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](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 `\foo` and `\baz`, so:

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

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

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

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:

```latex
\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.

```latex
\documentclass{ltxdoc}
\begin{document}
\DocInput{somedefs.dtx}
\end{document}
```

This is a \LaTeX\ 2ε package.

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.
The package which used the 'somedefs' package has an error.\%
\def\@providesdefinition#1{\@provides@definition#1\relax
\def\@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}%
\PackageError{somedefs}{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}