

The **somedefs** toolkit package

Alan Jeffrey

long time ago in a different century...

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:

```
\usepackage{fred}
```

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:

```
\usepackage[only,foo,baz]{fred}
```

and only the commands `\foo` and `\baz` would be defined.

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

```
\RequirePackage{somedefs}
```

You can then use four new commands:

- `\UseAllDefinitions` which says that all the commands in the file should be defined.
- `\UseSomeDefinitions` which says that only the commands specified by `\UseDefinition` should be defined.
- `\UseDefinition{<name>}` which says that the command `\name` should be defined.
- `\ProvidesDefinition{<definition>}` which provides one definition, of the form `\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 L^AT_EX 2_ε package.

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

<pre>\UseSomeDefinitions \UseAllDefinitions \UseDefinition \ProvidesDefinition \@providesdefinition \@provides@definition \@unprovided@definition</pre>	<p>The package works by having <code>\UseDefinition{<name>}</code> define <code>\name</code> to be <code>\@unprovided@definition</code>. If <code>\UseSomeDefinitions</code> has been called, then <code>\UseDefinition</code> <code>\ProvidesDefinition</code> looks to see if <code>\name</code> is <code>\@unprovided@definition</code>. If <code>\UseAllDefinitions</code> has been called, then <code>\ProvidesDefinition</code> does nothing. If neither has been called, then <code>\ProvidesDefinition</code> produces an error message.</p> <pre>10 \def\UseSomeDefinitions{% 11 \let\ProvidesDefinition\@providesdefinition 12 } 13 \def\UseAllDefinitions{% 14 \let\ProvidesDefinition\@firstofone 15 } 16 \def\UseDefinition#1{%</pre>
---	--

```

17   \expandafter\let\csname#1\endcsname\@unprovided@definition
18 }
19 \def\ProvidesDefinition#1{%
20   \PackageError{somedefs}%
21     {No \noexpand\UseSomeDefinitions or \string\UseAllDefinitions}%
22     {The package which used the 'somedefs' package has an error.}%
23 }
24 \def\@providesdefinition#1{\@provides@definition#1\relax
25   \@provides@definition}
26 \def\@provides@definition#1#2#3\@provides@definition{%
27   \ifx#2\@unprovided@definition
28     #1#2#3%
29   \fi
30 }
31 \def\@unprovided@definition{%
32   \PackageError{somedefs}%
33     {Package 'somedefs' error: this command was never defined}%
34     {You have requested a command which does not exist.}%
35 }
36 \@onlypreamble\UseSomeDefinitions
37 \@onlypreamble\UseAllDefinitions
38 \@onlypreamble\UseDefinition
39 \@onlypreamble\ProvidesDefinition
40 \@onlypreamble\@providesdefinition
41 \@onlypreamble\@provides@definition

That's it!
42 </package>

```