The somedefs toolkit package

Alan Jeffrey

v0.03

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.

```
⟨*driver⟩
\documentclass{ltxdoc}
\begin{document}
\DocInput{somedefs.dtx}
\end{document}
⟨/driver⟩
```

This is a LaTeX 2_ε package.

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{somedefs}[1994/06/01 Toolkit for optional definitions]
```

`\UseSomeDefinitions` `\UseAllDefinitions` `\UseDefinition` `\ProvidesDefinition` `\@providesdefinition` `\@provides@definition` `\@unprovided@definition`

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.

```
\def\UseSomeDefinitions{%
   \let\ProvidesDefinition\@providesdefinition
}
\def\UseAllDefinitions{%
   \let\ProvidesDefinition\@firstofone
}
\def\UseDefinition#1{%
   \expandafter\let\csname#1\endcsname\@unprovided@definition
}
\def\ProvidesDefinition#1{%
   \PackageError{somedefs}%
     {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
}
\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⟩