The Occitan language module for babel

Cédric Valmary— cvalmary at yahoo dot fr

2016/02/04 – version 0.2

1 The Occitan language

Important notice: This language description file relies on functionalities provided by a modern TeX system distribution with pdfLaTeX working in extended mode (eTeX commands available); it should perform correctly also with XeLaTeX and LuaLaTeX; tests have been made also with the latter programs, but it was really tested in depth with babel and pdfLaTeX. Actually, even if it possible to use babel with XeLaTeX and LuaLaTeX, the latter programs have available the specific polyglossia package with the proper settings also for the Occitan language.

The Occitan language nowadays exists as a *koinè* for the multitude of Occitan varieties, each one with its dialects; the main varieties are Auvernhat, Gascon, Lemosin, Lengadocian, Niçard, Porovençal, Vivaroalpenc as they are named in the document published by the *Conselh de la lenga occitana* (see http://gianni.vacca.perso.sfr.fr/guilhemix/clo-sintesi.pdf). This Conselh takes care also of maintaining a common standard at least for spelling, but admits a variety of pronunciations and in some cases a variety of spellings; for example the Gascon variety changes the initial 'f' into an aspirated 'h'; this implies the necessity of distinguishing when the digraphs 'lh', 'nh', and 'sh' are really digraphs or are to be considered distinct letters that in Gascon are rendered as 'l·h', 'n·h', and 's·h' respectively. A similar situation takes place in the Vivaroalpenc variety, where the intervocalic 'd' vanishes. This language description file should not be concerned with these variant spellings; the hyphenation pattern files might not correctly handle all such situations so that some manual intervention might be necessary. The tests carried on so far apparently work correctly, but users are invited to let me know about possible necessary corrections.

The file occitan.dtx defines all the required and some optional language-specific macros for the Occitan language.

The features of this language definition file are the following:

1. The Occitan hyphenation is invoked, provided that the Occitan hyphenation pattern files were loaded when the specific format file was built.

`"`	inserts a compound word mark where hyphenation is legal; it allows etymological hyphenation which is recommended for technical terms, chemical names and the like; it does not work if the next character is represented with a control sequence or is an accented character.
`"|`	the same as the above without the limitation on characters represented with control sequences or accented ones.
`"<`	inserts open guillemets without trailing space.
`">`	inserts closed guillemets without leading space.
`"/`	allows hyphenation of both words connected with slash.
`".`	allows hyphenation of both words fragments connected with a half height dot (ponch interior in Occitan)

Table 1: Shorthands for the Occitan language. These shorthands are available only if command `\setactivedoublequote` is given after loading babel and before `\begin{document}`.

2. The language dependent infix words to be inserted by such commands as `\chapter`, `\caption`, `\tableofcontents`, etc. are redefined in accordance with the Occitan typographical practice.

3. Since Occitan can be easily hyphenated and Occitan practice allows to break a word before the last two letters, hyphenation parameters have been set accordingly, but a very high demerit value has been set in order to avoid word breaks in the penultimate line of a paragraph. Specifically the `\clubpenalty`, and the `\widowpenalty` are set to rather high values and `\finalhyphendemerits` is set to such a high value that hyphenation is strongly discouraged between the last two lines of a paragraph.

4. Some language specific shorthands have been defined so as to allow etymological hyphenation, specifically `"` inserts a break point at any word boundary that the typesetter chooses, provided it is not followed by an accented letter, and `"|` when the desired break point falls before an accented letter. As you can read in table 1, these shorthands are available only if they get activated with `\setactivedoublequote` after loading babel but before the `\begin{docuemnt}` statement. This is done in order to preserve the user from package conflicts: if s/he wants to use these facilities s/he must remember that conflicts may arise unless active characters are deactivated; this can be done with the babel command `\shorthadsoff{"}` (and reactivated with `\shorthandson{"}`) when its wise to do so; conflicts have been reported with package `xypic` and with `TikZ`, but the latter has its own library to deactivate all active characters, not just the double quotes, the only Occitan language specific activated character.

5. The shorthands `"<` and `">` insert the guillemets used in Occitan typography, but without any spacing as, on the opposite, it is done in French typography;

with the T1 font encoding the ligatures << and >> should insert such signs directly, but not all the virtual fonts that claim to follow the T1 font encoding actually contain the guillemets; with the OT1 encoding the guillemets are not available and must be faked in some way. By using the "< and "> shorthands (even with the T1 encoding) the necessary tests are performed and in case the guillemets are replaced with the ones contained in the Latin Modern fonts. At the same time if OpenType fonts are being used with XeLaTeX or LuaLaTeX, there are no problems with guillemets.

