianden was pressed by other towns not to agree with the
charter. It took Hasselt a long period of time to convince the
Bishop. After supporting the Bishop in a small war against the
Drents, the charter was released.
\stopmypar

\startmypar
1350
\mypar
Hasselt joins the Hanzepact to protect their international trade.
\mypar
The Hanzepact was of great importance for merchants in Hasselt. In
those days trading goods were taxed at every city, highway or
rivercrossing. After joining the Hanzepact duty free routes all over
Europe became available to Hasselt. However important the Hanzepact
was, Hasselt always stayed a minor member of the pact.
\stopmypar
```

After defining a paragraph with `\defineparagraphs` you can format the paragraph with `\setupparagraphs`. Next you can start your paragraph with `\start...` and end it with `\stop...`. A new paragraph starts with the name of your paragraph, in this case `\mypar`.

1252	Hasselt obtains its city charter from bishop Hendrik van Vianden.	Hendrik van Vianden was pressed by other towns not to agree with the charter. It took Hasselt a long period of time to convince the Bishop. After supporting the Bishop in a small war against the Drents, the charter was released.
------	---	--

- 1350 Hasselt joins the Hanzepact to protect their international trade. The Hanzepact was of great importance for merchants in Hasselt. In those days trading goods were taxed at every city, highway or rivercrossing. After joining the Hanzepact duty free routes all over Europe became available to Hasselt. However important the Hanzepact was, Hasselt always stayed a minor member of the pact.

Another example of paragraph formatting could look like this.

```
\defineparagraphs[chempar] [n=3,before=,after=,distance=1em]
\setupparagraphs [chempar] [1] [width=.12\textwidth]
\setupparagraphs [chempar] [2] [width=.4\textwidth]

\startchempar
Limekilns
\chempar
Hasselt has its own limekilns. These were build in 1504 and produced
quick lime up to 1956. Nowadays they are a tourist attraction.
\chempar
\chemical{CaCO_3,~,GIVES,~,CaO,~,+,~,CO_2}
\stopchempar
```



And it would come out like this:

Limekilns Hasselt has its own limekilns. These $\text{CaCO}_3 \rightarrow \text{CaO} + \text{CO}_2$ were build in 1504 and produced quick lime up to 1956. Nowadays they are a tourist attraction.

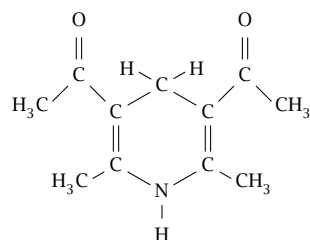
By the way you could also type it in a more cryptic way:

```
\chempar Limekilns
\\ Hasselt has its own limekilns. They were build in 1504 and
produced quick lime up to 1956. Nowadays they form a
tourist attraction.
\\ \chemical{CaCO_3,~,GIVES,~,CaO,~,+,~,CO_2} \\
```

The \\ are used as column separators and are essential.

The chemical module is explained in another manual, because not everybody is interested in chemical stuff. However, chemical structures always look impressive.

Columns



Compound A

CONTEXT relies on the `\pTeX` macros of M.J. Wichura to draw this kind of structure. Although the chemical module consists of only two or three commands, it takes some practice to get the right results. This is how the input looks:

```
\placeformula[-]  
\startformula  
\startchemical[scale=small,width=fit,top=3000,bottom=3000]  
  \chemical[SIX,SB2356,DB14,Z2346,SR3,RZ3,-SR6,+SR6,-RZ6,+RZ6]  
    [C,N,C,C,H,H,H]  
  \chemical[PB:Z1,ONE,Z0,DIR8,Z0,SB24,DB7,Z27,PE][C,C,CH_3,0]  
  \chemical[PB:Z5,ONE,Z0,DIR6,Z0,SB24,DB7,Z47,PE][C,C,H_3C,0]  
  \chemical[SR24,RZ24][CH_3,H_3C]  
  \bottext{Compound A}  
\stopchemical  
\stopformula
```

13

13 | Columns

Simple documents can be typeset in columns. If you precede a text fragment by `\startcolumns` and close the text fragment by `\stopcolumns` everything in between will be set in columns.

```
\startcolumns[...]=... \stopcolumns
```

```
\startcolumns[n=3,tolerance=verytolerant]
```

Hasselt is an old Hanseatic City, situated 12~km north of Zwolle at the river Zwartewater.

Columns

The city has a long history since obtaining the city charter around 1252. Part and parcel of this history can be traced back to a large number of monuments to be admired in the city center.

There you will find the St. Stephanus church, a late gothic church dating back to 1479 with a magnificent organ. The former Municipal Building is situated on The Market Place. Constituted between 1500 and 1550 it houses a large collection of weapons, amongst which one of the largest collection of black powder guns (haakhussen) in the whole world should be mentioned.

Furthermore there is a corn windmill 'The Swallow', dating back to 1748 as well as the 'Stenendijk', a unique embankment and the last shell limekiln in Europe still in full operation.

The city center with the townmoat adorned by lime-trees, the Van Stolkspark and the hustle and bustle at the docks are ideally suited for a stroll.

The area around Hasselt is also worth mentioning. In wintertime polder Mastenbroek harbours large numbers of geese. In summertime the hamlets Genne, Streukel and Celle\ -mui\ -den form, together with the very rare lapwing flowers (Lat. Fritillaria meleagris) found on the banks of the river Zwaterwater, the ideal surroundings for walking or cycling trips.

Hasselt also is a very important center for watersports. The lakes of northwest Overijssel, the river IJssel, the Overijsselse Vecht and the Randmeren are within easy reach from the yacht harbour 'De Molenwaard'. Sailing, fishing, swimming and canoeing can be fully enjoyed in Hasselt.

Furthermore some events of special interest should be mentioned. Every year at the end of August Hasselt celebrates the 'Eui Festival' (hay festival).

\stopcolumns

The result will be a three column text.

Hasselt is an old Hanseatic City, situated 12 km north of Zwolle at the river Zwartewater.

The city has a long history since obtaining the city charter around 1252. Part and par-

cel of this history can be traced back to a large number of monuments to be admired in the city center.

There you will find the St. Stephanus church, a late gothic church dating back to 1479

with a magnificent organ. The former Municipal Building is situated on The Market Place. Constituted between 1500 and 1550 it houses a large collection of weapons, amongst which one of the largest collection of black powder guns

Footnotes

(haakhussen) in the whole world should be mentioned.

Furthermore there is a corn windmill 'The Swallow', dating back to 1748 as well as the 'Stenendijk', a unique embankment and the last shell limekiln in Europe still in full operation.

The city center with the townmoat adorned by lime-trees, the Van Stolkspark and the hustle and bustle at the docks are ideally suited for a stroll.

The area around Hasselt is also worth mentioning. In winter-time polder Mastenbroek harbours large numbers of geese. In summertime the hamlets Genne, Streukel and Cellemuiden form, together with the very rare lapwing flowers (Lat. *Fritillaria meleagris*) found on the banks of the river Zwatewater, the ideal surroundings for walking or cycling trips.

Hasselt also is a very important center for watersports. The lakes of northwest Over-

jssel, the river IJssel, the Overijsselse Vecht and the Randmeren are within easy reach from the yacht harbour 'De Molenwaard'. Sailing, fishing, swimming and canoeing can be fully enjoyed in Hasselt.

Furthermore some events of special interest should be mentioned. Every year at the end of August Hasselt celebrates the 'Eui Festival' (hay festival).

If possible a new column can be enforced with `\column`. You can set up columns with:

```
\setupcolumns[...]=...]
```

If you want to annotate your text you can use `\footnote`. The command looks like this:

```
\footnote[ref]{...}
```

The bracket pair is optional and contains a logical name. The curly braces contain the text you want to display at the foot of the page.

The same footnote number can be called with its logical name.

```
\note[ref]
```

Definitions

If you have type this text:

The Hanse was a late medieval commercial alliance of towns in the regions of the North and the Baltic Sea. The association was formed for the furtherance and protection of the commerce of its members.⁴ This was the source of jealousy and fear among other towns that caused a number of wars.⁴ In the Hanse period there was a lively trade in all sorts of articles such as wood, wool, metal, cloth, salt, wine and beer.⁴ The prosperous trade caused an enormous growth of welfare in the Hanseatic towns.⁵ Hasselt is one of these towns.

It would look like this:

The Hanse was a late medieval commercial alliance of towns in the regions of the North and the Baltic Sea. The association was formed for the furtherance and protection of the commerce of its members.⁴ In the Hanse period there was a lively trade in all sorts of articles such as wood, wool, metal, cloth, salt, wine and beer.⁴ The prosperous trade caused an enormous growth of welfare in the Hanseatic towns.⁵

The footnote numbering is done automatically. The command `\setupfootnotes` enables you to influence the display of footnotes:

```
\setupfootnotes[...,...=...,...]
```

15 | Definitions

If you want to display notions, concepts and ideas in a consistent manner you can use:

```
\definedescription[...][...,...=...,...]
```

For example:

⁴ This was the source of jealousy and fear among other towns that caused a number of wars.

⁵ Hasselt is one of these towns.

```
\definedescription
  [concept]
  [location=serried,headstyle=bold,width=broad]
\concept{Hasselter juffer} A sort of biscuit made of puff pastry and
covered with sugar. It tastes very sweet. \par
```

It would look like this:

Hasselter juffer A sort of biscuit made of puff pastry and covered with sugar. It tastes very sweet.

But you can also choose other layouts:

Hasselter bitter

A very strong alcoholic drink (up to 40%) mixed with herbs to give it a special taste. It is sold in a stone flask and it should be served ijskoud (as cold as ice).

Euifeest A harvest home to celebrate the end of a period of hard work. The festivities take place in the third week of August.

If you have more than one paragraph in such a definition you can use a `\start...-\stop...` pair.

```
\definedescription
  [concept]
  [location=right,
   headstyle=bold,
   width=broad]
\startconcept{Euifeest} A harvest home to celebrate the end of a
period of hard work.
This event takes place at the end of August and lasts one week. The
city is completely illuminated and the streets are decorated. This
feast week ends with a {\em Braderie}.
\stopconcept
```

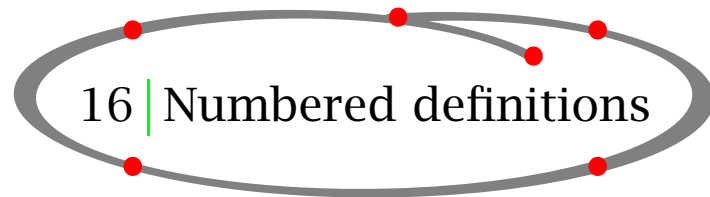
This would become:

A harvest home to celebrate the end of a period of hard work. **Euifeest**

This event takes place at the end of August and lasts one week. The city is completely illuminated and the streets are decorated. This feast week ends with a *Braderie*.

Layout is set up within the second bracket pair of `\definedescription[] []`. But you can also use:

```
\setupdescriptions[...][...=,...]
```

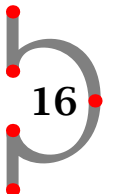


With `\defineenumeration` you can number text elements like remarks or questions. If you want to make numbered remarks in your document you use:

```
\defineenumeration[...1,...][.2.][...=...]
```

For example:

```
\defineenumeration
[remark]
[location=top,
text=Remark,
inbetween=\blank,
after=\blank]
```



Now the new commands `\remark`, `\subremark`, `\resetremark` and `\nextremark` are available and you can type remarks like this:

```
\remark In the early medieval times Hasselt was a place of
pilgrimage. The {\em Heilige Stede} (Holy Place) was torn down during
the Reformation. In 1930, after 300 years the {\em Heilige Stede} was
reopened.
```

```
\subremark Nowadays the {\em Heilige Stede} is closed again but once
a year an open air service is held on the same spot.
```

Remark 1

In the early medieval times Hasselt was a place of pilgrimage. The *Heilige Stede* (Holy Place) was torn down during the Reformation. In 1930, after 300 years the *Heilige Stede* was reopened.

Remark 1.1

Nowadays the *Heilige Stede* is closed again but once a year an open air service is held on the same spot.

You can reset numbering with `\resetremark` or `\resetsubremark` or increment a number with `\nextremark` or `\nextsubremark`. This is normally done automatically per chapter, section or whatever.

Outlined text

You can set up the layout of `\defineenumeration` with:

```
\setupenumerations[...][...=...]
```

You can also vary the layout of **Remark** and **Subremark** in the example above by:

```
\setupenumeration[remark][headstyle=bold]  
\setupenumeration[subremark][headstyle=slanted]
```

If a number becomes obsolete you can type:

```
\remark[-]
```

If the remark contains more than one paragraph you will have to use the command pair `\startremark ... \stopremark` that becomes available after defining **Remark** with `\defineenumeration[remark]`.

So the example above would look like this:

```
\startremark  
In the early medieval times Hasselt was a place of pilgrimage. The  
{\em Heilige Stede} (Holy Place) was torn down during the  
Reformation.  
  
After 300 years in 1930 the {\em Heilige Stede} was reopened.  
Nowadays the {\em Heilige Stede} is closed again but once a year an  
open air service is held on the same spot.  
\stopremark
```

Remark 1

In the early medieval times Hasselt was a place of pilgrimage. The *Heilige Stede* (Holy Place) was torn down during the Reformation.

After 300 years in 1930 the *Heilige Stede* was reopened. Nowadays the *Heilige Stede* is closed again but once a year an open air service is held on the same spot.

17 | Outlined text

You can outline a text with `\framed`. The command looks like this:

Outlined paragraphs

```
\framed[...,...=...,...]{...}
```

The bracket pair is optional and contains the set up parameters. The curly braces enclose the text. To be honest, the outlined text in the last paragraph was done with `\inframed`. This command takes care of the interline spacing.

```
\framed[height=3em,width=fit]{This box is bigger}
```

This becomes:

This box is bigger

Some other examples of `\framed` and its set up parameters are shown below. This time we use the in-line alternative `\inframed`.

