

# Page design

## Introduction

While processing a text  $\text{\TeX}$  makes use of the actual  $\text{\hspace}$  (width) and  $\text{\vsize}$  (height). As soon as  $\text{\vsize}$  is exceeded  $\text{\TeX}$ 's output routine is launched. The output routine deals with the typeset part — most of the time this will be a page. It takes care of typesetting the headers and footers, the page number, the backgrounds and footnotes, tables and figures. This rather complex process makes it obvious that the output routine actually makes use of more dimensions than  $\text{\hspace}$  and  $\text{\vsize}$ .

## Paper dimensions

With the command  $\text{\setuppapersize}$  the dimensions of the paper being used are defined. There is a difference between the dimensions for typesetting and printing.

```
\setuppapersize [...1...] [...2...]
                                OPTIONAL
1  A3 A4 A5 A6 letter ... CD IDENTIFIER landscape mirrored rotated 90 180 270
2  negative inherits from \setuppapersize
```

The dimensions of DIN formats are given in table 1.1.

format	size in mm	format	size in mm
A0	$841 \times 1189$	A5	$148 \times 210$
A1	$594 \times 841$	A6	$105 \times 148$
A2	$420 \times 594$	A7	$74 \times 105$
A3	$297 \times 420$	A8	$52 \times 74$
A4	$210 \times 297$	A9	$37 \times 52$

**Table 1.1** Default paper dimensions

There are a great number of standardized formats like B0–B9 and C0–C9. These formats are predefined in  $\text{Con}\text{\TeX}$ t as well. You can also use: *letter*, *legal*, *folio* and *executive*, *envelope 9–14*, *monarch*, *check*, *DL* and *CD*. Another series of predefined formats comprise the *RA* and *SRA* types of paper sizes.

A new format can be defined by:

```
\definepapersize [...1...] [...2...]
1  IDENTIFIER
2  width   = DIMENSION
   height  = DIMENSION
   offset  = DIMENSION
   scale   = NUMBER
```

For example CD was defined as:

```
\definepapersize[CD][width=12cm,height=12cm]
```

After defining CD you can type:

```
\setuppapersize[CD][A4]
```

This means that for typesetting ConT<sub>E</sub>Xt will use the newly defined size CD. The resulting, rather small page, is positioned on an A4 paper size. This second argument is explained in detail later.

ConT<sub>E</sub>Xt can also be used to produce screen documents. For that purpose a number of screen formats are available that relate to the screen dimensions. You can use: S3–S6. These generate screens with widths varying from 300 to 600 pt and a height of 3 / 4 of the width.

When one chooses another paper format than A4, the default settings are scaled to fit the new size.

All defined paper sizes can be used either in portrait or landscape orientation. You can tell ConT<sub>E</sub>Xt the orientation of the paper in the `\setuppapersize` command:

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

## Page texts

Page texts are texts that are placed in the headers, footers, margins and edges of the so called pagebody. This sentence is for instance typeset in the bodyfont in the running text. The fonts of the page texts are set up by means of different commands. The values of the parameters may be something like `style=bold` but `style=\ss\bf` is also allowed. Setups like `style=\ssbf` are less obvious because commands like `\cap` will not behave the way you expect.

Switching to a new font style (`\ss`) will cost some time. Usually this is no problem but in interactive documents where we may use interactive menus with dozens of items and related font switches the effect can be considerable. In that case a more efficient font switching is:

```
\setuplayout[style=\ss]
```

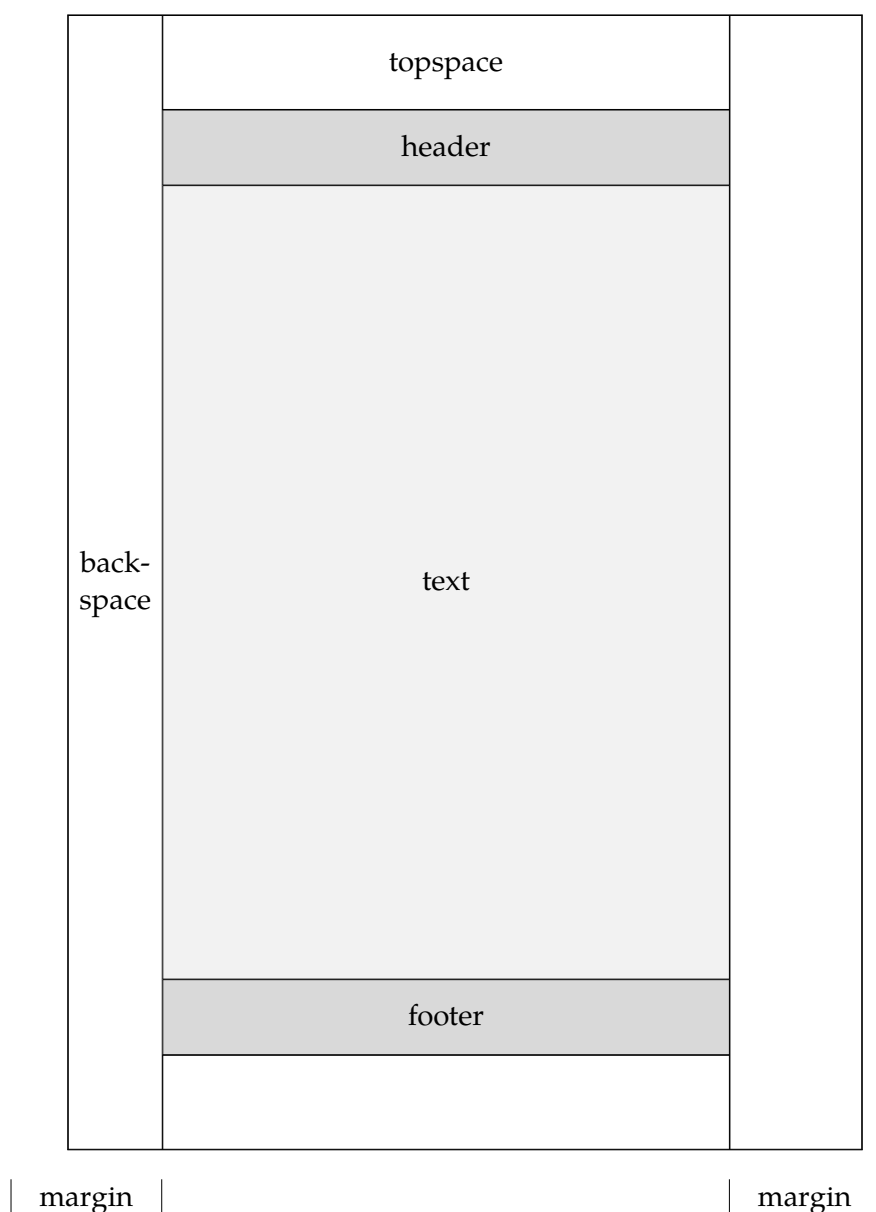
Border texts are setup by its command and the related key. For example footers may be set up with the key `letter`:

```
\setupfooter[style=bold]
```

## Page composition

In page composition we distinguish the main text area, headers and footers, and the margins (top, bottom, right and left). The main text flows inside the main text area. When defining a layout, one should realize that the header, text and footer areas are treated as a whole. Their position on the page is determined by the `topspace` and `backspace` dimensions (see picture 1.1).

The header is located on top and the footer below of the main text area. Normally, in the header and footer page numbers and running titles are placed. The left and/or right margins are often used for structural components like marginal notes and/or chapter and section numbers. The margins are located in the `backspace` (along the spine) and in the white space to the right/left



**Figure 1.1** The A4 typesetting area and margins (height = header + text + footer).

**left** of the main text area. Their width has *no* influence on the location of the typesetting area on the page. **right**

On the contrary, the height of the header and footer influences the height of the text area. When talking about the height, we think of the sum of the header, text and footer areas. This approach enables you to occasionally hide the header and/or footer, without introducing inconsistency in the layout.

The dimensions and location of all those areas are set up with `\setuplayout`.

Setting up the left or right margin has no influence on the typesetting area. In paper documents this parameter is only of use when keywords or other text are placed in the margin (hyphenation).

```
\setuplayout [...,*,...]

*   width           = DIMENSION fit middle
    height          = DIMENSION fit middle
    backspace       = DIMENSION
    topspace        = DIMENSION
    margin           = DIMENSION
    leftmargin       = DIMENSION
    rightmargin      = DIMENSION
    header           = DIMENSION
    footer           = DIMENSION
    top              = DIMENSION
    bottom           = DIMENSION
    leftedge         = DIMENSION
    rightedge        = DIMENSION
    headerdistance   = DIMENSION
    footerdistance   = DIMENSION
    topdistance      = DIMENSION
    bottomdistance   = DIMENSION
    leftmargindistance = DIMENSION
    rightmargindistance = DIMENSION
    leftedgedistance = DIMENSION
    rightedgedistance = DIMENSION
    horoffset        = DIMENSION
    veroffset        = DIMENSION
    style             = normal bold slanted boldslanted type cap small... COMMAND
    color             = IDENTIFIER
    marking           = on off color screen TEXT
    location          = left middle right bottom top singlesided doublesided
    scale             = DIMENSION
    nx               = NUMBER
    ny               = NUMBER
    dx               = DIMENSION
    dy               = DIMENSION
    lines            = NUMBER
    columns           = NUMBER
    columndistance    = DIMENSION
    grid              = yes no
    bottomspace       = DIMENSION
    cutspace          = DIMENSION
    textdistance      = DIMENSION
    textwidth         = NUMBER
    textmargin        = DIMENSION
    clipoffset        = DIMENSION
    page              = IDENTIFIER
    paper             = IDENTIFIER
```

For paper documents it is sufficient to set up the height, header, footer, top space and back space. For electronic and screen documents however we need some extra space for navigational tools (see chapter ??). In screen documents it is common practice to use backgrounds. Therefore it is also possible to set up the space between the text area and the header and footer on a page, and thereby visually separating those areas.

Parameter	Value	Comment
width	dimension	Determines the width of the typesetting area. Middle sets the white space right

		to the typesetting area to the value of the backspace. typeFit takes values set for margins, edges and margin and edge distances into account.
height	dimension	The height is the sum of the text height, header, footer, headerdistance, footerdistance. Middle sets the bottom white space to the value of the topspace. Fit calculates the text height based on the other vertical height-elements.
backspace	dimension	Backspace determines the left boundary of the typesetting area.
topspace	dimension	Topspace determines the top boundary of the typesetting area. Together backspace and topspace determine the left top corner of the typesetting area.
margin	dimension	Setting this parameters makes left and right margin equally large.
leftmargin	dimension	For documents with different size of the left and right margin, the left margin size is determined.
rightmargin	dimension	For documents with different size of the left and right margin, the right margin size is determined.
header	dimension	Determines the height of a running header. The header height is part of the height parameter.
footer	dimension	Determines the height of the footer. The footer height is part of the height parameter.
top	dimension	Makes space available in the topspace area. This parameter is not part of the text height.
bottom	dimension	Makes space available underneath the typesetting area. This parameter is not part of the text height.
leftedge	dimension	This space located left to the left margin is for screen documents only.
rightedge	dimension	This space located right to the right margin is for screen documents only.
headerdistance	dimension	All parameters ending on ...distance create white space between adjacent elements.
footerdistance	dimension	
leftmargindistance	dimension	
rightmargindistance	dimension	

leftedgedistance	dimension	
rightedgedistance	dimension	
topdistance	dimension	
bottomdistance	dimension	
horoffset	dimension	A horizontal offset moves the complete layout horizontally, starting from the place indicated by the parameter location.
veroffset	dimension	A vertical offset moves the complete layout vertically, starting from the place indicated by the parameter location.
style	normal bold slanted boldslanted type cap small... COMMAND	With the style parameter one can setup the general style of the font(s) used in the document.
marking	on off color screen TEXT	When this parameter is set to on, then crop marks are placed around the page. Color displays a color bar, whereas screen shows a gray-values bar.
location	left middle right bottom top singlesided doublesided duplex	location determines where the page is placed on the paper. It allows to typeset single and double sided documents and documents for duplex printing (see: 1.6).
scale	number	With scale it is possible to scale a page before placing it on the defined paper.
nx	number	In case that a given text should be placed multiple times on a defined paper, nx gives the number of pages on the x-axis and ny the number of pages on the y-axis.
ny	number	
dx	dimension	With dx and dy the distances of the pages indicated in nx and ny can be manipulated.
dy	dimension	
lines	number	Determines the textheight in terms of the number of lines-heights.
columns	number	
columndistance	dimension	
grid	yes no	Typsetting on the grid is activated with grid=on.
bottomspace	dimension	Bottomspace increases the white space at the bottom of the page without altering the page-layout.
cutspace	dimension	Cutspace increases the white space at the right side of the page without altering the page-layout.

textdistance	dimension
textwidth	dimension
textmargin	dimension
clipoffset	dimension
page	identifier
paper	identifier

In order to get information on the current settings the following commands can be issued:

```
\showframe [...]
          OPTIONAL
*   TEXT margin edge
```

The dimensions can be displayed by:

```
\showsetups
```

A multi-page combination of both is generated with:

```
\showlayout
```

The width of a text is available as `\hsize` and the height as `\vsize`. To be on the safe side one can better use ConTeXt's `\dimen`-registers `\textwidth` and `\textheight`, `\makeupwidth` and `\makeupheight`.

