dhua.sty

—

German Abbreviations Using Thin Space*

Uwe Lück†

September 19, 2011

Abstract/Zusammenfassung

dhua.sty provides commands for German phrase abbreviations such as 'd. h.' that are recommended to use a thin space—set-up commands `\newdhua` and `\newtwopartdhua` as well as commands for single cases (e.g., `\zB` for 'z. B.', to save you from typing `z.\,B.`). Package options are intended to support generating PDF and HTML from the same source, maybe automatically using `\xspace`.

Das Paket dhua bietet Befehle für sog. mehrgliedrige Abkürzungen, für die schmale Leerzeichen (Festabstände) empfohlen werden. In die englische Paketdokumentation sind deutsche Hinweise (kursiv) eingestreut.

Keywords: German typography; web typography, language support, macro programming

Contents

*This document describes version v0.11 of dhua.sty as of 2011/09/19.

†http://contact-ednotes.sty.de.vu

1 Installing, Calling, Usage

The file dhua.sty is provided ready, installation only requires putting it somewhere where TEX finds it (which may need updating the filename data base).[1]

Below the '`\documentclass`' line(s) and above '`\begin{document}`', you load dhua.sty (as usually) by

```
\usepackage{dhua}
```

or by

```
\usepackage[⟨option(s)⟩]{dhua}
```

with the option(s) ⟨*option(s)*⟩ described in Section 3 ('`[web]`', '`[xspace]`'). A few macros for single abbreviations are described in Section 5, the macros '`\newdhua`' and '`\newtwopartdhua`' for defining such abbreviation macros are described in Section 4.

Unten werden (i) Paketoptionen `[web]` *und* `[xspace]`*, (ii) die Makros* `\newdhua` *und* `\newtwopartdhua` *für die Definition einzelner Abkürzungsmakros sowie (iii) einzelne vordefinierte Abkürzungsmakros beschrieben.*

2 Package File Header (Legalize)

```
\ProvidesPackage{dhua}[2011/09/19 v0.11 German phrase abbrevs (UL)]
%% copyright (C) 2011 Uwe Lueck,
%% http://www.contact-ednotes.sty.de.vu
%% -- author-maintained in the sense of LPPL below.
%%
%% This file can be redistributed and/or modified under
%% the terms of the LaTeX Project Public License; either
%% version 1.3c of the License, or any later version.
%% The latest version of this license is in
%%    http://www.latex-project.org/lppl.txt
%% We did our best to help you, but there is NO WARRANTY.
%%
%% Please report bugs, problems, and suggestions via
%%
%%   http://www.contact-ednotes.sty.de.vu
%%
```

[1] http://www.tex.ac.uk/cgi-bin/texfaq2html?label=inst-wlcf

3 Package Options

3.1 Idea

You may say "I can

```
\newcommand{\ua}{u.\,a.\xspace}
```

myself." Yes, you can. The point of

```
\newtwopartdhua{\ua}{u}{a}
```

is that you can use that same defining instruction for all of the following *types of documents:* (i) *requiring* `\xspace` because in the source some '`\ua`' precedes a word without control space ('`\␣`', as in '`\ua␣weil`'); (ii) *not* using '`\xspace`' (I use it with blog.sty where '`\xspace`' does not work); (iii) to be typeset for high-quality *printing* (thin space strongly recommended) (iv) to be displayed as HTML (web typography relevant, thin space somewhat dangerous). In order to use (almost) the *same source* for (iii) and (iv), the LaTeX document preamble just must load the `[web]` option for (iv) where it doesn't for (iii).

3.2 'web'

Paketoption `[web]` *verwendet ein geschütztes Leerzeichen normaler Breite anstelle des empfohlenen schmalen Festabstands.*

In web typography, '` `' (no-break space) may be used instead of a thin space because some browsers do not support the latter (the non-breakable one—which is not a *named* HTML entity, it is U+202F—and some even the breakable one—which is the named HTML entity ` `). You can decide for this choice by package option `[web]`. (Actually I use the package for direct HTML generation with blog.sty.)

3.3 'xspace'

With option `[xspace]`, the package's setup commands equip all the single abbreviation macros with a final '`\xspace`' from the xspace package in the LaTeX tools bundle.

