The stdclsdv package*

Peter Wilson, Herries Press†
Maintainer: Will Robertson
will dot robertson at latex-project dot org

2009/09/04

Abstract

The stdclsdv package is intended to be used by the authors of LaTeX packages that need to know about the sectional divisions provided by the document class.

Contents

1 Introduction

Several packages need to know the kind of sectional divisions provided by the document class.

The stdclsdv package provides a solution for for the case when the document class is one of the LaTeX standard classes (i.e., book, report, article, letter, slides, and classes derived from the article class, namely ltxdoc and proc). It also handles the koma classes.[1]

This manual is typeset according to the conventions of the LaTeX docstrip utility which enables the automatic extraction of the LaTeX macro source files [GMS94].

Section 2 describes the usage of the package. Commented source code for the package is in Section 3.

*This file has version number v1.1a, last revised 2009/09/04.

†This work was originally performed as a Guest Researcher at the National Institute of Standards and Technology.

[1] Courtesy of Joerg Jaspert Joerg.Jaspert@informatik.fh-fulda.de.

2 The stdclsdv package

The stdclsdv package provides several `\if...` macros which can be used to determine the kinds of sectional divisions supported by the current (standard) `\documentclass`.

The package is designed to work with the standard LaTeX document classes book, report, article, proc and ltxdoc class (which is based to a large extent on the article class).

`\ifSCDknownclass` `\ifSCDknownclass` is TRUE iff the document class is one of: book, report, article, letter, slides, proc, ltxdoc, scrbook, scrreprt, scrartcl or scrlettr. Otherwise it is FALSE.

`\ifSCDchapter` `\ifSCDchapter` is TRUE iff the document class defines a `\chapter` sectional division, otherwise it is FALSE.

`\ifSCDpart` `\ifSCDpart` is TRUE iff the document class defines a `\part` sectional division, otherwise it is FALSE.

`\ifSCDsection` `\ifSCDsection` is TRUE iff the document class defines a `\section` sectional division, otherwise it is FALSE.

`\ifSCDnodivs` `\ifSCDnodivs` is TRUE iff the document class has neither `\part` nor `\chapter` nor `\section` divisions, otherwise it is FALSE.

`\SCDquit` If the document class or divisioning is not handled by a package, it can be useful to skip all futher package code. This could be done using:

```
...
\ifSCDknownclass
  % normal processing
\else % just before end of package file
  % error/warning message about unknown class
\fi
\endinput
```

Alternatively, using the `\SCDQuit` command provides a clearer means of accomplishing this. The `\SCDquit` macro is defined to do nothing. It is intended to be used for prematurely ending a package file in the following manner:

```
\ifSCDknownclass\else
  \renewcommand{\SCDquit}{\endinput}
  % error/warning message about unknown class
\fi
\SCDquit % ends the file iff the class is unknown
% normal processing
...
\endinput
```

`\SCDCheckCommand` `\SCDCheckCommand` takes the same arguments as `\newcommand`, that is:

`\ifSCDSameDefinition` `\SCDCheckCommand{`⟨*command*⟩`}[`⟨*nargs*⟩`][`⟨*defarg*⟩`]{`⟨*definition*⟩`}`

`\SCDCheckCommand` is identical to the kernel `\CheckCommand` (see `ltdefns.dtx`) except that it sets `\ifSCDSameDefinition` to TRUE iff the definition given in

`\SCDCheckCommand` is the same as the current definition, otherwise `\ifSCDSameDefinition` is set to FALSE. (`\CheckCommand` issues a warning if the definitions are different).

3 The package code

Announce the name and version of the package, which requires LaTeX 2_ε.

```
⟨*usc⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{stdclsdv}[2009/09/04 v1.1a Sectional divisions]
```

In order to try and avoid name clashes with other packages, each macro name includes the character string `SCD` (Standard Class Division).

