# `parrun`
# User Guide*

Massimiliano Dominici

2004/02/06

**Abstract**

The package `parrun` provides a set of macros useful for typesetting several (two) streams of text running parallel on the same physical page in a vertical layout.

## Contents

## 1 Introduction

`parrun` It may occur, especially in anthologies and collections, that a translation and the original parallel text must not be disposed on two different pages or columns, but one *above* the other. The package `parrun` has been designed to handle this situation. The user interface is simple and provides two environments in order to define the respective streams of text and a command for placing them on the page, saving for each of them an exact fraction of the bodytext.

`multicol` Finally, as `parrun` showed himself, during tests, incompatible with package `multicol`, I had to introduce an explicit option for typesetting in two or more columns the second stream of text.

---

*This file has version number v1.1a dated 2004/02/06.

# 2 User interface

## 2.1 Environments

`fframe`
`sframe`

The way to use the two environments is easy and there are no additional arguments. The environments do just what their names say, *i.e.* they place in a `\box` the text enclosed between the initial and the final declaration, following the usual formulation `\begin{`⟨*env*⟩`}` ⟨*general text*⟩ `\end{`⟨*env*⟩`}`.

## 2.2 The command `\Place`

The actual arrangement on the page of the content of the two `boxes`, following the user instructions, is the task of the command `\Place`. There are two ways to use it:

`\Place*`

- the starred version (`\Place*`), without arguments, tries to automatically compute the optimal fraction of `\texteight` to be reserved for each `box`, so you need not to bore yourself with the whole matter.

`\Place`

- the unstarred version is provided for the case the algorithm of autocomputation fails, and takes two arguments, *i.e.*: the space to be reserved on the page for the two `boxes`, expressed in fractions of `\textheight`. So, typing `\Place{423}{57}`, you mean that the page will be divided in two regions, whose height is `0.423\textheight` and `0.57\textheight` respectively.

`\Place` checks wether the text is ill-balanced, with a stream keeping on running while the other is already finished, and, in the case of the unstarred version looks also after the total fraction of the text actually placed on the page checking it doesn't exceed the bodytext. In both cases compilation is halted and a warning appears giving possibility to get more help. Then compilation goes on till end and a table of the most important quantities used by the package is typed at the end of the log file, providing a useful hint for values to be used with the unstarred version in the case the autocomputation algorithm gives a bad result.

## 2.3 The option `multicol`

`multicol`

As already mentioned, the package provides internally an option `multicol` for typesetting the content of the second stream of text on two columns. The user needs only to load the option together with the package (`\usepackage[multicol]{parrun}`). Changing the value of counter `\cnum`, the user can, as a consequence, vary the number of columns.

`\cnum`

## 2.4 Limitations

Involving boxes containing large portions of text, the package deals with large dimensions. TeX has very loose limitations in the use of large dimensions, yet these limitations exist and to find himself involved in such problems is easier than one can think. Obviously the use of a multicolumn layout increases the possibility to overrun the limits, so this solution should be used with care. The only apology I can make is that the package has been designed for such tasks that request no large amounts of text stuffed in a box. Maybe a different approach should be taken into consideration. Besides, large font dimensions and small heights of the bodytext can have as result an oscillating behaviour of the output which remains unbalanced.

# 3 The code

```
⟨*parrun⟩
```

## 3.1 Package identification

```
\NeedsTeXFormat{LaTeX2e}[2001/06/01]
\ProvidesPackage{parrun}[2004/02/06 v1.1a Package for parallel text]
\RequirePackage{ifthen} \RequirePackage{calc}
```

## 3.2 Conditionals and the option declaration

```
\newif\ifmultic@l
\DeclareOption{multicol}{\multic@ltrue}
\ProcessOptions
```

## 3.3 The beginning

We declare the `\boxes` . . .

```
\newbox\ffram
\newbox\sfram
```

. . . the counters (`\cnum`, defining the number of columns in the option `multicol` is by default set equal to two). . .

```
\newcount\k
\newcount\cnum
\cnum=2
```

. . . and some useful lengths.

```
\newlength\flength
\newlength\slength
\newlength\ffrac
\newlength\sfrac
\newlength\nop
\newlength\total
\newlength\actualheight
\newlength\initskip
\setlength{\initskip}{0pt}
\newdimen\colframsep
\newdimen\h
\newdimen\test
\newdimen\temp
\colframsep=8pt
```

## 3.4 Environments

`fframe`
`sframe`

The two environments `fframe` and `sframe` place respectively in the two `\boxes` `ffram` and `sfram` the ⟨*general text*⟩ enclosed between the initial and the final declaration.

```
\newenvironment{fframe}{\global\setbox\ffram=\vbox\bgroup}{\vfill\egroup}
```

`sframe`

For what concerns `sframe` it has to distinguish if the option `multicol` is in force or not. In the first case he has to redefine `\hsize` so that the text can be distributed on the specified number of columns.

```
\newenvironment{sframe}{%
    \ifmultic@l \k=\cnum \advance\k by -1
        \dimen0=\textwidth \divide\dimen0 by
```

```
          \cnum \advance\dimen0 by -\k\colframsep
          \hsize=\dimen0
      \fi
      \global\setbox\sfram=\vbox\bgroup}
  {\vfill\egroup}
```

## 3.5 Commands for placing `boxes`

`\vsplit`

They are the commands directly charged to place the text on the page. They call, directly or indirectly, the primitive `\vsplit` (for further details on its functioning see TEXbook [1]).