Mit der Paketoption `[xspace]` *verwenden alle Abkürzungsmakros automatisch (– sie enden auf –)* `\xspace`*.*

3.4 Implementations

`\dhuaspace` stores the dot plus the inner space (TODO could be useful for Euro symbol without dot). The default setting is:

```
17  \newcommand*\dhuaspace{.\,}
```

`\DhuaSpace` is for getting a single token in `\edef`s:

```
\newcommand*{\DhuaSpace}{\noexpand\dhuaspace}
```

Option `[web]` uses the tilde (tie, '~') instead of '`\,`':

```
\DeclareOption{web}{\newcommand*\dhuaspace{.~}}
```

I reasoned as follows for this: (i) *Either* the HTML is generated from DVI, this is what TeX4ht does. I don't know exactly, but I assume that the combinatin of TeX4ht with a TEX run finally converts the tilde into '` `'. (ii) *Or* the HTML generator translates the TEX code into HTML in a more direct way. I don't know what all of these programs actually do, but they "should" translate '`~`' into '` `'. At least blog.sty does, for sure.

`\dhuaxspace` stores what closes the entire abbreviation definition; by default it is a dot only. Because German text should be typeset using '`\frenchspacing`', we do not care about the space factor:

```
\newcommand*{\dhuaxspace}{.}
\@ifdefinable{\DhuaXspace}{\let\DhuaXspace\dhuaxspace}
\DeclareOption{xspace}{%
    \AtEndOfPackage{\RequirePackage{xspace}}%%          %% 2011/09/09
    \renewcommand*{\dhuaxspace}{.\xspace}%
    \renewcommand*{\DhuaXspace}{\noexpand\dhuaxspace}}
```

—That were all options, processed now:

```
\ProcessOptions
```

4 Setup Commands

The syntax `\newdhua{`⟨*new-macro*⟩`}{`⟨*replace*⟩`}` is the same as for LATEX's standard '`\newcommand`' etc. defining user macros *without parameters*. However, '`\newdhua`' internally uses '`\edef`' in order to minimize the number of tokens in the actual internal replacement text. This additionally requires using '`\DhuaSpace`' for separating the one-word abbreviations in ⟨*replace*⟩.

```
\newcommand*{\newdhua}[2]{\@ifdefinable#1{%
%     \protected@edef#1{#2\DhuaXspace}%
    \let\protect\noexpand                          %% 2011/09/04
```

... an unusual meaning of `\protect`, CARE! I.e., '`\protect`' is used here to prevent expansion in the setup macros. It is not stored for future expansion with its usual robustification purpose.

```
    \edef#1{#2\DhuaXspace}%
    \let\protect\@typeset@protect
}}
```

Um ein Makro ⟨*neu*⟩ *für eine Abkürzung mit beliebig vielen Gliedern zu definieren, trennt man die einzelnen Glieder im* ⟨*Ergebnis*⟩*-Argument der Definition* `\newdhua{`⟨*neu*⟩`}{`⟨*Ergebnis*⟩`}` *durch* `\DhuaSpace` *(Beispiel unten).*

`\newtwopartdhua{`⟨*neu*⟩`}{`⟨*B.-1*⟩`}{`⟨*B.-2*⟩`}`

definiert dasselbe Makro ⟨neu⟩ wie

`\newdhua{`⟨*neu*⟩`}{`⟨*B.-1*⟩`\DhuaSpace`⟨*B.-2*⟩`}`

`\newtwopartdhua{`⟨*new*⟩`}{`⟨*letter-1*⟩`}{`⟨*letter-2*⟩`}` results in the same macro ⟨*new*⟩ as '`\newdhua{`⟨*new*⟩`}{`⟨*letter-1*⟩`\DhuaSpace`⟨*letter-2*⟩`}`':

```
33  \newcommand*{\newtwopartdhua}[3]{\newdhua#1{#2\DhuaSpace#3}}
```

5 Single Abbreviation Macros

The user may want to use (some of) the following single macro names for a different purpose or so; to this end, a file '`dhua.cfg`' may contain a different set of definitions. '`\IfFileExists`' works depending on whether a file '`dhua.cfg`' is found:

```
34  \IfFileExists{dhua.cfg}{%
35    \typeout{^^J * single definitions read from
36                \string`dhua.cfg\string'          %% 2011/09/19
37                *^^J}%
38    \input{dhua.cfg}}{%
```