1.1 The commented code

The macro `\LdfInit` takes care of preventing that this file is loaded more than once, checking the category code of the `@` sign, etc.

```
\LdfInit{occitan}{captionsoccitan}%
```

When this file is read as an option, i.e. by the `\usepackage` command, `occitan` will be an 'unknown' language in which case we have to make it known. So we check for the existence of `\l@occitan` to see whether we have to do something here.

```
\ifx\l@occitan\@undefined
    \@nopatterns{occitan}%
    \adddialect\l@occitan \z@\fi
```

The next step consists of defining commands to switch to (and from) the Occitan language.

`\captionsoccitan` The macro `\captionsoccitan` defines all strings used in the four standard document classes provided with LaTeX.

```
\addto\captionsoccitan{%
  \def\refname{Refer\'encias}%
  \def\abstractname{Resumit}%
  \def\bibname{Bibliografia}%
  \def\prefacename{Prefaci}%
  \def\chaptername{Cap\'itol}%
  \def\appendixname{Ann\`ex}%
  \def\contentsname{Ensenhador}%
  \def\listfigurename{Taula de las figuras}%
  \def\listtablename{Taula dels tabl\`eus}%
  \def\indexname{Ind\`ex}%
  \def\figurename{Figura}%
  \def\tablename{Tabl\`eu}%
  \def\partname{Partida}%
  \def\pagename{Pagina}%
  \def\seename{vejatz}%
  \def\alsoname{vejatz tanben}%
  \def\enclname{P\`e\c{c}a junta}%
  \def\ccname{c\`opia a}%
  \def\headtoname{A}%
  \def\proofname{Demostracion}%
```

```
  \def\glossaryname{Glossari}%
}%
```

`\dateoccitan` The macro `\dateoccitan` defines the command `\today` to typeset the current date according to the Occitan style of using the suitable prepositions before the month name and the year number; moreover the first day of the month is marked with the ordinal abbreviation, while the other day numbers are left without any ordinal indication.

```
\def\dateoccitan{%
  \def\today{%
  \ifcase \day\or
    1èr\else
    \number\day
  \fi~%
  \ifcase\month\or
    de~geni\‘er\or
    de~febri\‘er\or
    de~mar\c{c}\or
    d’abril\or
    de~mai\or
    de~jiunh\or
    de~julhet\or
    d’agost\or
    de~setembre\or
    d’octobre\or
    de~novembre\or
    de~decembre%
  \fi\space
  de~\number\year}}%
```

`\occitanhyphenmins` The occitan hyphenation patterns can be used with both `\lefthyphenmin` and `\righthyphenmin` set to 2.

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

`\extrasoccitan` `\noextrasoccitan` Lower the chance that clubs or widows occur; lower the chances that a line break takes place between the last two lines of a paragraph.

```
\addto\extrasoccitan{%
  \babel@savevariable\clubpenalty
  \babel@savevariable\widowpenalty
  \babel@savevariable\@clubpenalty
  \clubpenalty3000\widowpenalty3000\@clubpenalty\clubpenalty}%

\addto\extrasoccitan{%
  \babel@savevariable\finalhyphendemerits
  \finalhyphendemerits50000000}%
```

In order to enable the hyphenation of words such as “l’Occitània” we give the `’` a non-zero lower case code. This allows TEX to find the following hyphenation points `l’Oc-ci-tà-nia` instead of none.

```
\addto\extrasoccitan{%
  \lccode‘’=‘’}%
\addto\noextrasoccitan{%
  \lccode‘’=0}%
```

1.2 Support for etymological hyphenation

In Occitan etymological hyphenation is desirable with technical terms, chemical names, and the like. We reach this goal by means of shorthands tied to the active character ". The active double straight quote may conflict with other packages; so we set it as an optional facility.

Instead of a boolean switch we use a private counter so as to check at `\begin{document}` if this facility has to be activated. The default value is zero; anything different from zero means that the facility has to be activated; this is done with command `\setactivedoublequote` to be issued before `\begin{document}`

```
\newcount\oc@doublequoteactive \oc@doublequoteactive=\z@
\def\setactivedoublequote{\oc@doublequoteactive=\@ne}
\AtBeginDocument{%
  \unless\ifnum\oc@doublequoteactive=\z@
  \initiate@active@char{"}%
  \addto\extrasoccitan{\bbl@activate{"}\languageshorthands{occitan}}%
```

