The **somedefs** toolkit package

Alan Jeffrey

long time ago in a different century...

This file is maintained by the \LaTeX\ Project team. Bug reports can be opened (category tools) at https://latex-project.org/bugs.html.

**Overview**

This is an example ‘programmers toolkit’ package, for use by package writers. It allows package writers to provide options which switch definitions on and off. For example, a package **fred** might define a large number of commands, including \texttt{\textbackslash foo} and \texttt{\textbackslash baz}, so:

\begin{verbatim}
\usepackage{fred}
\end{verbatim}

would use a lot of memory, even if \texttt{\textbackslash foo} and \texttt{\textbackslash baz} were the only commands needed. However, if the author of **fred** used the **somedefs** package, then the user would be able to say:

\begin{verbatim}
\usepackage[only,foo,baz]{fred}
\end{verbatim}

and only the commands \texttt{\textbackslash foo} and \texttt{\textbackslash baz} would be defined.

To use the **somedefs** package in your own packages or classes, you say:

\begin{verbatim}
\RequirePackage{somedefs}
\end{verbatim}

You can then use four new commands:

- \texttt{\UseAllDefinitions} which says that all the commands in the file should be defined.
- \texttt{\UseSomeDefinitions} which says that only the commands specified by \texttt{\UseDefinition} should be defined.
- \texttt{\UseDefinition{⟨name⟩}} which says that the command \texttt{⟨name⟩} should be defined.
- \texttt{\ProvidesDefinition{⟨definition⟩}} which provides one definition, of the form \texttt{\definingcommand{\command}…}
For example, the package Fred could say:

\RequirePackage{somedefs}
\UseAllDefinitions
\DeclareOption{only}{\UseSomeDefinitions}
\DeclareOption*{\UseDefinition{\CurrentOption}}
\ProcessOptions
\ProvidesDefinition{\newcommand{\foo}{...}}
\ProvidesDefinition{\newcommand{\baz}{...}}

One of the commands \UseAllDefinitions or \UseSomeDefinitions should always be used. You may have some commands which need other commands, in which case you have to declare the options by hand. For example, if the command bar needs the command foo, you could say:

\DeclareOption{bar}{\UseDefinition{bar}\UseDefinition{foo}}

For a longer example of the use of the somedefs package, look at the rawfonts package.

Implementation

The driver for the documentation you’re now reading.

1 (*driver)
2 \documentclass{ltxdoc}
3 \begin{document}
4 \DocInput{somedefs.dtx}
5 \end{document}
6 (/driver)

This is a \LaTeX\textsuperscript{2e} package.

7 (*package)
8 \NeedsTeXFormat{LaTeX2e}
9 \ProvidesPackage{somedefs}[1994/06/01 v0.03 Toolkit for optional definitions]

The package works by having \UseDefinition{⟨name⟩} define \name to be \unprovided@definition. If \UseSomeDefinitions has been called, then \ProvidesDefinition looks to see if \name is \unprovided@definition. If \UseAllDefinitions has been called, then \ProvidesDefinition does nothing. If neither has been called, then \ProvidesDefinition produces an error message.

10 \def\UseSomeDefinitions{%
11 \let\ProvidesDefinition@providesdefinition\unprovided@definition
12 }
13 \def\UseAllDefinitions{%
14 \let\ProvidesDefinition@firstofone\unprovided@definition
15 }
16 \def\UseDefinition#1{%
17 \expandafter\let#cname#1\endcsname\unprovided@definition
18 }
19 \def\ProvidesDefinition#1{%
20 \PackageError{somedefs} %
21 \{No \noexpand\UseSomeDefinitions or \string\UseAllDefinitions\}%
The package which used the ‘somedefs’ package has an error.

\def\@providesdefinition#1{\@provides@definition#1\relax
\@provides@definition}
\def\@provides@definition#1#2#3\@provides@definition{\ifx#2\@unprovided@definition
#1#2#3\fi
\@unprovided@definition{\PackageError{somedefs}{Package 'somedefs' error: this command was never defined}{You have requested a command which does not exist.}}
\@onlypreamble\UseSomeDefinitions
\@onlypreamble\UseAllDefinitions
\@onlypreamble\UseDefinition
\@onlypreamble\ProvidesDefinition
\@onlypreamble\@providesdefinition
\@onlypreamble\@provides@definition

That’s it!

\end{package}