1 The Piedmontese language

The file `piedmontese.dtx`[1] defines all the language definition macros for the Piedmontese language[2].

The macro `\LdfInit` takes care of preventing that this file is loaded more than once, checking the category code of the `@` sign, etc. `\CurrentOption` is actually the name of language that was specified as an option in the call to babel.

```
\LdfInit{\CurrentOption}{captions\CurrentOption}
```

When this file is read as an option to the `\usepackage` command that loads `babel`, `piedmontese` could be an 'unknown' language, or better, a language who's patterns have not been loaded into the format file; in this case we have to make it known. So we check for the existence of `\l@piedmontese` to see whether we have to do something here. If the Piedmontese patterns have not (yet) been loaded in the format file, we prefer to declare `piedmontese` a dialect of `italian` rather than a dialect of `english`, although we resort to the latter language if also the Italian patterns have not been loaded into the format file: this unfortunately happens when a basic TEX system installation is being used.

```
\LdfInit\CurrentOption{captions\CurrentOption}
\ifx\l@piedmontese\@undefined
    \@nopatterns{piedmontese}
    \ifx\l@italian\@undefined
        \@nopatterns{italian}
        \adddialect\l@piedmontese\l@english
    \else
        \adddialect\l@piedmontese\l@italian
    \fi
\fi
```

`\captionspiedmontese` The macro `\captionspiedmontese` defines all strings used in the four standard documentclasses provided with LATEX.

```
\@namedef{captions\CurrentOption}{%
    \def\prefacename{Prefassion}%
    \def\refname{Riferiment}%
    \def\abstractname{Somari}%
    \def\bibname{Bibliograf\‘ia}%
    \def\chaptername{Cap\‘itol}%
    \def\appendixname{Gionta}%
    \def\contentsname{T\‘aula}%
    \def\listfigurename{Lista dle figure}%
    \def\listtablename{Lista dle tabele}%
    \def\indexname{T\‘aula anal\‘itica}%
    \def\figurename{Figura}%
    \def\tablename{Tabela}%
    \def\partname{Part}%
```

[1] The file described in this section has version number v.1.0 and was last revised on 2013/02/12.

[2] Language ISO 639 coded with letter identification `pms`. This file was created by Claudio Beccari; see the source file for more information.

```
    \def\enclname{Gionta/e}%
    \def\ccname{Con c\‘opia a}%
    \def\headtoname{P\"er}%
    \def\pagename{P\‘agina}%
    \def\seename{v\"ed}%
    \def\alsoname{v\"ed anche}%
    \def\proofname{Dimostrassion}%
    \def\glossaryname{Glossari}%
}
```

\datepiedmontese The macro \datepiedmontese redefines the command \today to produce Piedmontese dates. The proposition corresponding to 'of' has three spellings in Piedmontese, one general (ëd), one to be used in front of words starting with a group of consonants (dë), and a third one, that derives from the latter by vocalic elision (d') to be used in front of words starting with a vowel.

```
\@namedef{date\CurrentOption}{%
    \def\today{\number\day\space \space\ifcase\month\or
      \"ed gen\’e\or \"ed fevr\’e\or \"ed mars\or d’avril\or
      \"ed maj\or \"ed giugn\or \"ed luj\or d’agost\or
      d\"e st\‘ember\or d’ot\’ober\or \"ed nov\‘ember\or
      d\"e dz\‘ember\fi\space dal\space\number\year}}
```

\piedmontesehyphenmins This macro is used to store the correct values of the hyphenation parameters \lefthyphenmin and \righthyphenmin.

```
\providehyphenmins{\CurrentOption}{\tw@\tw@}
```

\extraspiedmontese \noextraspiedmontese The next step consists of defining commands to switch to (and from) the Piedmontese language.

The macro \extraspiedmontese will perform all the extra definitions needed for the Piedmontese language. The macro \noextraspiedmontese is used to cancel the actions of \extraspiedmontese. In particular we set pretty high default values for widow and club lines and very high demerits to avoid that the last line starts with the second half of a hyphenated word. We also assign a non zero value \lccode to the apostrophe that in Piedmontese is being used for marking a vocalic elision and for apocope; by giving it a non zero value, the hyphenation algorithm treats the phrase formed by the article or articulated preposition and the following term as a single word and the patterns for Piedmontese take care of avoiding line breaks right after the apostrophe.

```
\expandafter\addto\csname extras\CurrentOption\endcsname{%
  \babel@savevariable\clubpenalty
  \babel@savevariable\widowpenalty
  \babel@savevariable\@clubpenalty
  \clubpenalty3000\widowpenalty3000\@clubpenalty\clubpenalty}%
\expandafter\addto\csname extras\CurrentOption\endcsname{%
  \babel@savevariable\finalhyphendemerits
  \finalhyphendemerits50000000}%
\expandafter\addto\csname extras\CurrentOption\endcsname{%
  \lccode‘’=‘’}%
```

```
\expandafter\addto\csname noextras\CurrentOption\endcsname{%
  \lccode‘’=0}%
```

The double straight quote " is made active for the current language. Its purpose is primarily to produce a diacritic hyphen; meanwhile it is convenient to let it perform some other little tasks, such as to insert an etymologic line break, to simplify the `\slash` command, and to ease the introduction of the double raised open quotes, that are cumbersome to insert with the Italian keyboard.

The main activation of the " sign is done through the babel command `\declare@shorthand` that checks if the " has been used in math or in text mode; if in text mode, a service macro `\pms@next` is defined that upon execution lets a future token be aliased by the implicit token c.s. `\pms@temp`, while the actual work is demanded to `\pms@cwm`.

```
\initiate@active@char{"}%
\expandafter\addto\csname extras\CurrentOption\endcsname{%
\bbl@activate{"}\languageshorthands{piedmontese}}%
\declare@shorthand{piedmontese}{"}{%
\ifmmode
    \def\pms@next{’’}%
\else
    \def\pms@next{\futurelet\pms@temp\pms@cwm}%
\fi
\pms@next
}%
```

After defining another service macro `\pms@@cwm` for introducing a discretionary break that may be inserted in the middle of a word, allowing another break point, but allowing hyphenation in both letter strings that sit at either side of this break point. The the `\pms@cwm` macro is defined to perform the necessary actions depending on the nature of the token memorized into the temporary c.s. `\pms@temp`. The `\@gobble` macro must be repeated at each test, otherwise it gobbles the `\fi` of the outer test. An `\expandafter` construct might reduce this repetition.

```
\DeclareRobustCommand*{\pms@cwm}{\let\pms@@next\relax
\ifcat\noexpand\pms@temp a%
    \def\pms@@next{\pms@@cwm}%
\else
    \ifx\pms@temp/%
        \def\pms@@next{\bbl@allowhyphens/\bbl@allowhyphens\@gobble}%
    \else
        \ifx\pms@temp-%
           \def\pms@@next{\bbl@allowhyphens-\bbl@allowhyphens\@gobble}%
        \else
            \ifx\pms@temp"%
                \def\pms@@next{‘‘\expandafter\@gobble\string}%
            \fi
        \fi
    \fi
\fi
\pms@@next}%
```

\noextraspiedmontese This done, we are ready to prepare the switch back to another language"

```
\expandafter\addto\csname noextras\CurrentOption\endcsname{%
\bbl@deactivate{"}}
```

The macro `\ldf@finish` takes care of looking for a configuration file, setting the main language to be switched on at `\begin{document}` and resetting the category code of `@` to its original value.

```
\ldf@finish\CurrentOption
```