```
\leftaligned
  {\inframed[width=fit]{People in Hasselt}}
\midaligned
  {\inframed[height=1.5cm,frame=off]{have a}}
\rightaligned
  {\inframed[background=screen]{historic background}}
```

This leads to:

People in Hasselt

have a

historic background

The `\framed` command is very sophisticated and it is used in many macros. The command to set up frames is:

```
\setupframed[...,...=...,...]
```

18 | Outlined paragraphs

Complete paragraphs can be outlined with:

```
\startframedtext[...][...=...,...] ... \stopframedtext
```

```
\definefloat[intermezzo]
```

```
\placeintermezzo[here][block:bridge]{An intermezzo.}
```

```
\startframedtext[width=.8\makeupwidth]
```

It was essential for Hasselt to have a bridge across the river Zwarte Water. The bishop of Utrecht gave Hasselt his consent in 1486.

```
\blank
```

Other cities in the neighbourhood of Hasselt were afraid of the toll money to be paid when crossing this bridge so they prevented the construction.

```
\stopframedtext
```

This example also illustrates the command `\definefloat`. You can find more information on this command in paragraph 37.2. The `\blank` is necessary to enforce a blank line.

It was essential for Hasselt to have a bridge across the river Zwarte Water. The bishop of Utrecht gave Hasselt his consent in 1486.

Other cities in the neighbourhood of Hasselt were afraid of the toll money to be paid when crossing this bridge so they prevented the construction.

Intermezzo 18.1 An intermezzo.

The outlining can be set up with:

```
\setupframedtexts[...=...,...]
```

19 | Margin texts

It is very easy to put text in the margin. You just use `\inmargin`.


```
\inmargin[.1.][ref]{.2.}
```

You may remember one of the earlier examples:

```
\inmarge
  {\externalfigure
    [hass23g]
    [width=\marginwidth]}
```



This would result in a figure in the margin. You can imagine that it looks quite nice in some documents. But be careful. The margin is rather small so the figure could become very marginal.

A few other examples are shown in the text below.

The Ridderstraat (Street of knights) \inmarge{Street of\\Knights} is an obvious name. In the 14th and 15th centuries, nobles and prominent citizens lived in this street. Some of their big houses were later turned into poorhouses \inright{poorhouse} and old peoples homes.

Up until \inleft[low]{\tfc 1940}1940 there was a synagog in the Ridderstraat. Some 40 Jews gathered there to celebrate their sabbath. During the war all Jews were deported to Westerbork and then to the extermination camps in Germany and Poland. None of the Jewish families returned. The synagog was knocked down in 1958.

The commands \inmargin, \inleft and \inright all have the same function. In a two sided document \inmargin puts the margin text in the correct margin. The \\ is used for line breaking. The example above would look like this:

**Street of
Knights**

The Ridderstraat (Street of knights) is an obvious name. In the 14th and 15th centuries, nobles and prominent citizens lived in this street. Some of their big houses were later turned into poorhouses and old peoples homes.

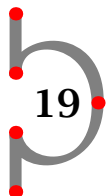
poorhouse

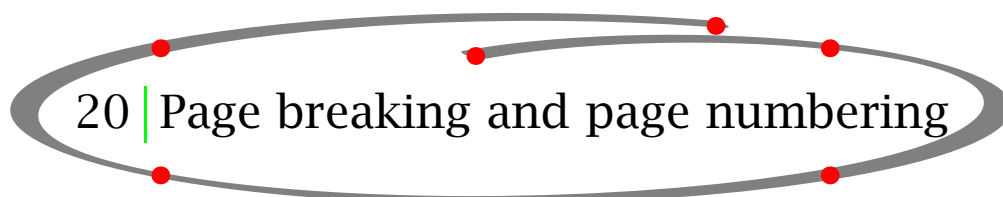
1940

Up until 1940 there was a synagog in the Ridderstraat. Some 40 Jews gathered there to celebrate their sabbath. During the war all Jews were deported to Westerbork and then to the extermination camps in Germany and Poland. None of the Jewish families returned. The synagog was knocked down in 1958.

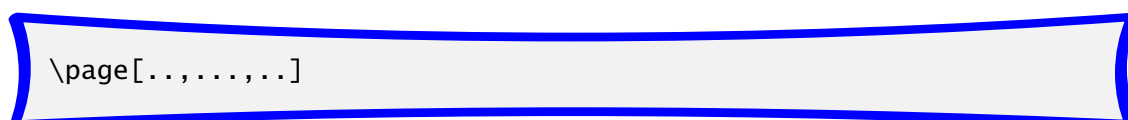
You can set up the margin text with:

```
\setupinmargin
```





A page can be enforced or blocked by:



The options can be stated within the brackets. The options and their meaning are presented in table 20.1

Option	Meaning
yes	enforce a page
makeup	enforce a page without filling
no	no page
preference	prefer a new page here
bigpreference	great preference for a new page here
left	next page is a left handside page
right	next page is a right handside page
disable	following commands have no effect
reset	following commands do have effect
empty	insert an empty page
last	add pages till even number is reached
quadruple	add pages till a multiple of four is reached



Table 20.1 Page options.

Page numbering happens automatically but you can enforce a page number with:

`\page[25]`

Sometimes it is better to state a relative page number like [+2] of [-2].

The position of the page numbers on a page depend on your own preferences and if it concerns a one sided or double sided document. Page numbering can be set up with:



```
\setuppagenumbering[...]=...]
```

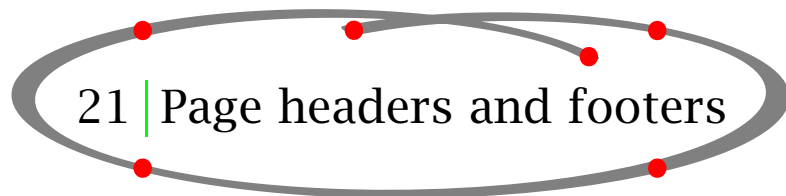
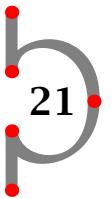
The preferences are placed within the brackets.

Tables or figures may take up a lot of space. The placing of these text elements can be postponed till the next page break. This is done with:

```
\startpostponing ... \stoppostponing
```

```
\startpostponing
\placefigure
  {A postponed figure.}
  {\externalfigure[hass16g][width=\textwidth]}
\stoppostponing
```

The figure will be placed at the top of next page and will cause minimal disruption of the running text.



In some cases you want to give your page a page header and footer. The commands to do this are:

```
\setupfootertexts[.1.][.2.][.3.]
```

```
\setupheadertexts[.1.][.2.][.3.]
```

The first bracket pair is used for the location of the footer or header (text, edge etc). Footer and header are placed within the second and third bracket pairs. In a double sided document the fourth and fifth bracket pairs are used for footer and header on the



Figure 21.1 A postponed figure.

left-hand side page and the right-hand page. In most cases you can omit the last two bracket pairs.

```
\setupfootertexts[Manual][section]
```

In this case the text *Manual* will appear in the left-hand side corner and the title of the actual section on the right-hand side of the page. This footer will change with the beginning of a new section.

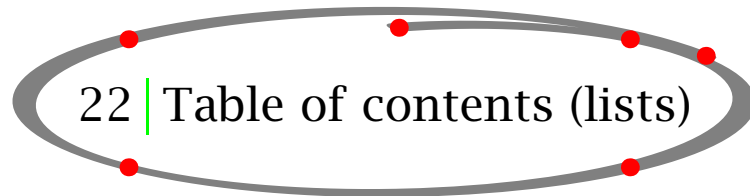
You can set up the head- and footline with:

```
\setupheader[...][...,...=,...]
```

```
\setupfooter[...][...,...=,...]
```

If you want to leave out the page header and footer you can type:

`\noheaderandfooterlines`



A table of contents contains chapter numbers, chapter titles and page numbers and can be extended with sections, sub sections, etc. A table of contents is generated automatically by typing:

`\placecontent`

Which table of contents is produced depends on the location of this command in your document. At the start of the document it will generate a list of chapters, sections etc. But at the top of a chapter:

`\chapter{Hasselt in Summer}`

`\placecontent`

`\section{Hasselt in July}`

`\section{Hasselt in August}`

it will only produce a list of (sub) section titles with the corresponding section numbers and page numbers.

The command `\placecontent` is available after definition with:

`\definecombinedlist[.1.][...2,...][...=...,...]`

This command and `\definelist` allows you to define your own lists necessary for structuring your documents.

The use of this command and its related commands is illustrated for the table of contents.

`\definelist[chapter]`

`\setuplist`

`[chapter]`

`[before=\blank,`

`after=\blank,`

`style=bold]`



Table of contents (lists)

```
\definelist[section]
\setuplist
  [section]
  [alternative=d]
```

Now there are two lists of chapters and sections and these will be combined in a table of contents with the command `\definecombinedlist`.

```
\definecombinedlist
  [contents]
  [chapter,section]
  [level=subsection]
```

Now two commands are available: `\placecontent` and `\completecontent`. With the second command the title of the table of contents will be added to the table of contents itself.

The layout of lists can be varied with the parameter `alternative`.

Alternative	Display
a	number - title - page number
b	number - title - spaces - page number
c	number - title - dots - page number
d	number - title - page number (continuing)
e	reserved for interactive purposes
f	reserved for interactive purposes

Table 22.1 Alternatives for displaying lists.

Lists are set up with:

```
\setuplist[...][...]=...]
```

```
\setupcombinedlist[...][...]=...]
```

If you want to change the layout of the generated table of contents you'll have to remember that it is a list.

```
\setupcombinedlist
  [content]
  [alternative=c,
```

Table of contents (lists)

```
aligntitle=no,  
width=2.5cm]
```

This will result in a somewhat different layout than the default one.

Lists are called up and placed with:

```
\placelist[...,...,...][...,...=...,...]
```

So if you want a table of content you type:

```
\placecontent[level=section]
```

or

```
\completecontent[level=section]
```

only the sections will be displayed. You might need this option when you have a well structured document that has sub sub sub sub sub sections and you don't want those in the table of contents.

A long list or a long table of contents will use up more than one page. To be able to force page breaking you can type:

```
\completecontent[2.2,8.5,12.3.3]
```

A page break will occur after section 2.2 and 8.5 and sub section 12.3.3.

In some cases you want to be able to write your own text in an automatically generated list. This is done with

```
\writetolist[.1.]{.2.}{.3.}
```

```
\writebetweenlist[.1.]{.2.}
```

For example if you want to make a remark in your table of contents after a section titled *Hotels in Hasselt* you can type:

```
\section{Hotels in Hasselt}  
\writebetweenlist[section]{\blank}  
\writetolist[section]{}{---under construction---}  
\writebetweenlist[section]{\blank}
```



It is possible to generate one or more registers. By default the command `\index` is available. If you want to add a word to the index you type:

```
\index{town hall}
```

The word *town hall* will appear as an index entry. An index is sorted in alphabetical order by an auxilliary program. Sometimes the index word does not appear in normal alphabetic order. For example, entries such as symbols have to provide extra sorting information in order to produce a correct alphabetical list:

```
\index[minus]{$-$}
```

Sometimes you have sub- or sub sub entries. These can be defined as follows:

```
\index{town hall+location}
\index{town hall+architecture}
```

You can generate your indexlist with:

```
\placeindex
```

or

```
\completeindex
```

The command `\index` is a predefined `CONTEXT` command, but of course you can also define your own registers.

```
\defineregister[.1.][.2.]
```

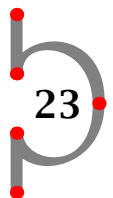
For example if you want to make a new register based on the streets in Hasselt you could type:

```
\defineregister[street][streets]
```

Now a new register command `\street` is available. An new index entry could be `\street{Ridderstraat}`. To produce a list of entries you could now use:

```
\placestreets
\completestreets
```

You can alter the display of the registers with:




```
\setupregister[.1.][.2.][...]=...]
```

24 | Synonyms

In many documents people want to use specific words consistently throughout the document. To enforce consistency the command below is available.

```
\definesynonyms[.1.][.2.][.3.][.4.]
```

The first bracket pair contains the singular form of the synonym, and the second contains the plural form. The third bracket pair contains a command.

For example the command `\abbreviation` is defined by:

```
\definesynonyms[abbreviation][abbreviations][\infull]
\setupsynonyms[style=cap]
```

Now the command `\abbreviation` is available and can be used to state your abbreviations:

```
\abbreviation{ANWB}{Dutch Automobile Association}
\abbreviation{VVV}{Bureau of Tourist Information}
\abbreviation{NS}{Dutch Railways}
```

If you would type:

The Dutch `\VVV` (`\infull{VVV}`) can provide you with the tourist information on Hasselt.

You would obtain something like this:

The Dutch `vvv` (Bureau of Tourist Information) can provide you with the tourist information on Hasselt.

The list of synonyms or abbreviations is best defined in the set up area of your input file for maintenance purposes. You can also store this kind of information in an external file, and load the file (e.g. `abbrev.tex`) with:

```
\input abbrev.tex
```

If you want to put a list of the abbreviations used in your document you can type:

```
\placelistofabbreviations
```

or

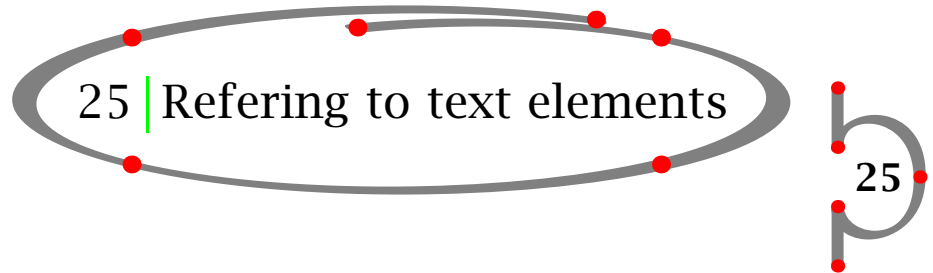
```
\completelistofabbreviations
```

A complete and sorted list with abbreviations and their meaning is produced.

The typesetting of synonyms can be influenced with:

```
\setupsynonyms
```

Look at chapter 9 for another application of synonyms.



For referring to one location in a document from another you can use the command:

```
\in{.1.}{.2.}[ref]
```

The curly braces contain text, the brackets contain a logical label. If you have written a chapter header like this:

```
\chapter[hotel]{Hotels in Hasselt}
```

then you can refer to this chapter with:

```
\in{chapter}[hotel]
```

After processing the chapter number is available and the reference could look something like: *chapter 23*. You can use `\in` for any references to text elements like chapters, sections, figures, tables, formulae etc.

Another example:

There are a number of things you can do in Hasselt:

```
\startitemize[n,packed]
```

```
\item swimming
```

```
\item sailing
```

```
\item[hiking] hiking  
\item biking  
\stopitemize
```

Activities like `\in{activity}[hiking]` described on `\at{page}[hiking]` are very tiring.

This would look like this:

There are a number of things you can do in Hasselt:

1. swimming
2. sailing
3. hiking
4. biking

Activities like activity 3 described on page 57 are very tiring.

As you can see, it is also possible to refer to pages. This is done with:

```
\at{.1.}{.2.}[ref]
```

For example with:

```
\at{page}[hiking]
```

This command can be used in combination with:

```
\pagereference[ref]
```

and

```
\textreference[ref]{...}
```

If you want to refer to the chapter *Hotels in Hasselt* you could type:

Look in `\in{chapter}[hotel]` on `\at{page}[hotel]` for a complete overview of accomodation in `\pagereference[accomodation]Hasselt`.

A chapter number and a page number will be generated when processing the input file. On another spot in the document you can refer to accomodation with `\at{page}[accomodation]`.

You can also define a set of labels separated by commas.

Color

```
\placefigure  
  [here]  
  [fig:canals,fig:boats]  
  {A characteristic picture of Hasselt.}  
  {\externalfigure[hass08g][width=5cm]}
```

There are many canals in Hasselt (see \in{figure}[fig:canals]).

.
. .
.

Boats can be moored in the canals of Hasselt (see
\in{figure}[fig:boats]).

This might look like this:



Figure 25.1 A characteristic picture of Hasselt.

There are many canals in Hasselt (see figure 25.1). . . . Boats can be moored in the canals of Hasselt (see figure 25.1).

Text can be set in color.

