

Replacing L^AT_EX_{2 ϵ} standard classes with KOMA-Script

Yuri Robbers, Markus Kohm and Rasmus Pank Roulund

Email yuri.robbers@gmail.com
Website <http://www.komascript.de/>

Abstract KOMA is a complete replacement of the standard L^AT_EX_{2 ϵ} classes. It is aimed more at European typography but is easily configurable. Some of KOMA's extensions and ways to configure document layout are also available in other document classes, such as the L^AT_EX_{2 ϵ} standard classes. This paper discusses just a few parts of KOMA, especially those that have to do with page layout and with writing letters.

1 Introduction

KOMA-Script is a complex bundle of classes intended to replace the L^AT_EX_{2 ϵ} standard classes. The standard look and feel of KOMA-Script is rather different from the standard classes and is inspired by European typographical standards. Since the whole design philosophy of KOMA-Script is new, it offers all the commands that are offered by the standard classes (except for the `letter` and `slides` class) but not vice versa. Generally speaking the KOMA classes offer far more options for adaptation and configuration than standard L^AT_EX_{2 ϵ} . Using the KOMA classes, it should be relatively easy to obtain the look and feel that you like.

Some of the KOMA-Script options are also made available in packages that can easily be loaded in your own favourite class (for example the AMS classes) by including a `\usepackage{}` command with one of the special KOMA-Script packages (see below for details).

Most modern T_EX-distributions already include KOMA-Script; but if your installation does not, then download a copy from CTAN://macros/latex/contrib/koma-script or from <http://developer.berlios.de/project/koma-script3/>. Installation instructions are rather straightforward and covered in detail in the included README and INSTALL files.

2 Page layout using `typearea.sty`

2.1 Some theoretical considerations

When looking at a page of printed text, one may notice that it consists of the actual text and the margins. Sometimes there are headers and footers as well in between the main text body and the top and bottom margin respectively. When the printed text is a book, there was some additional white space that was printed but is hidden in the binding. Setting up a page with optimal margins and text area is called *page layout*.

Page layout doesn't depend only on fashions that change over time and from one culture to another; it also have a firm basis in human visual perception. Some texts are easier to read than others, and the two most important factors determining readability are page layout and the choice of typeface (Bringhurst, 2004).

KOMA-Script implements a series of modern European page layouts using its `typearea` package. The basic tenets of this series are as follows:

- (a) the size of the top margin compared to the bottom margin should be as 1:2;
- (b) for a single-sided document, the left and right margins should have equal width;
- (c) for a double-sided document, the total "inner" margin (i.e., the right-hand margin of the left page plus the left-hand margin of the right page) should equal the width of the "outer" margins (an example of this is shown in figure 5 on the fifth page, page 46, of the document by Willi Egger at http://www.ntg.nl/maps/pdf/30_13.pdf).

Please note that, when talking about page, we refer to the *visible* page, which may or may not be equal to the sheet of paper used to print the page on. In a book, for example, the sheet of paper contains both the page and the binding area.

Knowing how to set the margins in relation to each other is of course not enough. The text width and height should also be determined. The optimal line width is determined by many factors, including:

- typeface and font size used
- interline spacing

- language of the text and average word length
- number of columns
- page dimensions

Larger fonts allow wider lines than smaller fonts; serified fonts guide the eye along the text and hence allow for wider lines than sans-serifed fonts. More space between lines makes it easier to skip from the end of one line to the beginning of the next and, hence, allow for wider lines; but bear in mind that there is an optimum somewhere and increasing line space beyond this optimum will hinder rather than help the eye. Average word length and language play a role too. Generally, a text in a western language, typeset with a standard serified font of size 10 to 12 and the standard L^AT_EX line space should contain no more than, on average, fifteen words per line. This usually translates to 60 or 70 characters per line, including spaces and punctuation. Finally, the available space on the page plays an important role. One can imagine that the optimal line width for a paperback novel and a glossy magazine or even a newspaper would be rather different.

