The classlist package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2008/08/11 v1.3

Abstract

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

Contents

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 implemenation how to use these.

2 Implementation

```
1 ⟨*package⟩
```

Package identification.

```
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{classlist}%
4   [2008/08/11 v1.3 Record loaded classes (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}%
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
```

```
\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.

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/
```

[1] ftp://ftp.ctan.org/tex-archive/

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 History

[2005/06/19 v1.0]

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

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

[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.

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.