```
\setupcolors[...]=...]
```

The use of colors has to be activated by:

Background text

```
\setupcolor[state=start]
```

Now the basic colors are available (red, green and blue).

```
\startcolor[red]
```

```
Hasselt is a very \color[green]{colorful} town.
```

```
\stopcolor
```

Hasselt is a very colorful town.

On a black and white printer you will see only grey shades. In an electronic document these colors will be as expected.

You can define your own colors with:

```
\definecolor[...] [...,...=...,...]
```

For example:

```
\definecolor[darkred] [r=.5,g=.0,b=.0]
```

```
\definecolor[darkgreen] [r=.0,g=.5,b=.0]
```

Now the colors darkred and darkgreen are available.



27

27 | Background text

To emphasize a section you can use backgrounds. A background is set with the command pair:

```
\startbackground ... \stopbackground
```

An example can illustrate the use:

```
\setupbackground[background=screen,corner=round]
```

```
\startbackground
```

Hasselt has produced a few well known people. Only recently it turned out that Kilian van Rensselaer played a prominent role in the foundation of the State of New York.

```
\stopbackground
```

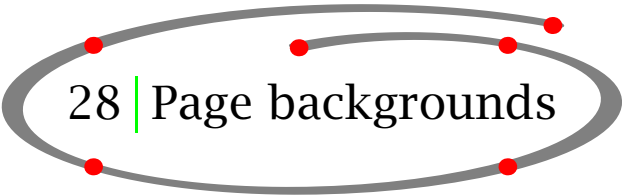
This would be displayed as:

Hasselt has produced a few well known people. Only recently it turned out that Kilian van Rensselaer played a prominent role in the foundation of the State of New York.

Backgrounds can span multiple pages. With:

`\setupbackground[...]=...,...]`

you can vary the display of the backgrounds.



The page background can also be set, with:

`\setupbackgrounds[.1.][...,.2.,...][...]=...,...]`



The first two bracket pairs are used to define the page areas. The last bracket pair is used for set up.

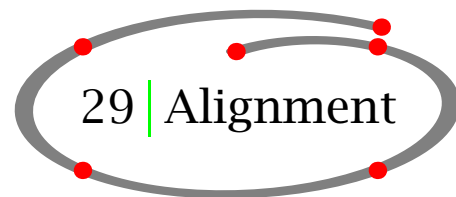
	left edge	left margin	text	right margin	right edge
top					
header					
text					
footer					
bottom					

Figure 28.1 The page areas defined in `\setupbackgrounds`.

Alignment

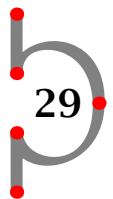
If you want to have backgrounds in the gray areas of the page layout of figure 28.1 you type:

```
\setupbackgrounds
[header,text,footer]
[leftmargin,text,rightmargin]
[background=screen]
```



Horizontal and vertical alignment can be set up with:

```
\setupalign[...]
```



Single lines can be aligned with:

```
\rightaligned{}
\leftaligned{}
\midaligned{}
\leftaligned {Hasselt was built on a sandhill.}
\midaligned {Hasselt was built on the crossing of two rivers.}
\rightaligned {Hasselt's name stems from hazelwood.}
```

After processing this would look like:

Hasselt was built on a sandhill.

Hasselt was built on the crossing of two rivers.

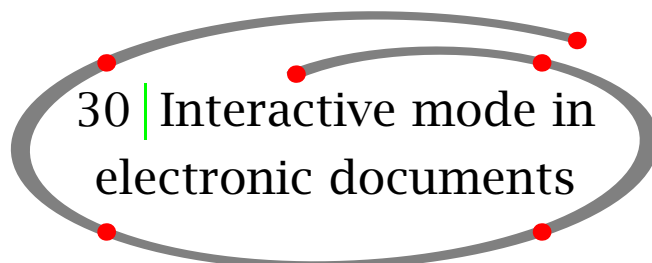
Hasselt's name stems from hazelwood.

Alignment of section is done with:

```
\startalignment ... \stopalignment[...]
```

In case of alignment you can specify a tolerance and the direction (vertical or horizontal). Normally the tolerance is `verystrict`. In columns you could specify `verytolerant`. The tolerance in this manual is:

`\setuptolerance[horizontal,verystrict]`



30.1 Introduction

Nowadays documents can be made electronically available for consulting on a computer and displaying on a computer screen.

Interaction means that you can click on active areas and jump to the indicated areas. For example if you consult a register you can click on a (active) page number and you will jump to the corresponding page.

Interaction relates to:

- active chapter numbers in table of content
- active page numbers in registers
- active page numbers, chapter numbers and figure numbers in internal references to pages, chapters, figures etc.
- active titles, page numbers, and chapter numbers in external references to other interactive documents
- active menus as navigation tools

Interactivity depends on the program you use to view the interactive document. We assume that you will use `PDFTEX` for producing a PDF document directly or use Acrobat Distiller to convert a `POSTSCRIPT` file into a PDF one. It is obvious that you will then use Acrobat Reader, Acrobat Exchange, or `GHOSTVIEW` for viewing.

`CONTEXT` is a very powerful system for producing electronic or interactive PDF documents. However only a few standard features are described in this chapter. As the authors of this manual are planning to make all `CONTEXT` related manuals electronically (sources included) available, reverse engineering is one of the options to become more acquainted with the possibilities of `CONTEXT`.

30.2 Interactive mode

The interactive mode is activated by:




```
\setupinteraction[...,...=...,...]
```

For example:

```
\setupinteraction
  [state=start,
   color=green,
   style=bold]
```

The hyper links are now generated automatically and the active words are displayed in bold green.

The interactive document is considerably bigger (in MB's) than its paper cousin because hyperlinks consume space. You will also notice that processing time becomes longer. Therefore it is advisable to de-activate the interactive mode as long as your document is under construction.

30.3 Interaction within a document

Earlier you have seen how to make a reference with `\in` and `\at`. You may have wondered why you had to type `\in{chapter}[chap:introduction]`. In the first place *chapter* and its corresponding chapter number will not be separated at line breaking. In the second place the word *chapter* and its number are typeset differently in the interactive mode. This gives the user a larger clickable area.

In interactive mode there is one other command that has little meaning in the paper variant.

```
\goto{.1.}{.2.}[ref,...]
```

The curly braces contain text, the brackets contain a reference (logical name, location).

In `\goto{Hasselt}[fig:cityplan]` all streets are build in a circular way.

In the interactive document *Hasselt* will be green and active. You will jump to a map of Hasselt.

30.4 Interaction between documents

It is possible to link one document to another. First you have to state that you want to refer to another document. This is done by:

```
\useexternaldocument[.1.][.2.][.3.]
```

The first bracket pair must contain a logical name of the document, the second pair the file name of the other document and the third pair is used for the title of the document.

For referring to these other documents you can use:

```
\from[ref]
```

The curly braces contain text and the brackets contain the reference.

Look at the example below.

```
\useexternaldocument[hia][hasbook][Hasselt in August]
```

Most tourist attractions are described in \from[hia]. A description of the Eui||feest is found in \from[hia::eui feest]. A description of the \goto{Eui||feest}[hia::eui feest] is found in \from[hia]. The eui||feest is described on \at{page}[hia::eui feest] in \from[hia]. See for more information \in{chapter}[hia::eui feest] in \from[hia].

The \useexternaldocument is usually typed in the set up area of your input file.

After processing your input file (at least two times to get the references right), and the file hasbook.tex, you will have two PDF documents. The references above have the following meaning:

- \from[hia] will produce the active title you gave in the third bracket pair of \useexternaldocument and is linked to the first page of hasbook.pdf
- \from[hia::eui feest] will produce an active title and is linked to the page where chapter Eui-feest begins
- \goto{Eui||feest}[hia::eui feest] will produce an active word *Eui-feest* and is linked to the page where chapter Eui-feest begins
- \at{page}[hia::eui feest] will produce an active word *page* and page number and is linked to that page
- \in{chapter}[hia::eui feest] will produce on active word *chapter* and chapter number and is linked to that chapter

As you can see the :: separates the (logical) file name and the destination in that file.

30.5 Menus

You can define navigation tools with:

```
\defineinteractionmenu[.1.][.2.][...,...=...,...]
```

The first bracket pair is used for a logical name that can be used to recall the menu. The second pair contains the location on the screen. The third pair is used for setting up the menu.

A typical menu definition might look like this:

```
\setupcolors
  [state=start]

\setupinteraction
  [state=start,
   menu=on]

\defineinteractionmenu
  [mymenu]
  [right]
  [state=start,
   align=middle,
   background=screen,
   frame=on,
   width=\marginwidth,
   style=smallbold,
   color=]

\setupinteractionmenu
  [mymenu]
  [{Content[content]},
   {Index[index]},
   {\vfill},
   {Stop[ExitViewer]}]
```

This will produce a menu on the right hand side of every screen. The menu buttons contain the text *Content*, *Index* and *Stop* with respectively the following functions: jump to the table of contents, jump to the index and leave the viewer. The labels to obvious destinations like *content* and *index* are predefined. Other predefined destinations are *FirstPage*, *LastPage*, *NextPage* and *PreviousPage*.

An action like *ExitViewer* is necessary to make an electronic document self containing. Other predefined actions you can use are *PrintDocument*, *SearchDocument* and *PreviousJump*. The meaning of these actions is obvious.

Menus are set up with:

```
\setupinteractionmenu[...,...,...][...,...=...,...][...,{...[ref,...]},...]
```

31 | Fonts and font switches

31.1 Introduction

The default font in `CONTEXT` is the *Computer Modern Roman* (`cmr`). You can also use *Lucida Bright* (`lbr`) as a full alternative and symbols of the *American Mathematical Society* (`ams`). Standard PostScript fonts (`pos`) are also available.

31.2 Fontstyle and size

You select the font family, style and size for a document with:

```
\setupbodyfont[...,...,...]
```

If you typed `\setupbodyfont[sansserif,9pt]` in the setup area of the input file your text would look something like this.

For changes in mid-document and on section level you should use:

```
\switchtobodyfont[...,...,...]
```

On November 10th (one day before Saint Martensday) the youth of Hasselt go from door to door to sing a special song and they accompany themselves with a `{\em foekepot}`. And they won't go away before you give them some money or sweets. The song goes like this:

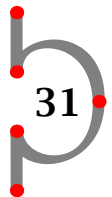
```
\startnarrower
\switchtobodyfont[small]
\startlines
Foekepotterij, foekepotterij,
```

Geef mij een centje dan ga'k voorbij.
 Geef mij een alfje dan blijf ik staan,
 'k Zak nog liever naar m'n arrenmoeder gaan.
 Hier woont zo'n rieke man, die zo vulle gèven kan.
 Gèf wat, old wat, gèf die arme stumpers wat,
 'k Eb zo lange met de foekepot elopen.
 'k Eb gien geld om brood te kopen.
 Foekepotterij, foekepotterij,
 Geef mij een centje dan ga'k voorbij.
`\stoplines`
`\stopnarrower`

Notice that `\startnarrower ... \stopnarrower` is also used as a begin and end of the fontswitch. The function of `\startlines` and `\stoplines` in this example is obvious.

On November 10th (one day before Saint Martensday) the youth of Hasselt go from door to door to sing a special song and they accompany themselves with a *foekepot*. And they won't go away before you give them some money or sweets. The song goes like this:

Foekepotterij, foekepotterij,
 Geef mij een centje dan ga'k voorbij.
 Geef mij een alfje dan blijf ik staan,
 'k Zak nog liever naar m'n arrenmoeder gaan.
 Hier woont zo'n rieke man, die zo vulle gèven kan.
 Gèf wat, old wat, gèf die arme stumpers wat,
 'k Eb zo lange met de foekepot elopen.
 'k Eb gien geld om brood te kopen.
 Foekepotterij, foekepotterij,
 Geef mij een centje dan ga'k voorbij.



If you want an overview of the available font family you can type:

`\showbodyfont[cmr]`

[cmr]													
	<code>\tf</code>	<code>\sc</code>	<code>\sl</code>	<code>\it</code>	<code>\bf</code>	<code>\bs</code>	<code>\bi</code>	<code>\tfx</code>	<code>\tfxx</code>	<code>\tfa</code>	<code>\tfb</code>	<code>\tfc</code>	<code>\tfd</code>
<code>\rm</code>	Ag	AG	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
<code>\ss</code>	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
<code>\tt</code>	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

31.3 Style and size switch in commands

In a number of commands one of the parameters is `character` to indicate the desired typestyle. For example:

`\setuphead[chapter][style=\tfd]`

In this case the character size for chapters is indicated with a command `\tfd`. But instead of a command you could use the predefined options that are related to the actual typeface:

```
normal bold slanted boldslanted type mediaeval
small smallbold smallslanted smallboldslanted smalltype
capital cap
```

31.4 Local font style and size

In the running text (local) you can change the *typestyle* into roman, sans serif and teletype with `\rm`, `\ss` and `\tt`.

You can change the *typeface* like italic and boldface with `\sl` and `\bf`.

The *typesize* is available from 4pt to 12pt and is changed with `\switchtobodyfont`.

The actual style is indicated with `\tf`. If you want to change into a somewhat greater size you can type `\tfa`, `\tfb`, `\tfc` and `\tfd`. An addition of a, b, c and d to `\sl`, `\it` and `\bf` is also allowed.