The optimal text height is influenced more by our sense of aesthetics than by readability. Nowadays the optimal text height is considered to best be linked to the text width.

When we take all these considerations together, we end up with an algorithm for dividing a page into equally sized rectangles. Suppose we assume nine rectangles horizontally, and nine vertically for a single-sided document (see again figure 5 on the fifth page, page 46, at http://www.ntg.nl/maps/pdf/30_13.pdf). This is called a *division factor* of nine. Mentally shift the gray area of the figure half a rectangle to the right; we would then have a left and right margin of one and a half rectangle each, a top margin of one rectangle, and a bottom margin of two rectangles. Our typearea would then be six rectangles wide and six rectangles high. For the two-page spread of a double sided document all heights would remain the same, but the vertical margins would change. The inner margin would consist of one rectangle, and the outer margin of two rectangles (as shown in the referenced figure). The text width would thus still be six rectangles, so changing from single-sided to double-sided does not change line breaks.

We can use any factor other than nine, as long as it is at least four. The reason for the lower limit of four is that, using the margin size rules described

above, `typearea` would be one rectangle high and one rectangle wide (which would already probably look silly); a smaller division factor would leave no space for the `typearea`.

The ideal division factor depends on the page size, the chosen font, etc. For an A4-sized page width a 12pt text as described above, one would probably take a division factor of 15. There are, however ways of having \LaTeX take care of determining an optimal division factor.

2.2 Implementing the theory using `typearea.sty`

KOMA-Script implements two options to control the page layout according to the theory described above. This is done using the package `typearea.sty`, which can be loaded with any document class. The KOMA-Script classes load this package automatically, so when using a KOMA document class, these

options can be given in the optional part of the `\documentclass[]{}{}` command. The options are called `BCOR` (Binding `COR`rection) and `DIV` (`DIV`ision factor).

Figure 1 shows an example using the standard $\text{\LaTeX}2_{\epsilon}$ report class. We use A4-paper to print a double sided report with the Bookman fonts, yielding an optimal division factor of 15, and we need a binding correction of 12mm because our report will be bound with glue. The first page of the resulting output in figure 2

There are some additional ways of using the `DIV` option. One of them is using `calc` rather than an explicit `DIV`-factor. This will tell \LaTeX to take care of determining an optimal `DIV`-factor. This makes life a lot easier for those users who lack the typographical (or mathematical) background to determine the optimum line width themselves. Please see figure 3 for an adaptation of our previous example to include this useful option.

A complete list of ways to use the `DIV`-option is:

DIV<factor> Set the division factor to be equal to the number <factor>.

```
\documentclass[a4paper,twoside]{report}
\usepackage{bookman}
\usepackage[DIV15,BCOR12mm]{typearea}
\usepackage{lipsum}
\begin{document}
\lipsum[1-8]
\end{document}
```

Figure 1: The source for a basic example of the use of `typearea.sty`

DIVcalc Calculate the optimal division factor for the current page and font settings.

DIVclassic Calculate a division factor for the current page and font settings that approximates the mediaeval page layout as closely as possible.

DIVcurrent Re-calculate the page layout using the DIV-factor that is currently in use.

DIVdefault Calculate the page layout with the standard value for the current page and font settings, or — lacking a standard value — use *DIVcalc*.

DIVlast Repeat the calculations for the page layout using the same DIV-argument that was used previously.

These uses of the DIV-option seem to imply that it is possible to do the calculations of DIV-factor and page layout multiple times in the same document. This is indeed the case, but we can of course only include the `typearea.sty` package once. If we want to do the necessary calculations again later, we use the command `\typearea[BCOR]{DIV}`. For example, if we want to use *DIVclassic* with an 8mm binding correction, we can use the command `\typearea[8mm]{classic}`. (Note that, as opposed to the options given when including `typearea.sty` with the `\usepackage` command, we do *not* include the letters DIV or BCOR explicitly in argument of the `\typearea` command.)

The option `current` as used in conjunction with DIV is also available for BCOR, for re-calculating the page layout including the existing BCOR value. The other DIV-options are *not* available with BCOR.

