The underscore package

Donald Arseneau*

13 September 2006, version 1.0

Features

The package alters the command _ (which normally prints an underscore character or facsimile) so that the hyphenation of constituent words is not affected, and hyphenation is permitted after the underscore. For example,

```
compound\_fracture
```

hyphenates as

```
com- pound\_- frac- ture
```

If you prefer the underscore to break without a hyphen (but still with the same rules as for explicit hyphen-breaks) then use the [nohyphen] package option.

A simple "_" acts just like "_" in text mode, but makes a subscript in maths mode, so

```
activation_energy $E_a$
```

is printed as

activation_energy E_a

Both forms use an underscore character if the font encoding contains one (e.g., with \usepackage[T1]{fontenc} or typewriter fonts in any encoding), but they use a rule if there is no proper character (just as unmodified LaTeX does).

Deficiencies

The skips and penalties ruin any kerning with the underscore character (when a character is used). However, there doesn't seem to be much, if any, such kerning in the EC fonts, and there is never any kerning with a rule.

You must avoid "_" in file names and in cite or ref tags, or you must use the babel package, with its active-character controls, or you must give the [strings] option, which attempts to redefine several commands (and may not work perfectly). Even without the [strings] option or babel, you can use occasional underscores like: "\include{file\string_name}".

*Documentation file assembled by Robin Fairbairns

Option [strings]

The default operation is quite simple and needs no customization; but you must avoid using "_" in any place where LaTeX uses an argument as a string of characters for some control function or as a name. These include the tags for \cite and \ref, file names for \input, \include, and \includegraphics, environment names, counter names, and placement parameters (like [t]). The problem with these contexts is that they are 'moving arguments' but LaTeX does not 'switch on' the "\protect mechanism" for them.

If you need to use the underscore character in these places, the package option [strings] is provided to redefine commands that take such a string argument so that protection is applied (with \protect made to be \string). The list of commands this provision affects is given in \UnderscoreCommands, with \do before each one; plus several others covering \input, \includegraphics, \cite, \ref, and their variants. Not included are many commands regarding font names, anything with counter names, environment names, page styles, and versions of \ref and \cite defined by external packages (e.g., \vref and \citeyear).

You can add to the list of supported commands by defining \UnderscoreCommands before loading this package; e.g.

```
\usepackage{chicago}
\newcommand{\UnderscoreCommands}{%   (\cite already done)
  \do\citeNP \do\citeA \do\citeANP \do\citeN \do\shortcite
  \do\shortciteNP \do\shortciteA \do\shortciteANP \do\shortciteN
  \do\citeyear \do\citeyearNP
}
\usepackage[strings]{underscore}
```

Not all commands can be supported this way! Only commands that take a string argument *first* can be protected. One optional argument before the string argument is also permitted, as exemplified by \cite: both \cite{tags} and \cite[text]{tags} are allowed. A command like \@addtoreset which takes two counter names as arguments could not be protected by listing it in \UnderscoreCommands.

When you use the [strings] *option, you must load this package* ***last*** (or nearly last). There are two reasons for this requirement:

1. The redefinitions done for protection must come after other packages define their customized versions of those commands.

2. The [strings] option requires the "_" character to be activated immediately in order for the cite and ref tags to be read properly from the .aux file as plain strings, and this catcode setting might disrupt other packages.

The babel package implements a protection mechanism for many commands, and will be a complete fix for most documents without the [strings] option. Many add-on packages are compatible with babel, so they will get the strings protection also. However, there are several commands that are not covered by babel, but can easily be supported by [strings] and \UnderscoreCommands mechanism. Beware the potential conflict using both [strings] and babel (though none have been reported, yet); load babel last.

Implementation notes

The first setting of "_" to be an active character is performed in a local group so as to not interfere with other packages. The catcode setting is repeated with "\AtBeginDocument" so the

definition is in effect for the text. However, the catcode setting is repeated immediately when the [strings] option is detected.

The definition of the active "_" is essentially:

```
\ifmmode \sb \else \BreakableUnderscore \fi
```

where \sb retains the normal subscript meaning of _ and where \BreakableUnderscore is essentially _. The rest of the definition handles the \protection without causing \relax to be inserted before the character.

\BreakableUnderscore uses \nobreak\hskip\z@skip to separate the underscore from surrounding words, thus allowing TEX to hyphenate them, but preventing free breaks around the underscore. Next, it checks the current font family, and uses the underscore character from tt fonts or otherwise \textunderscore (which is a character or rule depending on the font encoding). After the underscore, it inserts a discretionary hyphenation point as \usc@dischyph, which is usually just \- except that it still works in the tabbing environment; if the [nohyphen] option is in effect, the empty discretionary \discretionary{}{}{} is used instead. After that, another piece of non-breaking interword glue is inserted.

Ordinarily, the comparison \ifx\f@family\ttdefault will fail because \ttdefault is 'long' whereas \f@family is not[1], but \ttdefault is redefined to be non-long \AtBeginDocument.

The _ command is then defined to use \BreakableUnderscore.

If the [strings] option has not been given, that is all!

Under the [strings] option, the list of special commands is processed to:

- retain the original command as \US_⟨*command*⟩ (e.g., \US_ref)
- redefine the command as \US@prot\US_command for ordinary commands (\US@prot\US_ref) or as \US@protopt\US_command when an optional argument is possible (e.g., \US@protopt\US_bibitem).
- self-protecting commands (e.g., \cite) retain their self-protection.

Diagnosing the state of the pre-existing command is done by painful contortions involving \meaning.

\US@prot and \US@protopt read the argument, process it with \protect enabled, then invoke the saved \US_command.

[1] the package author says "boooo hisss" about this...