`\v@idb@xtwo`
`\b@xbalance`
`\v@idb@xbalance`
`\rigidbalance`

There are five macros: two for the option `multicol`, two for the package without options, and the last one shared (obviously it's `\v@idb@xtwo`, typesetting a void `\box` at the bottom of the page, when `\sfram` is void). Arguments are passed by `\Place`, consisting in fractions of `\textheight`, but, as can be easily seen, in the two macros used by `multicol` (`\b@xbalance` and `\v@idb@xbalance`) only two arguments are visible. The third has been swallowed by `\rigidbalance`, a macro to be examined in the following.

```
  \newcommand{\b@xbalance}{%
              \vbox to\initskip{}
              \vsplit\ffram to \flength
              \smallskip \hrule \smallskip
              \rigidbalance}

  \newcommand{\v@idb@xbalance}{%
              \vbox to\initskip{}
              \vbox to \flength{}
              \smallskip \hrule \smallskip
              \rigidbalance}

  \newcommand{\v@idb@xone}{%
              \vbox to\initskip{}
              \vbox to \flength{}
              \smallskip \hrule \smallskip
              \vsplit\sfram to \slength}

  \newcommand{\v@idb@xtwo}{%
              \vbox to\initskip{}
              \vsplit\ffram to \flength
              \smallskip \hrule \smallskip
              \vbox to \slength{}}

  \newcommand{\splitb@x}{%
              \vbox to\initskip{}
              \vsplit\ffram to \flength
              \smallskip \hrule \smallskip
              \vsplit\sfram to \slength}
```

`\rigidbalance`
`\dosplits`
`\dobalance`

This macro has been taken, with remarkable adjustments, from the Appendix "Dirty Tricks" of the TEXbook ([1]) and is needed to make sure that also on the last page the columns have all the same height. Namely, it checks the height of the content of the `box` and on the base of the outcome it calls the macro `\dosplits` which splits the text in columns of normal height, or the macro `\dobalance` which sets the height of the columns to the height of the remaining content of `\sfram` divided the number of columns and then splits the text.

```
\newcommand{\rigidbalance}{\hsize=\textwidth \k=\cnum
    \ifdim\ht\sfram>\cnum\slength
        \myline{\splittopskip=\h \vbadness=10000 \hfilneg
        \valign{##\vfil\cr\dosplits}}
    \else
        \myline{\temp=\ht\sfram \advance\temp by \baselineskip
        \divide\temp by \cnum \splittopskip=\h \vbadness=10000
        \hfilneg \valign{##\vfil\cr\dobalance}}
    \fi}
\newcommand{\dosplits}{\ifnum\k>0 \noalign{\hfil}
    \splitoff\global\advance\k-1\cr\dosplits\fi}
\newcommand{\splitoff}{
    \vsplit\sfram to \slength}
\newcommand{\dobalance}{\ifnum\k>0 \noalign{\hfil}
    \finalbalance\global\advance\k-1\cr\dobalance\fi}
\newcommand{\finalbalance}{
    \vsplit\sfram to \temp}
```

`\myline` I had also to define a new command `\myline` restoring the original meaning of `\line`, because it has been modified in the LaTeX format.

```
\newcommand{\myline}{\hbox to\hsize}
```

`\m@kelayout` The macro `\m@kelayout` executes tests on the content of `\ffram` and `\sfram` and on the base of their outcome calls a particular subroutine for placing `boxes` and let appear on the terminal any needed warning. All this regarding the presence or the absence of the option `multicol`.

```
\newcommand{\m@kelayout}{%
\ifmultic@l
    \whiledo{\ht\ffram>0 \AND \ht\sfram>0}{\b@xbalance}
    \ifthenelse{\ht\ffram>0 \AND \ht\sfram=0}{\whiledo{\ht\ffram>0
    \AND \ht\sfram=0}{\v@idb@xtwo}
        \sec@nderror}
    \ifthenelse{\ht\ffram=0 \AND \ht\sfram>0}{\whiledo{\ht\ffram=0
    \AND \ht\sfram>0}{\v@idb@xbalance}
        \sec@nderror}
\else
    \whiledo{\ht\ffram>0 \AND \ht\sfram>0}{\splitb@x}
    \ifthenelse{\ht\ffram>0 \AND \ht\sfram=0}{\whiledo{\ht\ffram>0
    \AND \ht\sfram=0}{\v@idb@xtwo}
        \sec@nderror}
    \ifthenelse{\ht\ffram=0 \AND \ht\sfram>0}{\whiledo{\ht\ffram=0
    \AND \ht\sfram>0}{\v@idb@xone}
        \sec@nderror}
\fi
\UsefulLengthsTable}
```