These options may seem rather esoteric at first sight, so it is probably best to give an example of their use. See the source in figures 4 and 5. Here we use

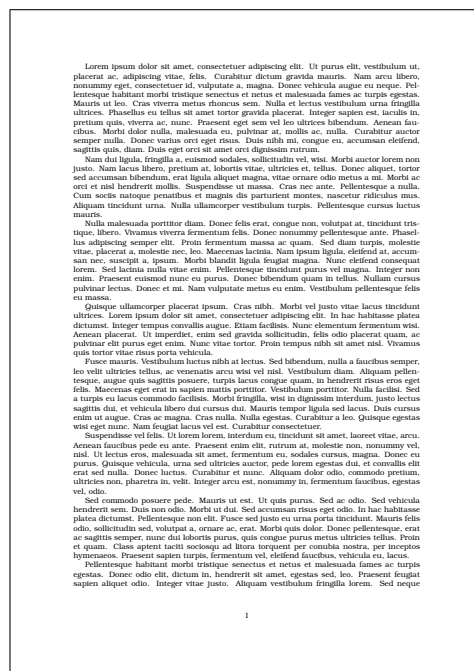


Figure 2: The first page of the output for a basic example of the use of `typearea.sty`.

the KOMA-Script replacement for the report class, which is called `screprpt`. We also use the default Computer Modern or Latin Modern font, but we want to increase the line spacing to 1.5, using the package `setspace`.

You may notice that the default paper size for KOMA-Script is already A4, and hence this does not need to be included with the `\documentclass` options. Paper size can be changed using any of `letterpaper`, `legalpaper`, `executivepaper`, `aXpaper`, `bXpaper`, `cXpaper`, `dXpaper` and `landscape`. Please note that the `X` in, for example, `aXpaper` is any reasonable integer. So it is possible to make a small booklet on paper one-fourth the size of A4-paper using the `a6paper`-declaration.

You may also notice that the resulting output is markedly different from the output for our first example as presented in figure 2. These differences are caused by the change in line spacing and the change in font. It is probably a good idea to do some experimenting yourself in order to get a feeling for the way the KOMA-Script page layout mechanism works.

Of course we have only been looking at very basic page layout options so far, and we have ignored the fact that for most documents it may not be all that clear what the actual margin is. This may seem a trivial issue. It isn't. Consider headers and footers. Do they belong to the actual text, or to the margin, or perhaps a bit to both? Things get even more complex when we start including marginal notes (the $\text{\LaTeX}2_{\epsilon}$ command `\marginpar{}` establishes these notes).

Should one of these elements be empty, then obviously it ought to be included with the margin. Similarly, a footer that contains no more than a page number is optically almost empty and should thus be considered part of the margin. If,

```
\documentclass[a4paper,twoside]{report}
\usepackage{bookman}
\usepackage[DIVcalc,BCOR12mm]{typearea}
\usepackage{lipsum}
\begin{document}
\lipsum[1-8]
\end{document}
```

Figure 3: The adapted source for our basic example, now using the `DIVcalc` option.

```
\documentclass[10pt,twoside,BCOR12mm,DIVcalc]{screprpt}
\usepackage{setspace}
\onehalfspacing
\typearea[current]{last}
\usepackage{lipsum}
\begin{document}
\lipsum[1-8]
\end{document}
```

Figure 4: The source for an example demonstrating the use of `DIVlast` and `BCORcurrent`.

however, the header is full of text (perhaps even underlined), then it is better to include it with the text body rather than with the margin. KOMA-Script is not able to decide whether headers, footers or marginal notes should be considered part of the text body or part of the margin. By default it considers these textual elements to be part of the margins, but this behaviour can be changed using the following options to the typearea command:

```
headinclude
headexclude
footinclude
footexclude
mpinclude
mpexclude
```

Using the *include* version of any of these options makes the respective textual element a part of the body; the *exclude* version makes it part of the margin. The size of the page header is 1.25 times the height of a regular line, but this can be changed using the headlines option with the typearea or usepackage command. For example, the command

```
\usepackage[1.8headlines]{typearea}
```

sets the header to a height of 1.8 line height.

