A Babel language definition file for Breton

breton.dtx v2.0.1, 2026-05-17

Jacques André
Jacques.Andre35@gmail.com

Contents

1 The Breton language

The file `breton.ldf`[1] defines all the language-specific macros for the Breton language.
There are not really typographic rules for the Breton language. It is a regional language (it's one of the celtic languages) which is spoken in Brittany (West of France). So we have a synthesis between Breton typographic rules and French typographic rules. The characters `:`, `;`, `!` and `?` get a whitespace automatically before them as in French.
This version (v2.0.1, 2026-05-17) supports new encoding for "the" character C'H, and uses new caption names. Furthermore, compatibilities between pdfTeX, lualatex and xelatex have been upgraded with the deep help of Daniel Flipo.

1.1 Basic interface

The `breton` language for Babel works for lualatex, xelatex and pdflatex.
Requirements:

1. user's documents *must* be coded in UTF-8, also for pdflatex.
2. pdflatex users *must* use fonts in T1 encoding, this means adding a `\usepackage[T1]{fontenc}` command in the preamble.

1.1.1 Punctuation

High punctuation (`:;!?`) in Breton requires, as in French, some space to be added in front of it. The automatic insertion of these spaces is handled in two different ways: for the pdftex and xetex engines, active characters are used, while luatex provides a safer alternative.
The default behaviour of `breton.ldf` is to add a thin (nobreak, no stretch) space in front of `;!?` and an inter-word (stretchable, nobreak) space in front of the colon.
These spaces are customisable through two commands `\BRSetThinSpace` and `\BRSetColonSpace`. They both require three mandatory arguments which are decimal numbers specifying *width*, *stretch* and *shrink* relative to the relevant *fontdimens*. Examples:
`\BRSetColonSpace{1}{1}{1}` provides exactly an inter-word space.
`\BRSetThinSpace{0.5}{0}{0}` provides a thin space (half width of the inter-word space, no elasticity).
The automatic insertion of spaces in front of high punctuation is disabled for Typewriter fonts and hence in verbatim mode.
A command `\NoBRspacing` is provided to disable the automatic insertion of spaces in front of high punctuation if needed. It is meant to be used inside a group. For instance with pdflatex or xelatex, coding `{\NoBRspacing (!)}` or `{\NoBRspacing ??}` to avoid spurious spaces is recommended. Fortunately lualatex handles these cases flawlessly.

[1] The file described in this section has version number v2.0.1 and was last revised on 2026-05-17.

1.1.2 Caption names

The former list of caption names (such as `\chaptername`) has been revisited, mainly according to recommendations of OPLB. Furthermore, `\partname` is now numbered with ordinals (e.g. *Lodenn gentañ, Eil lodenn, ...*) instead of roman numbers (e.g. *Lodenn I, Lodenn II, ...*).
To the regular `\today` is added a new command `\deiziad{year}{month}{day}` that prints the corresponding date in breton.

2 The code

2.1 Initial setup

```
\ProvidesLanguage{breton}
```

The macro `\LdfInit` takes care of preventing that this file is loaded more than once (even if both options `breton` and `acadian` are used in the same document), checking the category code of the `@` sign, etc.

```
\LdfInit{breton}{captionsbreton}
```

Quit if Babel's version is less than 24.14.

```
\let\bbl@tempa\relax
\ifdefined\babeltags
\else
  \let\bbl@tempa\endinput
  \PackageError{breton.ldf}
        {babel-breton requires babel v.24.14.\MessageBreak
         Aborting here}
        {Please upgrade Babel!}
\fi
\bbl@tempa
```

Make sure that `\l@breton` is defined (fallback is language 0 US-english).

```
\ifdefined\l@breton
\else
  \@nopatterns{breton}
  \adddialect\l@breton0
\fi
```

Breton uses the standard values of `\lefthyphenmin` (2) and `\righthyphenmin` (3); let's provide their values though, as required by Babel.

```
\providehyphenmins{breton}{\tw@\thr@@}
```

`\extrasbreton` `\noextrasbreton`

The macro `\extrasbreton` will perform all the extra definitions needed for the Breton language. The macro `\noextrasbreton` is used to cancel the actions of `\extrasbreton`.

