The *somedefs* toolkit package

Alan Jeffrey

long time ago in a different century...

This file is maintained by the \TeX\ 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{\verb|\foo|} and \texttt{\verb|\baz|}, so:

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

would use a lot of memory, even if \texttt{\verb|\foo|} and \texttt{\verb|\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{\verb|\foo|} and \texttt{\verb|\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{\verb|\name|} should be defined.
- \texttt{\ProvidesDefinition{⟨definition⟩}} which provides one definition, of the form \texttt{\definingcommand{\command}...}
For example, the package \texttt{fred} could say:

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

One of the commands \texttt{\UseAllDefinitions} or \texttt{\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 \texttt{\bar} needs the command \texttt{\foo}, you could say:

\begin{verbatim}
\DeclareOption{\bar}{\UseDefinition{\bar}\UseDefinition{\foo}}
\end{verbatim}

For a longer example of the use of the \texttt{somedefs} package, look at the \texttt{rawfonts} package.

\section*{Implementation}

The driver for the documentation you're now reading.

\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{somedefs}[1994/06/01 v0.03 Toolkit for optional definitions]
\UseSomeDefinitions\UseAllDefinitions\UseDefinition\ProvidesDefinition]\Unprovided\Definition\Provide\Definition\@provides\definition\@unprovided\definition
\end{verbatim}

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

{/package}