It is also possible to set the type area to a particular fixed size and have the typearea.sty package calculate the right DIV-value to achieve this. For this use the command

```
\areaset[BCOR]{textwidth}{textheight}.
```

Should you, for example, desire to typeset a book of poetry with square text bodies of 15cm by 15cm and a binding correction of 1cm, use the command

```
\areaset[1cm]{15cm}{15cm}.
```

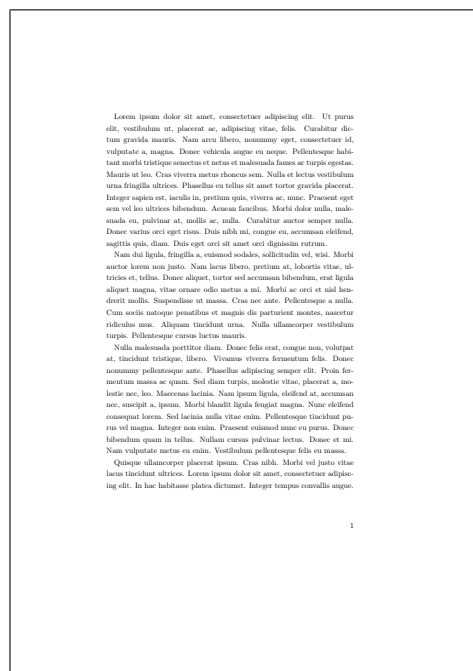


Figure 5: First page of the result for the example of figure 4.

This will give you a text body of the desired size and adjust the margins accordingly.

Should you require particular margin sizes that do not fit into the system described above, then it is better to use the `geometry`-package (Umeki, 2000).

3 A short remark on headers and footers

KOMA-Script has a style file named `scrpage2.sty` that allows all sorts of manipulations, both simple and complex, of headers and footers. It easily surpasses the capabilities of the better known `fancyhdr.sty`. Since this is a large and complicated style file with many options, we think it is better to just make the reader aware of its existence here and refer to the included documentation for now. We intend to write another paper dealing exclusively with the `scrpage2.sty` package for a future issue of *The PracT_EX Journal*.

4 An application: writing letters

This section is intended to show the power of using KOMA-Script. We could have chosen an article, report or book, but we have decided to show a letter, using the `sclttr2` class. We chose a letter for several reasons. First of all, the standard L^AT_EX_{2 ϵ} letter class is not used all that often since it looks rather outdated and does not lend itself well to customisation. The `sclttr2` class is quite the opposite: it looks quite modern already, and it is rather easy to make significant customisations and modifications. Also

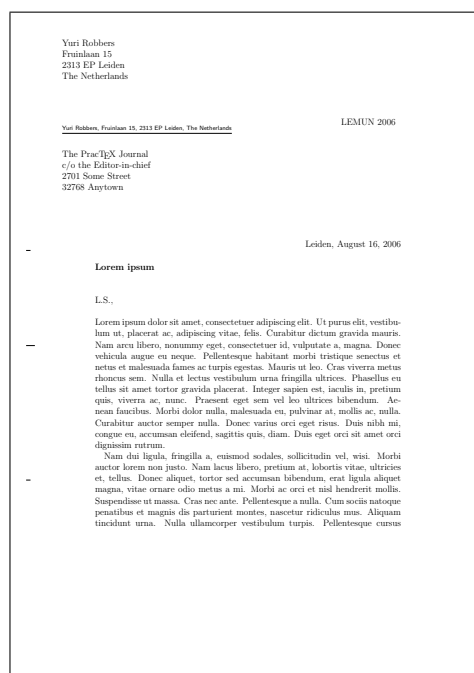


Figure 6: The first page of the output for our very basic KOMA-Script letter from figure 8.

letters tend to be rather short by nature, so it easy to include a complete example in this paper. Finally, while there are quite a few replacements for the other standard classes, there are very few available replacements for the letter class.

