

The `numspell` package

v1.4 (2020/11/11)

Tibor Tómacs

tomacs.tibor@uni-eszterhazy.hu

1 Introduction

The aim of the `numspell` package is to spell the cardinal and ordinal numbers from 0 to $10^{67} - 1$ (i.e. maximum 66 digits).

The supported languages are English (British and American), French, German, Hungarian and Italian. The spelling will happen in the current language.

The `numspell` package requires the services of the `xstring` and `iflang` packages.

Load the package as usual, with

```
\usepackage{numspell}
```

2 Commands

```
\numspell[zeros]{num}
```

Spelling the cardinal number $n = \langle num \rangle \cdot 10^{\langle zeros \rangle}$, where $0 \leq n \leq 10^{67} - 1$. The default value of *zeros* is 0. For example

```
\numspell{12000} → twelve thousand  
\numspell[3]{12} → twelve thousand  
\numspell[6]{12} → twelve million  
\numspell[63]{1} → one vigintillion
```

```
\thenumspell
```

The `\numspell` stores the result in this command. For example

```
\numspell{12000}; \thenumspell → twelve thousand; twelve thousand  
\numspell{1}; \numspell{2}; \thenumspell → one; two; two
```

```
\numspellsave{name}
```

It generates the `\thenumspell<name>` command, which saves the current `\thenumspell`. For example

```

\numspell{1};
\numspellsave{MyNum}
\numspell{2};
\thenumspell;
\thenumspellMyNum

```

one; two; two; one

`\numspelledashspace{<length>}`

In the number spelling, the spaces around the dashes are flexibility for the optimal hyphenation. Its value is 0pt plus *<length>*. The default value of *<length>* is 2pt. For example

```

\selectlanguage{magyar}
\numspell{6512312354762547162546254756}\[2mm]
\numspelledashspace{10pt}
\numspell{6512312354762547162546254756}

```

hatkvadrilliárd - ötszázötvenkétkvadrillió - háromszázötvenkétrilliárd - háromszázötvennégytrillió - hétszázhatvankétbilliárd - ötszáznegyvenhétbillió - egyszázhatvankétmilliárd - ötszáznegyvenhatmillió - kétszázötvennégyezer - hétszázötvenhat

hatkvadrilliárd - ötszázötvenkétkvadrillió - háromszázötvenkétrilliárd - háromszázötvennégytrillió - hétszázhatvankétbilliárd - ötszáznegyvenhétbillió - egyszázhatvankétmilliárd - ötszáznegyvenhatmillió - kétszázötvennégyezer - hétszázötvenhat

`\numspell*{<zeros>}{<num>}`

It works like `\numspell`, but the number spelling will not be printed. In other words, the following two lines are equivalent:

```

\numspell[<zeros>]{<num>}
\numspell*{<zeros>}{<num>}\thenumspell

```

For example

```

\numspell*{1}
\numspellsave{MyNum}
\numspell*{2}
\thenumspell;
\thenumspellMyNum

```

two; one

`\Numspell[<zeros>]{<num>}`

It works like `\numspell`, but the first letter will be capital. For example

```

\Numspell{12000} → Twelve thousand
\Numspell[3]{12} → Twelve thousand
\Numspell[6]{12} → Twelve million
\Numspell[63]{1} → One vigintillion

```

`\Numspell*{<zeros>}{<num>}`

It works like `\Numspell`, but the number spelling will not be printed. In other words, the following two lines are equivalent:

```
\Numspell[⟨zeros⟩]{⟨num⟩}
\Numspell*[⟨zeros⟩]{⟨num⟩}\thenumspell
```

For example

```
\Numspell*{1}
\numspellsave{MyNum}
\Numspell*{2}
\thenumspell;
\thenumspellMyNum
```

Two; One

`\ordnumspell[⟨zeros⟩]{⟨num⟩}`

Spelling the ordinal number $n = \langle num \rangle \cdot 10^{\langle zeros \rangle}$, where $0 \leq n \leq 10^{67} - 1$. The default value of `⟨zeros⟩` is 0. For example

```
\ordnumspell{12000} → twelve thousandth
\ordnumspell[3]{12} → twelve thousandth
\ordnumspell[6]{12} → twelve millionth
\ordnumspell[63]{1} → one vigintillionth
```

`\ordnumspell*[⟨zeros⟩]{⟨num⟩}`

It works like `\ordnumspell`, but the number spelling will not be printed. In other words, the following two lines are equivalent:

```
\ordnumspell[⟨zeros⟩]{⟨num⟩}
\ordnumspell*[⟨zeros⟩]{⟨num⟩}\thenumspell
```