`\oc@cwm` The active character " is now defined for language `occitan` so as to perform different actions in math mode compared to text mode; specifically in math mode a double quote is inserted so as to produce a double prime sign, while in text mode the temporary macro `\oc@next` is defined so as to defer any further action until the next token category code has been tested.

```
  \declare@shorthand{occitan}{"}{%
    \ifmmode
      \def\oc@next{’’}%
    \else
      \def\oc@next{\futurelet\oc@temp\oc@cwm}%
    \fi
    \oc@next
  }%
\fi
```

The following statement must be conditionally executed after the above modification of the `\extraasoccitan` list; in facts at the “begin document” execution, the main language has already been set without the above modifications; therefore nothing takes place unless the occitan main language is selected again with the explicit command `\selectlanguage`; without this trick the active double quotes would remain inactive; of course `\languagename` contains the string `occitan` if this language was the main one; by testing this string, the suitable command may be issued again with the new settings and the double quotes become really active.

```
\ifdefstring{\languagename}{occitan}{\selectlanguage{occitan}}{\relax}
}%
```

\oc@cwm The \oc@next service control sequence is such that upon its execution a temporary variable \oc@temp is made equivalent to the next token in the input list without actually removing it. Yes, this is a point to be underlined: a token that has made equivalent with \futurelet to some other token, does not remove the latter; while a macro that is followed by a space ignores it and reads the first non-space token. This can be exploited in the following macros.

Such temporary token is then tested by the macro \oc@cwm and if it is found to be equivalent to a letter token (catcode=11), then it introduces the compound word separator control sequence \oc@allowhyphens whose expansion introduces a discretionary hyphen and an unbreakable zero space; otherwise the token is not a letter; then it is tested against $|_{12}$: if so a macro is defined that gobbles the token and introduces a compound word separator; otherwise two other tests are performed to see if guillemets have to be inserted, and in case suitable intelligent guillemet macros are introduced that gobble unwanted leading or trailing spaces; otherwise a test is made to see if the next char is a slash character, and in case a special discretionary break is inserted so as to maintain the slash while allowing the hyphenation of both words before and after the slash; otherwise another test is made to see if a period follows, and in case the period is gobbled and a special discretionary is inserted that introduces a hyphen sign if a line break occurs, or a centered dot otherwise; otherwise nothing is done.

```
\def\oc@@cwm{\bbl@allowhyphens\discretionary{-}{}{}\bbl@allowhyphens}%
\def\oc@@slash{\bbl@allowhyphens\discretionary{/}{}{/}\bbl@allowhyphens}%
\def\oc@ponchinterior{\nobreak
            \discretionary{-}{}{\mbox{$\cdot$}}\nobreak\hskip\z@skip}%
\def\oc@@oguil#1{\oc@oguil}\def\oc@@cguil#1{\oc@cguil}%

\DeclareRobustCommand*{\oc@cwm}{\let\oc@@next\oc@doublequote
\ifcat\noexpand\oc@temp a%
    \def\oc@@next{\oc@@cwm}%
\else
    \if\noexpand\oc@temp \string|%
        \def\oc@@next##1{\oc@@cwm}%
    \else
        \if\noexpand\oc@temp \string<%
            \def\oc@@next{\oc@@oguil}%
        \else
            \if\noexpand\oc@temp \string>%
                \def\oc@@next{\oc@@cguil}%
            \else
                \if\noexpand\oc@temp\string/%
                    \def\oc@@next##1{\oc@@slash}%
                \else
                    \if\noexpand\oc@temp\string.%
                        \def\oc@@next##1{\oc@ponchinterior}%
                    \fi
                \fi
            \fi
        \fi
```

```
    \fi
\fi
\oc@@next}%
```

By this definition of " if one types `macro"istrucion` the possible break points become ma-cro-is-tru-cion, while without the " mark they would be ma-crois-tru-cion, according to the phonetic rules such that the `macro` prefix is not taken as a unit. A chemical name such as `des"clor"fenir"amina"cloridrat` [1] is breakable as des-clor-fe-nir-ami-na-clo-ri-drat instead of des-clor-fe-ni-ra-mi-na-... Of course the use of this " functionality is useful if it is used for single words or to fine tune a final document. If a certain word that requires special break points appears quite often in a specific document, it would be much more convenient to specify these special break points in the argument of a `\hyphenation` command. See the babel documentation to set hyphenation exceptions for several languages in a specific document.

In other language description files a shorthand is defined so as to allow a break point without actually inserting any hyphen sign; examples are given such as input/output; actually if one wants to allow a breakpoint after the slash, it is much clearer to type `\slash` instead of `/` and LaTeX does everything by itself; here the shorthand `"/` was introduced to stand for `\slash` so that one can type `input"/output` and allow a line break after the slash.