When we are typesetting in one column of text `\textwidth` and `\makeupwidth` are identical. In case of a two columned text the `\textwidth` is somewhat less than half the `\makeupwidth`. The `\textheight` is the `\makeupheight` minus the height of the header and footer.

variable	meaning
<code>\makeupwidth</code>	width of a text
<code>\makeupheight</code>	height of a text
<code>\textwidth</code>	width of a column
<code>\textheight</code>	height – header – footer

**Table 1.2** Some `\dimen` variables

There are also other dimensions available like `\leftmarginwidth` and `\footerheight`, but be aware of the fact that you can only use these variables, you can not set them up. The width of a figure could for instance be specified as `width=.9\leftmarginwidth`.

Basically documents are typeset automatically. However, in some cases the output would become much better if a line would be moved to another page. For these situations you can adjust the layout temporarily (just for that page) by typing:

```

\adaptlayout [...1,...] [...2,...]
                OPTIONAL
1  NUMBER
2  height = DIMENSION max
    lines = NUMBER

```

The use of this command should be avoided inside a text, because after altering your document the adjustment could possibly not be necessary anymore. So, if you use this command, use it at the top of your document. For example:

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

The layout of page 21 and 38 will temporarily be 0.5 cm higher though the footer will be maintained at the same height. The numbers to be specified are the page numbers in the output file.

If the layout is disturbed you can reset the layout by:

```
\setuplayout[reset]
```

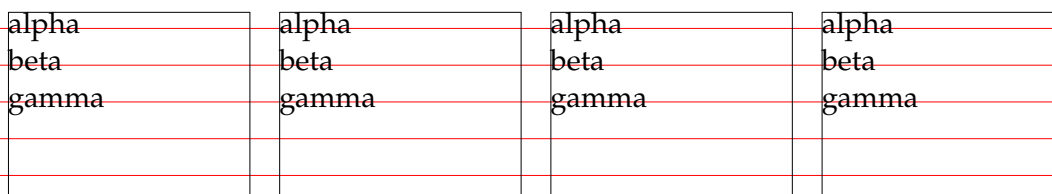
In some commands you can set up the parameters `width` and `height` with the value `fit`. In that case the width and height are calculated automatically.

On the next pages we will show a number of A5 page layouts centered on an A4. The default setups (dimensions) are adequate for standard documents like manuals and papers. The setup adjusts automatically to the paper size. Note the use of `middle` while setting up the parameters `width` and `height`.

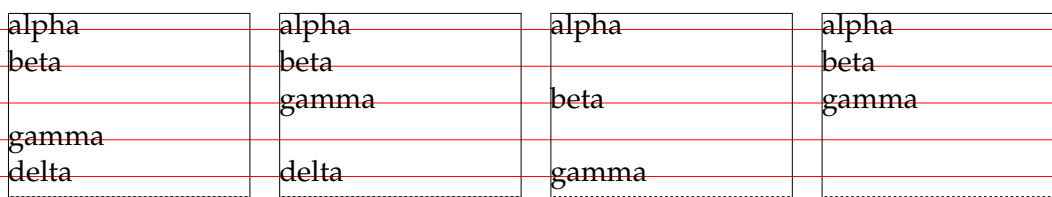


## 1.5 Grids

There are many ways to align text on a page. Look at the example below and notice the vertical alignment of the words and the white space between the words on the mini pages.



The first three alternatives result in an undesired output. The fourth alternative will lead to pages with unequal length. So we rather make the white space between the lines a little stretchable.<sup>1</sup>



A stretchable line spacing has the disadvantage that lines of two pages or two columns that are displayed close to each other, will seldom align. This is very disturbing for a reader.<sup>2</sup>

grid. The means to do this in TeX are very limited but ConTeXt has some features to support grid typesetting.<sup>3</sup>

<sup>1</sup> Hey, watch this. A footnote!

<sup>2</sup> Here! Another footnote.

<sup>3</sup> Finally, the last footnote!

In those situations we prefer to typeset on a

During typesetting on a grid the heads, figures, formulas and the running text are set on a fixed line spacing. If a typographical component for any reason is not placed on the grid one can snap this component to the grid with:

```
\placeongrid{\framed{This is like a snapshot.}}
```

This will result in:

[This is like a snapshot.](#)

This mechanism can be influenced with an argument:

```
\placeongrid[bottom]{\framed{Do you like the snapshot?}}
```

Now an empty line will appear below the framed text. Other parameters are: `top` and `both`. The last parameter divides the linespace between over and below the framed text.

Now the snapshot looks better.

These examples don't show pretty typesetting. The reason is that `\framed` has no depth because  $\TeX$  handles spacing before and after a line in a different way than text. Con $\TeX$ t has a solution to this:

```
\startlinecorrection
\framed{This is something for hotshots.}
\stoplinecorrection
```

The command `\startlinecorrection` tries to typeset the lines as good as possible and takes the use of grid in account.

This is something for hotshots.

Because line correction takes care of the grid we have to use yet another command to stretch the framed text:

```
\startlinecorrection
\framed{Anyhow it is good to know how this works.}
\stoplinecorrection
```

As you can see this results in somewhat more space:

Anyhow it is good to know how this works.

For test purposes one can display the grid with the command `\showgrid`. So grid related commands are:

```
\placeongrid [.1.] {.2.}
      OPTIONAL
1  inherits from \moveongrid
2  CONTENT
```

	1
\showgrid [... <sup>1</sup> ...] {... <sup>2</sup> }	2
OPTIONAL	3
1 reset top bottom none all lines frame nonumber right left	4
2 CONTENT	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
	22
	23
	24
	25
	26
	27
	28
	29
	30
	31
	32
	33
	34
	35
	36
	37
	38
	39
	40
	41
	42
	43
	44
	45
	46
	47

## 1.6 Printing

In an earlier section we used page and paper dimensions. In this section we will discuss how these two can be manipulated to yield a good output on paper.

In figure 1.3 and 1.4 we see some alternatives to manipulate the page composition by means of `\setuppapersize` and `\setuplayout`. So it is possible to put a page in a corner or in the middle of the paper, to copy a page and to use cutting marks.

When the parameter paper size is set to `landscape` width and height are interchanged. This is not the same as rotation! Rotation is done by typing 90, 180 and 270 in the first argument of `\setuppapersize`.

```
\setuppapersize[A5,landscape][A4]
```

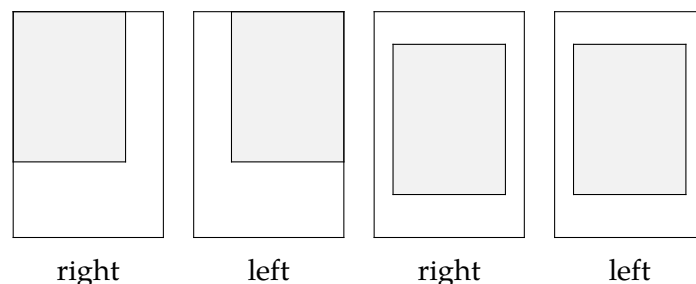
These examples don't show that we can correct for duplex printing. For example when we type:

```
\setuppapersize[A5][A4]
\setuplayout[location=middle,marking=on]
```

the front and back side will be placed in the middle of the paper. The markings enable you to cut the paper at the correct size. If we only want to cut twice, we type:

```
\setuppapersize[A5][A4]
\setuplayout[location=duplex]
```

This has the same meaning as `{duplex,left}`. At this setup ConTeXt will automatically move front and back side to the correct corner. In figure 1.2 we show both alternatives.

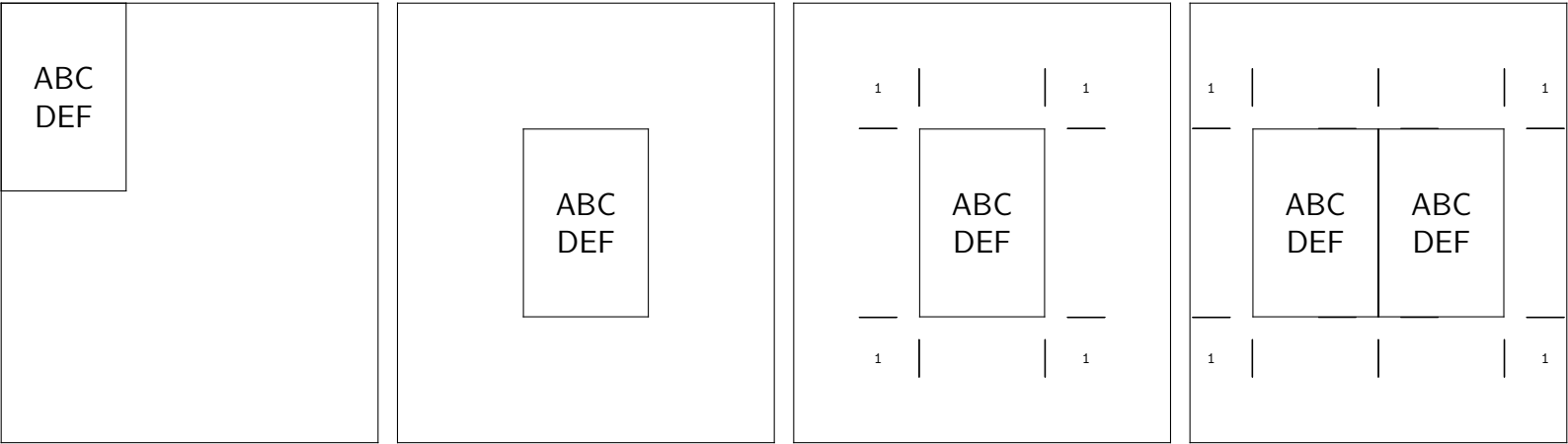


**Figure 1.2** Positioning the page on paper for cutting.

Rotating, mirroring, scaling, duplicating and placing pages on paper are independent operations. By combining these operations the desired effects can be reached. Rotating and mirroring and page and paper size are set up at the same time. The other operations are set up with `\setuplayout`.

## 1.7 Arranging pages

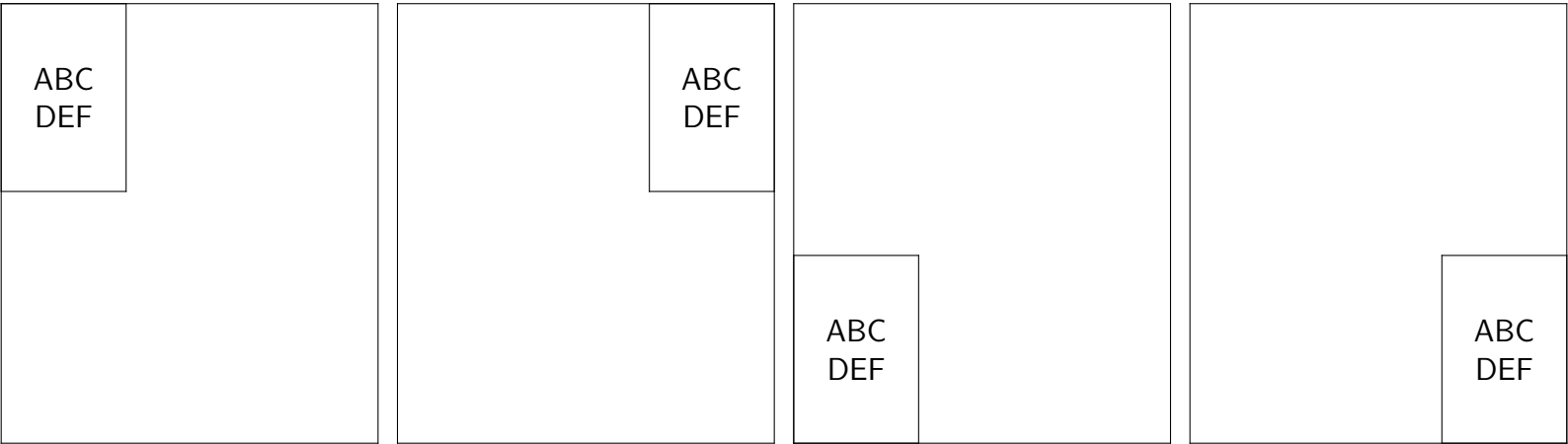
Simplified we can say that  $\text{\TeX}$  typesets pages. If the typeset material should become a book, then there are two options. Firstly the book will be produced on multiple sheets carrying only one page either on one or on both sides of the sheet. Second option is to produce arrangements of multiple pages per sheet of paper which will be folded into sections, using imposition schemes.