`\ifSCDknownclass` `\ifSCDpart` `\ifSCDchapter` `\ifSCDsection` `\ifSCDnodivs`

These are used when we need to decide what sectional divisions are supported by the document's class. We will assume assume an unknown class that has `\part` and `\section` but not `\chapter` divisions.

```
\newif\ifSCDknownclass\SCDknownclassfalse
\newif\ifSCDpart\SCDparttrue
\newif\ifSCDchapter\SCDchapterfalse
\newif\ifSCDsection\SCDsectiontrue
\newif\ifSCDnodivs\SCDnodivsfalse
```

Check the actual class.[2]

```
\@ifclassloaded{book}{\SCDknownclasstrue\SCDchaptertrue}{}
\@ifclassloaded{report}{\SCDknownclasstrue\SCDchaptertrue}{}
\@ifclassloaded{article}{\SCDknownclasstrue}{}
\@ifclassloaded{proc}{\SCDknownclasstrue}{}
\@ifclassloaded{ltxdoc}{\SCDknownclasstrue}{}
\@ifclassloaded{slides}{\SCDknownclasstrue
                        \SCDnodivstrue\SCDpartfalse\SCDsectionfalse}{}
\@ifclassloaded{letter}{\SCDknownclasstrue
                        \SCDnodivstrue\SCDpartfalse\SCDsectionfalse}{}
```

Joerg Jaspert supplied, via Email on 2000/07/16, the following code for the koma classes.

```
\@ifclassloaded{scrbook}{\SCDknownclasstrue\SCDchaptertrue}{}
\@ifclassloaded{scrreprt}{\SCDknownclasstrue\SCDchaptertrue}{}
\@ifclassloaded{scrartcl}{\SCDknownclasstrue}{}
\@ifclassloaded{scrlettr}{\SCDknownclasstrue
                          \SCDnodivstrue\SCDpartfalse\SCDsectionfalse}{}
```

If the class is not one of the standard classes, check if `\part` or `\chapter` or `\section` headings are provided.

```
\ifSCDknownclass\else
  \SCDnodivstrue
  \ifx\part\undefined
    \SCDpartfalse
  \else
```

[2] Thanks to David Carlisle for information on how to check on the class.

```
    \SCDnodivsfalse
  \fi
  \ifx\chapter\undefined\else
    \SCDchaptertrue \SCDnodivsfalse
  \fi
  \ifx\section\undefined
    \SCDsectionfalse
  \else
    \SCDnodivsfalse
  \fi
\fi
```

`\SCDquit` A macro that does nothing (see §2 for its intended usage).

```
\newcommand{\SCDquit}{}
```

`\ifSCDSameDefinition` Used to store the result of `\SCDCheckCommand`. TRUE iff the command has the given definition.

```
\newif\ifSCDSameDefinition
```

`\SCDCheckCommand` This is identical to the kernal `\CheckCommand` except that it sets the `\ifSCDSameCommand` flag rather than issuing a warning. See `ltdefns.dtx` for the coding for `\CheckCommand`.

```
\def\SCDCheckCommand{\@star@or@long%
  \SCDSameDefinitiontrue%             changed from CheckCommand
  \@SCD@check@command}
  \@onlypreamble\SCDCheckCommand
```

`\@SCD@check@command`

```
\def\@SCD@check@command#1#2#{\@SCD@check@c#1{#2}}
  \@onlypreamble\@SCD@check@command
```

`\@SCD@check@c`

```
\long\def\@SCD@check@c#1#2#3{%
  \expandafter\let\csname\string\reserved@a\endcsname\relax
  \renew@command\reserved@a#2{#3}%
  \@ifundefined{\string\reserved@a}%
   {\@SCD@check@eq#1\reserved@a}%
   {\expandafter\@SCD@check@eq
    \csname\string#1\expandafter\endcsname
    \csname\string\reserved@a\endcsname}}
  \@onlypreamble\@SCD@check@c
```

`\@SCD@check@eq`

```
\def\@SCD@check@eq#1#2{%
  \ifx#1#2\else
    \SCDSameDefinitionfalse  % changed from CheckCommand
  \fi}
  \@onlypreamble\@SCD@check@eq
```

The end of this package.

61 ⟨/usc⟩

References

[GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.

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.