We do not intend to show the full power of the `scr1ttr2.sty` package—merely to show some of its options and the effects these options have. This means we will not show all the commands for changing the look and feel of the letter, nor any of the advanced options such as mail-merge set of commands for easily creating standard letters that get sent with slight alternations sent to many people based on databases of recipients. We will show (without explanation) how to save various sets of settings in small data files so that they can be easily reused when sending out, for example, a personal letter (using your home address and perhaps a closing with your first name) or a business letter (using your work address, the company logo, and a digitised signature). For these options see the KOMA-Script documentation (and perhaps we'll write another paper in a future issue of *The LaTeX Journal*).

Please refer to figure 8 for the example source code of a basic letter (the output is shown in figures 6 and 7).

The `DIV`-option is set to 9, the fontsize to 12, and a few more options are set. The `enlargefirstpage` option is specifically intended for letter writing. Generally on multi-page letters the first page has different settings from the next pages, and adding the option `enlargefirstpage` enables those.

Next a new variable, `company`, is defined in addition to the many pre-defined variables. We use this variable to set a company name in our letter.

We include some packages for including the signature and the dummy text and set a whole bunch of `komavar` variables. These variables can be defined in the letter itself (as shown in this example), but it is generally advisable to create

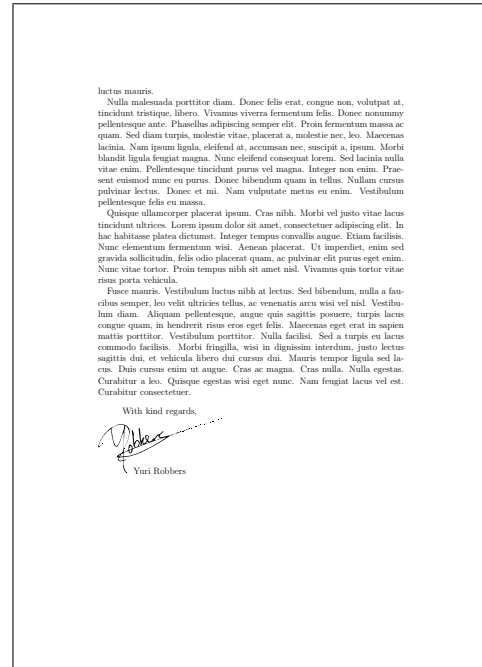


Figure 7: The second page of the output for our very basic KOMA-Script letter from figure 8.

```

\documentclass[%
  DIV=9,
  fontsize=12pt,
  pagesize,
  twoside,
  enlargefirstpage]{scrlltr2}

\newkomavar{company}
\setkomavar{location}{\ifkomavareempty{company}}{%
  \usekomavar*{company}%
  \usekomavar{company}}\}
}

\setkomavar{place}{Leiden}
\setkomavar{company}{LEMUN 2006}

\usepackage{graphicx}
\usepackage{epsf}
\usepackage{lipsum}

\nexthead{}

\setkomavar{fromlogo}{%
  \includegraphics[width=2cm]{LEMUN-logo2006}}
\setkomavar{fromname}{Yuri Robbers}
\setkomavar{fromaddress}{Fruinlaan 15\2313 EP Leiden\The Netherlands}
\setkomavar{fromphone}{+31-71-5121365}
\setkomavar{fromfax}{+31-71-5144543}
\setkomavar{fromemail}{robbers@lemun.org}
\setkomavar{fromurl}{http://www.lemun.org}
\setkomavar{frombank}{Postbank 9307157}
\pagestyle{empty}
\begin{document}
  \begin{letter}{%
    The Prac\TeX\ Journal\c/o the Editor-in-chief\2701 Some Street\32768 Anytown}
    \setkomavar{subject}{Lorem ipsum}
    \opening{L.S.,}

\lipsum[1-5]

    \closing{With kind regards,\}
\includegraphics[height=25mm]{yuri}\*[-20mm]
  }
  \end{letter}%
\end{document}

