

The Occitan language module for polyglossia

Cédric Valmary — `cvalmary at yahoo dot fr`

v.0.1 — 2016/02/04

Contents			
1 Usage	1	2.2 Option definitions	3
2 Documented code	3	2.3 The double quote active character	4
2.1 Initial settings	3	2.4 Occitan infix words	5
		2.5 Final clean-up	6

Abstract

This file describes the Occitan language module for *polyglossia*. It describes also the options that may be specified and their functionalities.

1 Usage

When selecting the Occitan language with *polyglossia* you have to use either

```
\setmainlanguage[babelshorthands]{occitan}  
or  
\setotherlanguage[babelshorthands]{occitan}
```

depending on the fact that Occitan is the main or a secondary document language. The option `babelshorhands` is, in facts, optional; if specified it defines the active double quote functionalities. See table 1. A few words are in order.

" This compound work marker is very useful. The hyphenation patterns developed for the Occitan language have been modelled on the French or Spanish ones especially for what concerns the separation between an ‘s’ followed by another consonant. Nevertheless the *Conselh de la lenga occitana* takes care of a common spelling for all the Occitan varieties, but leaves every speaker free to pronounce the way s/he likes best according to his/her dialect. Even if local pronunciations are allowed, the spelling varies a little between the Occitan varieties Auvernhat, Gascon, Lemosin, Lengadocian, Niçard, Provençal, and Vivaroalpenc. May be some particular spelling requires some manual intervention for a correct hyphenation. The hyphenation patterns have been developed taking into account some peculiarities of the Gascon and the Vivaroalpenc varieties, but better patterns will be created as long as suggestions/corrections arrive from the community of L^AT_EX users.

"	Followed by a single letter token inserts a compound word mark with the necessary discretionary break command and allows hyphenation of both strings that precede and follow this mark.
"	Behaves as " when the vertical bar is followed by a complex token (a control sequence) or anything different from a letter.
"<	Inserts open guillemets and eliminates space after «
">	Inserts closed guillemets and eliminates space before »
"/	Inserts a slash that allows hyphenation of both the preceding and the following word.
".	Inserts a centered dot (<i>ponch interior</i>) with a discretionary break that allows hyphenation of both word fragments.

Table 1: Occitan module shorthands

"| This shorthand should be useless within a `.tex` source file to be processed by UTF-8 aware engines as `XqLaTeX` and `LuaLaTeX`. Nevertheless it might be necessary to insert a discretionary break in a strange word that requires a real macro within it; in this case the "|" shorthand comes handy. The situation is different with 8-bit aware typesetting engines, because the `utf8` specified to the *inputenc* package changes every non-ASCII character into a LICR (LaTeX Internal Character Representation) which is substantially a macro; as such it is not recognised as a character by ", and this second compound word marker must be used.

"< and "> are used to set the guillemets with the proper spacing; French users generally leave in the source `.tex` file at least one space after the open ones and another space before the closed ones. This is supposed to be a bad practice for what concerns Occitan typesetting, therefore such commands take care of eliminating these unwanted spaces, while simplifying the keying.

". This is a very special shorthand; it is intended to distinguish, for example, 'sh' from 's-h' (and similarly for other such groups). For example *dis-har* is pronounced with a minimal pause between the sound of 's' and the aspired sound of 'h'; without the centered dot (*ponch interior*) the 'sh' is a digraph that is pronounced as the IPA phoneme /ʃ/. When *dis-har* (`dis".har` in the source file) gets hyphenated, it becomes *dis-har*.

This ends the few peculiarities of the `gloss-occitan` language description file.

2 Documented code

2.1 Initial settings

First we have to identify this file, and we start with the initial code written by Cédric Valmary.

```
1 %***** By Cédric Valmary
2 \ProvidesFile{gloss-occitan.ldf}[2016/02/04 v0.3 polyglossia:
3     module for Occitan]
```

Then we have to set up *polyglossia* in order to let the package know what language is handling; what is the name of the hyphenation pattern set; what are the minimum word fragment lengths of the first and respectively the last word fragment before or after a line break; the specific setting for punctuation spacing, the indentation of the first paragraph of a section; if *polyglossia* should use a special font family `\occitanfont` in case the user defined such a family.

```
4 \PolyglossiaSetup{occitan}{
5   hyphennames={occitan},
6   hyphenmins={2,2},
7   frenchspacing=true,
8   indentfirst=true,
9   fontsetup=true,
10 }
11 %*****
```

2.2 Option definitions

We now document the contributed extension required to create the optional functionality obtained from the double quote active character.

