

# The classlist package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2006/02/20 v1.2

## Abstract

This package records the loaded classes and stores them in a list.

## Contents

<b>1 Documentation</b>	<b>1</b>
1.1 Background	1
1.2 Usage	1
<b>2 Implementation</b>	<b>2</b>
<b>3 Installation</b>	<b>4</b>
3.1 Download	4
3.2 Bundle installation	4
3.3 Package installation	4
3.4 Refresh file name databases	5
3.5 Some details for the interested	5
<b>4 History</b>	<b>5</b>
[2005/06/19 v1.0]	5
[2005/06/19 v1.1]	5
[2006/02/20 v1.2]	6
<b>5 Index</b>	<b>6</b>

## 1 Documentation

### 1.1 Background

This packages is an answer of a newsgroup question:

```
Newsgroup: comp.text.tex
Subject:    Finding the Document Class
From:      Herber Schulz
Date:      18 Jun 2005 13:16:49 -0500
Message-ID: <herbs-D55DB9.13170418062005@news.isp.giganews.com>
```

### 1.2 Usage

Load this package before `\documentclass`:

```
\RequirePackage{classlist}
\documentclass[some,options]{whatever}
```

It then records the classes with options.

If used after `\documentclass`, `\@filelist` is parsed for classes. The additional data specified options and requested version is no longer available here.

`\MainClass` contains the first loaded class.

`\ClassList` stores the class entries, eg.

```
\ClassList → \ClassListEntry{myarticle}{a4paper}{}  
              \ClassListEntry{article}{}{}
```

`\ClassListEntry` has three arguments:

```
#1:  class name  
#2:  options given in \documentclass/\LoadClass  
#3:  requested version, not the version of class
```

`\PrintClassList` prints the list on screen it can be configured by

`\PrintClassListTitle` for the title and

`\PrintClassListEntry` for formatting the entries. See the implementation how to use these.

## 2 Implementation

```
1 (*package)
```

Package identification.

```
2 \NeedsTeXFormat{LaTeX2e}  
3 \ProvidesPackage{classlist}%  
4   [2006/02/20 v1.2 Record loaded classes (H0)]  
5 \let\ClassList\@empty  
6 \let\MainClassName\relax
```

Test, whether we are called before `\documentclass`.

```
7 \ifx\@classoptionslist\relax  
8   \let\CL@org@fileswith@pti@ns\@fileswith@pti@ns  
9   \def\@fileswith@pti@ns#1[#2]#3[#4]{%  
#1:  \@clsextension  
#2:  options of \documentclass/\LoadClass  
#3:  class name  
#4:  requested version  
10    \ifx#1\@clsextension  
11      \@ifl@aded#1{#3}{%  
12        \PackageInfo{classlist}{%  
13          Skipping class ‘#3’, because\MessageBreak  
14          this class is already loaded%  
15        }%  
16      }{%  
17        \@ifundefined{MainClassName}{%  
18          \def\MainClassName{#3}%  
19        }{}%  
20        \@temptokena\expandafter{%  
21          \ClassList  
22          \ClassListEntry{#3}{#2}{#4}%  
23        }%  
24        \edef\ClassList{\the\@temptokena}%  
25      }%  
26    \fi  
27    \CL@org@fileswith@pti@ns{#1}[{#2}]{#3}[{#4}]%  
28  }%  
29  \let\@@fileswith@pti@ns\@fileswith@pti@ns  
30 \else
```

Called after `\documentclass`.

```
31   \PackageInfo{classlist}{Use \string\@filelist\space method}%  
32
```

```

33 \let\ClassListEntry\relax
34 \expandafter\def\expandafter\CL@test
35 \expandafter#\expandafter1\@clsextension#2\@nil{%
36 \ifx\#2\%
Name does not contain \@clsextension
37 \else
38 \expandafter\CL@test@i\CL@entry\@nil
39 \fi
40 }%
41 \expandafter\def\expandafter\CL@test@i
42 \expandafter#\expandafter1\@clsextension#2\@nil{%
43 \ifx\#2\%
44 \@ifundefined{opt@\CL@entry}{%
45 }{%
46 \@ifundefined{MainClassName}{%
47 \let\MainClassName\CL@entry
48 }{%
49 }%
50 \edef\ClassList{%
51 \ClassList
52 \ClassListEntry{\CL@entry}{}}%
53 }%
54 }%
55 \else
Names with more than one \@clsextension are not supported.
56 \fi
57 }%
58 \@for\CL@entry:=\@filelist\do{%
59 \expandafter\expandafter\expandafter\CL@test\expandafter
60 \CL@entry\@clsextension\@nil
61 }%
62 \fi

\PrintClassListEntry

63 \providecommand*\PrintClassListEntry}[3]{%
64 \toks@{* #1}%
65 \typeout{\the\toks@}%
66 }

\PrintClassListTitle

67 \providecommand*\PrintClassListTitle){%
68 \typeout{Class list:}%
69 }

\PrintClassList

70 \providecommand*\PrintClassList){%
71 \begingroup
72 \let\ClassListEntry\PrintClassListEntry
73 \PrintClassListTitle
74 \ClassList
75 \endgroup
76 }

\CL@InfoEntry

77 \def\CL@InfoEntry#1#2#3{%
78 \advance\count@ by \@ne
79 \def\x{#2}%
80 \@onelevel@sanitize\x
81 \edef\CL@Info{%
82 \CL@Info
83 \noexpand\MessageBreak