```

Figure 8: The source code for a basic KOMA-Script letter. Note that the `setkomavar` and `newkomavar` statements could all be moved to a separate `.lco` file for easy reuse.

```

\documentclass[%
  asymTypB,
  DIV=9,
  fontsize=12pt,
  pagesize,
  twoside,
  enlargefirstpage]{scrlltr2}

\setkomavar{place}{Leiden}
\setkomavar{company}{LEMUN 2006}% new variable defined in asymTypB.lco

\usepackage{graphicx}
\usepackage{epsf}
\usepackage[english]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{luatex}
\usepackage{lipsum}

\nexthead{}

\setkomavar{fromlogo}{\includegraphics[width=2cm]{LEMUN-logo2006}}
\setkomavar{fromname}{Yuri Robbers}
\setkomavar{fromaddress}{Fruinlaan 15\2313 EP Leiden\The Netherlands}
\setkomavar{fromphone}{+31-71-5121365}
\setkomavar{fromfax}{+31-71-5144543}
\setkomavar{fromemail}{robbers@lemun.org}
\setkomavar{fromurl}{http://www.lemun.org}
\setkomavar{frombank}{Postbank 9307157}
\pagestyle{empty}
\begin{document}
  \begin{letter}{%
    The Prac\TeX\ Journal\c/o the Editor-in-chief\2701 Some Street\32768 Anytown}
    \setkomavar{subject}{Lorem ipsum}
    \opening{L.S.,}

\lipsum[1-5]

    \closing{With kind regards,\%*1cm]
\includegraphics[height=25mm]{yuri}\%*[-20mm]
  }
  \end{letter}%
\end{document}

```

Figure 9: The source code for a fancier KOMA-Script letter. Note that a .lco file named asymTypB.lco is included in the documentclass-command.

a so-called `.lco` file, a *letter class option* file. Any definitions made in this `.lco` file will be automatically loaded by adding its name (minus the `.lco`-extension) to the `\documentclass` options.

Then we begin the document and the letter (with addressee) and create an opening and a subject, put the main body (in this case using dummy text generated with `\lipsum`), and a closing, including a picture file with a signature. Then we close letter and document.

Note that the resulting letter is rather basic but does include fold marks, and has the addressee in the exactly the right spot for an ISO-compliant window envelope. There is even an underlined return address in tiny letters in the window.

Even without further explanation, you will probably be able to adapt this simple example for your own use. But let's see some more of the power of KOMA-Script and create a slightly fancier letter. As a matter of fact, let's go all out and use one of the fanciest pre-defined styles that can be downloaded as a `.lco`-file from <http://www.komascript.de>. The source of our fancier letter, including some commands to include the beautiful Lucida Bright fonts from Bigelow & Holmes, is shown in figure 9, and the source code of `asymTypB.lco` in the appendix (page 14). The two output pages are displayed in figures 10 and 11.

Most users will probably not be willing or able to create such complex `.lco` files as shown in the appendix; however, several of them are freely available at <http://www.komascript.de>, and simpler ones can easily be created even by beginning users using the KOMA-Script documentation. Help is available at the forum at <http://www.komascript.de>.

This fancy example letter creates a column with relevant information about the sender, including an optional logo, in the right-hand margin of the first page.

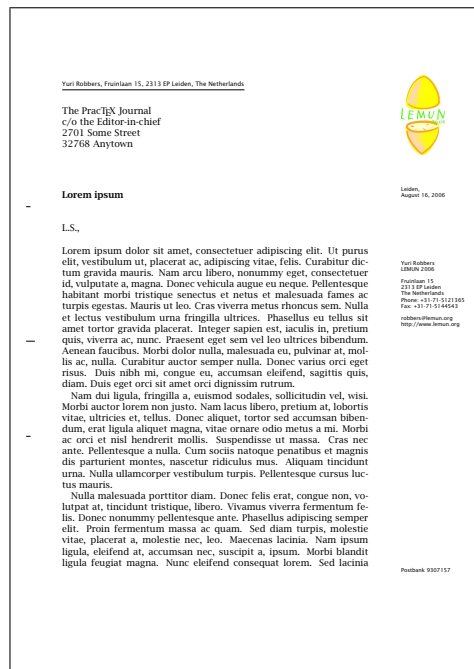


Figure 10: The first page of the output of our fancier KOMA-Script letter from figure 9.