location=middle

marking=on  
location=middle

marking=on  
location=middle  
nx=2

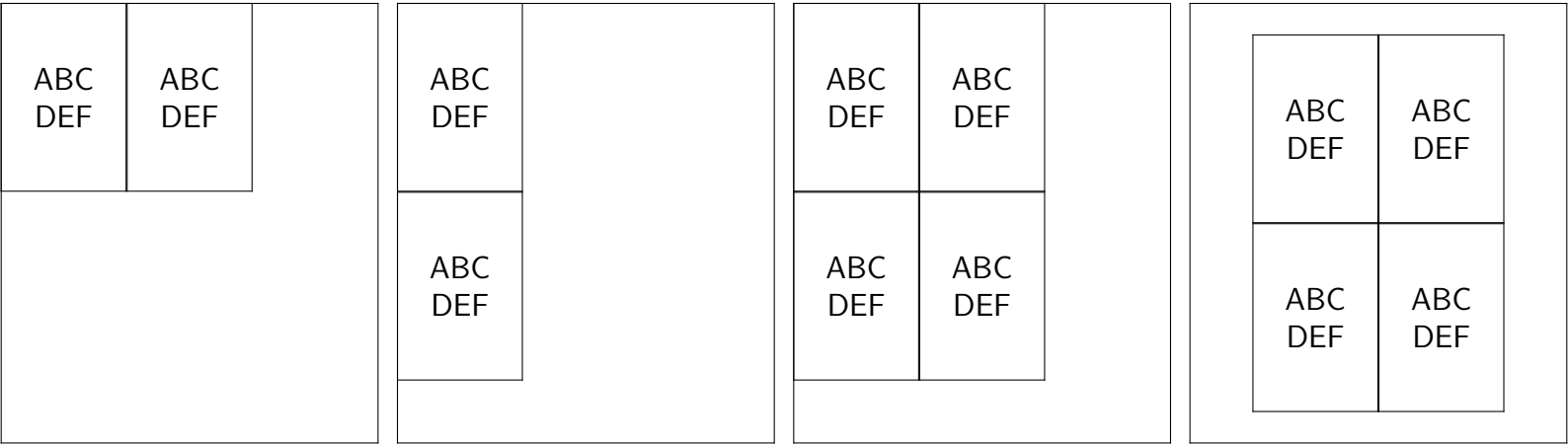


location=left

location=right

location=left,bottom

location=right,bottom

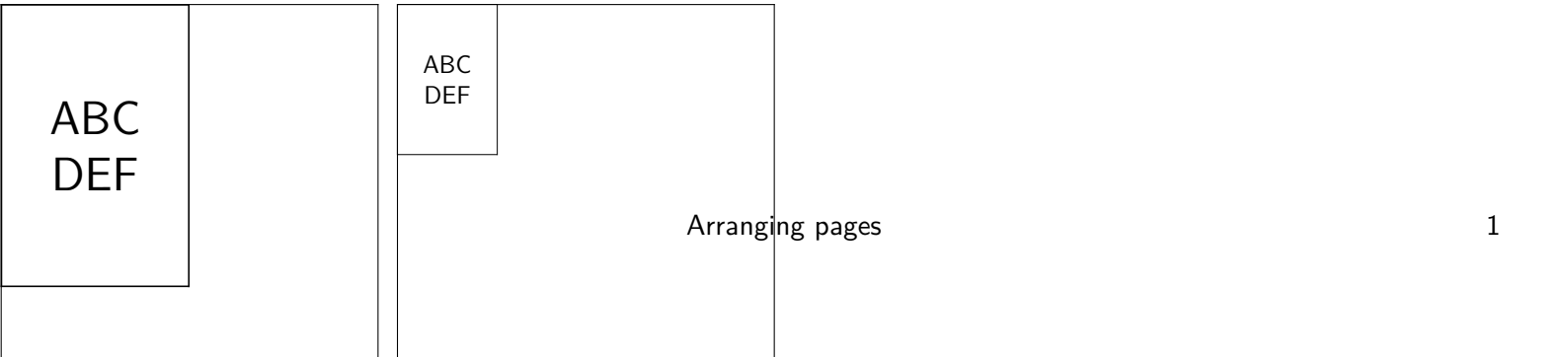


nx=2,ny=1

nx=1,ny=2

nx=2,ny=2

nx=2,ny=2  
location=middle



<div>ABC DEF</div>	<div>ABC DEF</div>	<div>ABC DEF</div>
landscape	landscape	landscape landscape
<div>ABC DEF</div>	<div>ABC DEF</div>	<div>DEF ABC</div>
90	90	90 90
<div>DEF ABC</div>	<div>DEF ABC</div>	<div>ABC DEF</div>
180	180	180 180
<div>DEF ABC</div>	<div>DEF ABC</div>	<div>ABC DEF</div>

ConT<sub>E</sub>Xt offers tools to achieve both options.

In the following table an overview is given about all currently available arranging schemes.

Key for \setuparranging	Meaning
[2SIDE]	2 pages next to each other single sided only!
[2TOP]	2 pages above each other, single sided only!
[1*8]	1 sheet 1 x 8 pages = 8 pages single sided!
[1*4]	1 sheet 1 x 4 pages = 4 pages single sided!
[1*2*Conference]	2 pages on top of each other, 1 page rotated
[1*4*Conference]	2 odd pages next to each other, even page rotated on top
[XY]	Arrangement in nx columns and ny rows, uses the setup \setuppaper [dx=, dy=, nx=, ny=]
[2UP]	2 pages next to each other, n sheets arranged for a single booklet!
[2DOWN]	2 pages above each other, n sheets arranged for a single booklet!
[2TOPSIDE]	2 odd pages on one side, 2 even pages verso, above each other
[2*16]	Section: one sheet 2 x 16 pages = 32 pages
[2*8]	Section: one sheet 2 x 8 pages = 16 pages
[2*8*Z]	Section: one sheet 2 x 8 pages = 16 pages, special folding: zig-zag
[2*6*Z]	Section: one sheet 2 x 6 pages = 12 pages, special folding: zig-zag
[2*4]	Section: one sheet 2 x 4 pages = 8 pages
[2*2]	Section: one sheet 2 x 2 pages = 4 pages
[2**2]	Section: one sheet 2 x 2 pages = 4 pages
[2*4*2]	Section of 16 pages: 2 sheets, 4 pages front and backside
[2*2*4]	Section of 16 pages: 4 sheets, 2 pages front and backside
[3SIDE]	3 odd pages recto, 3 even pages verso
[2*2*2]	Section: two sheets 2 x 2 pages = 8 pages
[2*2*3]	Section: three sheets 2 x 2 pages = 12 pages
[TRYPTICHON]	Leaflet: one sheet 2 x 3 pages = 6 pages
[DOUBLEWINDOW]	Leaflet: one sheet 2 x 4 pages = 8 pages
[ZFLYER-8]	Leaflet: one sheet 2 x 4 pages = 8 pages
[ZFLYER-10]	Leaflet: one sheet 2 x 5 pages = 10 pages
[ZFLYER-12]	Leaflet: one sheet 2 x 6 pages = 12 pages
[MAPFLYER-12]	Leaflet: one sheet 2 x 6 pages = 12 pages

When talking about book-printing the industry produces different kinds of sections, consisting commonly out of 32 or 16 pages. Consider, that sections of 32 pages may be quite thick. At binding if the sections are sewn and the spine is rounded the fore edge can become stepped. This is aesthetically less satisfying. Best results are normally obtained with sections of 16 pages.

For special purposes or in case of special papers also less than 16 pages per section are arranged.

The command to arrange pages with ConT<sub>E</sub>Xt is

For (standard) sections the following list of schemes is available:

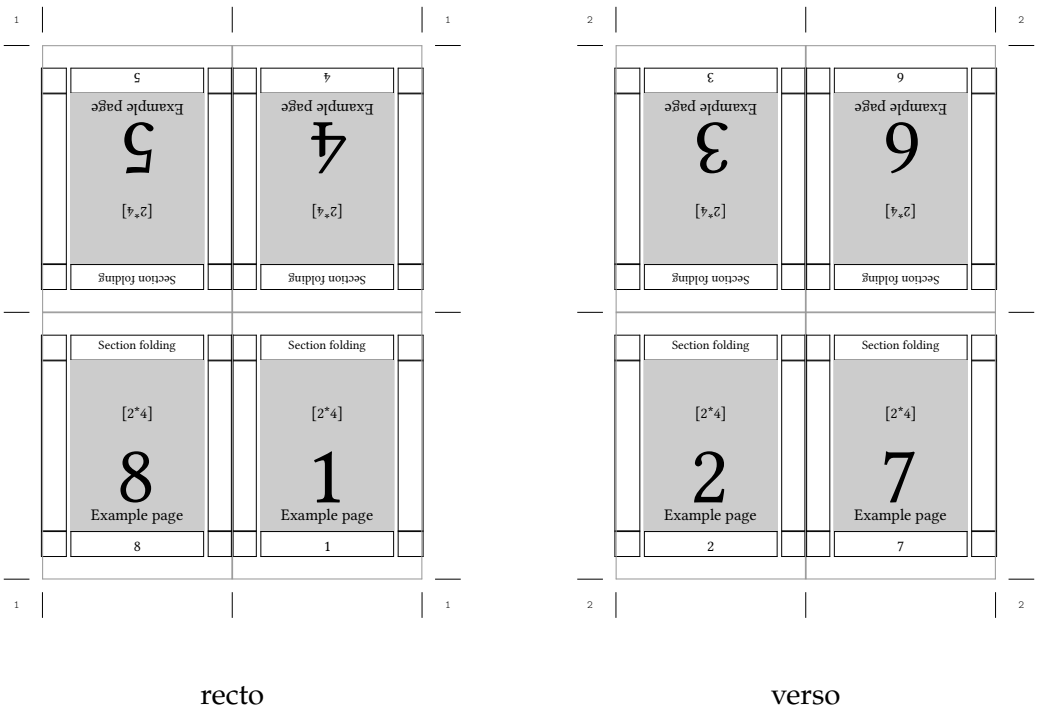
Arrangement	Result	Number of pages
\setuparranging [2*16]	section: one sheet 2 x 16 pages	= 32 pages
\setuparranging [2*8]	section: one sheet 2 x 8 pages	= 16 pages
\setuparranging [2*4]	section: one sheet 2 x 4 pages	= 8 pages

```
\setuparranging [2*2]    section: one sheet 2 × 2 pages  = 4 pages
\setuparranging [2**2]   section: one sheet 2 × 2 pages  = 4 pages
\setuparranging [2*8*Z]  section: one sheet 2 × 8 pages  = 16 pages, special folding: zig-zag
\setuparranging [2*6*Z]  section: one sheet 2 × 6 pages  = 12 pages, special folding: zig-zag
\setuparranging [2*4*2]  section: 2 sheets, 4 pages front = 16 pages
                           and backside
\setuparranging [2*2*4]  section: 4 sheets, 2 pages front = 16 pages
                           and backside
\setuparranging [2*2*2]  section: 2 sheets 2 × 2 pages  = 8 pages
\setuparranging [2*2*3]  section: 3 sheets 2 × 2 pages  = 12 pages
```

On the following pages we show pictures of arranged pages for the mentioned imposition schemes.

The above mentioned imposition schemes are meant for the professional printing industry.

But also with an office printer one can produce sections. Sections with less than 16 pages can be produced with the following folding schemes:



**Figure 1.5** 8 pages

The last two examples (Figure 1.6 and 1.7) differ only in the fact, that the verso side carries the two pages in reversed order.

The simplest version of a section is booklet-printing. In this case all pages are arranged in such a way, that with a single fold a booklet is formed.