```
{\tfc Mintage}
```

```
In the period from {\tt 1404} till {\tt 1585} Hasselt had its own
{\sl right of coinage}. This right was challenged by other cities,
but the {\switchtobodyfont[7pt] bishops of Utrecht} did not honour these
{\slb protests}.
```

The curly braces indicate begin and end of style or size switches.

Mintage

In the period from 1404 till 1585 Hasselt had its own *right of coinage*. This right was challenged by other cities, but the bishops of Utrecht did not honour these *protests*.

31.5 Redefining fontsize

For special purposes you can define your own fontsize.

```
\definebodyfont[...1,...][.2.][...,...=,...]
```

A definition could look like this:

```
\definebodyfont[10pt][rm][tfe=1br at 36pt]
{\tfe Hasselt!}
```

Now `\tfe` will produce 36pt characters saying:

Hasselt!

31.6 Small caps

Abbreviations like PDF (Portable Document Format) are printed in pseudo small caps. A small capital is somewhat smaller than the capital of the actual typeface. Pseudo small caps are produced with:

[setup cap is undefined]

If you compare PDF, `\cap{PDF}` and `\sc PDF` :

PDF and PDF and PDF

you can see the difference. The command `\sc` shows the real small caps. The reason for using pseudo small caps instead of real small caps is just a matter of taste.

31.7 Emphasized

To emphasize words consistently throughout your document you use:

`\em`

Emphasized words appear in a slanted style.

If you walk through Hasselt you should `{\bf \em watch out}` for `{\em Amsterdammers}`. An `{\em Amsterdammer}` is `{\bf \em not}` a person from Amsterdam but a little stone pilar used to separate sidewalk and road. A pedestrian should be protected by these `{\em Amsterdammers}` against cars but more often people get hurt from tripping over them.

If you walk through Hasselt you should ***watch out*** for *Amsterdammers*. An *Amsterdammer* is ***not*** a person from Amsterdam but a little stone pilar used to separate sidewalk and road. A pedestrian should be protected by these *Amsterdammers* against cars but more often people get hurt from tripping over them.

*An emphasize within an emphasize is normal again and a boldface emphasize looks like **this or this**.*

31.8 Teletype / verbatim

If you want to display typed text and want to keep your line breaking exactly as it is you use

`\starttyping ... \stoptyping`

In the text you can use:

```
\type{...}
```

The curly braces enclose the text you want in teletype. You have to be careful with `\type` because the line breaking mechanism does not work anymore.

You can set up the ‘typing’ with:

```
\setuptyping[...][...,...=...,...]
```

```
\setuptype[...=...]
```

32 | Composite characters

32

32.1 Introduction

In chapter 3 you have already seen that you have to type more than one token to obtain special characters like `#` `$` `%` `&` `_` `{` and `}`. Accented characters or foreign symbols have to be composed to produce the right typesetted character.

It is not within the scope of this manual to go into accented characters in math mode. See the \TeX Book by Donald E. Knuth on that subject.

32.2 Accents

Accented characters have to be composed in `CONTEXT`. Table 32.1 shows you how to do this. The character *u* is just an example here.

You don’t want `î` or `ĵ` so for an accented *i* and *j* you compose the characters as follows:

```
\{"\i}  î
```

```
\^{\j}  ĵ
```


You type	You get	You type	You get
\‘{u}	ù	\u{u}	ŭ
\’{u}	ú	\v{u}	ű
\^{u}	û	\H{u}	ẏ
\"{u}	ü	\t{uu}	üü
\~{u}	ũ	\c{u}	ұ
\={u}	ū	\d{u}	ұ
\.{u}	u	\b{u}	u

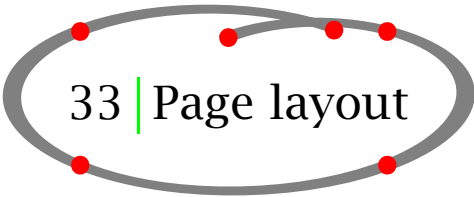
Table 32.1 Accents.

You type	You get	You type	You get
\oe	œ	\O	Ø
\OE	Œ	\l	ł
\ae	æ	\L	Ł
\AE	Æ	\SS	ß
\aa	å	?‘	ı
\AA	Å	!‘	ı
\o	ø		

Table 32.2 Foreign characters.

32.3 Foreign symbols

The composition of characters that appear in foreign languages is shown in table 32.2.



The page layout is defined by:

```
\setuplayout[...]=...]
```

Page layout

This command is typed in the set up area of your input file.

You have to familiarize yourself with the parameters that describe your page layout. A page is divided in a number of areas such as the text, margin, head, foot etc. The size of these areas can be set with `\setuplayout`.

The different areas in the page layout are shown in figure 33.1.

	left edge	left margin	text text	right margin	right edge
top					
header					
text					
footer					
bottom					

Figure 33.1 The page layout.

If you want to look at your page layout you can type the command `\showframe` and process one page or the whole file. The areas are shown in a number of frames.

The command `\showsetups` shows the values of the parameters. A combination of both commands is `\showlayout`.

The values of the layout parameters are available as commands (see table 33.2). This enables you to work more accurately when defining measures of columns, figures and tables. A few of these parameters are explained in table 33.1.

Commands	Meaning
<code>\makeupwidth</code>	width of the typing area
<code>\makeupheight</code>	height of the typing area
<code>\textwidth</code>	width of the text area
<code>\textheight</code>	height of the text area

Table 33.1 A few parameters as commands.

If you want to define the width of a column or the height of a figure you can do it relative to the `\makeupwidth` or `\makeupheight`. Changes in page width or page height will alter columns and figures proportionally.

```
\placefigure
[here]
[fig:steppable]
{A steppable.}
{\externalfigure[hass19g][width=.6\textwidth]}
```

After processing this would become:



Figure 33.2 A steppable.

The other distances and measures are shown in table 33.2.

The command `\setuplayout` is defined in the set up area of your input file, before the `\starttext` command. This means that the values of the parameters have a global effect and are default throughout the document. Nevertheless you might want to make slight changes in the page design for a number of pages.

```
\adaptlayout[21,38][height=+.5cm]
```

In this case page 21 and 38 have a height of .5 cm + default height.

For local changes in the page layout you can use:

```
\startlocal ... \stoplocal
```

For example:

```
\start
\startlocal
\setuplayout[height=+.5cm]
\stoplocal
```

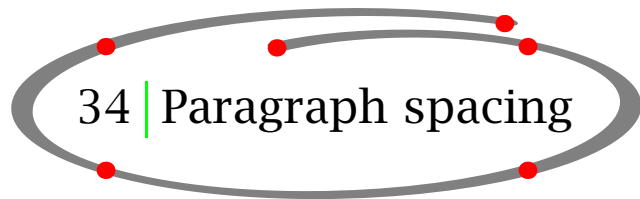
Hasselt has a completely different layout than most other cities because of its fortifications and moats.

`\stop`

It is advisable not to use these local changes to often. It is always better to alter the text than to change the page layout.

<code>topdistance</code>	<code>\topheight</code>
<code>topheight</code>	<code>\topheight</code>
<code>headerdistance</code>	<code>\headerdistance</code>
<code>headerheight</code>	<code>\headerheight</code>
<code>topspace</code>	<code>\topspace</code>
<code>headlevel</code>	<code>\headerlevel</code>
<code>backspace</code>	<code>\backspace</code>
<code>margindistance</code>	<code>\margindistance</code>
<code>marginwidth</code>	<code>\marginwidth</code>
<code>leftmarginwidth</code>	<code>\leftmarginwidth</code>
<code>rightmarginwidth</code>	<code>\rightmarginwidth</code>
<code>edgedistance</code>	<code>\edgedistance</code>
<code>edgewidth</code>	<code>\edgewidth</code>
<code>leftedgewidth</code>	<code>\leftedgewidth</code>
<code>rightedgewidth</code>	<code>\rightedgewidth</code>
<code>paperheight</code>	<code>\paperheight</code>
<code>paperwidth</code>	<code>\paperwidth</code>
<code>makeupheight</code>	<code>\makeupheight</code>
<code>makeupwidth</code>	<code>\makeupwidth</code>
<code>textheight</code>	<code>\textheight</code>
<code>textwidth</code>	<code>\textwidth</code>
<code>footerdistance</code>	<code>\footerdistance</code>
<code>footerheight</code>	<code>\footerheight</code>
<code>bottomdistance</code>	<code>\bottomdistance</code>
<code>bottomheight</code>	<code>\bottomheight</code>

Table 33.2 Parameters for page design.



34.1 Introduction

In `TEX` and `CONTEXT` the most important unit of text is the paragraph. You can start a new paragraph by:

- an empty line
- the `TEX` command `\par`

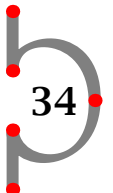
In your ASCII input file you should use empty lines as paragraph separators. This will lead to a clearly structured and well organized file and will prevent mistakes.

In situations where a command has to be closed explicitly you should use `\par`.

During one of the wars Hasselt lay under siege. After some time the city was famine stricken, everything edible was eaten. Except for one cow. The cow was kept alive and treated very well. `\par`

Once a day the citizens of Hasselt took the cow for a walk on the ramparts. The besiegers saw the well fed cow and became very discouraged. They broke up their camps and Hasselt was saved. `\par`

In the Hoogstraat in Hasselt there is a stone tablet with a representation of the cow that commemorates the siege and the wisdom of the citizens of Hasselt.



This could also be typed without `\pars` and a few empty lines.

During one of the wars Hasselt lay under siege. After some time the city was famine stricken, everything edible was eaten. Except for one cow. The cow was kept alive and treated very well.

Once a day the citizens of Hasselt took the cow for a walk on the ramparts. The besiegers saw the well fed cow and became very discouraged. They broke up their camps and Hasselt was saved.

In the Hoogstraat in Hasselt there is a stone tablet with a representation of the cow that commemorates the siege and the wisdom of the citizens of Hasselt.

34.2 Inter paragraph spacing

The vertical spacing between paragraphs can be specified by:



```
\setupwhitespace[...]
```

This document is produced with `\setupwhitespace[medium]`.

When inter paragraph spacing is specified there are two commands available that are seldom needed:

```
\nowhitespace  
\whitespace
```

When a paragraph consists of a horizontal line or a table like this one:

City	Area code
Hasselt	8060 - 8061

Sometimes spacing is suboptimal. For that purpose you could carry out a correction with:

```
\startlinecorrection ... \stoplinecorrection
```

So if you would type:

```
\startlinecorrection  
\framed{something framed}  
\stoplinecorrection
```

you will get a better output. Only use these commands if really needed!

something framed

Another command to deal with vertical spacing is:

```
\blank[...,...,...]
```

The bracket pair is optional and within the bracket pair you can type the amount of spacing. Keywords like `small`, `medium` and `big` are related to the fontsize.

In official writings Hasselt always has the affix `Ov`. This is an abbreviation for the province of `{\em Overijssel}`.

```
\blank[2*big]
```

The funny thing is that there is no other Hasselt in the Netherlands. So it is redundant.

```
\blank
```

Paragraph spacing

The affix is a leftover from the times that the Netherlands and Belgium were one country under the reign of King Philip II of Spain.

`\blank[2*big]`

Hasselt in Belgium lies in the province of Limburg. One wonders if the Belgian people write Hasselt (Li) on their letters.

The command `\blank` without the bracket pair is the default space.

The example would become:

In official writings Hasselt always has the affix Ov. This is an abbreviation for the province of *Overijssel*.

The funny thing is that there is no other Hasselt in the Netherlands. So it is redundant.

The affix is a leftover from the times that the Netherlands and Belgium were one country under the reign of King Philip II of Spain.

Hasselt in Belgium lies in the province of Limburg. One wonders if the Belgian people write Hasselt (Li) on their letters.

The default spacing can be set up with:

```
\setupblank[...]
```

If you want to suppress vertical spacing you can use:

```
\startpacked[...] ... \stoppacked
```

```
\defineparagraphs[city][n=2,before=,after=]
```

```
\city Hasselt (Ov) \\ Overijssel \\
```

```
\city Hasselt (Li) \\ Limburg    \\
```

```
\startpacked
```

```
\city Hasselt (Ov) \\ The Netherlands \\
```

```
\city Hasselt (Li) \\ Belgium        \\
```

```
\stoppacked
```

Hasselt (Ov)

Overijssel

Hasselt (Li)

Limburg

Hasselt (Ov)

The Netherlands

Hasselt (Li)

Belgium

It is not hard to imagine why there is also:

```
\startunpacked ... \stopunpacked
```

You can force vertical space with `\godown`. The distance is specified within the brackets.

