

dhua.sty

German Abbreviations Using Thin Space*

Uwe Lück[†]

September 16, 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

1	Installing, Calling, Usage	2
2	Package File Header (Legalize)	2
3	Package Options	3
3.1	Idea	3
3.2	‘web’	3
3.3	‘xspace’	3
3.4	Implementations	3
4	Setup Commands	4
5	Single Abbreviation Macros	5

*This document describes version [v0.1a](#) of `dhua.sty` as of 2011/09/16.

[†]<http://contact-ednotes.sty.de.vu>

1	INSTALLING, CALLING, USAGE	2
6	A Different Approach	6
7	The End	6
8	VERSION HISTORY	6
9	Colophon	6

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)

```

1 \ProvidesPackage{dhua}[2011/09/16 v0.1a German phrase abbrevs (UL)]
2 %% copyright (C) 2011 Uwe Lueck,
3 %% http://www.contact-ednotes.sty.de.vu
4 %% -- author-maintained in the sense of LPPL below.
5 %%
6 %% This file can be redistributed and/or modified under
7 %% the terms of the LaTeX Project Public License; either
8 %% version 1.3c of the License, or any later version.
9 %% The latest version of this license is in
10 %% http://www.latex-project.org/lppl.txt
11 %% We did our best to help you, but there is NO WARRANTY.
12 %%
13 %% Please report bugs, problems, and suggestions via
14 %%
15 %% http://www.contact-ednotes.sty.de.vu
16 %%

```

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 L^AT_EX 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 L^AT_EX 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 `\edefs`:

```
18 \newcommand*{\DhuaSpace}{\noexpand\dhuaSpace}
```

Option `\web` uses the tilde (tie, ‘~’) instead of ‘\,’:

```
19 \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.

`\dhuaSpace` 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:

```
20 \newcommand*{\dhuaSpace}{.}
21 \@ifdefinable{\DhuaXspace}{\let\DhuaXspace\dhuaSpace}
22 \DeclareOption{xspace}{%
23   \AtEndOfPackage{\RequirePackage{xspace}}}% %% 2011/09/09
24   \renewcommand*{\dhuaSpace}{.\xspace}%
25   \renewcommand*{\DhuaXspace}{\noexpand\dhuaSpace}}
```

—That were all options, processed now:

```
26 \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>`.

```
27 \newcommand*{\newdhua}[2]{\@ifdefinable#1{%
28   % \protected@edef#1{#2\DhuaXspace}%
29   \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.

```
30   \edef#1{#2\DhuaXspace}%
31   \let\protect\@typeset@protect
32 }}
```

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.-I>}{<B.-2>}
```

definiert dasselbe Makro $\langle neu \rangle$ wie

$$\backslash newdhua\{\langle neu \rangle\}\{\langle B.-1 \rangle\}\backslash DhuaSpace\langle B.-2 \rangle\}$$

$\backslash newtwopartdhua\{\langle new \rangle\}\{\langle letter-1 \rangle\}\{\langle letter-2 \rangle\}$ results in the same macro $\langle new \rangle$ as $\backslash newdhua\{\langle new \rangle\}\{\langle letter-1 \rangle\}\backslash DhuaSpace\langle letter-2 \rangle\}$:

```
33 \newcommand*\backslashnewtwopartdhua}[3]{\backslashnewdhua#1{\backslash#2\backslashDhuaSpace#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 \qtd{dhua.cfg} *^^J}%
36   \input{dhua.cfg}}{%
```

$\backslash idR$ exemplifies multi-part abbreviations, where ‘multi’ means “more than two”:

```
37 \newdhua{\idR}{\i\backslashDhuaSpace d\backslashDhuaSpace R}
38 % \show\idR
```

$\backslash idR$ erzeugt „i. d. R.“ – ein Anwendungsfall für $\backslash newdhua$. Nachfolgend wird nur noch $\backslash newtwopartdhua$ verwendet.

L^AT_EX actually defines $\backslash dh$ as something nordic (one of my earliest macro making experiences) so we are redefining it:

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

$\backslash dh$ wird hier undefiniert, um „d. h.“ zu bekommen. Die übrigen Makros sind „normale“ Anwendungsfälle von $\backslash newtwopartdhua$, man achte aber noch auf die Verwendung von $\backslash protect$.

“Normal” cases ($\backslash oae$ for ‘o. ä.’, $\backslash uae$ for ‘u. ä.’, $\backslash ua$ for ‘u. a.’, $\backslash vglu$ for ‘vgl. u.’, $\backslash vgl o$ for ‘vgl. o.’, $\backslash zB$ for ‘z. B.’, $\backslash zT$ for ‘z. T.’):

```
43 \newtwopartdhua{\oae}{o}{\protect\"a}
44 % \newtwopartdhua{\oae}{o}{\"a}
45 % \show\oae
```

—exemplifying use of $\backslash protect$ so the definition of ‘\oae’ has a single token ‘\’, not an expansion of ‘\’.

```

46 \newtwopartdhua{\ua} {u}{a}
47 \newtwopartdhua{\uae} {u}{\protect"a}
48 \newtwopartdhua{\vglu}{vgl}{u}
49 \newtwopartdhua{\vglo}{vgl}{o}
50 \newtwopartdhua{\zB} {z}{B}
51 \newtwopartdhua{\zT} {z}{T}
52 } %% 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>’), $\boxed{\backslash\text{abkii}_{\text{letter-1}}\text{letter-2}}$ could be preferred to typing the the two dots and ‘\,’:

```

53 \@ifdefinable\abkii{% %% w/o ‘protected’ 2011/09/09:
54 \edef\abkii#1#2{#1\ DhuaSpace#2\ DhuaXspace}}
55 % \show\abkii

```

So ‘\abkii_AB’ results in ‘A. B.’, saving you from ‘A.\,B.’.—You may create your own shorter alias $\langle\text{shal}\rangle$ for ‘\abkii’ by ‘\let\shal\abkii’. Perhaps ‘\II’: ‘\let\II\abkii’—‘\II_AB’—‘A. B.’. Is this better than ‘A.\,B.’?

7 The End

```

56 \endinput

```

8 VERSION HISTORY

```

57 v0.1 2011/09/13 renamed ‘dhusw’->‘dhua’
58 v0.1a 2011/09/14 doc. fix
59 2011/09/16 doc. of options much extended
60

```

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 ...