The classlist package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.com>

2016/05/16 v1.5

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

Contents

1 Documentation **1**
1.1 Background 1
1.2 Usage 2

2 Implementation **2**

3 Installation **4**
3.1 Download 4
3.2 Bundle installation 5
3.3 Package installation 5
3.4 Refresh file name databases 5
3.5 Some details for the interested 5

4 Catalogue **6**

5 History **7**
[2005/06/19 v1.0] 7
[2005/06/19 v1.1] 7
[2006/02/20 v1.2] 7
[2008/08/11 v1.3] 7
[2011/10/17 v1.4] 7
[2016/05/16 v1.5] 7

6 Index **7**

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>

*Please report any issues at https://github.com/ho-tex/oberdiek/issues

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.

`\MainClassName` 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 implemenation how to use these.

2 Implementation

```
1 ⟨*package⟩
```

Package identification.

```
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{classlist}%
4   [2016/05/16 v1.5 Record classes used in a document (HO)]
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}%
```

```
        }{}%
        \@temptokena\expandafter{%
          \ClassList
          \ClassListEntry{#3}{#2}{#4}%
        }%
        \edef\ClassList{\the\@temptokena}%
      }%
    \fi
    \CL@org@fileswith@pti@ns{#1}[{#2}]{#3}[{#4}]%
  }%
  \let\@@fileswith@pti@ns\@fileswith@pti@ns
\else
```

Called after `\documentclass`.

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

  \let\ClassListEntry\relax
  \expandafter\def\expandafter\CL@test
      \expandafter#\expandafter1\@clsextension#2\@nil{%
    \ifx\\#2\\%
```

Name does not contain `\@clsextension`

```
    \else
      \expandafter\CL@test@i\CL@entry\@nil
    \fi
  }%
  \expandafter\def\expandafter\CL@test@i
      \expandafter#\expandafter1\@clsextension#2\@nil{%
    \ifx\\#2\\%
      \@ifundefined{opt@\CL@entry}{%
      }{%
        \@ifundefined{MainClassName}{%
          \let\MainClassName\CL@entry
        }{%
        }%
        \edef\ClassList{%
          \ClassList
          \ClassListEntry{\CL@entry}{}{}%
        }%
      }%
    \else
```

Names with more than one `\@clsextension` are not supported.

```
    \fi
  }%
  \@for\CL@entry:=\@filelist\do{%
    \expandafter\expandafter\expandafter\CL@test\expandafter
        \CL@entry\@clsextension\@nil
  }%
\fi
```

`\PrintClassListEntry`

```
\providecommand*{\PrintClassListEntry}[3]{%
  \toks@{* #1}%
  \typeout{\the\toks@}%
}
```

`\PrintClassListTitle`

```
\providecommand*{\PrintClassListTitle}{%
  \typeout{Class list:}%
}
```

\PrintClassList

```
\providecommand*{\PrintClassList}{%
  \begingroup
    \let\ClassListEntry\PrintClassListEntry
    \PrintClassListTitle
    \ClassList
  \endgroup
}
```

\CL@InfoEntry

```
\def\CL@InfoEntry#1#2#3{%
  \advance\count@ by \@ne
  \def\x{#2}%
  \@onelevel@sanitize\x
  \edef\CL@Info{%
    \CL@Info
    \noexpand\MessageBreak
    (\the\count@) %
    #1 [\x]%
    \ifx\\#3\\%
    \else
      \space[#3]% hash-ok
    \fi
  }%
}

\AtBeginDocument{%
  \begingroup
    \count@=\z@
    \def\CL@Info{Class List:}%
    \let\ClassListEntry\CL@InfoEntry
    \ClassList
    \let\on@line\@empty
    \PackageInfo{classlist}{\CL@Info}%
  \endgroup
}
```

⟨/package⟩

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

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:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:tds/tds.pdf). Directories with `texmf` in their name are usually organized this way.

[1] http://ctan.org/pkg/classlist

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 TEX:

```
tex classlist.dtx
```

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 TEX distribution (teTEX, mikTEX, ...) relies on file name databases, you must refresh these. For example, teTEX users run `texhash` or `mktexlsr`.

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 LATEX. The `.dtx` chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), 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 pdfLaTeX:

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

4 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `classlist.xml`.

```
⟨*catalogue⟩
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='classlist'>
  <name>classlist</name>
  <caption>Record classes used in a document.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2005,2006,2008,2011'/>
  <license type='lppl1.3'/>
  <version number='1.5'/>
  <description>
    Load this package before \documentclass:
    <p/>
      <tt>\RequirePackage{classlist}</tt><br/>
      <tt>\documentclass[some,options]{whatever}</tt>
    <p/>
    After doing this, <tt>\MainClass</tt> contains the name of the
    first loaded class, <tt>\ClassList</tt> contains a set of triples
    &lt;class name&gt;, &lt;options directly requested&gt;, and
    &lt;version requested&gt;.  (The package may also be loaded after
    <tt>\documentclass</tt>, in which case some information is not
    available.)
    <p/>
    The package is part of the <xref refid='oberdiek'>oberdiek</xref>
    bundle.
  </description>
  <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/classlist.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/classlist.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/catalogue⟩
```

5 History

[2005/06/19 v1.0]

- First published version: CTAN and newsgroup comp.text.tex: "Re: Finding the Document Class"[2]

[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.

[2008/08/11 v1.3]

- Code is not changed.
- URLs updated.

[2011/10/17 v1.4]

- Documentation fix: \MainClass → \MainClassName.

[2016/05/16 v1.5]

- Documentation updates.

6 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; plain numbers refer to the code lines where the entry is used.

Symbols

\@@fileswith@pti@ns 29
\@classoptionslist 7
\@clsextension 10, 35, 42, 60
\@empty 5, 98
\@filelist 31, 58
\@fileswith@pti@ns 8, 9, 29
\@for 58
\@ifl@aded 11
\@ifundefined 17, 44, 46
\@ne 78
\@nil 35, 38, 42, 60
\@onelevel@sanitize 80
\@temptokena 20, 24
\\ 36, 43, 86

A

\advance 78
\AtBeginDocument 92

C

\CL@entry 38, 44, 47, 52, 58, 60
\CL@Info 81, 82, 95, 99
\CL@InfoEntry 77, 96
\CL@org@fileswith@pti@ns 8, 27
\CL@test 34, 59
\CL@test@i 38, 41
\ClassList 5, 21, 24, 50, 51, 74, 97, 120
\ClassListEntry 22, 33, 52, 72, 96
\count@ 78, 84, 94

D

\do 58

[2]Url: http://groups.google.com/group/comp.text.tex/msg/8ee9523c2dc13666

\documentclass 114, 117, 123

I

\ifx 7, 10, 36, 43, 86

M

\MainClass 119
\MainClassName 6, 18, 47
\MessageBreak 13, 83

N

\NeedsTeXFormat 2

O

\on@line 98

P

\PackageInfo 12, 31, 99
\PrintClassList 70
\PrintClassListEntry 63, 72
\PrintClassListTitle 67, 73
\providecommand 63, 67, 70
\ProvidesPackage 3

R

\RequirePackage 116

S

\space 31, 88

T

\the 24, 65, 84
\toks@ 64, 65
\typeout 65, 68

X

\x 79, 80, 85

Z

\z@ 94