```
\godown[...]
```

34.3 Indentation

You can set up the amount of the indentation with:

```
\setupindenting[...,...,...]
```

A reasonable indentation is achieved by:

```
\setupindenting[medium]
```

This will lead to indented paragraphs. By default, indentation after white space (as issued by `\blank`) is suppressed.

You can locally influence the indentation state by using

```
\indenting[...,...,...]
```

When for instance you say `never`, from that moment on indentation will be suppressed. Saying `geen`, only influences the next paragraph.

If you choose to use indentations, and at a certain place you explicitly *do not* want to indent, you can also say:

```
\noindenting
```


35 | Setup commands

Global commands are placed in the setup area of your input file, before `\starttext`. In appendix D there is a complete overview of the available commands and their parameters.

The set up commands all have the same structure. They look something like:

```
\setupparagraphs[.1.][.2.][...,...=...,...]
.1.      name
.2.      number each
style    normal bold slanted boldslanted type cap small... command
width    dimension
height   dimension
align    left right middle breedte
tolerance verystrict strict tolerant verytolerant
distance dimension
before   command
after    command
inner    command
command  command
rule     on off
```

A set up command consist of a more or less logical name and a number of bracket pairs. Bracket pairs may be optional and in that case the `[]` are typeset slanted `[]`. In the definition the bracket pairs may contain:

```
\setupacommand[.1.][.2.][...,...=...,...]
```

The commas indicate that a list of parameters can be enclosed. In the options list following the definition, the `.1.` and `.2.` show the possible options that can be set in the first and second bracket pair respectively. The parameters and their possible values are placed in the third bracket pair.

The default options and parameter values are underlined. Furthermore you will notice that some values are typeset in a slanted way: *section*, *name*, *dimension*, *number*, *command* and *text*. This indicates that you can set the value yourself.

<i>section</i>	a section name like chapter, paragraph etc.
<i>name</i>	an identifier (logical name)
<i>dimension</i>	an dimension with a unit in cm, pt, em, ex, sp or in
<i>number</i>	an integer

command a command
text text

36 | Defining commands / macros

CON_TEXT is a set of macros based on T_EX. T_EX is a programming language as well as a typographical system. This means that you can do the programming yourself if you need that kind of flexibility.

You can define a new command with:

```
\define[.1.]\command{.2.}
```

Without any further explanation this command is used in the following example.

You may have a well illustrated document and you are tired of typing:

```
\placefigure
  [here,force]
  [fig:logical name]
  {Caption.}
  {\externalfigure[filename][width=5cm]}
```

You could define your own command with a few variables like:

- logical name
- caption
- file name

Your command definition and call could look something like this:

```
\define[3]\myputfigure
  {\placefigure
    [here,force][fig:#1]
    {#2}\externalfigure[#3][width=5cm]}}

\myputfigure{lion}{The Dutch lion is a sentry.}{hass13g}
```

Between brackets [3] indicates that you want to use three variables #1, #2 and #3. In the command call \myputfigure you have to input these variables between curly braces. The result would be:



Figure 36.1 The Dutch lion is a sentry.

Very sophisticated commands can be programmed but this is left to your own inventiveness.

In addition to defining commands you can also define `\start ... \stop` command pairs.

```
\definestartstop[...] [..., ..=..., ...]
```

For example:

```
\definestartstop
[stars]
[commands={\inleft{\hbox to \leftmarginwidth
               {\leaders\hbox{$\star$}\hfill}}},
 before=\blank,
 after=\blank]

\startstars
{\em Hasselter Juffers} are sweet cookies but the name is no
coincidence. On July 21 in 1233 the {\em Zwartewaterklooster}
(Blackwater Monastery) was founded. The monastery was meant
for unmarried girls and women belonging to the nobility of
Hasselt. These girls and women were called {\em juffers}.
\stopstars
```

This will result in:

***** *Hasselter Juffers* are sweet cookies but the name is no coincidence. On July 21 in 1233 the *Zwartewaterklooster* (Blackwater Monastery) was founded. The monastery was meant for

unmarried girls and women belonging to the nobility of Hasselt. These girls and women were called *juffers*.

37 | Miscellaneous

37.1 Floating blocks

A block in `CONTEXT` is a text element, for example a table or a figure that you can process in a special way. You have already seen the use of `\placefigure` and `\placetable`. These are both examples of floating blocks. The floating mechanism is described in chapter 10 and 11.

You can define these kind of blocks yourself with:

```
\definefloat[.1.][.2.]
```

The bracket pairs are used for the name in singular and plural form. For example:

```
\definefloat[intermezzo][intermezzi]
```

Now the following commands are available:

```
\placeintermezzo[] [] {} {}
\startintermezzotext ... \stopintermezzotext
\placelistofintermezzi
\completelistofintermezzi
```

The newly defined floating block can be set up with:

```
\setupfloat[...][...,...=...,...]
```

You can set up the layout of floating blocks with:

```
\setupfloats[...=...,...]
```

You can set up the numbering and the labels with:

```
\setupcaption[...] [...,...=...,...]
```

These commands are typed in the set up area of your input file and will have a global effect on all floating blocks.

```
\setupfloats[location=middle]
\setupcaption[location=bottom,headstyle=boldslanted]
```

```
\placeintermezzo{An intermezzo.}
```

```
\startframedtext
```

At the beginning of this century there was a tramline from Zwolle to Blokzijl via Hasselt. Other means of transport became more important and just before the second world war the tramline was stopped.

Nowadays such a tramline would have been very profitable.

```
\stopframedtext
```

At the beginning of this century there was a tramline from Zwolle to Blokzijl via Hasselt. Other means of transport became more important and just before the second world war the tramline was stopped. Nowadays such a tramline would have been very profitable.

Intermezzo 37.1 An intermezzo.

37.2 Text blocks

Another type of block is a text block. A text block for example is one or more paragraphs you want to use several times.

You have to define a text block with:

```
\defineblock[...]
```

You give the name of text blocks between brackets; you can also type a list of names if you separate them by commas. For example you can define:

```
\defineblock[dutch]
```

After defining the text block the following command is available:

`\begin dutch ... \end dutch`

Text blocks are manipulated with:

`\hideblocks[...1,...][...2,...]`

`\useblocks[...1,...][...2,...]`

`\keepblocks[...1,...][...2,...]`

`\selectblocks[...1,...][...2,...][...=...]`

An example shows the possibilities of text blocks.

`\defineblock[dutch,english]`

`\hideblocks[dutch,english]`

`\begin english[dedemsvaart-e]`

After 1810 the Dedemsvaart brought some prosperity to Hasselt. All ships went through the canals of Hasselt and the shops on both sides of the canals prospered.

`\end english`

`\begin dutch[dedemsvaart-d]`

Sinds 1810 veroorzaakte de Dedemsvaart enige welvaart in Hasselt. Alle schepen voeren door de grachten en de winkels aan weerszijden van de gracht floreerden.

`\end dutch`

`\useblocks[english][dedemsvaart-e]`

After 1810 the Dedemsvaart brought some prosperity to Hasselt. All ships went through the canals of Hasselt and the shops on both sides of the canals prospered.

If you continue defining these blocks you could make a bilingual manual. For that purpose it is also possible to store the text blocks in an external file. This would look something like this:

`\setupblock[dutch][file=store-d]`

The Dutch text blocks are stored in `store-d.tex` and the text fragments can be called upon by their logical names.

37.3 Storing text for later use

You can store information temporarily for future use in your document with:

```
\startbuffer[...] ... \stopbuffer
```

For example:

```
\startbuffer[visit]
```

If you want to see what Hasselt has in store you should come and visit it some time.

```
\stopbuffer
```

```
\getbuffer[visit]
```

With `\getbuffer[visit]` you recall the stored text. The logical name is optional. With `\typebuffer[visit]` you get back the typeset version of the content of the buffer.

Buffers are set up with:

```
\setupbuffer[...] [..., ...=..., ...]
```

37.4 Hiding text

Text can be hidden with:

```
\starthiding ... \stophiding
```

The text in between will not be processed.

37.5 Lines

There are many commands to draw lines. For a single line you type:

```
\hairline
```

or:

```
\thinrule
```

For more lines you type:

```
\thinrules[...=...]
```

Text in combination with lines is also possible:

— **Hasselt – Amsterdam** —
 If you draw a straight line from Hasselt to Amsterdam you would have to cover a distance of almost 145 km.

If you draw two straight lines from Hasselt to Amsterdam you would have to cover a distance of almost 290 km.

Amsterdam _____

 _____ Hasselt

```
\starttextrule{Hasselt -- Amsterdam}
```

```
If you draw a straight line from Hasselt to Amsterdam you would have  
to cover a distance of almost 145 \Kilo \Meter.
```

```
\stoptextrule
```

```
If you draw two straight lines from Hasselt to Amsterdam you would  
have to cover a distance of almost 290 \Kilo \Meter.
```

```
Amsterdam \thinrules[n=3] Hasselt
```

You always have to be careful in drawing lines. Empty lines around `\thinrules` must not be forgotten and the vertical spacing is always a point of concern.

You can set up line spacing with:

```
\setupthinrules[...=...]
```

There are a few complementary commands that might be very useful.

```
\setupfillinrules[...=...=...]
```


`\setupfillinlines[...]=...]`

These commands are introduced in the examples below:

```
\setupfillinrules[width=2cm]
\setupfillinlines[width=3cm]
\fillinrules[n=1]{\bf name}
\fillinrules[n=3]{\bf adress}
\fillinline{Can you please state the \underbar{number} of cars used
            in your family.} \par
Strike out any word \overstrikes{in this text}\periods[18]
```

This will become:

name _____
adress _____

Can you please state the number of cars used in your family. _____

Strike out any word ~~in this text~~.....

These commands are used in questionnaires. Text that is struck out or underlined will not be hyphenated.

37.6 Super- and subscript in text

It is very easy to put ^{superscript} and _{subscript} in your text. What would you call this version ^{superscript}_{subscript}? It looks strange!

This ugly text was made with `\low{}`, `\high{}` and `\lohi{}`. The text was placed between the curly braces.

37.7 Date

You can introduce the system date in your text with:

`\currentdate`

37.8 Positioning

Sometimes you feel the need to position text on a page or within a text element. You can position text with:

```
\position(.1,.2){.3}
```

The parenthesis enclose the x, y coordinates, the curly braces enclose the text you want to position.

You can set up the x, y axes with:

```
\setuppositioning[...=...]
```

You can use units and scaling factors. An example will illustrate `\position`.

```
\def\dicefive%
{\framed
 [width=42pt,height=42pt,offset=0pt]
 {\setuppositioning
  [unit=pt,factor=12,xoffset=-11pt,yoffset=-8pt]%
  \startpositioning
   \position(1,1){$\bullet$}%
   \position(1,3){$\bullet$}%
   \position(2,2){$\bullet$}%
   \position(3,1){$\bullet$}%
   \position(3,3){$\bullet$}%
  \stoppositioning}}
\placefigure{This is five.}{\dicefive}
```

This is a rather complex example but it would look something like this.

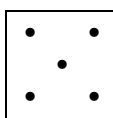


Figure 37.1 This is five.

37.9 Rotating text, figures and tables

In a number of cases you would like to rotate text or figures. You can rotate text and objects with:

```
\rotate[...=...]{...}
```

The first bracket pair is optional. Within that bracket pair you specify the rotation: `rotation=90`. The curly braces contain the text or object you want to rotate.

Hasselt got its municipal rights in 1252. From that time on it had the `\rotate[rotation=90]{right}` to use its own seal on official documents. This seal showed Holy Stephanus known as one of the first Christian martyrs, and was the `\rotate[rotation=270]{patron}` of Hasselt. After the Reformation the seal was redesigned and Stephanus lost his ‘holiness’ and was from that time on depicted without his aureole.

This results in a very ugly paragraph:

Hasselt got its municipal rights in 1252. From that time on it had the ^{right} to use its own seal on official documents. This seal showed Holy Stephanus known as one of the first Chris-
^{patron} tian martyrs, and was the of Hasselt. After the Reformation the seal was redesigned and Stephanus lost his ‘holiness’ and was from that time on depicted without his aureole.

You can rotate a figure just as easily:

```
\placefigure
{The 180 \Degrees\ rotated fishing port (de Vispoort).}
\rotate[rotation=180]{\externalfigure[hass15g][width=4cm]}
```

It is not always clear what you get when you rotate.



Figure 37.2 The 180° rotated fishing port (de Vispoort).

We can set up rotating with:

```
\setuprotate[...]=...,...
```

37.10 Carriage return

A new line can be enforced with:

```
\crlf
```

When a number of lines should be followed by *carriage return and line feed* you can use:

```
\startlines ... \stoplines
```

```
\startlines
```

```
.  
.   
.
```

```
\stoplines
```

On a wooden panel in the town hall of Hasselt you can read:

```
\startlines
```

```
Heimelijcken haet
```

```
eigen baet
```

```
jongen raet
```

```
Door diese drie wilt verstaen
```

```
is het Roomsche Rijck vergaen.
```

```
\stoplines
```

This little rhyme contains a warning for the magistrates of Hasselt: don't allow personal benefits or feelings to influence your wisdom in decision making.

On a wooden panel in the town hall of Hasselt you can read:

```
Heimelijcken haet
```

```
eigen baet
```

```
jongen raet
```

```
Door diese drie wilt verstaen
```

```
is het Roomsche Rijck vergaen.
```

This little rhyme contains a warning for the magistrates of Hasselt: don't allow personal benefits or feelings to influence your wisdom in decision making.

In a few commands new lines are generated by `\\`. For example if you type `\inmargin{in the\\margin}` then the text will be divided over two lines.

37.11 Hyphenation

When writing multi-lingual texts you have to be aware of the fact that hyphenation may differ from one language to another.

To activate a language you type:

```
\language[...]
```

Between the brackets you fill in nl, fr, en, de and sp.

To change from one language to another you can use the shorthand versions:

```
\nl \en \de \fr \sp
```

If you want to know more about Hasselt, the best book to read is probably
`{\nl \em Uit de geschiedenis van Hasselt}` by F.~Peereboom.

If you want to know more about Hasselt, the best book to read is probably *Uit de geschiedenis van Hasselt* by F. Peereboom.

If a word is wrongly hyphenated you can define points of hyphenation yourself. This is done in the set up area of your input file:

```
\hyphenation{his-to-ry}
```

37.12 Comment in input file

All text between `\starttext` and `\stoptext` will be processed while running `CONTEXT`. Sometimes however you may have text fragments you don't want to be processed or you want to comment on your `CONTEXT` commands.

If you precede your text with the percentage sign % it will not be processed.

