

# The storebox Package

Martin Scharrer  
[martin@scharrer.me](mailto:martin@scharrer.me)

CTAN: <http://www.ctan.org/pkg/storebox>

VC: [https://bitbucket.org/martin\\_scharrer/storebox](https://bitbucket.org/martin_scharrer/storebox)

Version v1.1 – 2011/10/21

## Abstract

This package allows the placement of identical content multiple times in a document while only storing it once in the output file. At the moment only  $\LaTeX$  compilers with native PDF output are supported (`pdflatex` or `lualatex`). For other  $\LaTeX$  compilers a fallback to `\savebox` is implemented.

## 1 Introduction

$\LaTeX$  provides box registers to save content and use it later in the document once or multiple times (or none at all). However, the box content is then written every time to the output file. The PDF file format provides a way to store material as object and reference it later. A similar technique is theoretically possible for PostScript output (but not yet implemented). This has the benefit that the content is really only stored once in the output file. However, if file compression is used for the final output file the size benefit might be very small if the content is only reused a low number of times.

This package provides “store boxes” which have the same user interface like normal  $\LaTeX$  “save boxes”, but only store the content once in the output file even if it is used several times. At the moment only PDF output is supported (i.e. `pdflatex` and `lualatex`). If the stored content is not used in the document after all it is not written to the PDF (at least the `pdfltex` manual says so). For any other  $\TeX$  and output format the package simply falls back to use the normal `\savebox` equivalents.

## 2 Options

The package should be loaded without any options and then automatically checks for PDF output (using the `ifpdf` package). It also provides two options `enable` and `disable` to explicitly switch the storebox feature on or off. The `disable` option is useful to see the resulting file size difference. The usage of the `enable` option should not be required and will lead to errors if used with unsupported  $\TeX$  formats.

### 3 Macros

```
\newstorebox{<\boxname>}
```

Because storeboxes are stored as PDF objects and not as T<sub>E</sub>X box registers it is not required to allocate a register for them. However, in fall-back mode the used control sequence <\boxname> must be defined as savebox. This macro defines <\boxname> simply as \relax (just as precaution) and is equal to \newsavebox in fall-back mode.

```
\storebox{<\boxname>}{<content>}
```

This stores the <content> as <\boxname> (which is actually just a numeric reference) for later use. The collectbox package is used to collect the <content> as box and not as macro argument and therefore it can contain verbatim and other special material. The braces can also be written in their explicit form \bgroup and \egroup and then split in two different macros or across the begin and end of an environment. It is possible to use \storebox multiple times with the same <\boxname>. This will create a new PDF object without affected the old one. In fall-back mode this macro is functional equal to \savebox (but then still uses collectbox).

```
\begin{storebox}{<\boxname>}  
  <content>  
\end{storebox}
```

This is the environment version of \storebox. Special care is taken to allow for an identical name. In fall-back mode this macro is equal to the lrbox environment.

```
\usestorebox{<\boxname>}
```

This macro typesets the stored content at the current position (as horizontal box in horizontal mode). This is realized by adding a PDF reference to the stored content. In fall-back mode this macro is equal to \usebox.

```
\ifstorebox
```

This if-switch is set to *true* if \storebox was successfully defined as intended but to *false* if the \savebox fall-back was used.

## 4 Example

```
\documentclass{article}

\usepackage{storebox}

\newstorebox{\mybox} % Not really required for PDF /
output
\begin{document}

\storebox{\mybox}{\verb+Supports verbatim #$\empty+}

\usestorebox{\mybox}
\usestorebox{\mybox}

\storebox\mybox\bgroup
Can also be split
\verb+\empty+
\egroup

\usestorebox{\mybox}
\usestorebox{\mybox}

\begin{storebox}{\mybox}
Or used as environment
(then will ignore leading and trailing spaces)
\end{storebox}

\usestorebox{\mybox}
\usestorebox{\mybox}

\end{document}
```

## 5 Implementation

```

1 %<!COPYRIGHT>
2 \ProvidesPackage{storebox}[%
3 %<!DATE>
4 %<!VERSION>
5 %<*DRIVER>
6     2099/01/01 develop
7 %</DRIVER>
8     Store and reuse boxes in a file size efficient /
9         way]
10
11 \DeclareOption{disable}{\let\ifstorebox\iffalse}
12 \DeclareOption{enable}{\let\ifstorebox\iftrue}
13 \ProcessOptions*
14 \edef\@tempa{\@optionlist{\@currname.\@currentext}}
15 \ifx\@tempa\empty
16     \RequirePackage{ifpdf}
17     \expandafter\let\csname ifstorebox\expandafter\
18         \endcsname\csname ifpdf\endcsname
19 \fi
20
21 \RequirePackage{collectbox}[2011/08/04]

```

\storebox

```

18 \newcommand*\storebox{%
19     \begingroup
20     \def\@tempa{storebox}%
21     \ifx\@currentenv\@tempa
22         \endgroup
23         \expandafter\@storebox@env
24     \else
25         \endgroup
26         \expandafter\@storebox
27     \fi
28 }
29
30 \ifstorebox

```

\@storebox

Macro version:

```

30 \def\@storebox#1{%
31     \begingroup

```

```

32     \@collectboxto\collectedbox{\pdfxform\
        collectedbox\endgroup\mathchardef#1=\
        pdflastxform}%
33 }

```

#### `\@storebox@env`

Environment version. Code adapted from `lrbox` environment. The group added by `\begin` and `\end` must be specially handled to allow for a local assignment.

```

34 \def\@storebox@env#1{%
35     \edef\@tempa{%
36         \setbox\collectedbox\hbox\bgroup%
37         \def\noexpand\the@storebox{\noexpand#1}%
38     }%
39     \@tempa
40     \begingroup
41     \aftergroup\@storebox@env@end
42     \@endpefalse
43     \color@setgroup
44     \begingroup
45     \def\@currenvir{storebox\empty}%
46     \ignorespaces
47 }

```

#### `\endstorebox`

```

48 \def\endstorebox{%
49     \unskip
50     \endgroup
51     \color@endgroup
52 }

```

#### `\@storebox@env@end`

This ends the box assignment and stores the box as PDF xform. Then the given control sequence is set to the xform number.

```

53 \def\@storebox@env@end{%
54     \edef\@tempa{%
55         \egroup
56         \pdfxform\collectedbox
57         \endgroup
58         \mathchardef\expandafter\noexpand\
            the@storebox=\pdflastxform
59     }%
60     \@tempa
61 }

```

`\newstorebox`

```
62 \newcommand*\newstorebox[1]{%
63   \@ifdefinable{#1}{\let#1\relax}%
64 }
```

`\usestorebox`

```
65 \newcommand*\usestorebox[1]{%
66   \mbox{\pdfrefxform#1}%
67 }
68 \else
```

`\@storebox`

Macro version:

```
69 \def\@storebox#1{\@collectboxto{#1}{}}
```

`\@storebox@env`

```
70 \def\@storebox@env{%
71   \edef\@currenvir{\@currenvir\noexpand\noexpand\
       noexpand\empty}%
72   \lrbox
73 }
```

`\endstorebox`

```
74 \def\endstorebox{%
75   \endlrbox
76   \edef\@currenvir{\@currenvir}%
77 }
```

`\newsavebox`

```
78 \@ifdefinable\newstorebox{%
79   \let\newstorebox\newsavebox
80 }
```

`\usestorebox`

```
81 \@ifdefinable\usestorebox{%  
82 \let\usestorebox\usebox  
83 }  
  
84 \fi
```