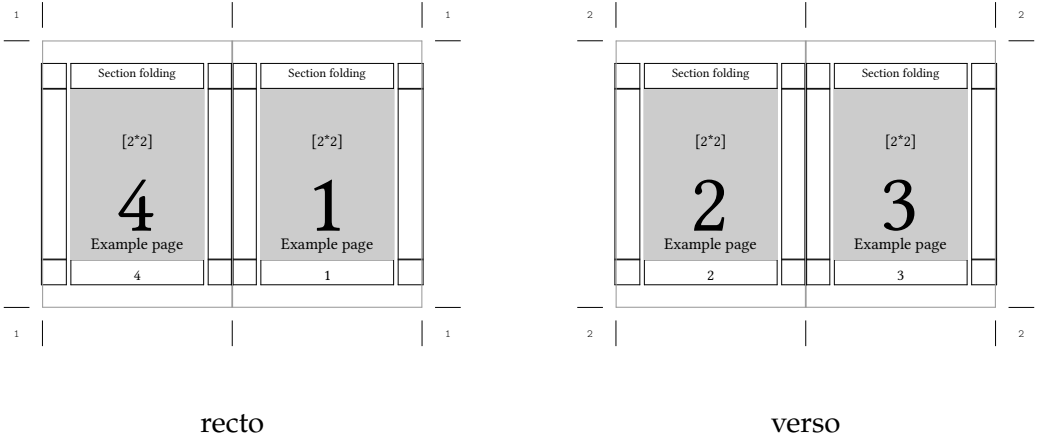


Figure 1.6 4 pages

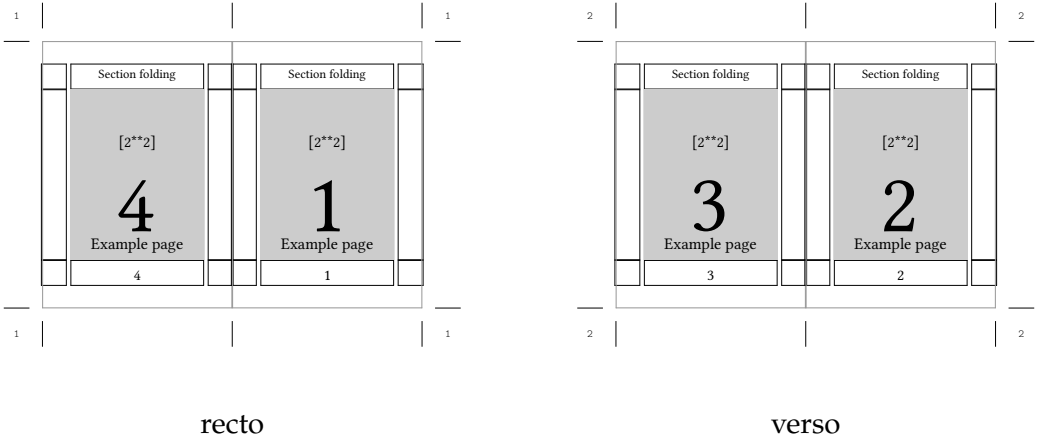


Figure 1.7 4 pages

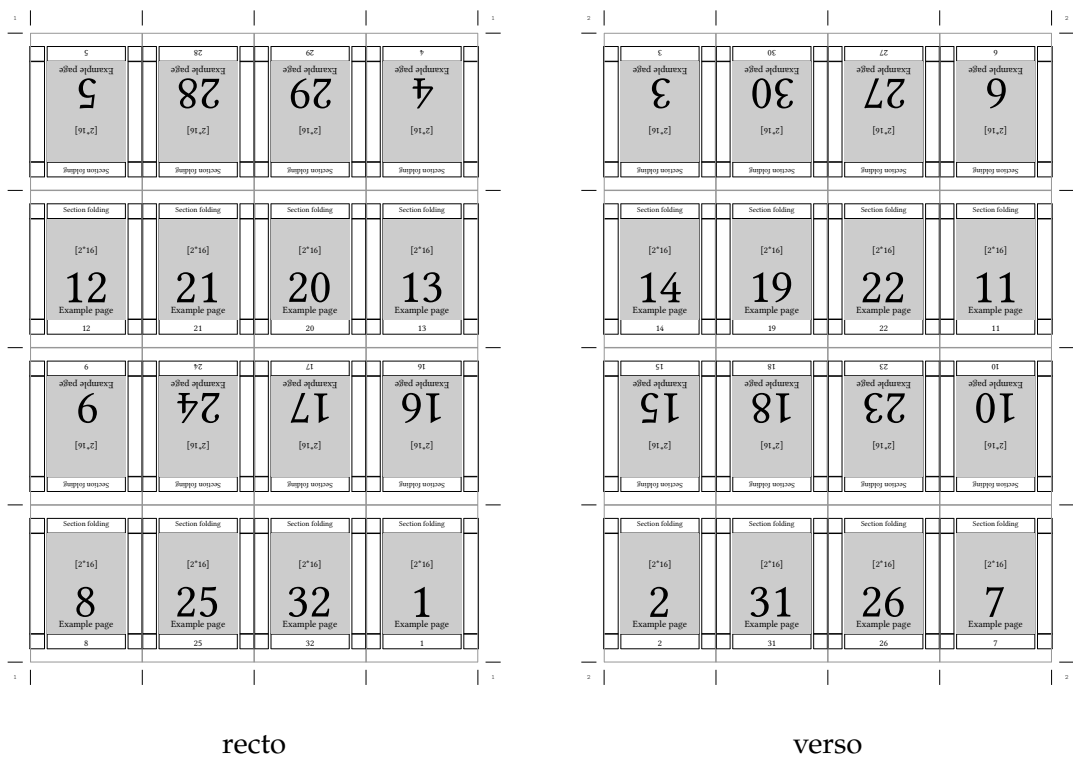


Figure 1.8 32 pages

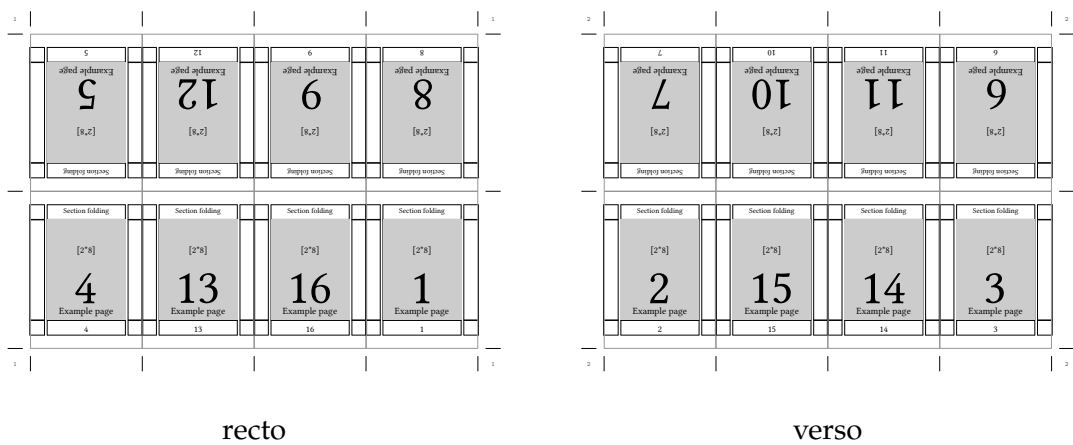


Figure 1.9 16 pages

Arrangement	Result	Number of pages
<code>\setuparranging [2UP]</code>	2 pages next to each other, n sheets arranged for a single booklet	
<code>\setuparranging [2DOWN]</code>	2 pages above each other, n sheets arranged for a single booklet	

‘2UP’ results in a booklet with the fold on the long egde of the page. ‘2DOWN’ gives a booklet with a short-edge binding of the pages.

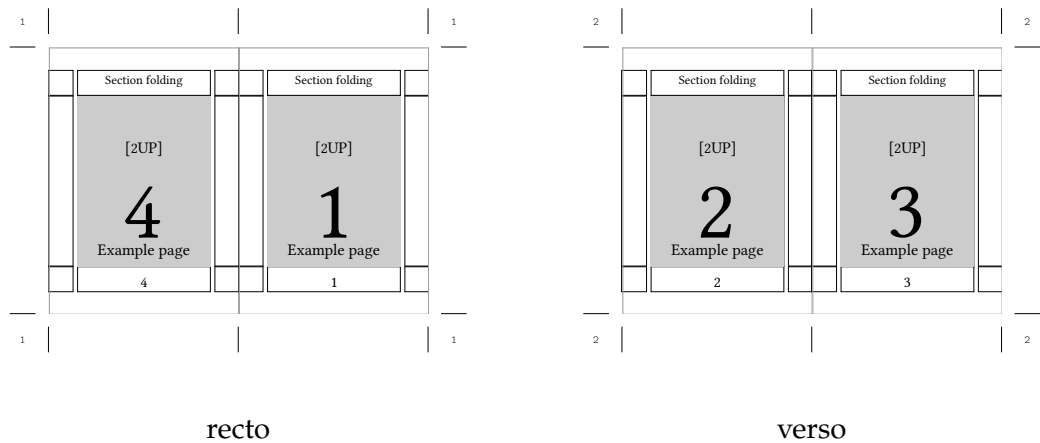


Figure 1.10 2 UP booklet: long edge binding

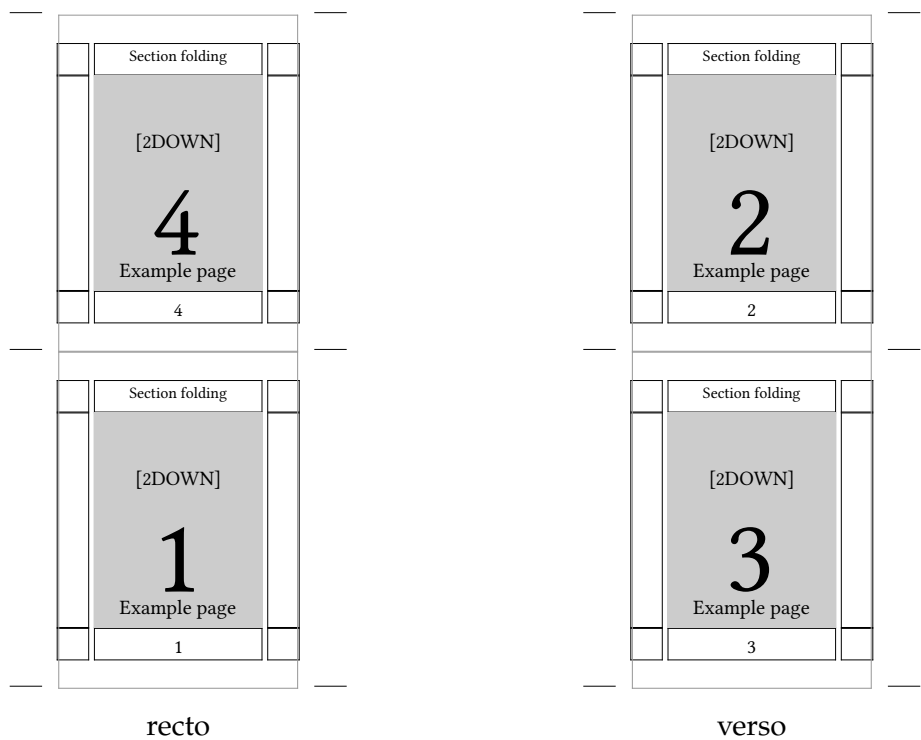
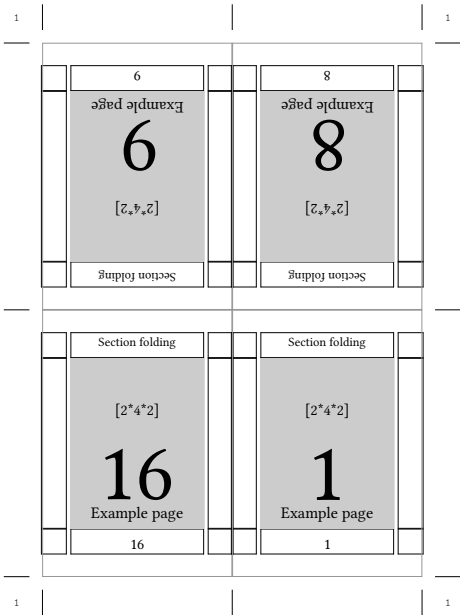


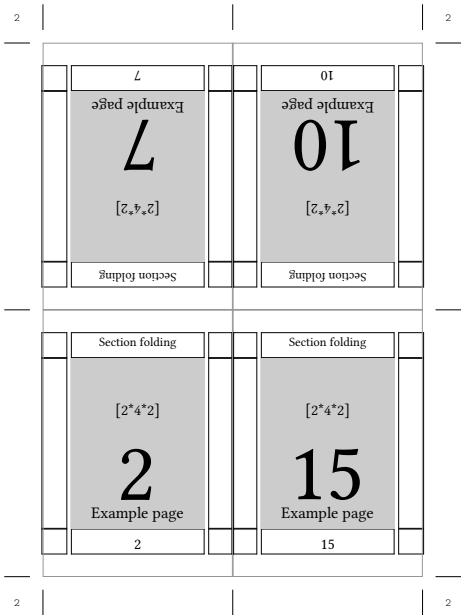
Figure 1.11 2 DOWN booklet: short edge binding

For those who want to print their own book with sections on the office printer ConT<sub>E</sub>Xt offers four schemes which use 2, 3 and 4 sheets of paper respectively to form a section.