```
% In very big documents you can use the command input for
% different files.
%
% For example:
%
% \input hass01.tex % chapter 1 on Hasselt
% \input hass02.tex % chapter 2 on Hasselt
% \input hass03.tex % chapter 3 on Hasselt
```

When you delete the % before `\input` the three files will be processed. The comment describing the contents of the files will not be processed.

37.13 Input of another tex file

In a number of situations you may want to insert other `TEX` files in your input file. For example, sometimes it is more efficient to specify `CONTEXT` sources in more than one file in order to be able to partially process your files.

Another file (with the name `another.tex`) can be inserted by:

```
\input another.tex
```

The extension is optional so this will work too:

```
\input another
```

The command `\input` is a \TeX command.




38 | Using modules

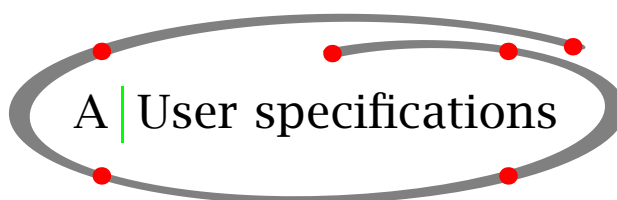
For reasons of efficiency the author decided to implement some functionality of \CONTEXT by means of external modules. At this moment you can load the following modules:

- `chemic` for typesetting chemical structures
- `units` for using SI units
- `pictex` for drawing pictures (is used in conjunction with module `chemic`)

Loading is done in the set up area of your input file and done by means of:



```
\usemodule[...,...,...]
```



When `CONTEXT` is run a number of predefined parameters is loaded. These parameters are set up in the file `cont-sys.tex`. Users can define their own preferences (housestyle) in this file. Be aware of the fact that `CONTEXT` has to be able to find this file. The readme file that goes with the distribution tells some more about site specific setups. The most important addition to this file probably concerns the output:

```
\setupoutput[pdftex]
```

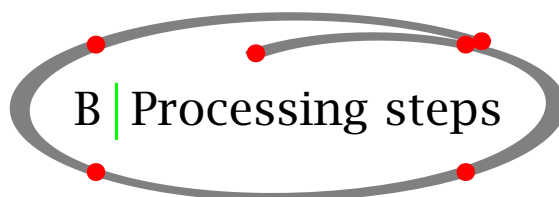
tells `CONTEXT` to produce PDF output instead of DVI, while

```
\setupoutput[dvipsonsone,dviwindo]
```

sets things up for those programs. By default DVIPS output is set up.



Processing steps



During processing `CONTEXT` writes information in the file `myfile.tui`. This information is used in the next pass. Part of this information is processed by the program `TEXUTIL`. Information on registers and lists are written in the file `myfile.tuo`. The information in this file is filtered and used (when necessary) by `CONTEXT`.

```
texutil --references filename
```

When `CONTEXT` cannot find a figure, you can generate an figure auxilliary file by saying:

```
texutil --figures *.*
```

or whatever specification suits.

When one wants to convert EPS illustrations to PDF one, there is:

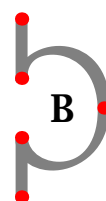
```
texutil --figures --epspage --epspdf
```

One can use `TEXEXEC` to run `CONTEXT`:

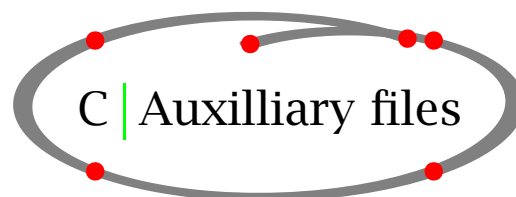
```
texexec filename
```

runs as many times as needed to get the references straight. One can also specify specific needs on the command line, for instance if two up formatting is needed or pdf code is output.

When in doubt, say `--help` and you get all the information needed to proceed.



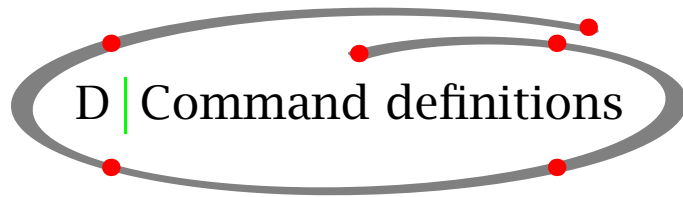
Auxilliary files



CONTEXT will produce a number of auxilliary files during processing. If your input file is called `myfile.tex` the following files may appear on your working directory.

File	Content	State
<code>myfile.tex</code>	text	do not remove
<code>myfile.tui</code>	input information	can be removed
<code>myfile.tuo</code>	output information	do not remove
<code>myfile.tub</code>	block information	do not remove
<code>myfile.tmp</code>	buffer information	can be removed
<code>texutil.tuf</code>	figure information	generate if needed
<code>myfile.dvi</code>	typeset text	can be removed
<code>myfile.ps</code>	printable text	can be removed
<code>myfile.pdf</code>	printable text	can be removed

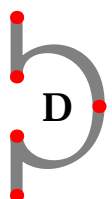




Here we summarize the commands we introduced in the previous chapters. This is just a selection of the whole repertoire of `CONTEXT` commands. Those who want to see them all can take a look at the more extensive manual.

Arguments that are typeset *slanted* are optional and can be omitted. The number points to the page where the command is explained. Black arrows indicate that the command is only of use in interactive documents and gray arrows tell us that additional functionality is provided in interactive mode. Keep in mind that we only show the commands we described in this manual, there are many more.

<code>\at{.1.}{.2.}[ref]</code>	◀ ▶ 57
.1. <i>text</i>	
<code>\blank[...,...,...]</code>	76
... <i>n*small n*medium n*big nowhite back white disable force reset line halfline formula fixed flexible</i>	
<code>\chapter[ref,...]{...}</code>	8
... <i>text</i>	
<code>\crlf</code>	90
<code>\define[.1.]\command{.2.}</code>	80
.1. <i>number</i>	



`\defineblock[...]`

83

... *name*

`\definebodyfont[...,.1,...][.2.][...,...=...,...]`

68

.1. 5pt ... 12pt default
 .2. rm ss tt mm hw cg
 tf *file*
 bf *file*
 sl *file*
 it *file*
 bs *file*
 bi *file*
 sc *file*
 ex *file*
 mi *file*
 sy *file*
 ma *file*
 mb *file*
 mc *file*

`\definecolor[...][...,...=...,...]`

59

... *name*
 r *text*
 g *text*
 b *text*
 c *text*
 m *text*
 y *text*
 k *text*
 s *text*

`\definecombinedlist[.1.][...,2,...][...,...=...,...]`

51

.1. *name*
 .2. *list*
 ..=.. see p 52: \setupcombinedlist



Command definitions

`\definedescription[...][...==,...]` 41

... *name*
 ..=.. see p 42: `\setupdescriptions`

`\defineenumeration[...1,...][.2.][...==,...]` 43

.1. *name*
 .2. *name*
 ..=.. see p 44: `\setupenumerations`

`\definefloat[.1.][.2.]` 82

.1. *singular name*
 .2. *plural name*

`\definehead[.1.][.2.]` 9

.1. *name*
 .2. *section*

`\defineinteractionmenu[.1.][.2.][...==,...]` ◀ ▶ 65

.1. left right top bottom *name*
 .2. left right top bottom *name*
 ..=.. see p 66: `\setupinteractionmenu`

`\defineparagraphs[...][...==,...]` 35

... *name*
 n *number*
 rule on off
 height *fit dimension*
 before *command*
 after *command*
 inner *command*
 distance *dimension*
 tolerance verystrict strict tolerant verytolerant
 align left right middle



Command definitions

`\defineregister[.1.][.2.]`

54

- .1. *singular name*
- .2. *plural name*

`\definestartstop[...] [..., ...=..., ...]`

81

- before *command*
- after *command*
- style *normal bold slanted boldslanted type cap small... command*
- commands *command*

`\definesynonyms[.1.][.2.][.3.][.4.]`

55

- .1. *singular name*
- .2. *plural name*
- .3. *command*
- .4. *command*

`\externalfigure[...] [..., ...=..., ...]`

23

- ... *file*
- ..=.. see p 22: `\useexternalfigure`

`\footnote[ref]{...}`

40

- ... *text*

`\framed[...] [..., ...=..., ...]{...}`

45

- ..=.. see p 45: `\setupframed`
- ... *text*

`\from[ref]`

◀ ▶ 64

- ... *text*



Command definitions

`\godown[...]` 78
... dimension

`\goto{.1.}{.2.}[ref,...]` ◀ ▶ 63
.1. text

`\hairline` 85

`\hideblocks[...1,...][...2,...]` 84
.1. name
.2. name

`\in{.1.}{.2.}[ref]` ◀ ▶ 56
.1. text

`\indenting[...,...]` 78
... never not no yes always first next

`\inmargin[.1.][ref]{.2.}` 47
.1. + - low
.2. text

`\keepblocks[...1,...][...2,...]` 84
.1. name
.2. all name



Command definitions

<code>\language[...]</code>	91
... <u>n</u> fr en uk de sp cz ..	
<code>\note[ref]</code>	◀ ▶ 40
<code>\page[...,..., ...]</code>	48
... <u>y</u> es markup no preference bigpreference left right disable last quadruple even odd blank empty reset	
<code>\pagereference[ref]</code>	57
<code>\placeformula[ref,...]{.1.}\$\$\$2. \$\$</code>	15
.1. <i>text</i> .2. <i>formula</i>	
<code>\placelist[...,..., ...][...,...=...,...]</code>	53
... <i>name</i> ..=.. see p 52: \setuplist	
<code>\position(.1.,.2.){.3.}</code>	88
.1. <i>number</i> .2. <i>number</i> .3. <i>text</i>	
<code>\rotate[...,...=...,...]{...}</code>	88
..=.. see p 89: \setuprotate	



`\section[ref,...]{...}`
 ... *text*

8

`\selectblocks[...1,...][...2,...][..=..]`
 .1. *name*
 .2. *name*
 criterium all *section*

84

`\setupalign[...]`

61

... width left right middle inner outer wide broad height bottom line reset

`\setupbackground[.....=...]`

60

leftoffset *dimension*
 rightoffset *dimension*
 topoffset *dimension*
 bottomoffset *dimension*
 state start stop
 ..=.. see p 45: \setupframed

`\setupbackgrounds[.1.][...2,...][.....=...]`

60

.1. top header text footer bottom page leftpage rightpage
 .2. leftedge leftmargin text rightmargin rightedge
 state start stop repeat
 command *command*
 ..=.. see p 45: \setupframed

`\setupblank[...]`

77

... normal standard line *dimension* big medium small fixed flexible



Command definitions

`\setupbodyfont[...,...]`

66

... *name* serif regular roman sans support sansserif mono type teletype
handwritten calligraphic 5pt ... 12pt

`\setupbuffer[...][...,...=,...]`

85

... *name*
before *command*
after *command*

`\setupcaption[...][...,...=,...]`

83

..=.. see p 24: `\setupcaptions`

`\setupcaptions[...,...=,...]`

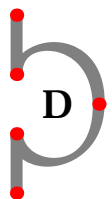
24

location top bottom none high low middle
width fit max *dimension*
headstyle normal bold slanted boldslanted type cap small... *command*
style normal bold slanted boldslanted type cap small... *command*
number yes no
inbetween *command*
align left middle right no
conversion numbers characters Characters romannumerals Romannumerals
way bytext bysection

`\setupcolors[...,...=,...]`

58

state start stop global local
conversion yes no always
reduction yes no
rgb yes no
cmyk yes no



Command definitions

`\setupcolumns[...]=...`

40

`n` *number*
`rule` *on* *off*
`height` *dimension*
`tolerance` *verystrict* *strict* *tolerant* *verytolerant*
`distance` *dimension*
`balance` *yes* *no*
`align` *yes* *no* *text*
`blank` *fixed* *halfline* *line* *flexible* *big* *medium* *small*
`option` *background*
`..=..` see p 45: `\setupframed`

`\setupcombinedlist[...][...]=...`

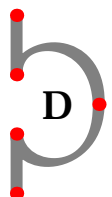
52

`...` *name*
`level` 1 2 3 4 *section* *current*
`..=..` see p 52: `\setuplist`

`\setupdescriptions[.....][...]=...`

42

`...` *name*
`headstyle` *normal* *bold* *slanted* *boldslanted* *type* *cap* *small*... *command*
`style` *normal* *bold* *slanted* *boldslanted* *type* *cap* *small*... *command*
`color` *name*
`width` *fit* *broad* *dimension*
`distance` *dimension*
`sample` *text*
`text` *text*
`align` *left* *middle* *right*
`margin` *standard* *yes* *no* *dimension*
`location` *left* *right* *top* *serried* *inmargin* *inleft* *inright* *hanging*
`hang` *fit* *broad* *number*
`before` *command*
`inbetween` *command*
`after` *command*
`indentnext` *yes* *no*



Command definitions

`\setupenumerations[...][...=...]` 44

<code>...</code>	<i>name</i>
<code>..=..</code>	see p 41: <code>\definedescription</code>
<code>location</code>	left right <u>top</u> serried inmargin inleft inright hanging
<code>text</code>	<i>text</i>
<code>levels</code>	<i>number</i>
<code>conversion</code>	<u>numbers</u> characters Characters romannumerals Romannumerals
<code>way</code>	<u>bytext</u> bysection
<code>blockway</code>	<u>yes</u> no
<code>sectionnumber</code>	yes no
<code>separator</code>	<i>text</i>
<code>stopper</code>	<i>text</i>
<code>coupling</code>	<i>name</i>
<code>couplingway</code>	global <u>local</u>
<code>number</code>	no <i>name</i>
<code>aligntitle</code>	no <u>yes</u>

`\setupfillinlines[...=...]` 87

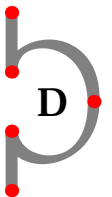
<code>width</code>	<i>dimension</i>
<code>margin</code>	<i>dimension</i>
<code>distance</code>	<i>dimension</i>
<code>before</code>	<i>command</i>
<code>after</code>	<i>command</i>

`\setupfillinrules[...=...]` 86

<code>width</code>	fit <i>dimension</i>
<code>distance</code>	<i>dimension</i>
<code>before</code>	<i>command</i>
<code>after</code>	<i>command</i>
<code>style</code>	normal bold slanted boldslanted type cap small... <i>command</i>
<code>n</code>	<i>number</i>
<code>separator</code>	<i>text</i>

`\setupfloat[...][...=...]` 82

<code>...</code>	<i>name</i>
<code>height</code>	<i>dimension</i>
<code>width</code>	<i>dimension</i>
<code>pageboundaries</code>	<i>list</i>
<code>..=..</code>	see p 45: <code>\setupframed</code>



`\setupfloats[...]=...]`

24

location	left right <u>middle</u>
width	fit <i>dimension</i>
before	<i>command</i>
after	<i>command</i>
margin	<i>dimension</i>
spacebefore	<i>n*small n*medium n*big none</i>
spaceafter	<i>n*small n*medium n*big none</i>
sidespacebefore	<i>n*small n*medium n*big none</i>
sidespaceafter	<i>n*small n*medium n*big none</i>
ntop	<i>number</i>
nbottom	<i>number</i>
nlines	<i>number</i>
..=..	see p 45: <code>\setupframed</code>

`\setupfooter[...][...]=...]`

50

...	see p 50: <code>\setupheader</code>
..=..	see p 50: <code>\setupheader</code>

`\setupfootertexts[.1.][.2.][.3.]`

49

.1.	<u>text</u> margin edge
.2.	<i>text section date mark pagenummer</i>
.3.	<i>text section date mark pagenummer</i>



Command definitions

`\setupfootnotes[...]=...]`

41

conversion	<u>numbers</u> characters Characters romannumerals Romannumerals
way	<u>bytext</u> bysection
location	<u>page</u> text columns
rule	on <u>off</u>
before	<i>command</i>
after	<i>command</i>
width	<i>dimension</i>
height	<i>dimension</i>
bodyfont	5pt ... 12pt <u>small</u> big
style	normal bold slanted boldslanted type cap small... <i>command</i>
distance	<i>dimension</i>
columndistance	<i>dimension</i>
margindistance	<i>dimension</i>
n	<i>number</i>
numbercommand	<code>\command#1</code>
..=..	see p 45: <code>\framed</code>

`\setupformulae[...]=...]`

17

location	left <u>right</u>
left	<i>text</i>
right	<i>text</i>



`\setupframed[...]=...]`

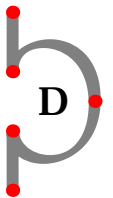
45

height	fit broad <i>dimension</i>
width	fit broad <i>dimension</i>
offset	none overlay <i>dimension</i>
location	low depth
option	<u>none</u> empty
strut	<u>yes</u> no
align	<u>no</u> left right middle normal
bottom	<i>command</i>
top	<i>command</i>
frame	<u>on</u> off
topframe	on <u>off</u>
bottomframe	on <u>off</u>
leftframe	on <u>off</u>
rightframe	on <u>off</u>
frameoffset	<i>dimension</i>
framedepth	<i>dimension</i>
framecorner	round <u>rectangular</u>
frameradius	<i>dimension</i>
framecolor	<i>name</i>
background	screen color <u>none</u>
backgroundscreen	<i>number</i>
backgroundcolor	<i>name</i>
backgroundoffset	<i>dimension</i>
backgrounddepth	<i>dimension</i>
backgroundcorner	round <u>rectangular</u>
backgroundradius	<i>dimension</i>
depth	<i>dimension</i>
corner	round <u>rectangular</u>
radius	<i>dimension</i>
...	<i>text</i>

`\setupframedtexts[...]=...]`

46

bodyfont	5pt ... 12pt small big
style	normal bold slanted boldslanted type small... <i>command</i>
left	<i>command</i>
right	<i>command</i>
before	<i>command</i>
after	<i>command</i>
inner	<i>command</i>
..=..	see p 45: <code>\setupframed</code>



`\setuphead[...][...,...=...,...]`

9

<code>...</code>	<i>section</i>
<code>style</code>	normal bold slanted boldslanted type cap small... <i>command</i>
<code>textstyle</code>	normal bold slanted boldslanted type cap small... <i>command</i>
<code>numberstyle</code>	normal bold slanted boldslanted type cap small... <i>command</i>
<code>number</code>	<u>yes</u> no
<code>ownnumber</code>	yes <u>no</u>
<code>page</code>	left right yes
<code>continue</code>	<u>yes</u> no
<code>header</code>	none empty high nomarking
<code>before</code>	<i>command</i>
<code>inbetween</code>	<i>command</i>
<code>after</code>	<i>command</i>
<code>alternative</code>	<u>normal</u> inmargin middle text
<code>command</code>	<code>\command#1#2</code>
<code>numbercommand</code>	<code>\command#1</code>
<code>textcommand</code>	<code>\command#1</code>
<code>prefix</code>	+ - <i>text</i>
<code>placehead</code>	<u>yes</u> no
<code>incrementnumber</code>	<u>yes</u> no <i>file</i>
<code>indentnext</code>	yes <u>no</u>
<code>file</code>	<i>name</i>
<code>expansion</code>	yes command <u>no</u>

`\setupheader[...][...,...=...,...]`

50

<code>...</code>	<u>text</u> margin edge
<code>state</code>	<u>normal</u> stop start empty high none nomarking <i>name</i>
<code>strut</code>	<u>yes</u> no
<code>style</code>	normal bold slanted boldslanted type cap small... <i>command</i>
<code>leftstyle</code>	normal bold slanted boldslanted type cap small... <i>command</i>
<code>rightstyle</code>	normal bold slanted boldslanted type cap small... <i>command</i>
<code>leftwidth</code>	<i>dimension</i>
<code>rightwidth</code>	<i>dimension</i>
<code>before</code>	<i>command</i>
<code>after</code>	<i>command</i>

`\setupheadertexts[.1.][.2.][.3.]`

49

<code>.1.</code>	<u>text</u> margin edge
<code>.2.</code>	<i>text</i> <i>section</i> date mark pagenumber
<code>.3.</code>	<i>text</i> <i>section</i> date mark pagenumber



Command definitions

`\setupheads[...]=...]`

10

sectionnumber yes no
 alternative normal inmargin middle text
 separator text
 command `\command#1#2`

`\setupindenting[...]`

78

... none small medium big next first *dimension*

`\setupinteraction[...]=...]`

◀ ▶ 63

state start stop
 menu on off
 page yes no
 openaction *reference*
 closeaction *reference*
 strut yes no
 width *dimension*
 height *dimension*
 depth *dimension*
 style normal bold slanted boldslanted type cap small...*command*
 color *name*
 contrastcolor *name*
 symbolset *name*
 title *text*
 subtitle *text*
 author *text*
 date *text*



Command definitions

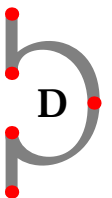
`\setupinteractionmenu[...][...][...][...][...][...][...][...][...]` 66

...	left right top bottom <i>name</i>
before	<i>command</i>
after	<i>command</i>
inbetween	<i>command</i>
left	<i>command</i>
right	<i>command</i>
middle	<i>command</i>
state	<u>start</u> stop none
style	normal <u>bold</u> slanted boldslanted type cap small... <i>command</i>
color	<i>name</i>
distance	overlay <i>dimension</i>
samepage	yes empty no none
unknownreference	yes empty no none
leftoffset	<i>dimension</i>
rightoffset	<i>dimension</i>
topoffset	<i>dimension</i>
bottomoffset	<i>dimension</i>
..=..	see p 45: \setupframed

`\setupitemize[.1.][...2.][...][...][...]`

14

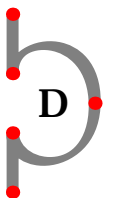
.1.	<i>number</i> each