```

```

84      (\the\count@) %
85      #1 [\x]%
86      \ifx\#3\%
87      \else
88      \space[#3]% hash-ok
89      \fi
90  }%
91 }

92 \AtBeginDocument{%
93   \begingroup
94     \count@=\z@
95     \def\CL@Info{Class List:}%
96     \let\ClassListEntry\CL@InfoEntry
97     \ClassList
98     \let\on@line\@empty
99     \PackageInfo{classlist}{\CL@Info}%
100  \endgroup
101 }

102 </package>

```

## 3 Installation

### 3.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[CTAN:macros/latex/contrib/oberdiek/classlist.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/classlist.pdf](#) Documentation.

**Bundle.** All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:macros/latex/contrib/oberdiek/oberdiek-tds.zip](#)

### 3.2 Bundle installation

**Unpacking.** Unpack the `oberdiek-tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek-tds.zip -d ~/texmf
```

**Script installation.** Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

### 3.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain- $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ :

```
tex classlist.dtx
```

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

**TDS.** Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
classlist.sty → tex/latex/oberdiek/classlist.sty
classlist.pdf → doc/latex/oberdiek/classlist.pdf
classlist.dtx → source/latex/oberdiek/classlist.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

### 3.4 Refresh file name databases

If your  $\text{T}_{\text{E}}\text{X}$  distribution (`te $\text{T}_{\text{E}}\text{X}$` , `mik $\text{T}_{\text{E}}\text{X}$` , ...) relies on file name databases, you must refresh these. For example, `te $\text{T}_{\text{E}}\text{X}$`  users run `texhash` or `mktextlsr`.

### 3.5 Some details for the interested

**Attached source.** The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk classlist.pdf unpack_files output .
```

**Unpacking with  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .** The `.dtx` chooses its action depending on the format:

**plain- $\text{T}_{\text{E}}\text{X}$ :** Run `docstrip` and extract the files.

**$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ :** Generate the documentation.

If you insist on using  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  for `docstrip` (really, `docstrip` does not need  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{classlist.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

**Generating the documentation.** You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$` :

```
pdflatex classlist.dtx
makeindex -s gind.ist classlist.idx
pdflatex classlist.dtx
makeindex -s gind.ist classlist.idx
pdflatex classlist.dtx
```

## 4 History

[2005/06/19 v1.0]

- First published version (`comp.text.tex`, CTAN).

[2005/06/19 v1.1]

- After `\documentclass` the package looks at `\@filelist` instead of aborting with error.

[2006/02/20 v1.2]

- DTX framework.
- Fix for \@@fileswith@pti@ns.

## 5 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.

Symbols		I	
\@@fileswith@pti@ns	29	\ifx	7, 10, 36, 43, 86
\@classoptionslist	7	M	
\@clsextension	10, 35, 42, 60	\MainClassName	6, 18, 47
\@empty	5, 98	\MessageBreak	13, 83
\@filelist	31, 58	N	
\@fileswith@pti@ns	8, 9, 29	\NeedsTeXFormat	2
\@for	58	O	
\@ifl@aded	11	\on@line	98
\@ifundefined	17, 44, 46	P	
\@ne	78	\PackageInfo	12, 31, 99
\@nil	35, 38, 42, 60	\PrintClassList	70
\@onelevel@sanitize	80	\PrintClassListEntry	63, 72
\@temptokena	20, 24	\PrintClassListTitle	67, 73
\\	36, 43, 86	\providecommand	63, 67, 70
A		\ProvidesPackage	3
\advance	78	S	
\AtBeginDocument	92	\space	31, 88
C		T	
\CL@entry	38, 44, 47, 52, 58, 60	\the	24, 65, 84
\CL@Info	81, 82, 95, 99	\toks@	64, 65
\CL@InfoEntry	77, 96	\typeout	65, 68
\CL@org@fileswith@pti@ns	8, 27	X	
\CL@test	34, 59	\x	79, 80, 85
\CL@test@i	38, 41	Z	
\ClassList	5, 21, 24, 50, 51, 74, 97	\z@	94
\ClassListEntry	22, 33, 52, 72, 96		
\count@	78, 84, 94		
D			
\do	58		