The addressee and an underlined return address in small print are again displayed at exactly the right place for sending the letter in an ISO-compliant window envelope. Note that, since a `.lco` file was created anyway, the definition of the company keyword was moved there too. This `.lco` can be reused for business letters of various styles. It is straightforward to include multiple `.lco`-files in one document.

5 Concluding remarks

We hope a little bit of the power and configurability of KOMA-Script has been demonstrated to the reader. Of course the scope of this paper is necessarily limited; much more has been explained in the book by one of the authors (Kohm & Morawski, 2005).

KOMA-Script offers a uniform interface to document layout, with a replacement class for each of the original $\text{\LaTeX}2_{\epsilon}$ -classes, except the `slides` class. The standard layout is well-thought out, modern-looking and based on European rather than American typographical ideas. KOMA-Script is highly configurable and adaptable. Several extension packages for standard $\text{\LaTeX}2_{\epsilon}$ are no longer necessary when using the KOMA-script classes. Should one, however, decide to use another class, KOMA-Script still has much to offer through its classes: `typearea.sty`, `scrpage.sty`, `scrtime.sty` (has not been mentioned before but is used for displaying system time), `scrdate.sty` (used for displaying system time and date respectively in \LaTeX -documents), `verb+scraddr.sty+` (used for handling address files), and `scrfile.sty` (adds additional hooks pre and post inputting files and loading classes or packages). These classes offer advanced ways of manipulating document layout using powerful options and commands that are easy to learn.

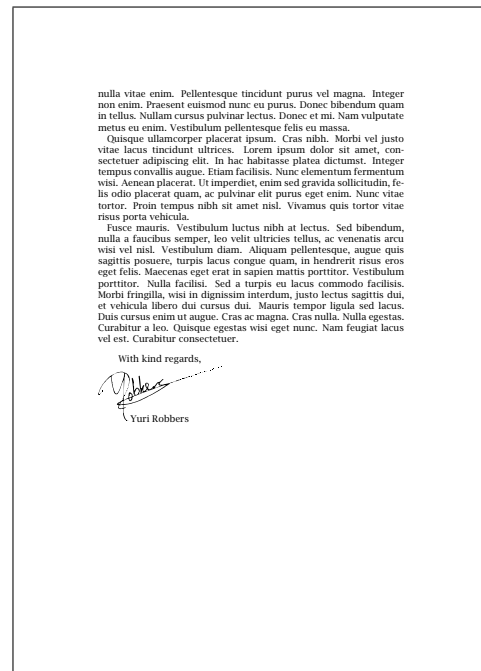


Figure 11: The second page of the output of our fancier KOMA-Script letter from figure 9.

We urge readers to give KOMA-Script a try and to experiment with the many available options.

We thank David Walden and the anonymous referees for their helpful comments, and we thank Karl Berry for solving a font problem.

References

BRINGHURST, R. (2004). *The Elements of Typographic Style, 2nd Edition*. Vancouver: Hartley & Marks.

KOHN, M. & MORAWSKI, J.-U. (2005). *KOMA-Script. Eine Sammlung von Klassen und Paketen für L^AT_EX₂ ϵ* . Heidelberg: DANTE.

UMEKI, H. (2000). The *geometry* package. CTAN://macros/latex/contrib/geometry/.