We set up the necessary machinery for the module option `babelshorthands`. We set it as a boolean key that does not require the explicit value `true` when it is specified to the module. The option must be tied to the Occitan language, so we also define its prefix `occitan@`. We simultaneously use the switch `\ifsystem@babelshorthands` in order to set the boolean key to `true` or `false`.

```
12 \define@boolkey{occitan}[occitan@]{babelshorthands}[true]{}
13
14 \ifsystem@babelshorthands
15   \setkeys{occitan}{babelshorthands=true}
16 \else
17   \setkeys{occitan}{babelshorthands=false}
18 \fi
```

At this point, in order to use the *babel* machinery to define active characters, we test it if it was already loaded by testing the definiteness of a specific macro. If the module `babelsh.def` was not loaded, we load it, then start preparing the ground to define the double quote `"` as an active character.

```
19 \ifcsundef{initiate@active@char}{%
20 \input{babelsh.def}%
21 \initiate@active@char{"}%
```

```
22 }{}
```

Now we are ready to assign a definition to the double quote " active character. The " active char is supposed to do a certain small collection of actions, different in math mode compared to text mode; therefore we define a service macro `\xpgoc@next` with a different meaning depending on the typesetting mode. Notice that in text mode the definition assigns to a token the meaning of the token that upon expansion of the macro follows directly `\xpgoc@cwm`. The assignment with `\futurelet` is executed before `\xpgoc@cwm` therefore it can pick up also the first space token that possibly follows the expansion of "; a macro would ignore such space.

```
23 \def\occitan@shorthands{%
24   \bbl@activate{"}%
25   \def\language@group{occitan}%
26   \declare@shorthand{occitan}{"}{%
27     \relax\ifmmode
28       \def\xpgoc@next{' '%}
29     \else
30       \def\xpgoc@next{\futurelet\xpgoc@temp\xpgoc@cwm}%
31     \fi
32   \xpgoc@next}%
33 }
```

2.3 The double quote active character

We now define a couple of service macros; `\xpgoc@@cwm` expands to an absolute `\nobreak` macro that forbids any line break; then a normal discretionary (the long definition with three arguments, is made through a primitive command, but if we used the standard `\-` control character, we would get the same performance); finally we put another `\nobreak` command and a zero width glob of glue; this zero-width, zero-stretch, zero-shrink glob of glue does not interfere with typesetting but is the actual trick that lets the typesetting engine understand that the incoming string of letters has to be treated as a word, so that the hyphenation algorithm continues working after the discretionary break.

Similarly the macro `\xpgoc@ponchinterior` works in the same way, but the discretionary break contains a non empty third argument that contains a box which in turn contains the centered dot.

```
34 \def\xpgoc@@cwm{\nobreak\discretionary{-}{-}\nobreak\hskip\z@skip}
35 \def\xpgoc@ponchinterior{%
36   \nobreak\discretionary{-}{-}{\mbox{${\cdot}}}\nobreak\hskip\z@skip}
```

Eventually we can make a chain of conditional statements that check what the `\xpgoc@temp` let token actually represents. We define another service macro `\xpgoc@@next` macro with different definitions depending on the status of the various tests. We first let it to a `\relax` no-op, in case that none of the tests is true; eventually we use its expansion that will do different actions depending on the tests. Notice that all the used conditionals expand their arguments; therefore it is necessary to use `\noexpand` in order to maintain the integrity of the tokens

to be compared. Notice also that the service macro is sometimes defined as an argument-less macro, and sometimes as a macro with one compulsory argument; in this latter case, since we are making definitions within another definition we have to double the hash sign. In the latter case it will ignore any spaces following it and get the first non blank token; in most cases it will gobble the first non blank token and discard it.

```

37 \def\xpgoc@cwm{\let\xpgoc@@next\relax
38 \ifcat\noexpand\xpgoc@temp a%
39 \def\xpgoc@@next{\xpgoc@cwm}%
40 \else
41 \if\noexpand\xpgoc@temp \string|%
42 \def\xpgoc@@next##1{\xpgoc@cwm}%
43 \else
44 \if\noexpand\xpgoc@temp \string<%
45 \def\xpgoc@@next##1{\ignorespaces}%
46 \else
47 \if\noexpand\xpgoc@temp \string>%
48 \def\xpgoc@@next##1{\unskip}>%
49 \else
50 \if\noexpand\xpgoc@temp\string/%
51 \def\xpgoc@@next##1{\slash}%
52 \else
53 \if\noexpand\xpgoc@temp\string.%
54 \def\xpgoc@@next##1{\xpgoc@ponchinterior}%
55 \fi
56 \fi
57 \fi
58 \fi
59 \fi
60 \fi
61 \xpgoc@@next}