But what is really important in some Occitan varieties is the macro `\oc@ponchinterior` that inserts the special discretionary break that inserts a hyphen sign if a line break takes place or a centered dot otherwise.

Attention: the expansion of " takes place before the actual expansion of OT1 or T1 accented sequences such as `\‘{a}`; therefore this etymological hyphenation facility works as it should only when the semantic word fragments *do not start* with an accented letter; this in Occitan is almost always avoidable, because very rarely accented vowels start a new syllable and the only consonant that carries a diacritic mark, ‘ç’, is already taken care of by the hyphenation pattern files. In this case the special shorthand `"|` may be used that performs exactly as " normally does, except that the | sign is removed from the token input list: `kilo\"orsted` or `kiloörsted` gets hyphenated as `ki-loör-sted`; but `kilo"|örsted` gets hyphenated correctly as `ki-lo-ör-sted` The `"|` macro is necessary because, even with a suitable option specified to the `inputenc` package, the letter ‘ö’ does not have category code 11, as the ASCII letters do, because of the LICR (LaTeX Internal Character Representation); the LICRs are the set of intermediate macros that have to be expanded in order to fetch the proper glyph in the output font or to build up a composite glyph if it is not available in the output font.

1.3 Extra advanced macros

We need to perform some tests that require some smart control-sequence handling; therefore we call the `etoolbox` package that allows us more testing functionality.

[1] Not all " signs are necessary, but all have been indicated in order to mark in an explicit way the various components of the technical compound word.

à		á	
è		é	
	ï	í	
ò		ó	
	ü	ú	
			ç

Table 2: Specific Occitan characters

There are no problems with this package that can be invoked also by other ones before or after babel is called; the `\RequirePackage` mechanism is sufficiently smart to avoid reloading the same package more than once. But we have to delay this call, because `occitan.ldf` is being read while processing the options passed to babel and while options are being scanned and processed it is forbidden to load packages; we delay it at the end of processing the babel package itself.

```
111 \AtEndOfPackage{\RequirePackage{etoolbox}}
```

1.4 Accents

Most of the other language description files introduce a number of shorthands for inserting accents and other language specific diacritical marks in a more comfortable way compared with the lengthy standard TEX conventions. I don't know if every user has a specific keyboard layout or keyboard driver dedicated to the Occitan language; it is possible that Occitan users, living in countries where the Occitan language is actually used (France, Italy, and Spain), use the more common local national keyboard and have already available the suitable keys for entering (some of)the Occitan non ASCII characters; table 2 displays the lowercase special Occitan characters. Among the national keyboards it seems that the Italian one is the least suited to typeset in Occitan; it actually is not fully functional with Italian itself (it misses any two-key combination to enter accented uppercase letters). In any case the recent distributions of the Windows operating systems (may be from Win7 on, certainly from Win8) and the Apple OS X have a virtual keyboard application that allows to enter any glyph by clicking on the virtual keys; all operating systems have a Character Table Viewer that allows to enter any UNICODE glyph (very uncomfortable if needed to enter long stretches of text, but...). I suppose that the French national keyboard is the most comfortable to use when entering Occitan text, thanks to the fact that the French language uses many diacritics.

The best solution, may be, consists in using a smart editor that accepts shorthand definitions such that, for example, by striking `"a` one gets directly ä on the screen and the same string is saved into the `.tex` file; the same smart editor might be capable of translating the accented characters into the standard TEX sequences when writing a file to disk (for the sake of file portability), and to transform the standard TEX sequences into the corresponding signs when loading a `.tex` file from disk to screen memory. Such smart editors do exist and can be downloaded from the CTAN archives.

1.5 Guillemets or French double quotes

Although the T1 font encoding ligatures solve the problem, there are some circumstances where even the T1 font encoding cannot be used, either because the author/typesetter wants to use the OT1 encoding, or because s/he uses a font set that does not comply completely with the T1 font encoding; some virtual fonts, for example, are supposed to implement the double Cork font encoding but actually miss some glyphs; one such virtual font set is given by the `ae` virtual fonts, because they are supposed to implement such double font encoding by using virtual fonts that map the `CM` fonts to a T1 font scheme; the type 1 PostScript version of the `CM` fonts do exist, therefore one believes to be able of using them with pdfLaTeX; but since the `CM` fonts do not contain the guillemets, neither do the `AE` ones. Since guillemets do not exist in any OT1 encoded `cm` Latin font, their glyphs must be substituted with something else that fakes them. Therefore if the OT1 encoding is being used the T1 encoded Latin Modern font guillemets are used, otherwise the current font ones are actually used.