In Breton, the quote character (U+27 or U+2019 or even 02BC) is a letter in expressions like `C'hwevrer` (February). Breton hyphenation patterns need to provide entries for this kind of words. This means that its `\lccode` has to be non null in Breton for proper hyphenation of those expressions, and has to be reset to null when exiting Breton.

```
\def\extrasbreton{%
    \babel@savevariable{\lccode"27}%
    \lccode"27="27
    \ifnum\bbl@engine > \z@
      \babel@savevariable{\lccode"2019}%
      \lccode"2019="2019
    \fi
}
```

One more thing \extrasbreton needs to do is to make sure that “Frenchspacing’’ is in effect. \noextrasfrench will switch “Frenchspacing’’ off again if necessary.
A new conditional is used to switch off, if needed, automatic insertion of spaces in front of ; ! ?, :, « and ».

```
\newif\ifBR@spacing
\addto\extrasbreton{\bbl@frenchspacing \BR@spacingtrue}
\addto\noextrasbreton{\bbl@nonfrenchspacing \BR@spacingfalse}
```

2.2 Breton apostrophe U+02BC

Strangely, some organisms, such as the Public Office for Breton Language (OPLB, Office Public de la Lange Bretonne, Ofis Publik ar Brezhoneg), recommends to encode the character C’H with the U+02BC apostrophe. As this character is missing from most fonts, it’s safer to replace it with a regular apostrophe.
Substitution of char U+02BC into U+0027 (pdftex) or U+2019 (unicode engines).
This is for the pdftex engine; caveat: the substitution holds for all languages, not just Breton.

```
\ifcase\bbl@engine
  \DeclareUnicodeCharacter{02BC}{\char"0027}
```

This is for the luatex engine.

```
\or
  \babelprehyphenation{breton}{ ’ }{ string = ’ , remove }
```

This is for the xetex engine.

```
\or
  \def\@gobblenext#1{}
  \babelcharclass{breton}{apos}{’}
  \babelinterchar[apos]{breton}{default, boundary}{apos}{’\@gobblenext}
\fi
```

2.3 Punctuation

\BRguillspace
\BRcolonspace
\BRthinspace

Let’s define default values for punctuation spaces.

```
\newcommand*{\BRcolonspace}{\space}
\newcommand*{\BRthinspace}{\hskip .5\fontdimen2\font \relax}
\newcommand*{\BRguillspace}{\hskip .8\fontdimen2\font
                              plus .3\fontdimen3\font
                             minus .8\fontdimen4\font \relax}
```

\BRSetGuillSpace
\BRSetColonSpace
\BRSetThinSpace

This command makes it easy to fine tune \BRthinspace and \BRcolonspace in Breton. It requires three mandatory arguments which are decimal numbers specifying *width*, *stretch* and *shrink* relative to the relevant *fontdimens*. For instance \BRSetColonSpace}{0.5}{0}{0} defines \BRcolonspace as a thinspace instead of a word space.

```
\newcommand*{\BRSetColonSpace}[3]{%
  \ifnum\bbl@engine=\@ne
    \SetTransformValue{breton}{colon.natural}{#1}%
    \SetTransformValue{breton}{colon.minus}{#2}%
    \SetTransformValue{breton}{colon.plus}{#3}%
  \else
    \renewcommand{\BRcolonspace}{%
       \hskip #1\fontdimen2\font
        plus  #2\fontdimen3\font
        minus #3\fontdimen4\font \relax}%
  \fi}
\newcommand*{\BRSetThinSpace}[3]{%
  \ifnum\bbl@engine=\@ne
    \SetTransformValue{breton}{thin.natural}{#1}%
    \SetTransformValue{breton}{thin.minus}{#2}%
    \SetTransformValue{breton}{thin.plus}{#3}%
  \else
    \renewcommand{\BRthinspace}{%
       \hskip #1\fontdimen2\font
        plus  #2\fontdimen3\font
        minus #3\fontdimen4\font \relax}%
  \fi}
\newcommand*{\BRSetGuillSpace}[3]{%
  \ifnum\bbl@engine=\@ne
    \SetTransformValue{breton}{guill.natural}{#1}%
    \SetTransformValue{breton}{guill.minus}{#2}%
    \SetTransformValue{breton}{guill.plus}{#3}%
  \else
    \renewcommand{\BRguillspace}{%
       \hskip #1\fontdimen2\font
        plus  #2\fontdimen3\font
        minus #3\fontdimen4\font \relax}%
  \fi}
```