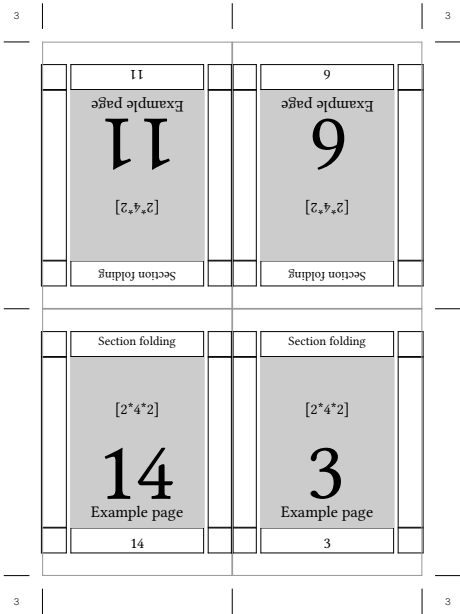
Arrangement	Result	Number of pages
<code>\setuparranging [2*4*2]</code>	section: 2 sheets, 4 pages front and backside =	16 pages
<code>\setuparranging [2*2*4]</code>	section: 4 sheets, 2 pages front and backside =	16 pages
<code>\setuparranging [2*2*2]</code>	section: 2 sheets 2 × 2 pages =	8 pages
<code>\setuparranging [2*2*3]</code>	section: 3 sheets 2 × 2 pages =	12 pages



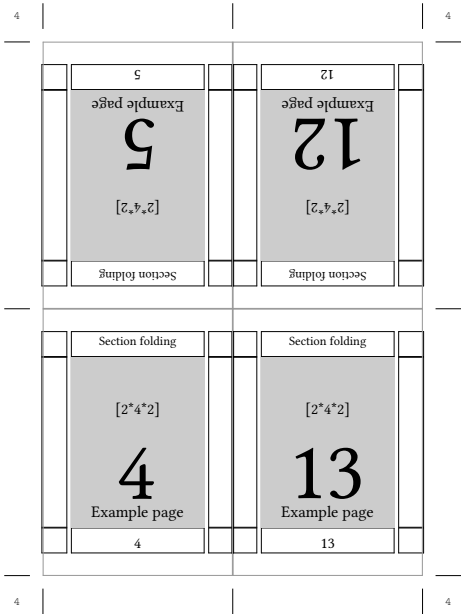
1<sup>st</sup> sheet recto



1<sup>st</sup> sheet verso

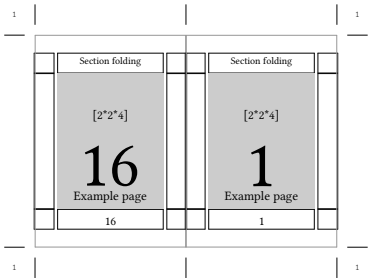


2<sup>nd</sup> sheet recto

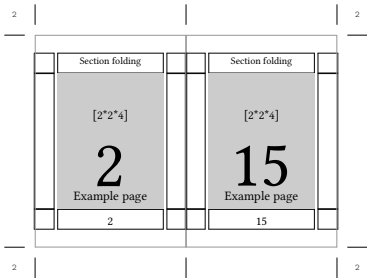


2<sup>nd</sup> sheet verso

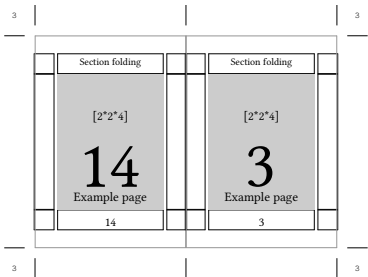
Figure 1.12 16 pages, 2 sheets



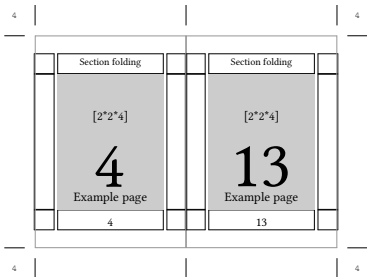
1<sup>st</sup> sheet recto



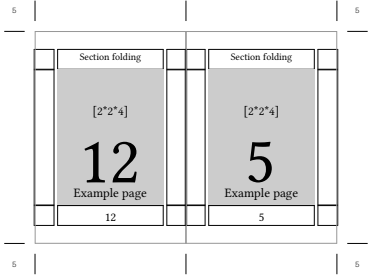
1<sup>st</sup> sheet verso



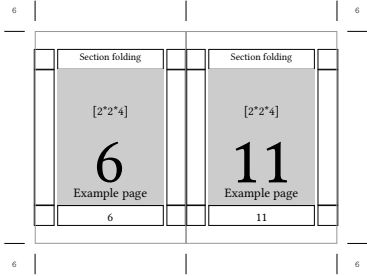
2<sup>nd</sup> sheet recto



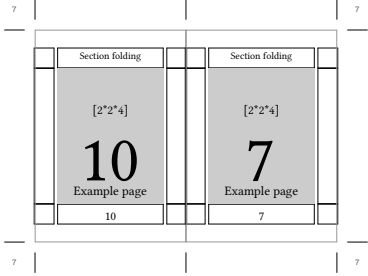
2<sup>nd</sup> sheet verso



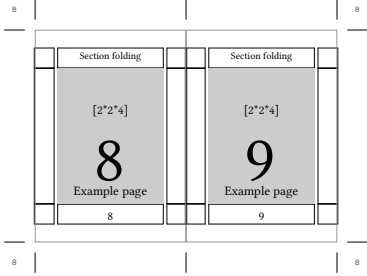
3<sup>rd</sup> sheet recto



3<sup>rd</sup> sheet verso



4<sup>th</sup> sheet recto



4<sup>th</sup> sheet verso

Figure 1.13 16 pages, 4 sheets

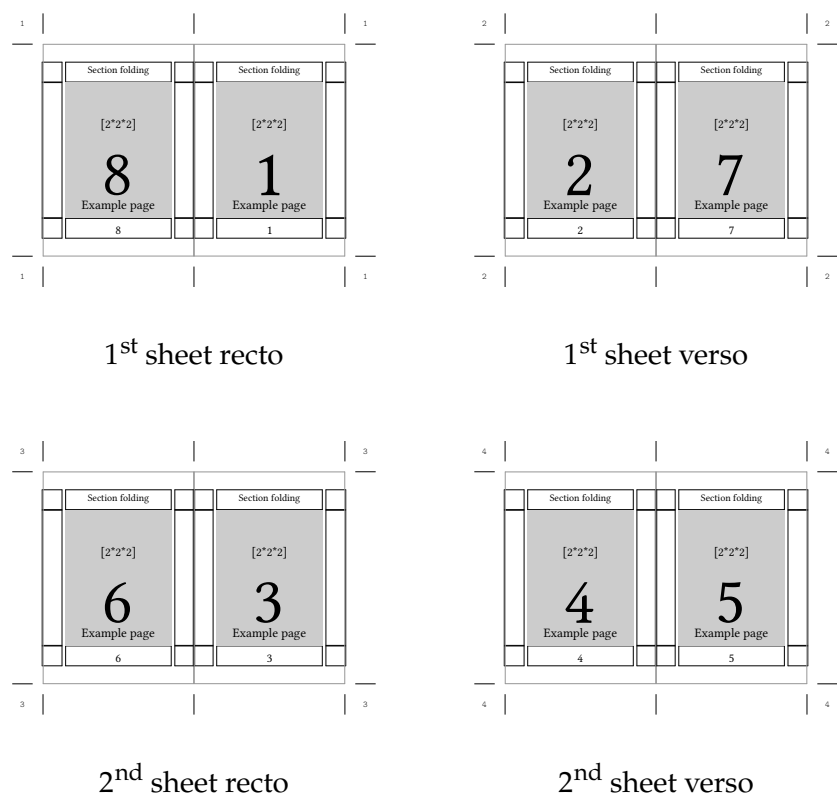
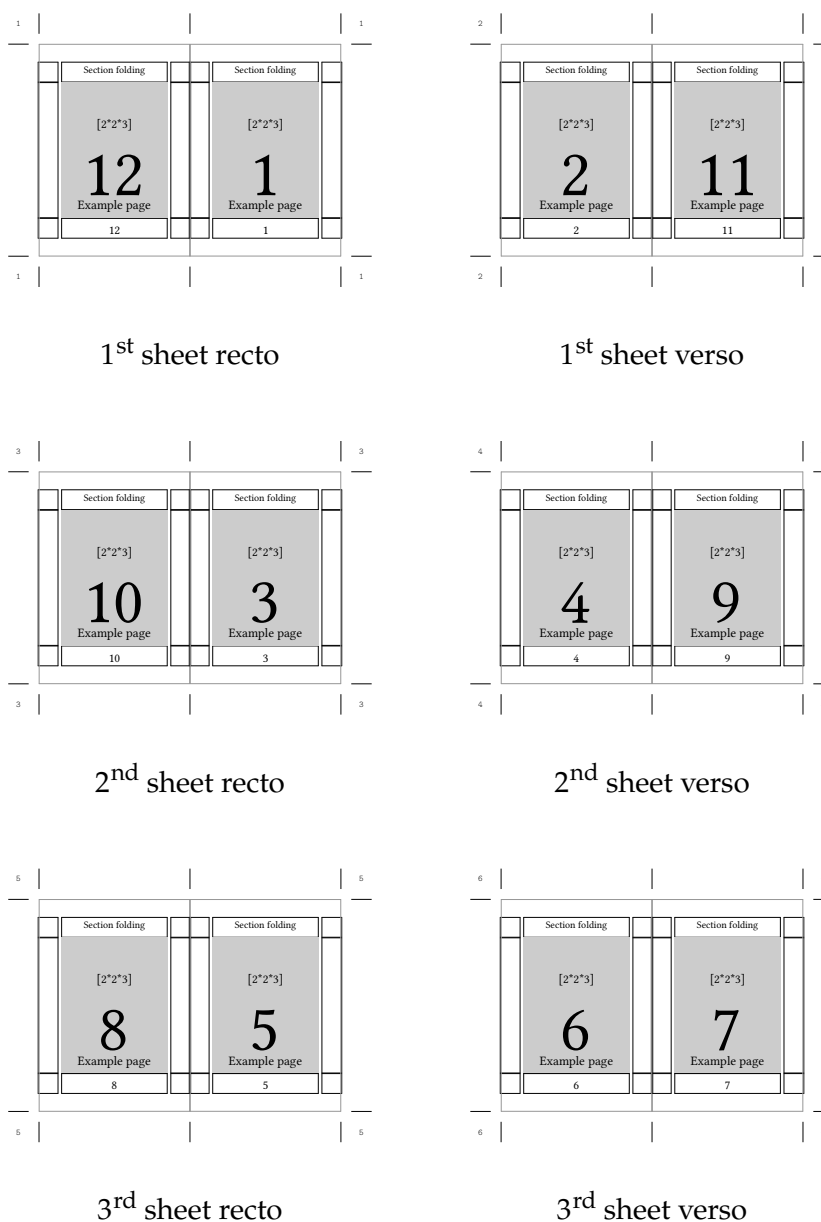


Figure 1.14 8 pages, 2 sheets



**Figure 1.15** 12 pages, 3 sheets

Yet another way to print sections is to use z-folding, which is a zig-zag folding combined with a single fold in the spine. ConTeXt comes with two types of sections, one with 12 pages and one with 16 pages.

Next to the imposition schemes involving folding ConTeXt offers possibilities to arrange pages in such a way, that after cutting the pile of sheets book blocks can be assembled. The resulting book block consists of loose sheets of paper and will be glued along the spine to prepare e.g. a paperback.

ConTeXt has an arranging scheme for two odd pages above each other and two even pages on the backside of the sheet. In order to build the book block the sheets need to be cut and the two piles must be merged.