For example

```
\ordnumspell*{1}
\numspellsave{MyNum}
\ordnumspell*{2}
\thenumspell;
\thenumspellMyNum
```

second; first

`\Ordnumspell[⟨zeros⟩]{⟨num⟩}`

It works like `\ordnumspell`, but the first letter will be capital. For example

```
\Ordnumspell{12000} → Twelve thousandth
\Ordnumspell[3]{12} → Twelve thousandth
\Ordnumspell[6]{12} → Twelve millionth
\Ordnumspell[63]{1} → One vigintillionth
```

`\Ordnumspell*[⟨zeros⟩]{⟨num⟩}`

It works like `\Ordnumspell`, but the number spelling will not be printed. In other words, the following two lines are equivalent:

```
\Ordnumspell[⟨zeros⟩]{⟨num⟩}
\Ordnumspell*[⟨zeros⟩]{⟨num⟩}\thenumspell
```

For example

```
\Ordnumspell*{1}
\numspellsave{MyNum}
\Ordnumspell*{2}
\thenumspell;
\thenumspellMyNum
```

Second; First

3 Commands for English language

If the `english`, `british`, `ukenglish` or `UKenglish` language is active, then the number spelling will happen in British English. But it will be in American English, if the `american`, `usenglish` or `USenglish` language is active.

`\numspellUS`

Using British English, you can rechange the number spelling to American English by this command.

`\numspellGB`

Using American English, you can rechange the number spelling to British English by this command.

4 Commands for French language

The following commands only work, if `french` language is active.

`\numspellpremiere`

By default, `\ordnumspell{1}` → premier,
but `\numspellpremiere\ordnumspell{1}` → première

`\numspellpremier` (default)

```
\numspellpremiere\ordnumspell{1};
\numspellpremier\ordnumspell{1}
```

première ; premier

5 Commands for Hungarian language

The following commands only work, if `magyar` or `hungarian` language is active.

`\anumspell[⟨zeros⟩]{⟨num⟩}`

It works like `\numspell`, but the number spelling will start with Hungarian definite article.
For example

```
\anumspell{1} → az egy  
\anumspell{2} → a kettő
```

`\anumspell*[⟨zeros⟩]{⟨num⟩}`

It works like `\anumspell`, but the number spelling will not be printed. In other words, the following two lines are equivalent:

```
\anumspell[⟨zeros⟩]{⟨num⟩}  
\anumspell*[⟨zeros⟩]{⟨num⟩}\thenumspell
```

For example

```
\anumspell*{1}  
\numspellsave{MyNum}  
\anumspell*{2}  
\thenumspell;  
\thenumspellMyNum
```

a kettő; az egy

`\Anumspell[⟨zeros⟩]{⟨num⟩}`

It works like `\anumspell`, but the first letter will be capital.

`\Anumspell*[⟨zeros⟩]{⟨num⟩}`

It works like `\anumspell*`, but the first letter will be capital.

`\aordnumspell[⟨zeros⟩]{⟨num⟩}`

It works like `\ordnumspell`, but the number spelling will start with Hungarian definite article. For example

```
\aordnumspell{1} → az első  
\aordnumspell{2} → a második
```

`\aordnumspell*[⟨zeros⟩]{⟨num⟩}`

It works like `\aordnumspell`, but the number spelling will not be printed. In other words, the following two lines are equivalent:

```
\aordnumspell[⟨zeros⟩]{⟨num⟩}  
\aordnumspell*[⟨zeros⟩]{⟨num⟩}\thenumspell
```

For example

```
\aordnumspell*{1}  
\numspellsave{MyNum}  
\aordnumspell*{2}  
\thenumspell;  
\thenumspellMyNum
```

a második; az első

`\Aordnumspell[⟨zeros⟩]{⟨num⟩}`

It works like `\ordnumspell`, but the first letter will be capital.

`\Aordnumspell*[⟨zeros⟩]{⟨num⟩}`

It works like `\aordnumspell*`, but the first letter will be capital.

6 Commands for Italian language

The following commands only work, if `italian` language is active.