`\UserDefWidths` The macro `\UserDefWidths` translates the arguments given by the user in the relative dimensions, executes the check on the resulting amount of the single fractions and, in case they exceed the dimension of `\textheight` calls the consequent warning; then it calls `\m@kelayout`.

```
\newcommand{\UserDefWidths}[2]{
    \setlength{\actualheight}{%
    \textheight-2\smallskipamount-\topskip-\baselineskip}
    \flength=.#1\actualheight \slength=.#2\actualheight
  \test=\flength \advance\test by \slength
  \ifdim\test>\actualheight
    \firsterror
  \fi
  \m@kelayout
}
```

`\AutoCompute` The macro `\AutoCompute`, just as the name says, tries to automatically compute the optimal division of the page in the two regions where the boxes will be placed, so that the text will be well balanced. Actually the macro does not take in consideration the whole `\textheight` during computation, but a newly defined dimension, `\actualheight`, standing for the *actual* height of the bodytext (in its definition is also involved the length `\initskip`, which can be used to insert an additional space between the header and the first box). During computation, scaling is needed to avoid exceeding `\maxdimen`, especially when the multicolumn option is active.

```
\newcommand{\AutoCompute}{%
    \setlength{\actualheight}{%
    \textheight-2\smallskipamount-\topskip-\baselineskip-\initskip}
        \ifmultic@l
            \setlength{\total}{\cnum\ht\ffram+\ht\sfram}
            \divide\total by 10 \divide\actualheight by 10%
            \setlength{\nop}{1pt*\ratio{\total}{\cnum\actualheight}}
            \setlength{\flength}{1pt*\ratio{\ht\ffram}{\nop}}
            \setlength{\slength}{1pt*\ratio{\ht\sfram}{\cnum\nop}}
        \else
            \setlength{\total}{\ht\ffram+\ht\sfram}
            \divide\total by 10 \divide\actualheight by 10%
            \setlength{\nop}{1pt*\ratio{\total}{\actualheight}}
            \setlength{\flength}{1pt*\ratio{\ht\ffram}{\nop}}
            \setlength{\slength}{1pt*\ratio{\ht\sfram}{\nop}}
        \fi
    \setlength{\ffrac}{1pt*\ratio{\flength}{10\actualheight}}
    \setlength{\sfrac}{1pt*\ratio{\slength}{10\actualheight}}
    \m@kelayout
}
```

`\Place` The macro `\Place` is merely a switcher between `\AutoCompute` and `\UserDefWidths`.

```
\newcommand{\Place}{%
    \@ifstar{\AutoCompute}{\UserDefWidths}}
```

### 3.6 Warnings

```
\newcommand{\UsefulLengthsTable}{%
\AtEndDocument{\immediate\write\m@ne{******************************************^^J%
                                     ffrac=\the\ffrac, sfrac=\the\sfrac^^J%
                                     flength=\the\flength, slength=\the\slength^^J%
                                     ******************************************}}
}
\newcommand{\firsterror}{\PackageError{parrun}{%
        Warning: Text fractions exceeding \string\textheight}
        {The total dimension of
        the single fractions of text exceeds \string\textheight.^^J%
        You probably should reconsider the parameters in \string\Place.^^J%
        However, if you are sure of what you have done, you can go on.^^J%
        Luck!}}
\newcommand{\sec@nderror}{\PackageError{parrun}{%
            Warning: the box still contains some text}
            {Your text is not well balanced. Probably you'll get a bad ^^J%
            output. You should reconsider your document's layout.^^J%
            You will find at the end of the log file some useful^^J%
            length for dealing with.}
}
⟨/parrun⟩
```

# References

[1] Donald Knuth. *The TEXbook* Addison–Wesley, Reading, MA, 1996.

# Change History

v1.0
  General: First public release. . 1
v1.1
  General: Added autocomputation, some macro names changed. . . . . . . 1
  `\AutoCompute`: new macro `\AutoCompute` added. . . 6
  `\Place`: new macro `\Place` added. . . . . . . . . . . . . . 6
  `\UserDefWidths`: name changed from `\Frame` to `\UserDefWidths`; some adjustments made. . . . . 5
v1.1a
  General: Bug fixed. . . . . . . . 1

# Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.