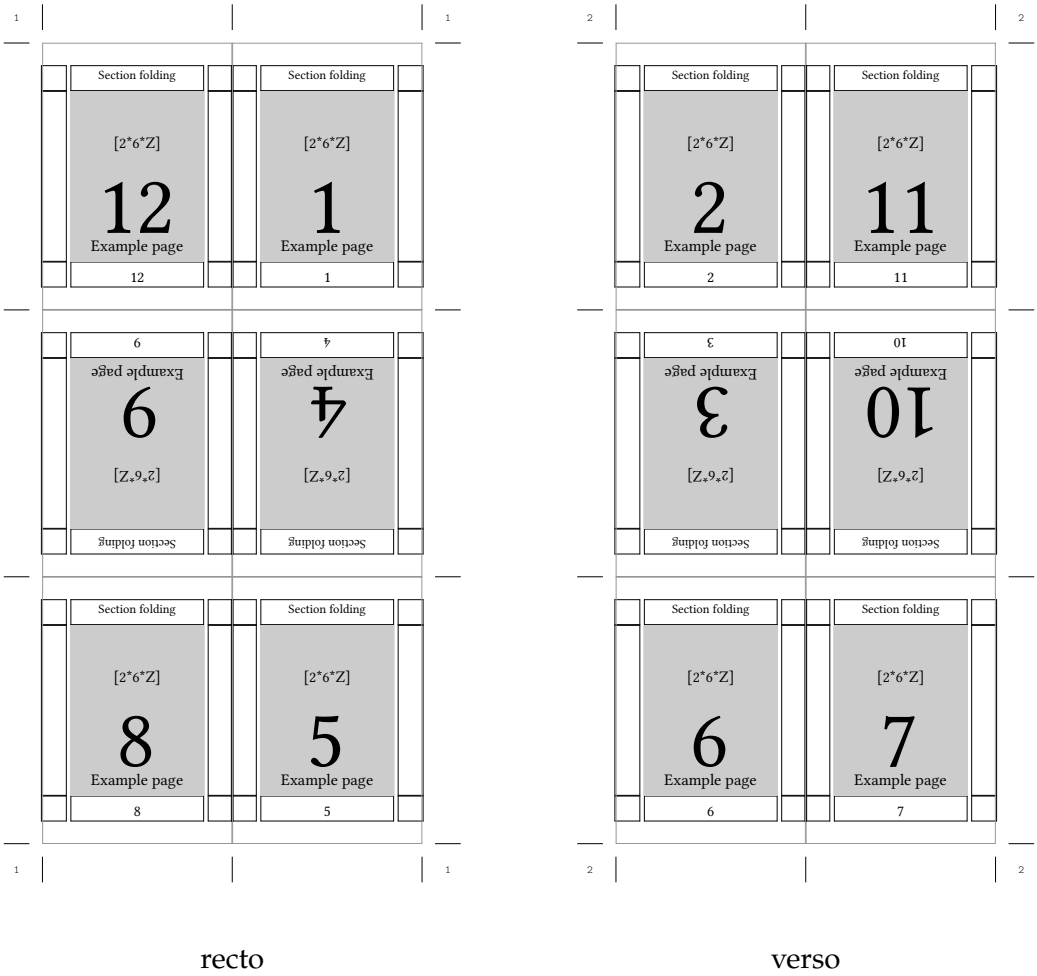


Figure 1.16 12 pages z-folding



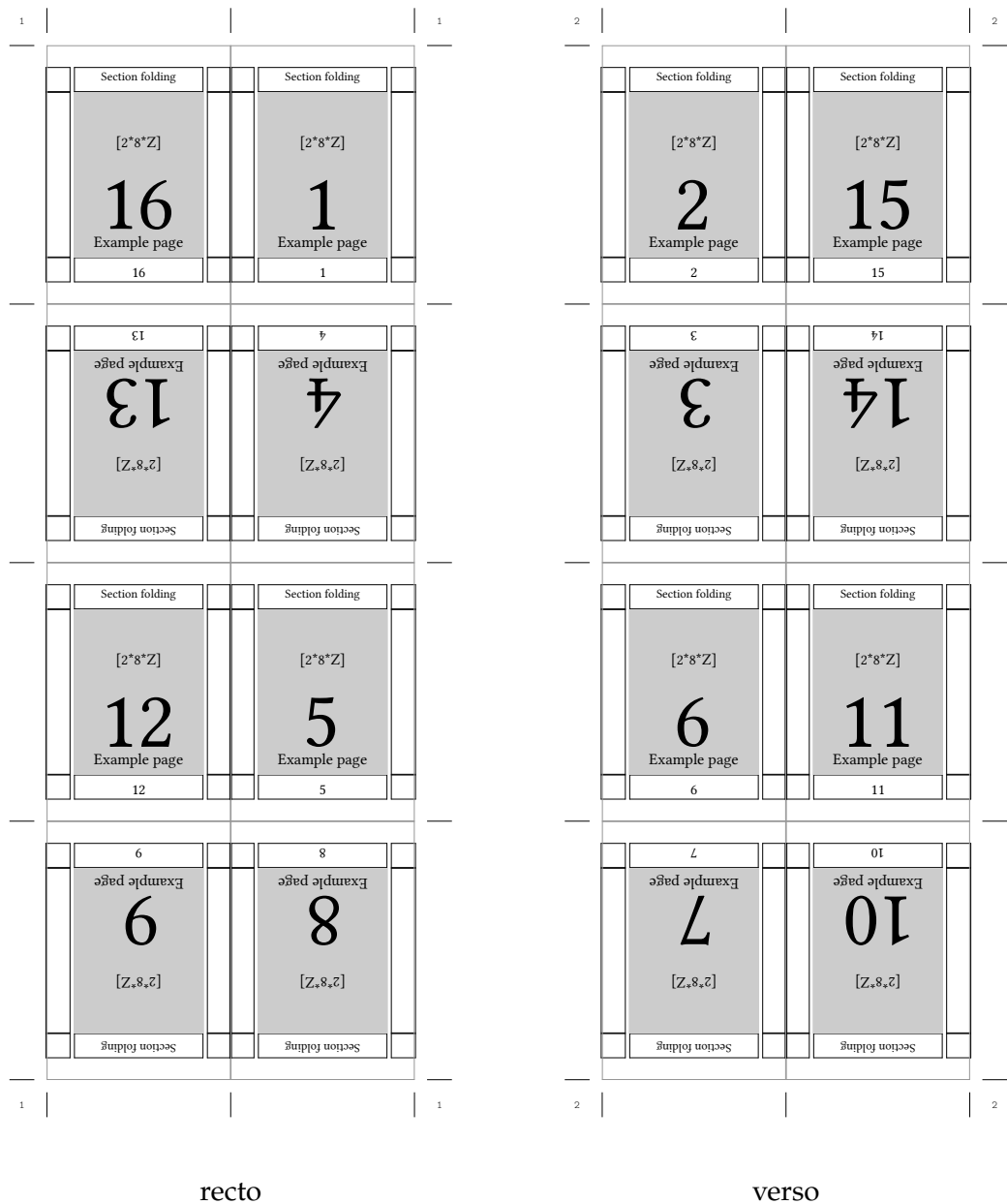


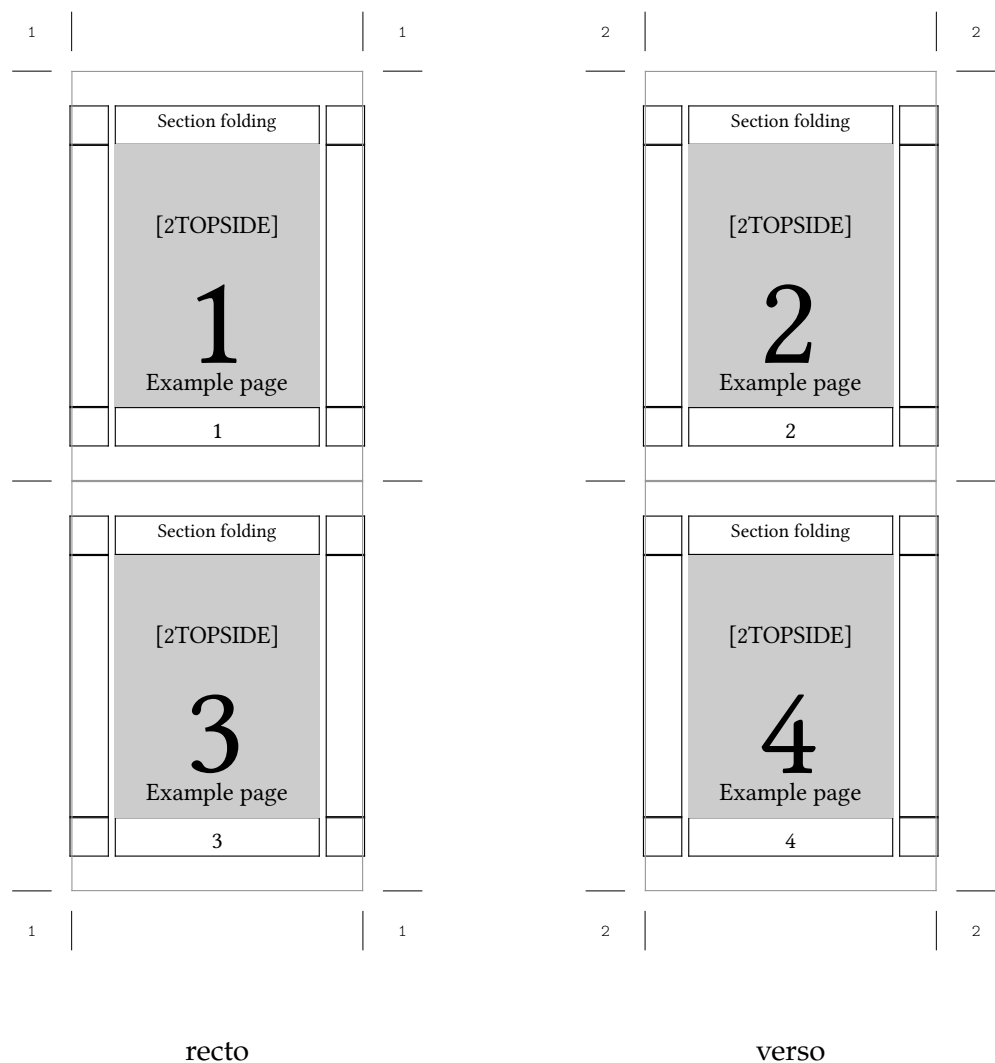
Figure 1.17 16 pages z-folding

Arrangement	Result	Number of pages
<code>\setuparranging [2TOPSIDE]</code>	recto 2 odd pages, verso 2 even pages = per sheet	4 pages

The following schemes can be used for the preparation of handouts from presentations. They also can be used to assemble book blocks after cutting and merging the piles.

The first scheme arranges 4 pages on the front side of the sheet.

The second scheme puts two pages on the front side of a sheet next to each other.



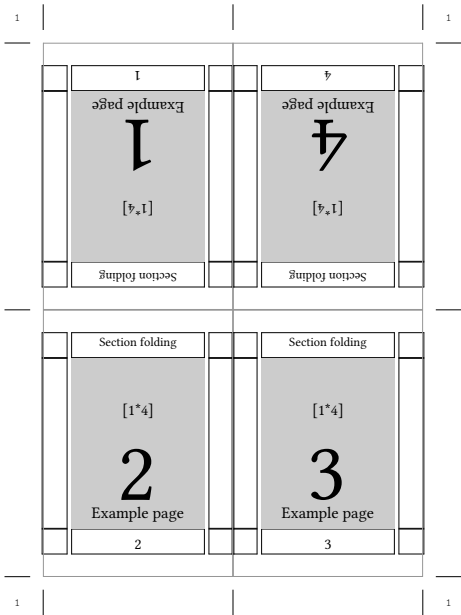
**Figure 1.18** 4 pages, 1 sheet

The third scheme works like the previous one but instead of putting the pages next to each other the pages are placed on top of each other.

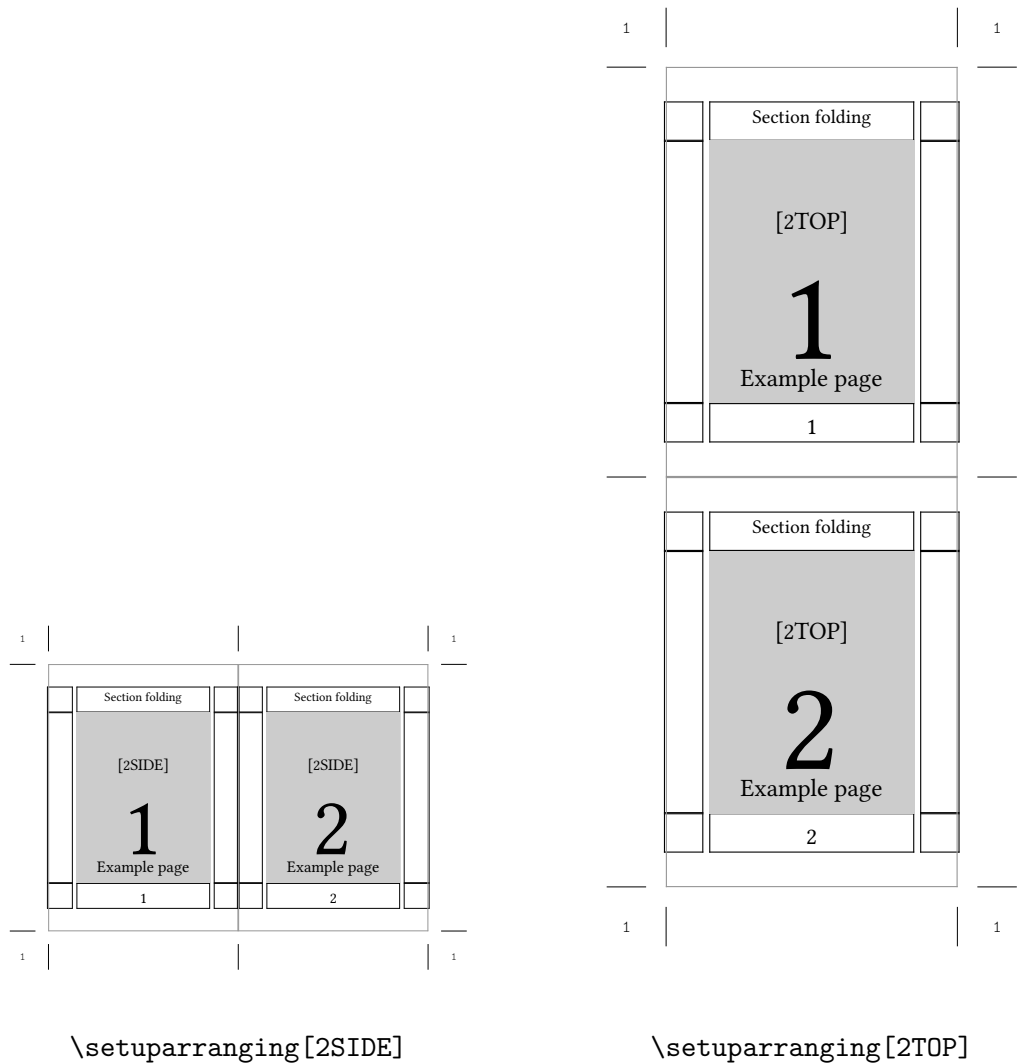
Arrangement	Result	Number of pages
<code>\setuparranging [1*4]</code>	one sheet recto 4 pages =	4 pages
<code>\setuparranging [2SIDE]</code>	one sheet recto 2 pages =	2 pages
<code>\setuparranging [2TOP]</code>	one sheet recto 2 pages =	2 pages