`\idR` exemplifies multi-part abbreviations, where 'multi' means "more than two":

```
39    \newdhua{\idR}{i\DhuaSpace d\DhuaSpace R}
40  %   \show\idR
```

`\idR` *erzeugt „i. d. R." – ein Anwendungsfall für* `\newdhua`*. Nachfolgend wird nur noch* `\newtwopartdhua` *verwendet.*

LaTeX actually defines `\dh` as something nordic (one of my earliest macro making experiences) so we are *re*defining it:

```
41    \PackageWarning{dhua}{Redefining \string\dh}
42    \let\dh\relax
43    \newtwopartdhua{\dh}{d}{h}
44  %   \show\dh
```

`\dh` *wird hier* umdefiniert*, um „d. h." zu bekommen. Die übrigen Makros sind „normale" Anwendungsfälle von* `\newtwopartdhua`*, man achte aber noch auf die Verwendung von* `\protect`*.*

"Normal" cases (`\oae` for 'o. ä.', `\so` for 's. o.', `\su` for 's. u.', `\uae` for 'u. ä.', `\ua` for 'u. a.', `\vglu` for 'vgl. u.', `\vglo` for 'vgl. o.', `\zB` for 'z. B.', `\zT` for 'z. T.'):

```
45    \newtwopartdhua{\oae}{o}{\protect\"a}
46  % \newtwopartdhua{\oae{o}{\"a}
47  %   \show\oae
```

—exemplifying use of `\protect` so the definition of '`\oae`' has a single token '`\"`', not an expansion of '`\"`'.

```
48    \newtwopartdhua{\so}  {s}{o}
49    \newtwopartdhua{\su}  {s}{u}
50    \newtwopartdhua{\ua}  {u}{a}
51    \newtwopartdhua{\uae} {u}{\protect\"a}
52    \newtwopartdhua{\vglu}{vgl}{u}
53    \newtwopartdhua{\vglo}{vgl}{o}
54    \newtwopartdhua{\zB}  {z}{B}
55    \newtwopartdhua{\zT}  {z}{T}
56  }                                     %% Closes \IfFileExists
```

6 A Different Approach

Statt für „A. B.“ ein Makro zu definieren, kann man auch „`\abkii␣AB`*“ tippen.*

I also thought that, instead of defining an abbreviation macro (perhaps '`\`⟨*letter-1*⟩⟨*letter-2*⟩'), `\abkii␣`⟨*letter-1*⟩⟨*letter-2*⟩ could be preferred to typing the the two dots and '`\,`':

```
57  \@ifdefinable\abkii{%               %% w/o 'protected' 2011/09/09:
58      \edef\abkii#1#2{#1\DhuaSpace#2\DhuaXspace}}
59  %   \show\abkii
```

So '`\abkii␣AB`' results in 'A. B.', saving you from '`A.\,B.`'.—You may create your own shorter alias ⟨*shal*⟩ for '`\abkii`' by '`\let`⟨*shal*⟩`\abkii`'. Perhaps '`\II`': '`\let\II\abkii`'—'`\II␣AB`'—'A. B.'. Is this better than '`A.\,B.`'?

7 The End

```
60  \endinput
```

8 VERSION HISTORY

```
61  v0.1    2011/09/13  renamed 'dhusw'->'dhua'
62  v0.1a   2011/09/14  doc. fix
63          2011/09/16  doc. of options much extended
64  v0.11   2011/09/19  don't use \qtd with .cfg-\typeout; \so, \su;
65                      doc. fix \zB
66
```

9 Colophon

The English part of the documentation exemplifies a new (2011/09/09) function of niceverb.sty v0.44: automatically enclose inline TEX code in single quotation marks after '`\AddQuotes`'. I needed especially much time for this because group nesting spans several documentation pages.

I spent much time with a special environment '`{german}`' for the present purpose: the indent of the following paragraph was missing—until I added an

empty documentation line. (Same with standard '`{sloppypar}`' environment, I don't understand it, tried '`\@endpefalse`' in vain.) I don't like babel ...

The German parts use niceverb's '`\DontAddQuotes`' because of a different frequency of TEX code. Even in the English parts I considered the single quotation marks bad and avoided them using LATEX's '`\verb`'.

And my terms 'phrase abbreviation' and 'abbreviation macro' may be bad, please help me ...