`\numspellitmasculine` (default)

The ordinal numbers will be printed in masculine form. For example
`\ordnumspell{1}` → primo

`\numspellitfeminine`

The ordinal numbers will be printed in feminine form. For example
`\numspellitfeminine\ordnumspell{1}`;
`\numspellitmasculine\ordnumspell{1}`
prima; primo

7 Examples

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[magyar,italian,ngerman,french,english]{babel}
\usepackage{numspell}
\usepackage[group-separator={,}]{siunitx}
\begin{document}
\def\mynum{123456789}
\noindent
In British English the spelling of \num{\mynum} is
\emph{\numspell{\mynum}}.
\par\smallskip\noindent
In American English the spelling of \num{\mynum} is
\foreignlanguage{american}{\em\numspell{\mynum}}.
\par\smallskip\noindent
In French the spelling of \num{\mynum} is
\foreignlanguage{french}{\em\numspell{\mynum}}.
\par\smallskip\noindent
In German the spelling of \num{\mynum} is
\foreignlanguage{ngerman}{\em\numspell{\mynum}}.
\par\smallskip\noindent
In Hungarian the spelling of \num{\mynum} is
\foreignlanguage{magyar}{\em\numspell{\mynum}}.
```

```

\par\smallskip\noindent
In Italian the spelling of \num{\mynum} is
\foreignlanguage{italian}{\em\numspell{\mynum}}.
\end{document}

```

In British English the spelling of 123,456,789 is *one hundred and twenty-three million, four hundred and fifty-six thousand, seven hundred and eighty-nine*.

In American English the spelling of 123,456,789 is *one hundred twenty-three million, four hundred fifty-six thousand, seven hundred eighty-nine*.

In French the spelling of 123,456,789 is *cent vingt-trois millions quatre cent cinquante-six mille sept cent quatre-vingt-neuf*.

In German the spelling of 123,456,789 is *ehundertdreiundzwanzig Millionen vierhundertsechs- undfünfzigtausendsiebenhundertneunundachtzig*.

In Hungarian the spelling of 123,456,789 is *százhuszonhárommillió-négyszázötvenhatezer-hétszáz-nyolcvankilenc*.

In Italian the spelling of 123,456,789 is *centoventitré milioni quattrocentocinquantaseimila settecentottantanove*.

```

\documentclass{article}
\usepackage{numspell}
\newcounter{mycount}
\begin{document}
The
\makeatletter
\@whilenum\value{mycount}<51
\do{\ordnumspell{\themycount}\stepcounter{mycount},\ }\dots
\makeatother
\end{document}

```

The zeroth, first, second, third, fourth, fifth, sixth, seventh, eighth, ninth, tenth, eleventh, twelfth, thirteenth, fourteenth, fifteenth, sixteenth, seventeenth, eighteenth, nineteenth, twentieth, twenty-first, twenty-second, twenty-third, twenty-fourth, twenty-fifth, twenty-sixth, twenty-seventh, twenty-eighth, twenty-ninth, thirtieth, thirty-first, thirty-second, thirty-third, thirty-fourth, thirty-fifth, thirty-sixth, thirty-seventh, thirty-eighth, thirty-ninth, fortieth, forty-first, forty-second, forty-third, forty-fourth, forty-fifth, forty-sixth, forty-seventh, forty-eighth, forty-ninth, fiftieth, ...

```

\documentclass{article}
\usepackage{numspell}
\newcounter{mycount}
\def\themycount{\numspell{\arabic{mycount}}}
\begin{document}
\Numspell{0},
\makeatletter
\@whilenum\value{mycount}<30
\do{\stepcounter{mycount}\themycount,\ }\dots

```

```
\makeatother
\end{document}
```

Nought, one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty, twenty-one, twenty-two, twenty-three, twenty-four, twenty-five, twenty-six, twenty-seven, twenty-eight, twenty-nine, thirty, ...

8 Limitations

Do not use the `\numspell`, `\numspell*`, `\Numspell`, `\Numspell*`, etc. commands inside sectioning commands and `\MakeUppercase`. An example for the illustration of the problem:

```
\documentclass{article}
\usepackage{hyperref}
\usepackage{numspell}
\pagestyle{headings}
\begin{document}
\section{The \ordnumspell{123} factor}
\MakeUppercase{\numspell{123}}
\newpage
Text
\end{document}
```

The following mistakes occur:

1. On the page 1: “one hundred and twenty-three”
Required: “ONE HUNDRED AND TWENTY-THREE”
2. On the heading: “*THE one hundred and twenty-third FACTOR*”
Required: “*THE ONE HUNDRED AND TWENTY-THIRD FACTOR*”
3. On the pdf bookmark: “The 123 factor”
Required: “The one hundred and twenty-third factor”

The solution

```
\documentclass{article}
\usepackage{hyperref}
\usepackage{numspell}
\pagestyle{headings}
\begin{document}
\ordnumspell*{123}
\section{The \thenumspell\ factor}
\numspell*{123}
\MakeUppercase{\thenumspell}
\newpage
Text
\end{document}
```