There are a couple of arranging schemes for special purposes. The first one places 8 pages on the recto side of the paper. It is intentioned for single sided prints only. The arrangement is made in such a way, that it is possible to fold the paper into a booklet, where while turning the pages now empty pages are shown.

Arrangement	Result	Number of pages
<code>\setuparranging [1*8]</code>	“section”: one sheet 1 × 8 pages =	8 pages



**Figure 1.19** 4 pages, singlesided, 1 sheet



**Figure 1.20** 2 pages, single sided, 1 sheet

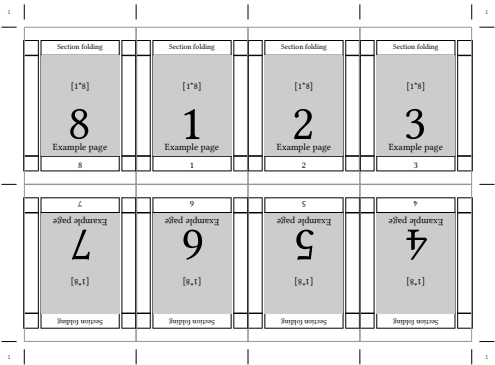


Figure 1.21 8 pages, single sided, 1 sheet

For those who will have to produce name-card displays for e.g. conferences or for the preparation of menu-displays in a restaurant the following schemes might be of use.

Arrangement	Result
<code>\setuparranging [1*2*Conference]</code>	one sheet 2 pages on top of each other, 1 page rotated
<code>\setuparranging [1*4*Conference]</code>	one sheet 2 odd pages next to each other, even page rotated on top

There are diary systems, where three pages are place next to each other. The following scheme provides this arranging scheme:

Arrangement	Result	Number of pages
<code>\setuparranging [3SIDE]</code>	3 odd pages recto, 3 even pages verso =	6 pages

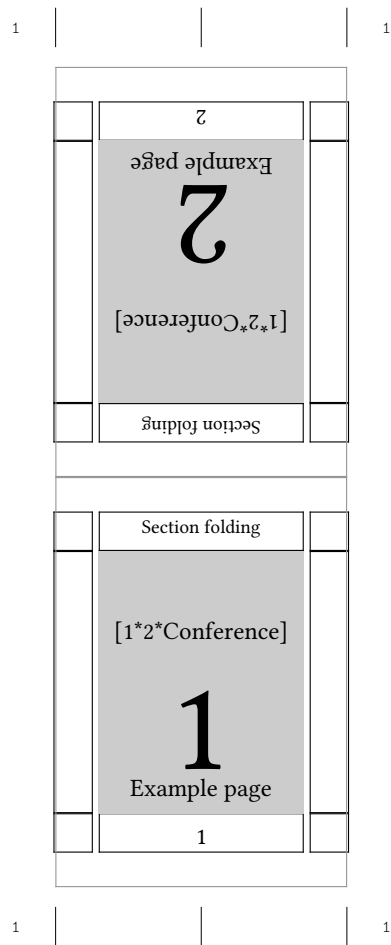
ConT<sub>E</sub>Xt also arrange pages for the production of flyers. There is a great variety of such flyers. ConT<sub>E</sub>Xt supports flyers with 6, 8, 10 and 12 pages. It is also possible to make a flyer with 12 pages which is folded like a map.

Arrangement	Result	Number of pages
<code>\setuparranging [TRYPTICHON]</code>	Leaflet: one sheet 2 × 3 pages =	6 pages
<code>\setuparranging [DOUBLEWINDOW]</code>	Leaflet: one sheet 2 × 4 pages =	8 pages
<code>\setuparranging [ZFLYER-8]</code>	Leaflet: one sheet 2 × 4 pages =	8 pages
<code>\setuparranging [ZFLYER-10]</code>	Leaflet: one sheet 2 × 5 pages =	10 pages
<code>\setuparranging [ZFLYER-12]</code>	Leaflet: one sheet 2 × 6 pages =	12 pages
<code>\setuparranging [MAPFLYER-12]</code>	Leaflet: one sheet 2 × 6 pages =	12 pages

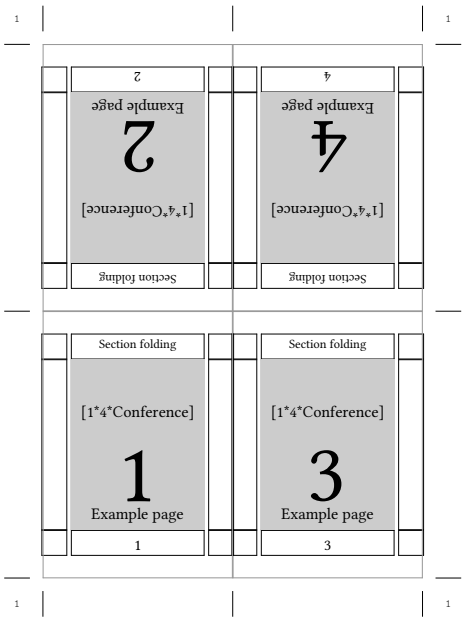
As a representative of the Z-folded flyers the flyer with 8 pages is shown.

Last but not least is the X-Y-arrangement of pages. This scheme is intended for the placement of a number of pages in sequence on a single sided sheet of paper e.g. on sheets carrying labels or for the placement of other information which must return several times on a sheet.

Before issuing the command `\setuparranging[XY]` the xy-arrangement must be setup. For this purpose the command `\setuppapaper[...]` is used.

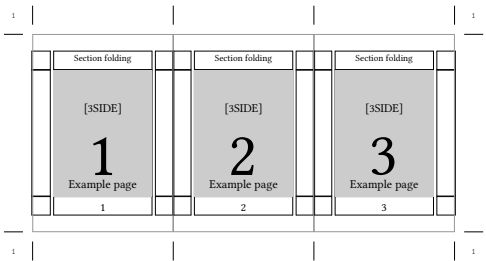
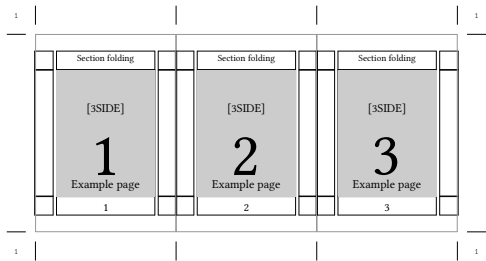


1 card with 2 pages



1 card with 4 pages

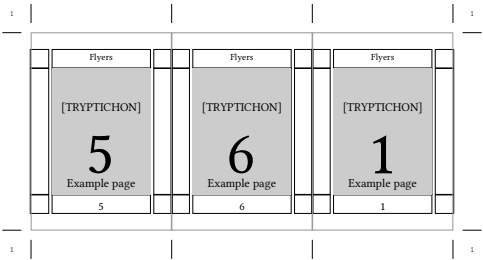
Figure 1.22 Display cards



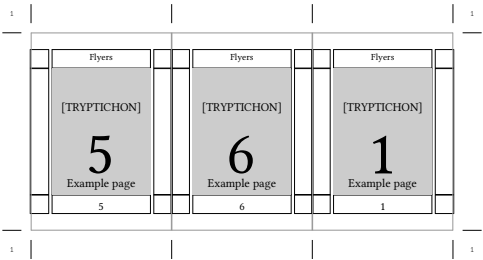
3 pages recto

3 pages verso

Figure 1.23 3 pages per side

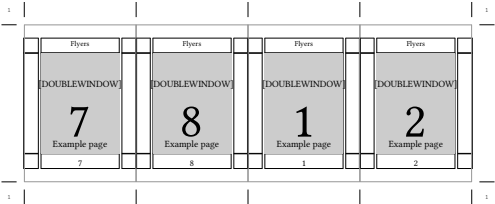


3 pages recto

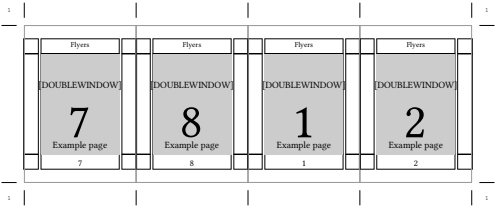


3 pages verso

Figure 1.24 Tryptichon type of flyer

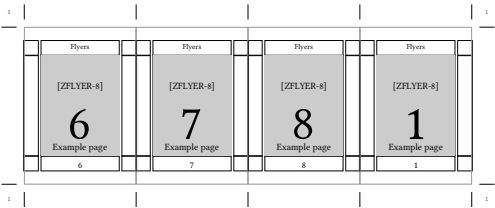


4 pages recto

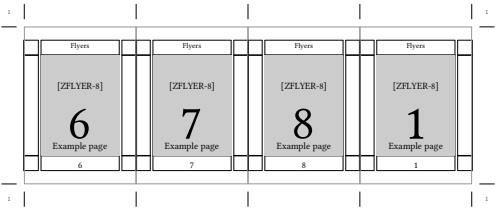


4 pages verso

Figure 1.25 Double window type of flyer



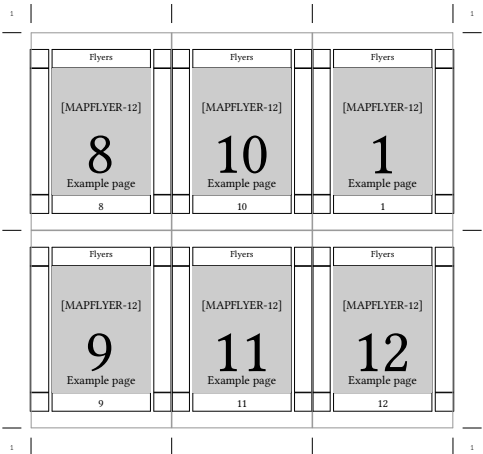
4 pages recto



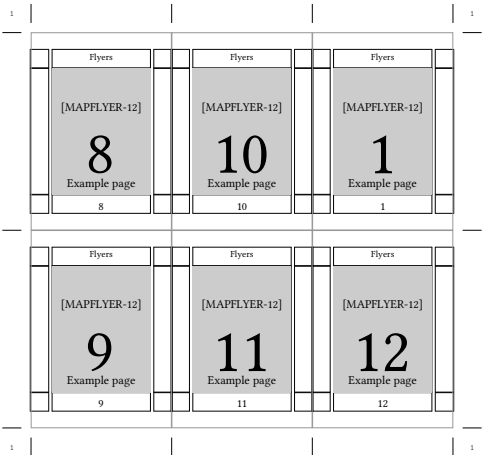
4 pages verso

Figure 1.26 Z-folded type of flyer





6 pages recto



6 pages verso

Figure 1.27 Map type of flyer

```
\setuppaper [...,*...]  
  
* paper      = IDENTIFIER  
  page      = IDENTIFIER  
  nx        = NUMBER  
  ny        = NUMBER  
  width     = DIMENSION  
  height    = DIMENSION  
  topspace  = DIMENSION  
  backspace = DIMENSION  
  option    = max fit
```

'nx' denominates the number of pages in the x-direction and 'ny' determines the number of pages in the y-direction. With 'dx' and 'dy' the whitespace between the pages in x and y direction can be set.