A The contents of the `asymTypB.lco`-file.

```
\ProvidesFile{asymTypB.lco}%
    [2005/04/09 v0.2a unsupported LCO-file]%
\LoadLetterOption{DINmtext}%
\setlength{\oddsidemargin}{\useplength{toaddrhpos}}%
\addtolength{\oddsidemargin}{-1in}%
% Take care that the shift stays intact even after recalculating the page
% layout (see Kohm & Morawski 2005, section C.7)
\l@addto@macro{\@typearea@end}{%
    \setlength{\oddsidemargin}{\useplength{toaddrhpos}}%
    \addtolength{\oddsidemargin}{-1in}%
}
\ifx\AtBeginDocument\@notprerr
    \KOMAOptions{DIV=last}%
\else
    \AtBeginDocument{%
        \g@addto@macro{\@typearea@end}{%
```

```

        \setlength{\oddsidemargin}{\useplength{toaddrhpos}}%
        \addtolength{\oddsidemargin}{-1in}%
    }%
} %
\fi
% continue as per description
\@setplength{firstheadvpos}{0pt}%
\@setplength{firstheadwidth}{\paperwidth}%
\@setplength{firstfootvpos}{\paperheight}%
\@addtoplength[-]{firstfootvpos}{\useplength{toaddrvpos}}%
\@addtoplength{refvpos}{-1.5\baselineskip}%
\newkomavar{company}%
\@newplength{infocolwidth}%
% Kohm & Morawski 2005, C.7. Modifikationen (Modifications)
\ifdim \textwidth<0.666\paperwidth
    \@setplength{infocolwidth}{.22222\paperwidth}%
\else
    \@setplength{infocolwidth}{0.1667\paperwidth}%
\fi
% continue as per description
\firsthead{%
    \fontsize{7}{8}\sffamily
    \hspace*{\fill}%
    \begin{picture}(0,0)%
        \put(0,0){\parbox[t]{\useplength{infocolwidth}}{%
            \vspace{\useplength{toaddrvpos}}%
            \usekomavar{fromlogo}%
        }}%
    }%
}%
\put(0,0){\parbox[t]{\useplength{infocolwidth}}{%
    \raggedright
    \vspace{\useplength{refvpos}}%
    \vspace{\useplength{refaftervskip}}%
    \usekomavar{place}\usekomavar{placeseparator}\\
    \usekomavar{date}\\[10\baselineskip]

```

```

\usekomavar{fromname}
\ifkomavareempty{company}{}{%
  \\\
  \usekomavar{company}%
}\[\baselineskip]
\usekomavar{fromaddress}\\\
\usekomavar*{fromphone}\usekomavar{fromphone}\\\
\usekomavar*{fromfax}\usekomavar{fromfax}%
\[\baselineskip]
\usekomavar{fromemail}\\\
\usekomavar{fromurl}\\\
}%
}%
\put(0,0){\parbox[t]{\useplength{infocolwidth}}{%
  \raggedright
  \vspace{\useplength{firstfootvpos}}%
  \vspace{-\footskip}%
  \usekomavar{frombank}\\\%
}%
}%
\end{picture}%
\hspace*{\useplength{infocolwidth}}%
}%
\l@addto@macro\@firstheadfootfield{\setkomavar{date}{}}
\nexthead{%
  \begingroup
  \@tempwattrue
  \if@twoside\ifodd\number\value{page}
    \else\@tempwafalse\fi\fi
  \if@tempwa
    \makebox[0pt][l]{%
      \hspace*{-\oddsidemargin}\hspace{-1in}%
      \hspace{\paperwidth}%
      \hspace{-\useplength{infocolwidth}}%
      \normalfont\fontsize{7}{8}\sffamily
    }
  }

```



```

\begin{picture}(0,0)
  \put(0,0){\parbox[t]{\useplength{infocolwidth}}{%
    \vspace*{-\topmargin}\vspace{-1in}%
    \vspace{-\headheight}%
    \vspace{\useplength{toaddrvpos}}%
    \begin{tabular}{@{}c@{}}
      \usekomavar{fromlogo}\\
      \usekomavar{company}\\
    \end{tabular}%
  }}%
\end{picture}%
}
\fi
\endgroup
}
\pagestyle{headings}%
\endinput

```