2.3.1 Punctuation for the pdftex engine

`\bbl@engine` is a Babel command which expands to 0, 1 or 2 according to the engine used for compilation, resp. pdftex, luatex, xetex. For pdftex the four high punctuation characters are made active.

```
\ifcase\bbl@engine
  \initiate@active@char{:}
  \initiate@active@char{;}
  \initiate@active@char{!}
  \initiate@active@char{?}
```

We tune the amount of space before `;` `!` `?` and `:`. In horizontal mode, if a space has

been typed before ‘;’ we remove it. Then a non-breaking \BRthinspace is inserted.

```
\declare@shorthand{breton}{;}{%
  \ifBR@spacing
    \ifhmode
      \ifdim\lastskip>1sp \unskip \fi
      \penalty\@M \BRthinspace
    \fi
  \fi
```

Now we can insert a ; character.

```
  \string;}
```

The next three definitions are very similar.

```
\declare@shorthand{breton}{!}{%
  \ifBR@spacing
    \ifhmode
      \ifdim\lastskip>1sp \unskip \fi
      \penalty\@M \BRthinspace
    \fi
  \fi
  \string!}
\declare@shorthand{breton}{?}{%
  \ifBR@spacing
    \ifhmode
      \ifdim\lastskip>1sp \unskip \fi
      \penalty\@M \BRthinspace
    \fi
  \fi
  \string?}
\declare@shorthand{breton}{:}{%
  \ifBR@spacing
    \ifhmode
      \ifdim\lastskip>1sp \unskip \fi
      \penalty\@M \BRcolonspace
    \fi
  \fi
  \string:}
```

When the active characters appear in an environment where their Breton behaviour is not wanted they should give an ‘expected’ result. Therefore we define shorthands at system level as well.

```
\declare@shorthand{system}{:}{\string:}
\declare@shorthand{system}{!}{\string!}
\declare@shorthand{system}{?}{\string?}
\declare@shorthand{system}{;}{\string;}
```

We specify that the Breton group of shorthands should be used when switching to Breton.

```
\addto\extrasbreton{\languageshorthands{breton}%
```

These characters are ‘turned on’ once, later their definition may vary. Don’t misunderstand the following code: they keep being active all along the document, even when leaving Breton.

```
    \bbl@activate{:}\bbl@activate{;}%
    \bbl@activate{!}\bbl@activate{?}%
  }
  \addto\noextrasbreton{%
    \bbl@deactivate{:}\bbl@deactivate{;}%
    \bbl@deactivate{!}\bbl@deactivate{?}%
  }
```

French quotes (« and ») also require some additional space;

```
  \DeclareUnicodeCharacter{00AB}{\BR@og}
  \DeclareUnicodeCharacter{00BB}{\BR@fg}
  \newcommand*{\BR@og}{%
    \ifBR@spacing \guillemotleft\penalty\@M\BRguillspace\ignorespaces
    \else \guillemotleft
    \fi}
  \newcommand*{\BR@fg}{%
    \ifBR@spacing \ifdim\lastskip>\z@\unskip\fi \penalty\@M\BRguillspace
       \guillemotright
    \else \guillemotright
    \fi}
```