Arrangement	Result	Number of pages
<code>\setuparranging [XY] +</code> <code>\setuppaper [dx=,dy=,nx=,ny=]</code>	$nx \times ny$ pages, single sided =	$n \times m$ pages

There is culprit in arranging pages. If multiple layers of paper are folded, the outermost paper will require more width because it has to turn around the inner paper layers. This effect occurs as well in the spine folds as also in the head folds. How much width is required depends on the number of folds and the thickness of the paper. In professional book printing this effect is accounted for by displacing the pages depending on their position in horizontal and vertical direction. The result is that there will be a perfect look-through registering of all pages. There are no simple rules to indicate the required amount of displacement. Mostly it is a matter of experience to set up the page shift information.

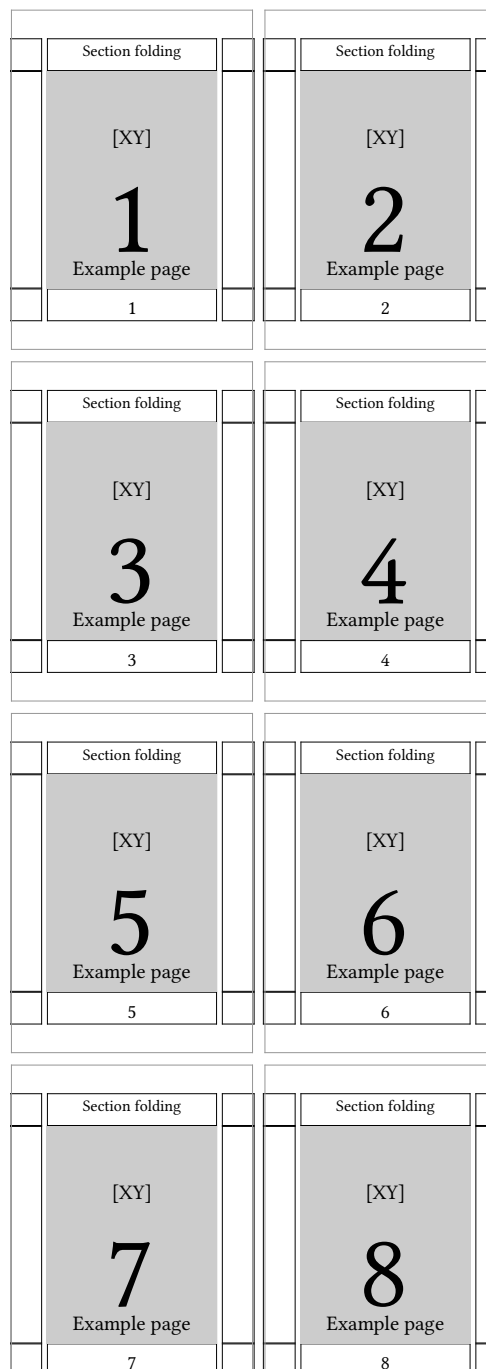
ConTeXt is equipped with a mechanism, which allows to move pages on a sheet apart from each other in horizontal as well as in vertical direction. The mechanism is build on two shift-lists, one for horizontal and one for vertical page shifting. The mechanism works through cycling over the lists which contain a shift amount for each page in a section. For filling in such a shift-list knowledge and understanding the position of a page on the printed sheet is necessary.

In order to use a horizontal shift list this list must be defined and setup.

For a section of 16 pages a horizontal shift list is filled in where for each page the amount of displacement is given. Such a list could look as follows:

```
\definepageshift[Hor][horizontal]
  [0.25mm, %1
  -0.25mm, %2
  0.15mm, %3
  -0.15mm, %4
  0.05mm, %5
  -0.05mm, %6
  0mm, %7
  0mm, %8
  0mm, %9
  0mm, %10
  0.05mm, %11
  -0.05mm, %12
  0.15mm, %13
  -0.15mm, %14
  0.25mm, %15
  -0.25mm] %16
```

For illustration purposes the following list for horizontal page-shift with exaggerated values is used in a Z-folding with 12 pages.



**Figure 1.28** 8 pages, singlesided, 1 sheet, XY-arrangement

```
\definepageshift[Hor][horizontal]
    [1mm,    %1
    -1mm,    %2
    0.5mm,   %3
    -0.5mm,  %4
    0mm,     %5
    0mm,     %6
```

```

0mm,      %7
0mm,      %8
0.5mm,    %9
-0.5mm,   %10
1mm,      %11
-1mm]     %12

```

In a similar fashion also vertical shift lists can be defined.

```

\definepageshift[Vert][vertical]
  [1.5mm, %1
  1.25mm, %2
  0.75mm, %3
  1.0mm,  %4
  1.0mm,  %5
  0.75mm, %6
  1.25mm, %7
  1.5mm,  %8
  1.5mm,  %9
  1.25mm, %10
  0.75mm, %11
  1.0mm,  %12
  1.0mm,  %13
  0.75mm, %14
  1.25mm, %15
  1.5mm]  %16

```

For each page in a section the shift amount must be indicated. The above presented list has exaggerated values just for making clear what happens:

While arranging these lists can be used in the following way:

Only one list is used:

```
\setuppageshift[paper][Hor]
```

or

```
\setuppageshift[paper][Vert]
```

Both lists are used:

```
\setuppageshift[paper][Hor][Vert]
```

The next examples show the cooperation of the commands `\setuppapersize`, `\setuplayout` and `\setuparranging`.

```

\setuppapersize      [A7][A3,mirrored] %negative creates an out of memory error
in Acrobat 8.2.2. on the MAC OSX 10.6.3
\setuparranging      [2*8,rotated,doublesided]
\setuppagenumbering  [alternative=doublesided]

```

With the above shown preamble you get sections of 16 pages of the size of A7, where both sides of the A3 paper carry 8 pages [2\*8]. For two reasons the A7 pages must be rotated on the

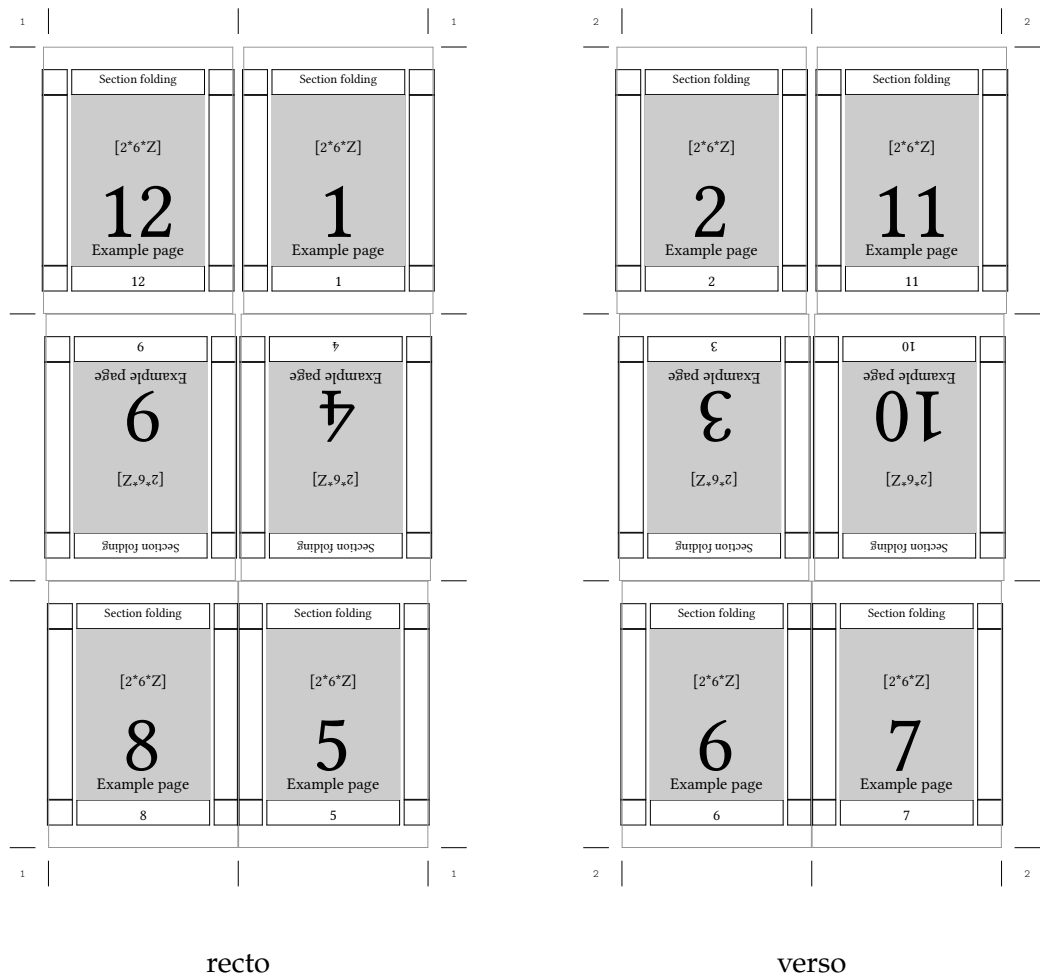


Figure 1.29 Horizontal page-shift

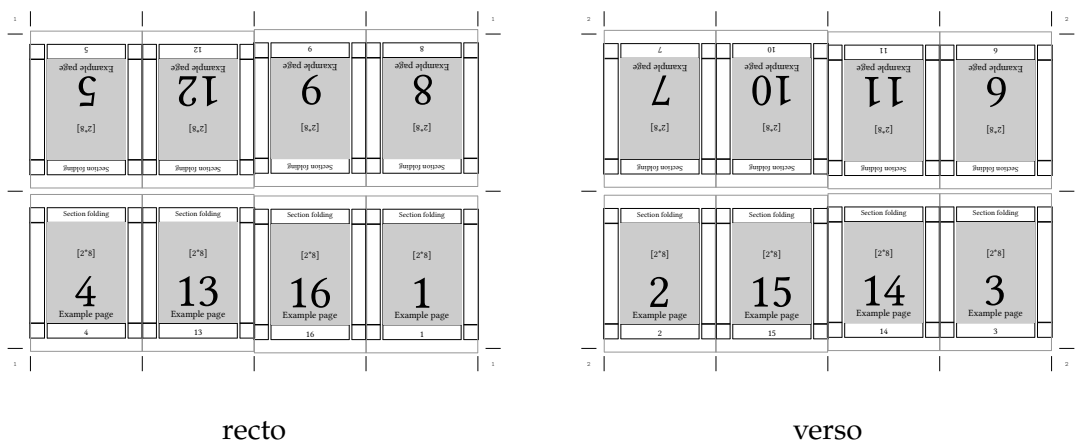


Figure 1.30 Vertical page-shift

paper. First in this imposition scheme there will be 4 A7 pages next to each other so they need to be aligned along the long edge of the A3. Secondly and this is important for book-printing,

the grain direction of the paper must be in the direction of the spine i.e. in the height of the A7. Since A3 has its grain direction normally along the short edge it is correct to rotate the A7 pages. Further more there is the ‘doublesided’ directive in the `\setuparranging` command. This is to rotate the whole content of the verso side of the A3 paper by 180° in order to enable automatic double sided printing on the printing machine. `\setuppagenumbering` tells ConT<sub>E</sub>Xt to use a doublesided lay-out, resulting in left and right pages.

Yet there is inside the `\setuppapersize` command the directive ‘mirrored’. Using this directive, the content of the A3 paper is mirrored along the long edge of the paper, this results in mirrored typeset text.

```
\setuppapersize      [A5] [A3]
\setuparranging      [2UP,rotated,doublesided]
\setuppagenumbering  [alternative=doublesided]
```

What this does is placing two A5 pages side by side on a A3 sheet of paper. Both the page and the paper are in portrait orientation. Because A5 fits better on a A3 when the page is rotated the `\setuparranging` command carries the ‘rotated’ directive. The resulting sheet of paper will be printed on an automatic double-sided printing machine. Often these machines require, that the verso side of the paper is printed reversed, this is achieved with ‘doublesided’ in the `\setuparranging` command.

Instead of using the ‘rotated’ directive in `\setuparranging` you can also say:

```
\setuppapersize      [A7] [A3,landscape]
\setuparranging      [2*8,doublesided]
\setuppagenumbering  [alternative=doublesided]
```

You rotate the A3 paper by means of the ‘rotated’ directive in `\setuppapersize`.

There is one thing which should be kept in mind when using `\setuparranging`: T<sub>E</sub>X compilations with ConT<sub>E</sub>Xt are most of the time multi-pass runs. If there is a table of content or other lists, this information is stored in auxiliary files or tables in LuaT<sub>E</sub>X. In order not to loose the content of those lists it is important to run the file first without the `\setuparranging` command enabled. If all went well, run the file a single time with the `\setuparranging` command enabled.

## 1.8 Logo types

Logos were removed in mkiv.