.2.	<u>standard</u> <i>n</i> *broad <i>n</i> *serried packed unpacked stopper joinedup atmargin inmargin autointro loose
margin	no standard <i>dimension</i>
width	<i>dimension</i>
distance	<i>dimension</i>
factor	<i>number</i>
items	<i>number</i>
before	<i>command</i>
inbetween	<i>command</i>
after	<i>command</i>
beforehead	<i>command</i>
afterhead	<i>command</i>
headstyle	normal bold slanted boldslanted type cap small... <i>command</i>
marstyle	normal bold slanted boldslanted type cap small... <i>command</i>
symstyle	normal bold slanted boldslanted type cap small... <i>command</i>
stopper	text
n	<i>number</i>
symbol	<i>number</i>
align	left right <u>normal</u>
indentnext	<u>yes</u> no



`\setuplayout[...]=...]`

71

<code>width</code>	<i>dimension</i> fit middle
<code>height</code>	<i>dimension</i> fit middle
<code>backspace</code>	<i>dimension</i>
<code>topspace</code>	<i>dimension</i>
<code>margin</code>	<i>dimension</i>
<code>leftmargin</code>	<i>dimension</i>
<code>rightmargin</code>	<i>dimension</i>
<code>header</code>	<i>dimension</i>
<code>footer</code>	<i>dimension</i>
<code>top</code>	<i>dimension</i>
<code>bottom</code>	<i>dimension</i>
<code>leftedge</code>	<i>dimension</i>
<code>rightedge</code>	<i>dimension</i>
<code>headerdistance</code>	<i>dimension</i>
<code>footerdistance</code>	<i>dimension</i>
<code>topdistance</code>	<i>dimension</i>
<code>bottomdistance</code>	<i>dimension</i>
<code>leftmargindistance</code>	<i>dimension</i>
<code>rightmargindistance</code>	<i>dimension</i>
<code>leftedgedistance</code>	<i>dimension</i>
<code>rightedgedistance</code>	<i>dimension</i>
<code>horoffset</code>	<i>dimension</i>
<code>veroffset</code>	<i>dimension</i>
<code>style</code>	normal bold slanted boldslanted type cap small... <i>command</i>
<code>marking</code>	on <u>off</u> color
<code>location</code>	left middle right bottom top <u>singlesided</u> doublesided
<code>scale</code>	<i>dimension</i>
<code>nx</code>	<i>number</i>
<code>ny</code>	<i>number</i>
<code>dx</code>	<i>dimension</i>
<code>dy</code>	<i>dimension</i>
<code>lines</code>	<i>number</i>
<code>grid</code>	yes <u>no</u>



\setuplist[...,...][...,...]

52

...	<i>name</i>
state	<u>start</u> stop
alternative	a <u>b</u> c d none command
coupling	on <u>off</u>
criterium	<i>section</i> <u>local</u> previous all
pageboundaries	<i>list</i>
style	normal bold slanted boldslanted type cap small... <i>command</i>
numberstyle	normal bold slanted boldslanted type cap small... <i>command</i>
textstyle	normal bold slanted boldslanted type cap small... <i>command</i>
pagestyle	normal bold slanted boldslanted type cap small... <i>command</i>
color	<i>name</i>
command	\command#1#2#3
numbercommand	\command#1
textcommand	\command#1
pagecommand	\command#1
interaction	<u>sectionnumber</u> text pagenummer all
before	<i>command</i>
after	<i>command</i>
left	<i>text</i>
right	<i>text</i>
label	yes <u>no</u>
prefix	<u>yes</u> no
pagenummer	<u>yes</u> no
sectionnummer	yes no
aligntitle	yes <u>no</u>
margin	<i>dimension</i>
width	<i>dimension</i>
height	<i>dimension</i> fit broad
depth	<i>dimension</i> fit broad
offset	<i>dimension</i>
distance	<i>dimension</i>
separator	<i>text</i>
symbol	none 1 2 3 ...
expansion	yes <u>no</u> command
maxwidth	<i>dimension</i>



Command definitions

`\setuppagenumbering[...]=...]`

49

alternative	<u>singlesided</u> <u>doublesided</u>
location	header footer left right <u>middle</u> margin marginedge inleft inright
conversion	<u>numbers</u> characters Characters romannumerals Romannumerals
style	normal bold slanted boldslanted type cap small... <i>command</i>
left	text
right	text
way	bytext bysection <u>bypart</u>
text	text
numberseparator	text
textseparator	text
sectionnummer	yes no
separator	text
state	<u>start</u> stop
command	<code>\command#1</code>

`\setupparagraphs[.1.][.2.][...]=...]`

35

.1.	<i>name</i>
.2.	<i>number</i> each
style	normal bold slanted boldslanted type cap small... <i>command</i>
width	<i>dimension</i>
height	<i>dimension</i>
align	left right middle <u>breedte</u>
tolerance	verystRICT strict <u>tolerant</u> verytolerant
distance	<i>dimension</i>
before	<i>command</i>
after	<i>command</i>
inner	<i>command</i>
command	<i>command</i>
rule	on <u>off</u>

`\setuppositioning[...]=...]`

88

state	<u>start</u> overlay
unit	<u>cm</u> pt em mm ex sp in
factor	<i>number</i>
scale	<i>number</i>
offset	<u>yes</u> no
xstep	<u>absolute</u> relative
ystep	<u>absolute</u> relative



Command definitions

`\setupregister[.1.][.2.][...]=...]`

55

.1.	<i>singular name</i>
.2.	<i>name</i>
n	<i>number</i>
balance	yes <u>no</u>
align	yes <u>no</u>
style	normal bold slanted boldslanted type cap small... <i>command</i>
pagestyle	normal bold slanted boldslanted type cap small... <i>command</i>
textstyle	normal bold slanted boldslanted type cap small... <i>command</i>
indicator	<u>yes</u> no
coupling	yes <u>no</u>
sectionnummer	yes no
criterium	section local all <u>part</u>
distance	<i>dimension</i>
symbol	1 2 ... n a ... none
interaction	<u>pagenumber</u> text
expansion	yes command <u>no</u>
referencing	<u>on</u> off
command	<code>\command#1</code>
location	left <u>middle</u> right
unknownreference	<u>empty</u> none

`\setuprotate[...]=...]`

89

rotation	<i>number</i>
..=..	see p 45: <code>\setupframed</code>

`\setupsectionblock[...][...]=...]`

8

...	<i>name</i>
number	<u>yes</u> no
page	yes <u>right</u>
before	<i>command</i>
after	<i>command</i>



Command definitions

`\setuptables[...]=...]`

32

distance	small <u>medium</u> big
bodyfont	5pt ... 12pt small big
HL	<i>number</i> small medium big none
VL	<i>number</i> small medium big none
commands	<i>command</i>
background	screen color <u>none</u>
backgroundscreen	<i>number</i>
backgroundcolor	<i>name</i>

`\setupthinrules[...]=...]`

86

distance	small <u>medium</u> big
n	<i>number</i>
before	<i>command</i>
after	<i>command</i>
height	<i>dimension</i>
depth	<i>dimension</i>

`\setuptype[...]=...]`

70

space	on <u>off</u>
option	slanted normal <u>none</u>

`\setuptyping[...][...]=...]`

70

...	file typing <i>name</i>
space	on <u>off</u>
page	<u>yes</u> no
option	slanted normal commands color <u>none</u>
before	<i>command</i>
after	<i>command</i>
margin	<i>dimension</i> standard
evenmargin	<i>dimension</i>
oddmargin	<i>dimension</i>
blank	<i>dimension</i> small medium big standard halfline line
escape	<u>/</u>
indentnext	<u>yes</u> no
palet	<i>name</i> <u>colorpretty</u>



Command definitions

<code>\setupwhitespace[...]</code> ... <u>none</u> small medium big line fixed fix <i>dimension</i>	76
<code>\startalignment ... \stopalignment[...]</code> ..=.. see p 61: \setupalign	61
<code>\startbackground ... \stopbackground</code>	59
<code>\startbuffer[...]</code> ... <code>\stopbuffer</code> ... <i>name</i>	85
<code>\startcolumns[...]=...]</code> ... <code>\stopcolumns</code> ..=.. see p 40: \setupcolumns	38
<code>\startcombination[...]</code> ... <code>\stopcombination</code> ... <i>n*m</i>	22
<code>\startformula ... \stopformula</code> ... <i>formula</i>	15
<code>\startframedtext[...][...]=...]</code> ... <code>\stopframedtext</code> ... left right middle ..=.. see p 46: \setupframedtexts	46
<code>\starthiding ... \stophiding</code>	85



Command definitions

`\startitemize[...][...=...]` ... `\stopitemize` 10

... a A K A n N m r R K R *number* continue standard *n**broad *n**serried packed stopper
joinedup atmargin inmargin intro columns
..=.. see p 14: `\setupitemize`

`\startlegend[.1.]\leg.2.\.3.\.4.\\\leg` ... `\stoplegend` 18

.1. two
.2. text
.3. text
.4. text

`\startlinecorrection` ... `\stoplinecorrection` 76

`\startlines` ... `\stoplines` 90

`\startlocal` ... `\stoplocal` 73

`\startpacked[...]` ... `\stoppacked` 77

... blank

`\startpostponing` ... `\stoppostponing` 49

`\starttable[...]` ... `\stoptable` 25

... *text name*

`\starttyping` ... `\stoptyping` 69



Command definitions

<code>\startunpacked ... \stopunpacked</code>	78
<code>\subject[ref,...]{...}</code> ... text	9
<code>\subsection[ref,...]{...}</code> ... text	8
<code>\subsubject[ref,...]{...}</code> ... text	9
<code>\switchtobodyfont[...,...,...]</code> ... 5pt ... 12pt small big global	66
<code>\textreference[ref]{...}</code> ... text	57
<code>\thinrule</code>	86
<code>\thinrules[..=..]</code> ..=.. see p 86: \setupthinrules	86
<code>\title[ref,...]{...}</code> ... text	9

Command definitions

`\type{...}`

70

... *text*

`\useblocks[...1,...][...2,...]`

84

.1. *name*

.2. *name*

`\useexternaldocument[.1.][.2.][.3.]`

◀ ▶ 64

.1. *name*

.2. *file*

.3. *text*

`\useexternalfigure[.1.][.2.][...,...=...,...]`

22

.1. *name*

.2. *file*

scale *number*

factor *max fit broad*

wfactor *number max broad fit*

hfactor *number max broad fit*

width *dimension*

height *dimension*

frame *on off*

preset *yes no*

preview *yes no*

repeat *yes no*

object *yes no*

type *eps mps pdf tif png jpg mov tex*

method *eps mps pdf tif png jpg mov tex*

`\usemodule[...,...,...]`

92

... *name*



Command definitions

`\writebetweenlist[.1.]{.2.}`

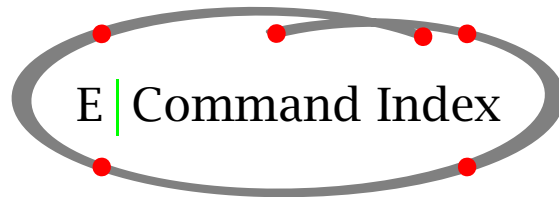
53

.1. *section name*

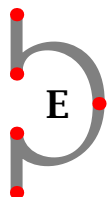
`\writetolist[.1.]{.2.}{.3.}`

53

.1. *section name*



<code>\abbreviation</code> 55	<code>\em</code> 69
<code>\adaptlayout</code> 71	<code>\en</code> 90
<code>\at</code> 56, 63	<code>\externalfigure</code> 19
<code>\bf</code> 68	<code>\fact</code> 17
<code>\blank</code> 75	<code>\footnote</code> 40
<code>\cap</code> 69	<code>\FR</code> 25
<code>\chapter</code> 8	<code>\fr</code> 90
<code>\color</code> 58	<code>\framed</code> 44
<code>\column</code> 38	<code>\from</code> 63
<code>\completecontent</code> 51	<code>\getbuffer</code> 85
<code>\completeindex</code> 54	<code>\goto</code> 63
<code>\completelistofabbreviations</code> 55	<code>\hairline</code> 85
<code>\completeregister</code> 54	<code>\head</code> 10
<code>\crlf</code> 89	<code>\hideblocks</code> 83
<code>\currentdate</code> 87	<code>\high</code> 87
<code>\DC</code> 25	<code>\HL</code> 25
<code>\de</code> 90	<code>\in</code> 56, 63
<code>\defineblock</code> 83	<code>\indenting</code> 78
<code>\definebodyfont</code> 68	<code>\index</code> 54
<code>\definecolor</code> 58	<code>\inframed</code> 44
<code>\definecombinedlist</code> 51	<code>\infull</code> 55
<code>\definedescription</code> 41	<code>\inleft</code> 46
<code>\defineenumeration</code> 43	<code>\inmargin</code> 46
<code>\definefloat</code> 82	<code>\input</code> 91
<code>\definelist</code> 51	<code>\inright</code> 46
<code>\defineparagraphs</code> 35	<code>\item</code> 10
<code>\defineregister</code> 54	<code>\language</code> 90
<code>\definesymbol</code> 10	<code>\leftlines</code> 61
<code>\definesynonyms</code> 55	<code>\leg</code> 17
<code>\DL</code> 25	
<code>\DR</code> 25	



Command Index

\loadabbreviations 55
\lohi 87
\LOW 25
\low 87
\LR 25

\margintitle 46
\midaligned 61
\MR 25

\NC 25
\nl 90
\noheaderandfooterlines 49
\noindenting 78
\nowhitespace 75
\NR 25

\page 48
\pagereference 56
\par 75
\paragraph 8, 75
\percent 18
\permille 18
\placecontent 51
\placefigure 19
\placeformula 15
\placeindex 54
\placeintermezzo 82
\placelistofabbreviations 55
\placeregister 54
\placetable 25
\position 87

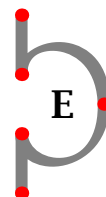
\rightaligned 61
\rm 68
\rotate 88

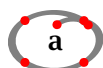
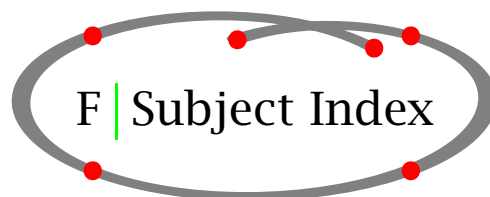
\setupalign 61
\setupbackground 59
\setupbackgrounds 60
\setupblank 75

\setupblock 83
\setupbodyfont 66
\setupbuffer 85
\setupcaptions 19, 25, 82
\setupcolors 58
\setupcolumns 38
\setupcombinedlist 51
\setupdescriptions 41
\setupenumerations 43
\setupfigures 19
\setupfloat 82
\setupfloats 19, 25, 82
\setupfooter 49
\setupfootertexts 49
\setupfootnotes 40
\setupformulae 15
\setupframed 44
\setupframedtext 45
\setuphead 8
\setupheader 49
\setupheadertexts 49
\setupheads 8
\setupindenting 78
\setupinteraction 62
\setupitemize 10
\setuplayout 71
\setuplinecorrection 75
\setuplist 51
\setuppositioning 87
\setuppagenumbering 48
\setupparagraphs 35
\setupregister 54
\setupsynonyms 55
\setuptables 25
\setupthinrules 85
\setuptolerance 61
\setuptype 69
\setuptyping 69
\setupwhitespace 75
\showlayout 71
\showmakeup 71

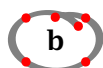
Command Index

<code>\showsetups</code>	71	<code>\subsubject</code>	8
<code>\sl</code>	68	<code>\switchtobodyfont</code>	66
<code>\sp</code>	90		
<code>\SR</code>	25	<code>\tfa</code>	68
<code>\ss</code>	68	<code>\tfb</code>	68
<code>\startbackground</code>	59	<code>\tfc</code>	68
<code>\startbuffer</code>	85	<code>\tfd</code>	68
<code>\startchemical</code>	35	<code>\thinrule</code>	85
<code>\startcolumns</code>	38	<code>\thinrules</code>	85
<code>\startcombination</code>	19, 25	<code>\THREE</code>	25
<code>\startfact</code>	17	<code>\title</code>	8
<code>\startfiguretext</code>	19	<code>\tt</code>	68
<code>\startformula</code>	15	<code>\TWO</code>	25
<code>\startframedtext</code>	45	<code>\type</code>	69
<code>\starthiding</code>	85	<code>\typebuffer</code>	85
<code>\startitemize</code>	10		
<code>\startlegend</code>	17	<code>\unit</code>	18, 55
<code>\startlines</code>	89	<code>\useblocks</code>	83
<code>\startlocal</code>	71	<code>\useexternaldocument</code>	63
<code>\startpacked</code>	75	<code>\useexternalfigures</code>	19
<code>\startpostponing</code>	48		
<code>\starttable</code>	25	<code>\VL</code>	25
<code>\starttextrule</code>	85		
<code>\starttyping</code>	69	<code>\whitespace</code>	75
<code>\startunpacked</code>	75	<code>\writebetweenlist</code>	51
<code>\subject</code>	8	<code>\writetolist</code>	51
<code>\subparagraph</code>	8		





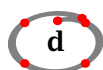
accents 70
alignment 61
AMS 66
auxilliary files 95



background
 page areas 60
 text 59



carriage return 89
color 58
columns 35, 38
columns in itemize 10
comment 91
Computer Modern Roman 66
cont-sys.tex 93



date 87
definition 41
Acrobat Distiller 62
dvi-file 5



emphasized 69
Acrobat Exchange 62

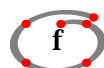
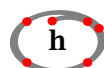
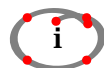


figure 19
floating blocks 19, 25, 82
font
 size 66
 style 66
fontsize 68
footnote 40
foreign symbols 71
formula 15



headers 8
hiding text 85
hyphenation 90



indentation 78
input file 3
 processing 5
input other \TeX -files 91
interaction
 external 63
 internal 63
interactive mode 62
inter paragraph spacing 75
itemize 10



label 56

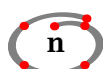


Subject Index

language 90
layout 71, 79
legends 17
lines 85
list 51
Lucida Bright 66



margin text 46

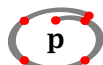


numbered definition 43

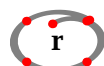


outline

paragraph 45
text 44



page areas 60
page breaking 48
page design 71
pagefooter 49
page header 49
page numbering 48
paragraph 75
indentation 78
pdf-file 5
% in input file 91
positioning 87
ps-fonts 66



Acrobat Reader 62
referring 56
register 54
rotating 88



set up 79
SI-unit 18
small caps 69
special characters 6
storing text 85
subscript 87
superscript 87
synonyms 55

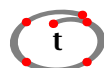
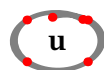


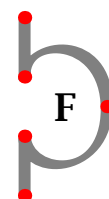
table of contents 51
tables 25
text blocks 83
T_EXUTIL 94
tub-file 95
tuf-file 95
tui-file 95
tuo-file 94, 95
type 69

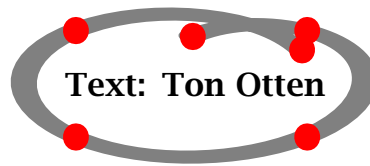


units 18

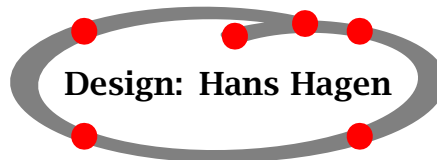


verbatim 69

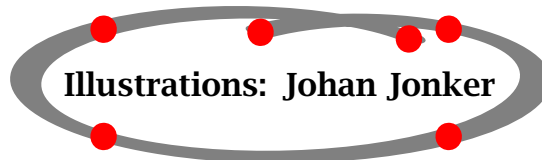




Text: Ton Otten



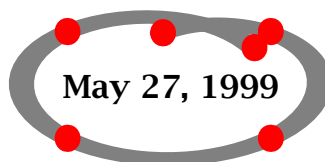
Design: Hans Hagen



Illustrations: Johan Jonker



**PRAGMA ADE
Ridderstraat 27
8061GH Hasselt NL
+31 (0)38 477 53 69
pragma@pi.net**



May 27, 1999