2.3.2 Punctuation for the luatex engine

For the LuaTeX engine, we use the “transforms’’ from Babel’s kernel and we exclude monospaced fonts from their application.

```
\or
  \babelprehyphenation[label=breton.space, fonts=rm sf]{\CurrentOption}
    { [^|{(}{[}{007B}!{?}] () [;!{?}] () }
    { { insert, penalty = 10000 },
      { insert, spacefactor= {high.natural|.5} {high.plus|0}
                             {high.minus|0}, data = 1 },
      {} }
  \babelprehyphenation[label=breton.space, fonts=rm sf]{\CurrentOption}
    { [^{(}{[}{007B}!{?}] () | [;!{?}] () }
    { { insert, penalty = 10000 },
      { spacefactor= {high.natural|.5} {high.plus|0}
                     {high.minus|0}, data = 2 },
      {} }
  \babelprehyphenation[label=breton.space, fonts=rm sf]{\CurrentOption}
    { [^|{(}{[}{007B} ] () : () }
    { { insert, penalty = 10000 },
      { insert, spacefactor= {colon.natural|1} {colon.plus|1}
```

```
                            {colon.minus|1}, data = 1 },
    {} }
\babelprehyphenation[label=breton.space, fonts=rm sf]{\CurrentOption}
  { [^{(}{[}{007B} ] () |: () }
  { { insert, penalty = 10000 },
    { spacefactor= {colon.natural|1} {colon.plus|1}
                   {colon.minus|1}, data = 2 },
    {} }
\babelprehyphenation[label=breton.space, fonts=rm sf]{\CurrentOption}
  { [^|]()»() }
  { { insert, penalty = 10000 },
    { insert, spacefactor= {guill.natural|.8} {guill.plus|.3}
                           {guill.minus|.8}, data = 1 },
    {} }
\babelprehyphenation[label=breton.space, fonts=rm sf]{\CurrentOption}
  { [^|]()|»() }
  { { insert, penalty = 10000 },
    { spacefactor= {guill.natural|.8} {guill.plus|.3}
                   {guill.minus|.8}, data = 2 },
    {} }
\babelprehyphenation[label=breton.space, fonts=rm sf]{\CurrentOption}
  { «[^|] }
  { {},
    { insert, penalty = 10000 },
    { insert, spacefactor= {guill.natural|.8} {guill.plus|.3}
                           {guill.minus|.8}, data = 1 },
    {} }
\babelprehyphenation[label=breton.space, fonts=rm sf]{\CurrentOption}
  { «|[^|] }
  { {},
    { insert, penalty = 10000 },
    { spacefactor= {guill.natural|.8} {guill.plus|.3}
                   {guill.minus|.8}, data = 1 },
    {} }
```

2.3.3 Punctuation for the xetex engine

For the xetex engine, inter-character rules are defined to insert spaces before the ‘high punctuation’ characters.

```
\or
  \babelcharclass{breton}{thin}{;!?}
  \babelcharclass{breton}{colon}{:}
  \babelinterchar{breton}{default,boundary}{thin}{%
    \ifBR@spacing
       \ifhmode
          \ifdim\lastskip>1sp \unskip \fi
```

```
          \penalty\@M\BRthinspace
        \fi
     \fi}%
  \babelinterchar{breton}{default,boundary}{colon}{%
    \ifBR@spacing
       \ifhmode
          \ifdim\lastskip>1sp \unskip \fi
          \penalty\@M\BRcolonspace
        \fi
     \fi}%

```

This is for French quote characters:

```
  \babelcharclass{breton}{og}{«}
  \babelcharclass{breton}{fg}{»}
  \babelinterchar{breton}{og}{default,boundary}{%
    \ifBR@spacing \penalty\@M\BRguillspace \ignorespaces\fi}
  \babelinterchar{breton}{default,boundary}{fg}{%
    \ifBR@spacing \ifdim\lastskip>\z@ \unskip \fi
       \penalty\@M\BRguillspace\fi}
\fi
```

2.3.4 Punctuation switches common to all engines

Automatic insertion of spaces in front of high punctuation is canceled in for Typewriter fonts and hence in verbatim. For luatex, nothing has to be done as Typewriter fonts are excluded from Babel transforms. For pdftex and xetex, this is done by switching the `\ifBR@spacing` flag. As `\ttfamily` action may be global (when not issued inside a group), `\rmfamily` and `\sffamily` need to be patched too.

```
\ifnum\bbl@engine=\@ne
\else
  \AddToHook{cmd/ttfamily/after}{\BR@spacingfalse}
  \AddToHook{cmd/rmfamily/after}{\BR@spacingtrue}
  \AddToHook{cmd/sffamily/after}{\BR@spacingtrue}
\fi
```

