The FixLtxHyph package
A small fix in order to hyphenate emphasized words after a vocalic elision in Catalan, French, Italian and Romansh

Claudio Beccari

v.0.3 2012/02/17

Contents

Abstract

This file fixes a small feature of the hyphenation algorithm used by the TEX system typesetting engines that manifests itself only with those languages that use the apostrophe for marking a vocalic elision. This small package was set up to fix this little undesirable feature in Italian, but it was extended to Catalan, French and the future implementation of the fourth official Swiss language Rumantsch Grischun (Romansh in English).

1 What is the feature to be fixed

The four languages Catalan, French, Italian and Romansh use the apostrophe for marking the vocalic elision of the ending vowel at the end of prepositions, articles, articulated prepositions, definite adjectives, and other words playing similar rôles when they just precede nouns, adjectives, verbs, numerals, that start with a vowel.

This feature is common to most Romance languages (from West to East) from Catalan and Valencian, to French, Langue d'oc, Occitan, Provençal, Vivaroalpin, Italian, Piedmontese, Lombard, Romansh, Ladin, Friulian; up to now only Catalan, French and Italian are handled by the TEX system programs; at the same time most of these languages are minority ones and are being protected by local legislation or are supported by specific cultural or linguistic institutions; only Romansh has got a national/federal legal status in Switzerland and is being used in legal and official documents in the Swiss Confederation, not only in its area of everyday use, the Kanton Graubunden or Canton Grigioni or Chantun Grischun

(where seven Romansh varieties are being spoken, besides Swiss German, Italian, and other languages).

This spelling rule is very rigorous in French; I suppose it is also a rigorous rule in Catalan and Romansh, but I am not that familiar with these languages even if I can understand their written forms. In Italian it used to be a rigorous rule many years ago, but nowadays it is less frequently used when plurals are involved. Nevertheless apostrophes are practically the only para-alphabetic sign you see in an Italian text except for letters and punctuation marks.

In order to hyphenate correctly these word combinations all four languages have to declare the apostrophe, that has a category code of 12, as a glyph with non zero lower case code. In facts all four languages declare:

```
\lccode‘\’=‘\’
```

or something equivalent. With this little trick, the typesetting engine considers the apostrophe as a valid word character and treats the whole string as a single word; the patterns of these languages, of course, take into consideration also the apostrophe so that the resulting correct line break points are easily found[1]:

Catalan	`d’aquesta`	d’a-ques-ta
French	`l’électricité`	l’élec-tri-ci-té
Italian	`dell’eleganza`	del-l’e-le-gan-za
Romansh	`l’identitad`	l’i-den-ti-tad

So where is the problem? It emerges when the second part of the string is emphasized, because in this case no hyphenation takes place:

Catalan	`d’\emph{aquesta}`	d’*aquesta*
French	`l’\emph{électricité}`	l’*électricité*
Italian	`dell’\emph{eleganza}`	dell’*eleganza*
Romansh	`l’\emph{identitad}`	l’*identitad*

This behavior is easily explained, so that it is not to be considered a bug, but a feature; a feature that is annoying only when using the above four named languages. The point is that all TEX system typesetting engines consider a word to be that character string starting after a character invalid in a word and finishing

[1]For typesetting Romansh and hyphenating it, initially I wrote myself a short language definition file, where, among other things, Romansh was considered as a dialect of Italian. In this way I did not need to build up the specific Romansh patterns; of course this was done only for thesting this small package. With my great surprise, my first feeling was that Italian patterns did a pretty good job (the common Latin ancestor helps a lot, of course); I spotted only very few final syllables without a vowel (which is wrong) and a couple of digraphs wrongly divided. After this “successful” start, I made up the Romansh hyphenation patters by just modifying the Italian ones in order to correct the few errors mentioned above; I checked my Romansh patterns aginst the hyphenation rules described in the *Grammatica per l’instrucziun dal rumantsch grischun* published by the Department of romance languages and literature of the Friburg University, Switzerland. The patterns I produced work pretty well, but they are not built up for etimological hyphenation, therefore some prefixex or compound words might be divided without respecting the compound word barrier. At the date of writing these notes, the Romansh patterns and the language description file have been uploaded to the polyglossia and babel teams, but they are not yet available for the general user.

with the first token invalid in a word. Notice that when the hyphenating algorithm comes to work the command `\emph` has already been expanded and it ends up with the qualifications of the selected font; therefore a string such as ␣`d'aquesta`␣ (starting after a space and ending before the following space) is made up of valid characters; but ␣`d'\emph{aquesta}`␣ is a "word" starting after a space and ending before a space, but containing a font change. And this makes the word invalid for hyphenation. The TEXbook is clear on this respect: "If a suitable letter is found [as a starting character], let it be in font f. ... TEX continues to scan forward until coming to something that's not one of the following three "admissible items": (1) a character in font f whose `\lccode` is not zero; (2) a ligature formed entirely from characters of type (1); (3) an implicit kern. ... Notice that all these letters are in font f."

This was a specific programming choice decided by Donald Knuth together with Frank Liang, his PhD student who developed the hyphenation algorithm implemented in the typesetting engines of the TEX system[2]. As all such decisions, it is a compromise between accuracy and speed. And remember that at the beginning `tex` the program was used essentially with English, a language that does not use accented letters and uses elision in a much different way as the one we are speaking here. The problem did non exist and, I suppose, it will never exist in English.

2 The solutions

As a compromise I decided to solve the problem in an automatic way only when the second part of the "word" to be hyphenated is emphasized. I suppose it is the most frequent situation, although no one can avoid thinking to other situations; for example; the second part of such "word" after the apostrophe is bolded, is colored, is written in another font selected on purpose or is in another alphabet, it is in italics, with no choice on the inclination; in such cases the solution is manual and remains manual, because there are too many possibilities and it is cumbersome to deal with all of them.

