

The fnumprint package*

Robin Schneider

ypid23@aol.de

August 27, 2012

Abstract

The fnumprint package can decide to typeset a number either as number or as word name (only in English or German).

Fork me on GitHub: <https://github.com/ypid/latex-packages/tree/master/fnumprint>

Contents

Abstract	1
1 Introduction	1
2 Usage	2
3 Examples	2
4 ToDo	2
5 Implementation	2
5.1 Language checking	3
5.2 Macro definition	3

1 Introduction

The fnumprint package defines two macros which decide to typeset a number either as Arabic number or as word name for the number. If the number is between zero and twelve (including zero and twelve) then the word name will be used. This package uses the `zahl2string` package to convert a number in the word name in German. The conversion of a number (0–12) to the English word name is implemented by this package. If the number is outside of the range then it will be typesetted with the `numprint` package.

*This document corresponds to fnumprint v1.1, dated 2012/08/27.

2 Usage

Just load the package placing

```
\usepackage{fnumprint}
```

in the preamble of your L^AT_EX 2_ε source file. In this case the last by `fnumprint` supported language given as class option will be used. You can overwrite this with a package option like this:

```
\usepackage[english]{fnumprint}
```

\fnumprintc The macro `\fnumprintc` [*eins*] {*L^AT_EX counter name*} takes a name of a LaTeX counter as not optional parameter and typesets it's value. The optional parameter is only active if the German language was selected. In this case it can be used to typeset “ein” instead of “eins”. This is for example necessary if you would like to typeset “ein Jahr”.

\fnumprint The macro `\fnumprint` [*eins*] {*number*} is like the `\fnumprintc` macro but it takes a number or a macro that expands to a number. A T_EX counter can also be used with this macro.

3 Examples

macro	expanded macro
<code>\fnumprint{-1}</code>	−1
<code>\fnumprint{0}</code>	zero
<code>\fnumprint{1}</code>	one
<code>\fnumprint{10}</code>	ten
<code>\fnumprint{12}</code>	twelve
<code>\fnumprint{13}</code>	13
<code>\fnumprint{\the\month}</code>	eight
<code>\fnumprint{\the\day}</code>	27
<code>\fnumprintc{page}</code>	two

4 ToDo

Here are some things that could be useful.

- Provide more features implemented by the `zahl2string` package and create macros which implement these features for the English language.

5 Implementation

This package depends on these packages.

```
1 \RequirePackage{xifthen}
2 \RequirePackage{numprint}
```

5.1 Language checking

I used a counter `fnumprint@language` to save the (last) selected language. Meaning from the counter values:

value	meaning
0	no supported language selected
1	German selected
2	English selected

```

3 \newcounter{fnumprint@language}
4 \setcounter{fnumprint@language}{0}
5 \DeclareOption{german}{\setcounter{fnumprint@language}{1}}
6 \DeclareOption{ngerman}{\setcounter{fnumprint@language}{1}}
7 \DeclareOption{english}{\setcounter{fnumprint@language}{2}}
8 \DeclareOption{USenglish}{\setcounter{fnumprint@language}{2}}
9 \DeclareOption{american}{\setcounter{fnumprint@language}{2}}
10 \DeclareOption{UKenglish}{\setcounter{fnumprint@language}{2}}
11 \DeclareOption{british}{\setcounter{fnumprint@language}{2}}
12 \DeclareOption{canadian}{\setcounter{fnumprint@language}{2}}
13 \DeclareOption{australian}{\setcounter{fnumprint@language}{2}}
14 \DeclareOption{newzealand}{\setcounter{fnumprint@language}{2}}
15 \ProcessOptions\relax

```

If none of the supported languages was selected a package warning will appear.

```

16 \ifthenelse{\value{fnumprint@language} = 0}{
17   \PackageWarning{\@currname}{No supported language selected}
18   \MessageBreak
19   This package supports only English and German
20   \MessageBreak
21   There will be no word names printed}
22 }{

```

The `zahl2string` package will only be loaded if it is necessary.

```

23 \ifthenelse{\value{fnumprint@language} = 1}{\RequirePackage{zahl2string}}{}

```

5.2 Macro definition

`\ns@en@neunzehn` This internal marco expands to the English word name for a number. It only goes from 0 to 19. It will only be defined if it is necessary.

```

24 \ifthenelse{\value{fnumprint@language} = 2}{
25   \newcommand{\ns@en@neunzehn}[1]{%
26     \ifcase#1 zero\or one\or two\or three\or four\or five\or six\or%
27     seven\or eight\or nine\or ten\or eleven\or twelve\or thirteen\or%
28     fourteen\or fifteen\or sixteen\or seventeen\or%
29     eighteen\or nineteen\fi%
30   }
31 }{}
32 }

```

`\fnumprint` Here is the `\fnumprint` macro definition. It takes one not optional parameter. The parameter must be a number or a macro which expands to a number. It can

also take a optional parameter which replaces the German word “eins” with “ein”.

```

33 \DeclareRobustCommand{\fnumprint}[2][]{%
34   \ifthenelse{-1 < #2 \AND #2 < 13}{%
35     \ifthenelse{\value{fnumprint@language} = 1}{%
36       \ifthenelse{\equal{#1}{ein}}{\ns@neunzehn{#2}}{\numstr{#2}}%
37     }{%
38       \ifthenelse{\value{fnumprint@language} = 2}{%
39         \ns@en@neunzehn{#2}%
40       }{\numprint{#2}}%
41     }%
42   }\numprint{#2}}%
43 }

```

`\fnumprintc` Here is the `\fnumprintc` macro definition. It takes one not optional parameter. The parameter must be the name of a L^AT_EX counter. It can also take a optional parameter which replaces the German word “eins” with “ein”.

```

44 \DeclareRobustCommand{\fnumprintc}[2][]{%
45   \ifthenelse{-1 < \value{#2} \AND \value{#2} < 13}{%
46     \ifthenelse{\value{fnumprint@language} = 1}{%
47       \ifthenelse{\equal{#1}{ein}}{\ns@neunzehn{\value{#2}}}{\numstring{#2}}}%
48     \ifthenelse{\value{fnumprint@language} = 2}{%
49       \ns@en@neunzehn{\value{#2}}%
50     }{\cntprint{#2}}%
51   }%
52   }\cntprint{#2}}%
53 }
54 \endinput

```

Change History

v0.1		v1.0a
General: Initial version	1	General: Optimized 1
v1.0		v1.0b
General: Wrote this documenta-		General: Optimized 1
tion and the L ^A T _E X-package and		v1.1
added support for English	1	General: Added optional parameter
		“ein” 1

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

C

`\cntprint` 50, 52

F	N	
		\numprint 40, 42
\fnumprint 2, <u>33</u>	\ns@en@neunzehn ...	\numstr 36
 <u>24</u> , 39, 49	
\fnumprintc 2, <u>44</u>	\ns@neunzehn 36, 47	\numstring 47