`\BRNoAutoSpacing` The following command disables automatic spacing for high punctuation. It is engine independent and is meant to be used inside a group. The faked definition of `\texorpdfstring` will be overwritten by `hyperref.sty`.

```
\providecommand\texorpdfstring[2]{#1}
\DeclareRobustCommand*{\NoBRspacing}{%
  \texorpdfstring%
    {\ifnum\bbl@engine=1
     \else
        \shorthandoff{;:!?}%
      \fi
```

```
    }%
    {}%
}
```

2.4 Caption names

The next step consists in defining the Breton equivalents for the LaTeX caption names. New implementation for caption names.

```
\StartBabelCommands*{breton}{captions}
      [unicode, fontenc=TU EU1 EU2, charset=utf8]
  \SetString{\refname}{Daveennoù}
  \SetString{\contentsname}{Taolenn ar pennadoù}
  \SetString{\listfigurename}{Listenn ar lunioù}
  \SetString{\listtablename}{Listenn an taolennoù}
  \SetString{\enclname}{Pezh stag}
  \SetStringLoop{ordinal#1}{% change 02BC to 2019
    Kentañ,Eil,Trede,Pevare,Pempvet,C’hwec’hvet,Seizhvet,Eizhvet,%
    Navvet,Dekvet,Unnekvet,Daouzekvet,Trizekvet,Pevarzekvet,Pemzekvet,%
    C’hwezekvet,Seitekvet,Triwec’hvet,Naontekvet,Ugentvet,%
    Unanvet warn-ugent,Eil warn-ugent,Trede warn-ugent,%
    Pevare warn-ugent,Pempvet warn-ugent,C’hwec’hvet warn-ugent,%
    Seizhvet warn-ugent,Eizhvet warn-ugent,Navvet warn-ugen,Tregontvet}
\StartBabelCommands*{breton}{captions}
  \SetString{\prefacename}{Rakskrid}
  \SetString{\refname}{Daveenno\'u}
  \SetString{\abstractname}{Diverradur}
  \SetString{\bibname}{Levrlennadur}
  \SetString{\chaptername}{Pennad}
  \SetString{\appendixname}{Stagadenn}
  \SetString{\contentsname}{Taolenn ar pennado\'u}
  \SetString{\listfigurename}{Listenn ar lunio\'u}
  \SetString{\listtablename}{Listenn an taolenno\'u}
  \SetString{\indexname}{Meneger}
  \SetString{\figurename}{Lun}
  \SetString{\tablename}{Taolenn}
  \SetString{\enclname}{Pez stag}
  \SetString{\ccname}{Eilad da}
  \SetString{\headtoname}{Da}
  \SetString{\pagename}{pajenn}
  \SetString{\seename}{gwelet}
  \SetString{\alsoname}{gwelet ivez}
  \SetString{\proofname}{Anatadur}
  \SetString{\glossaryname}{Geriaoueg}
```

`\part{}` is printed in Breton as “Lodenn gentañ’’ (first), “Eil lodenn’’ (second), “Trede lodenn’’ (third) and so on.

```
  \SetString{\Partnameord}{Lodenn}
  \SetString{\partnameord}{lodenn}
  \SetStringLoop{ordinal#1}{%
    Kenta\~n,Eil,Trede,Pevare,Pempvet,C'hwec'hvet,Seizhvet,Eizhvet,%
    Navvet,Dekvet,Unnekvet,Daouzekvet,Trizekvet,Pevarzekvet,Pemzekvet,%
    C'hwezekvet,Seitekvet,Triwec'hvet,Naontekvet,Ugentvet,%
    Unanvet warn-ugent,Eil warn-ugent,Trede warn-ugent,%
    Pevare warn-ugent,Pempvet warn-ugent,C'hwec'hvet warn-ugent,%
    Seizhvet warn-ugent,Eizhvet warn-ugent,Navvet warn-ugen,Tregontvet}
  \SetString{\partname}{%
     \ifcase\value{part}\Partnameord
     \or \Partnameord\space genta\~n
     \else \csname ordinal\romannumeral\value{part}\endcsname
           \space \partnameord
     \fi
     \def\thepart{\unskip}}
\EndBabelCommands
```