But manual or automatic, how should we proceed? Simply we must convince that the starting letter must not be the start of the part preceding the apostrophe, but what follows it. This is simple: it suffices to put after the apostrophe a unbreakable, zero width glob of glue; TEX starts looking for a potential starting letter after the glue. Therefore the manual solution consists in defining a short macro such as the following one:

```
\newcommand\hz{\nobreak\hskip\z@skip}
```

or, if you want to avoid setting this short command into a personal `.sty` file, simple change `\z@skip` with `0pt`. You will then have to modify the font changing phrase into something such as:

```
... d'\hz\textbf{aquesta} ...
```

[2] I have been told that LuaTEX is developing a different algorithm that eliminates this feature.

The `\hz`, whose name reminds the sentence "Horizontal skip of an unbreakable Zero width glob of glue", finishes the preceding word and sets the grounds for starting the search of a new starting letter of another word; it will be found after the font selections code introduced in the horizontal list by the selected font identification.

The automatic solution, on the opposite, implies a small but substantial modification of the `\emph` command. In facts the text command uses the text declaration `\em`; on turn `\em` is a robusta commanda, that is it is defined as `\protext\em␣`: it would be very unwise to modify a protected command, so it is necessary to modifiy the "protected" one, and this operation is not trivial because of the space in the name of this macro. In any case if we find out how, we must add `\hz` to the definition of `\em␣` before its argument, the real text to emphasize, is processed.

This small package does exactly this, only for the four named languages, and only if they are used, and only with the `\emph` command. The `\hz` command is available to the user in a global way, so that when this package is loaded, the manual solution remains valid for every language, although in very unlikely situations.

It has been tested with the four languages, and apparently it works as expected; it has been throughly tested in all situations with Italian; it should work properly also in French and in Romansh. The adopted solution does not fiddle with active characters and therefore it does not interfere with the internal workings and settings of Catalan and the other languages.

3 Installation

With modern TEX distributions these instructions are superfluous; should you need to install by hand, download from CTAN in a scratch directory (possibly create one, and after finishing, delete the whole directory with its contents) run this file `fixltxhyph.dtx` through `pdflatex`; you get two files and move them as follows:

- Move all the files in the following directories on your disk; if you don't already have those directories, create them.
- These directories should be created in your personal `texmf` tree; if you don't have one, create it; how to do this and where to root it depends on your operating system; before doing any change to your hard disk, please read carefully the TeX Live or the MiKTeX documentations in order to find out what a personal tree is.
- Move `fixltxhyph.dtx` to `.../texmf/source/latex/FixLtxHyph/`;
- Move `fixltxhyph.pdf` to `.../texmf/doc/latex/FixLtxHyph/`;
- Move `fixltxhyph.sty` to `.../texmf/tex/latex/FixLtxHyph/`;
- if your distribution requires it, refresh the file name database.

You are now ready to use the package by simply invoking it in the preamble of your documents:

```
\usepackage{fixltxhyph}
```

sectionAknowledgements I wish to thank Lorenzo Pantieri who tested the preliminary and the actual versions of this package and directly or indirectly helped debugging the code, especially in the preliminary version that used active characters and was particularly buggy. Another big thank to Enrico Gregorio who spotted the protection problem of the `\em` command.

4 The documented code

We start by identifying the package and the necessary format file:

```
⟨*style⟩
\ProvidesPackage{fixltxhyph}[2011/12/25 v.0.1 Small fix for hyphenating
emphasized words preceded by vocalic elision]
\NeedsTeXFormat{LaTeX2e}[2011/06/27]
```

Then we make sure that the package `babel` has already been loaded; otherwise we warn the user and exit; no patches can be made to an unknown package.

```
\@ifpackageloaded{babel}{}{%
\PackageWarning{FixLtXHyph}{This package must be loaded after babel}%
\endinput}
```

We need the package `etoolbox` in order to perform any action on control sequences that contain spaces in their names.

```
\RequirePackage{etoolbox}
```

We define a very short command `\hz` in order to have available a handy command for inserting an unbreakable zero width glob of glue in case we needed to do some sort of patching by hand.

```
\newcommand\hz{\nobreak\hskip\z@skip}
```

We make patches only if one or more of the four languages Catalan, French, Italian, or Romansh has been invoked as an option to `babel`; if none of these options had been selected, evidently the user had his/her thoughts focused on other details, so that s/he missed the point that this patch is necessary only for the above mentioned four languages. In any case no harm should happen if from now on nothing gets done, except for the definition of `hz` that remains available to the user.

The next bit of code defines some aliases in order to keep the original meaning of the declaration `\em`; in order to patch an alias, so as to be able to set the proper definitions only for the named four languages and to rest the original situation when a change of language takes place.

```
\letcs{\FLH@originalem}{em }
\let\FLH@newem\FLH@originalem
\preto\FLH@newem{\hz}
```

We then use a repetition cycle based on a list of language names; if the language with one of the listed names has been invoked as an option to `babel`, then the patched `\em␣` definition is made the default, while when changing language the original definition is restored:

```
\def\@tempB{catalan,french,italian,romansh}
\@for\@tempA:=\@tempB\do{%
\expandafter\ifx\csname captions\@tempA\endcsname\relax\else
\expandafter\addto\csname extras\@tempA\endcsname{\cslet{em }{\FLH@newem}}%
\expandafter\addto\csname noextras\@tempA\endcsname{\cslet{em }{\FLH@originalem}}\fi}
```

Eventually the documented file is terminated and its final commands are issued.

```
\endinput
⟨/style⟩
```