`\oc@oguil` `\oc@cguil` A new macro that the user may possibly use, if the default does not meet his/her requirements, is defined so as to chose which family the guillemets should be taken from; but through the `\DeclareTextCommand`, we tie the definitions of the macros `\oc@oguil` and `\oc@cguil` to the current actual encoding.

```
\def\GuillemetsFrom#1#2#3#4{%
  \DeclareFontEncoding{#1}{}{}%
  \DeclareTextCommand{\oc@oguil}{T1}{%
    {\fontencoding{#1}\fontfamily{#2}\selectfont\char#3\ignorespaces}}%
  \DeclareTextCommand{\oc@cguil}{T1}{\ifdim\lastskip>\z@\unskip\fi%
    {\fontencoding{#1}\fontfamily{#2}\selectfont\char#4}}%
  \DeclareTextCommand{\oc@oguil}{OT1}{%
    {\fontencoding{#1}\fontfamily{#2}\selectfont\char#3\ignorespaces}}%
  \DeclareTextCommand{\oc@cguil}{OT1}{\ifdim\lastskip>\z@\unskip\fi%
    {\fontencoding{#1}\fontfamily{#2}\selectfont\char#4}}}
```

This macro requires four arguments with the syntax:

`\GuillemetsFrom{`⟨*encoding*⟩`}{`⟨*family*⟩`}{`⟨*open guill. slot*⟩`} {`⟨*closed guill. slot*⟩`}`

where ⟨*encoding*⟩ and ⟨*family*⟩ identify the font family name of that particular encoding from which to get the substitution guillemets; ⟨*open guill. slot*⟩ and ⟨*closed guill. slot*⟩ are the (preferably) decimal slot addresses of the opening and closing guillemets the user wants to use. For example if the T1-encoded Latin Modern fonts are desired, the specific command should be

```
\GuillemetsFrom{T1}{lmr}{19}{20}
```

We define a default macro for using the current font in encoding T1:

```
\def\T@unoGuillemets{\DeclareRobustCommand*{\oc@oguil}{<<\ignorespaces}%
    \DeclareRobustCommand*{\oc@cguil}{\ifdim\lastskip>\z@\unskip\fi>>}}%
```

Notice that the above macro is strictly tied to the T1 encoding; it won't do anything if the default encoding is not the T1 one. Therefore if the `AE` font collection

is being used it would be a good idea to issue the command (shown above as an example) in order to get the proper guillemets; of course using directly the LM fonts instead of the AE ones would be a much better idea.

Now we set a boolean variable and test the default family; if such family has a name that starts with the letters "ae" then we have no built in guillemets; of course if the AE fonts are chosen after the babel package is loaded, the test does not perform as expected.

```
\def\get@ae#1#2#3!{\def\oc@ae{#1#2}}%
\def\@ifT@one@noGuil{\expandafter\get@ae\f@family!%
\ifdefstring{\oc@ae}{ae}}%
```

Now we can set some real settings; first we start by testing the encoding; if the encoding is OT1 we substitute the missing guillemets with the Latin Modern ones and issue a message; then we test if the font family is the AE one and we set again the Latin Modern ones and issue another message[2]; otherwise we set the commands valid for the T1 encoding, that work well also with the TeX Ligatures of the OpenType fonts.

```
\AtBeginDocument{\normalfont
  \ifdefstring{\cf@encoding}{OT1}{%
    \GuillemetsFrom{T1}{lmr}{19}{20}%
    \GenericWarning{occitan.ldf\space}{%
    File occitan.ldf warning: \MessageBreak\space\space\space
    With OT1 encoding guillemets are taken form the
    \MessageBreak\space\space\space
    T1 encoded Latin Modern fonts\MessageBreak\space\space\space
    \MessageBreak\space\space}%
  }{%
    \ifdefstring{\cf@encoding}{T1}{%
      \@ifT@one@noGuil{%
        \GuillemetsFrom{T1}{lmr}{19}{20}%
        \GenericWarning{occitan.ldf\space}{%
        File occitan.ldf warning: \MessageBreak\space\space\space
        The AE font collection does not contain the guillemets
        \MessageBreak\space\space\space
        Using Latin Modern guillemets instead
        \MessageBreak\space}
      }{%
    \T@unoGuillemets}}{\T@unoGuillemets}
  }%
}
```

1.6 Finishing commands

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.

[2]Notice that it is impossible to check if the slots 19 and 20 of the AE fonts are defined by means of the eTeX macro \iffontchar, because they are actually defined as black squares!

```
\ldf@finish{occitan}%
```