2.5 Date in Breton

\deiziad
\today

```
\StartBabelCommands*{breton}{date}
  \SetStringLoop{month#1name}{%
     Genver, C'hwevrer, Meurzh, Ebrel, Mae, Mezheven,
     Gouere, Eost, Gwengolo, Here, Du, Kerzu}
  \SetString\today{\deiziad{\year}{\month}{\day}}
\EndBabelCommands
\newcommand*{\deiziad}[3]{%
  {\ifnum1=#3{\kentan}\space \else {\number#3} a viz\fi
  \csname month\romannumeral#2name\endcsname
  \ifx#1\@empty\else\space\number#1\fi}}
```

2.6 Extra utilities

Some handy macros for numbering. If n>5, use n\vet.

\kentan
\eil
\re
\trede
\pevare
\vet
\pempvet

```
\def\kentan{1\textsuperscript{añ}}
\def\eil{2\textsuperscript{l}}
\def\re{\textsuperscript{e}}
\def\trede{3\re}
\def\pevare{4\re}
\def\vet{\textsuperscript{vet}}
\def\pempvet{5\vet}
```

Some definitions for special characters.
We won't define \tilde as a Text Symbol not to conflict with the macro \tilde for math mode and will use the breton name \tildenninstead.
Note that \boi may *not* be used in math mode, its name in math mode is \backslash. In Breton backslash is \kilveskell.
\degre can be poorly accessed by the command \r{} for ring accent. \degre is \derez in Breton.
"^" alone (\textasciicircum) is coded \tiredkognek.

```
\AtEndOfPackage{\RequirePackage{scalefnt}}
\if\bbl@engine>\z@
  \providecommand*{\textbackslash}{{\char"005C}}
  \providecommand*{\textasciicircum}{{\char"005E}}
  \providecommand*{\textasciitilde}{{\char"007E}}
  \DeclareRobustCommand*{\derez}{°}
\else
  \DeclareRobustCommand*{\derez}{\textdegree}
\fi
\DeclareRobustCommand*{\boi}{\textbackslash}% keep it for french users
\DeclareRobustCommand*{\kilveskell}{\textbackslash}% OPLB
\DeclareRobustCommand*{\tiredkognek}{\textasciicircum}% OPLB
\DeclareRobustCommand*{\tildenn}{\textasciitilde}% OPLB
```

\Ker
\kerbihan

Stroked K (and k), \Ker (and \kerbihan), give an Unicode glyph if it exists in font, else overlapped K/.

```
\NewDocumentCommand{\Ker}{O{.5}O{.7}}{
  \ifnum\bbl@engine > \z@
    \iffontchar\font "A742 \char"A742 \relax
    \else
      \settowidth{\@tempdima}{K}%
  \makebox[0pt][l]{\hskip#1\@tempdima\raisebox{-.1ex}{\scalefont{#2}/}}K%
    \fi
  \else
    \settowidth{\@tempdima}{K}%
  \makebox[0pt][l]{\hskip#1\@tempdima\raisebox{-.1ex}{\scalefont{#2}/}}K%
  \fi
}%K barrennet
\NewDocumentCommand{\kerbihan}{O{.6}O{.5}}{%
  \ifnum\bbl@engine > \z@
    \iffontchar\font "A743 \char"A743 \relax
    \else
      \settowidth{\@tempdima}{k}%
  \makebox[0pt][l]{\hskip#1\@tempdima\raisebox{-.1ex}{\scalefont{#2}/}}k%%
    \fi
  \else
    \settowidth{\@tempdima}{k}%
  \makebox[0pt][l]{\hskip#1\@tempdima\raisebox{-.1ex}{\scalefont{#2}/}}k%%
```

```
  \fi
}%k barrennet
```

2.7 Clean up and exit

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

```
\ldf@finish{breton}
```