```

Before going on we have to define what to delete when leaving the Occitan typesetting, so that another language may start working without any residue of the Occitan settings. In particular the double quote " active char must be deactivated.

```

62 \def\nooccitan@shorthands{%
63 \@ifundefined{initiate@active@char}{\bb1@deactivate{}}%
64 }

```

2.4 Occitan infix words

We can resume the original code for defining the infix Occitan words and to typeset the date.

```

65 %***** By Cédric Valmary
66 \def\captionsoccitan{%
67 \def\refname{Referéncias}%
68 \def\abstractname{Resumit}%
69 \def\bibname{Bibliografia}%
70 \def\prefacename{Prefaci}%

```

```

71 \def\chaptername{Capítol}%
72 \def\appendixname{Annèx}%
73 \def\contentsname{Ensenhador}%
74 \def\listfigurename{Taula de las figuras}%
75 \def\listtablename{Taula dels tablèus}%
76 \def\indexname{Indèx}%
77 \def\figurename{Figura}%
78 \def\tablename{Tablèu}%
79 %\def\thepart{}%
80 \def\partname{Partida}%
81 \def\pagename{Pagina}%
82 \def\seenname{vejatz}%
83 \def\alsoname{vejatz tanben}%
84 \def\enclname{Pèça junta}%
85 \def\ccname{còpia a}%
86 \def\headtoname{A}%
87 \def\proofname{Demostracion}%
88 \def\glossaryname{Glossari}%
89 }
90 \def\dateoccitan{%
91 \def\occitanmonth{\ifcase\month\or
92 de~genièr\or
93 de~febrièr\or
94 de~març\or
95 d'abril\or
96 de~mai\or
97 de~junh\or
98 de~julhet\or
99 d'agost\or
100 de~setembre\or
101 d'octobre\or
102 de~novembre\or
103 de~decembre\fi
104 }%
105 \def\occitanday{\ifcase\day\or
106 1èr\else% primièr
107 \number\day\fi% all other numbers
108 }%
109 \def\today{\occitanday\space \occitanmonth\space de~\number\year}%
110 }
111 %*****

```

2.5 Final clean-up

polyglossia requires that at `\begin{document}` time certain values are saved. Actually this is a prudential action: the user might have changed the normal values, so before setting the values valid for Occitan typesetting, its better to save the general values. The actual values that require saving are the club and widow penalties, together with the final demerits; the latter ones are the demerits in hy-

phenating the penultimate line of a paragraph so as not to terminate with a last line composed with a single syllable. The internal value `\@clubpenalty` must be saved, because sometimes it does not equal that of `\clubpenalty`

```
112 \let\xpgoc@savedvalues\empty
113 \AtEndPreamble{% the user or the class might define different values
114   \edef\xpgoc@savedvalues{%
115     \clubpenalty=\the\clubpenalty\space
116     \@clubpenalty=\the\@clubpenalty\space
117     \widowpenalty=\the\widowpenalty\space
118     \finalhyphendemerits=\the\finalhyphendemerits}
119 }
```

Eventually we define the definitive `\noextras@occitan` macro to undo everything that was done for setting up the typesetting of the Occitan language.

```
120 \def\noextras@occitan{%
121   \lccode\string"2019=\z@
122   \nooccitan@shorthands
123   \xpgoc@savedvalues
124 }
```

For setting up Occitan typesetting *polyglossia* requires two different settings: the general ones and the specific settings for typesetting in line.

```
125 \def\blockextras@occitan{%
126   \lccode\string"2019=\string"2019
127   \clubpenalty=3000 \@clubpenalty=3000 \widowpenalty=3000
128   \finalhyphendemerits=50000000
129   \ifoccitan@babelshorthands\occitan@shorthands\fi
130 }
131
132 \def\inlineextras@occitan{%
133   \lccode\string"2019=\string"2019
134   \ifoccitan@babelshorthands\occitan@shorthands\fi
135 }
```

This ends the documentation of the Occitan language module for *polyglossia*.