The **eukdate** package

Andrew Gilbert Moschou
andmos@gmail.com

Wednesday, 18 March 2009
v. 1.04

1 Introduction

`\today` This package redefines `\today`, to typeset `\today`'s date in, the format *Day, Date Month Year* (here, called 'UK format'), e.g., 'Wednesday, 18 March 2009', as is preferred in many parts of the world outside the US. Thus, `\maketitle` uses the UK format (instead of the default US format, e.g. March 18, 2009, as used in the `article` class).

From *Hart's Rules* (Hart, H., 1983. *Hart's Rules for Compositors and Readers at the University Press Oxford*, Thirty-ninth edition, Completely revised, Originally compiled by Horace Hart, M.A., Printer to the University, 1883–1915, Oxford University Press, Oxford, pp. 18–19.):

> In dates, print 25 June 1978.[1] Omit comma between month and year: 'The delay after November 1967 was due to an oversight.'
>
> ---
>
> [1] This will not apply to quotations, nor to reprints of documents. As to the form May 19, 1862, Sir James Murray said, 'This is not logical: 19 May 1862 is. *Begin* at day, *ascend* to month, *ascend* to year; not *begin* at month, *descend* to day, then *ascend* to year.'

This package also determines `\today`'s day of the week, which still makes the package useful even to Americans who might use the `usdate` option. The day of the week is calculated according to Zeller's congruence:

$$w = \begin{cases} d + \left\lfloor \frac{26}{10}(m+1) \right\rfloor + y + \left\lfloor \frac{y}{4} \right\rfloor + \left\lfloor \frac{c}{4} \right\rfloor - 2c \pmod 7 & \text{Gregorian calendar} \\ d + \left\lfloor \frac{26}{10}(m+1) \right\rfloor + y + \left\lfloor \frac{y}{4} \right\rfloor + 5 - c \pmod 7 & \text{Julian calendar} \end{cases}$$

where d is the day of the month, m is the number of the month, y is the two digit year, c is the century (so the four digit year is given by $100c + y$) and w represents the day of the week (0 for Saturday through to 6 for Friday); January and February are counted as months 13 and 14 of the previous year.

The ordinal indicator of the date (in e.g. 18^{th}, 18th) is absent because its use is outdated.

2 Usage

`\weekday`
`\monthname`

The package is invoked with `\usepackage{eukdate}`. This package also defines `\weekday` to typeset `\today`'s day of the week and `\monthname` to typeset the name of `\today`'s month ('Wednesday' and 'March' when this document was typeset).

2.1 Options

no-weekday Suppresses the weekday when typesetting `\today`, for example, '18 March 2009'. `\weekday` is still defined and can be used in the document.

ukdate Typesets `\today` in UK format, 'Wednesday, 18 March 2009'. This option is applied by default, if the `usdate` option is not specified.

usdate Typesets `\today` in US format, 'Wednesday, March 18, 2009'.

gregorian Calculates the day of the week according to the Gregorian calendar, which is the calendar that is generally used today. This option is applied by default, if the `julian` option is not specified.

julian Calculates the day of the week according to the Julian calendar. However, since `\today`'s date is not *today*'s date, as intepreted using the Julian calendar, this option is of limited use and can be safely ignored for every day use.

3 Why a new solution?

It is natural to ask 'Why a new solution to this problem?' because surely, somebody has already dealt with the issue. This is true but other solutions tend to be cumbersome. A similar effect can be achieved with `\usepackage[australian]{babel}` or `\usepackage[english,cleanlook]{isodate}`, but these instances do not include the day of the week. The day of the week can be calculated separately with scripts like `dow` or `dayofweek`, but neither is automatically integrated into the definition of `\today`. The `datetime` package can do this with the `\longdate` command, but there are minor format issues, which can be fixed only by defining a new format.

This package was created to offer a simple method to set the date using a format according to *my* preferences in a way that is portable to each system. It supports only a handful of date formats and in no way aims to be comprehensive.

4 eukdate.sty

```
\newif\if@euk@noweekday
\newif\if@euk@usdate
\newif\if@euk@julian
\DeclareOption{no-weekday}{\@euk@noweekdaytrue}
\DeclareOption{ukdate}{\@euk@usdatefalse}
\DeclareOption{usdate}{\@euk@usdatetrue}
```

```
\DeclareOption{gregorian}{\@euk@julianfalse}
\DeclareOption{julian}{\@euk@juliantrue}
\ExecuteOptions{ukdate,gregorian}
\ProcessOptions\relax
```

4.1 Zeller's congruence

Actually, the equivalent congruence:

$$w = \begin{cases} d + \left\lfloor \frac{26}{10}\left(m+1\right)\right\rfloor + y + \left\lfloor \frac{y}{4} \right\rfloor + \left\lfloor \frac{c}{4} \right\rfloor + 5c \pmod 7 & \text{Gregorian calendar} \\ d + \left\lfloor \frac{26}{10}\left(m+1\right)\right\rfloor + y + \left\lfloor \frac{y}{4} \right\rfloor + 5 + 6c \pmod 7 & \text{Julian calendar} \end{cases}$$

is used instead of Zeller's, as claimed, to simplify the algorithm for the modulo operation.

```
\newcounter{euk@date}
\newcounter{euk@month}
\newcounter{euk@century}
\newcounter{euk@year}
\setcounter{euk@date}{\number\day}
\setcounter{euk@month}{\number\month}
\setcounter{euk@century}{\number\year}
\divide\c@euk@century by 100
\setcounter{euk@year}{\number\year}
\addtocounter{euk@year}{-\theeuk@century00}
\ifnum\number\theeuk@month<3
  \addtocounter{euk@month}{13}
  \addtocounter{euk@year}{-1}
\else
  \addtocounter{euk@month}{1}
\fi
\multiply\c@euk@month by 26
\divide\c@euk@month by 10
\addtocounter{euk@date}{\theeuk@month}
\addtocounter{euk@date}{\theeuk@year}
\divide\c@euk@year by 4
\addtocounter{euk@date}{\theeuk@year}
\if@euk@julian
  \addtocounter{euk@date}{6\theeuk@century}
  \addtocounter{euk@date}{5}
\else
  \addtocounter{euk@date}{5\theeuk@century}
  \divide\c@euk@century by 4
  \addtocounter{euk@date}{\theeuk@century}
\fi
```

The modulo operation.

```
\loop\ifnum6<\theeuk@date
  \addtocounter{euk@date}{-7}
\repeat
```

4.2 Definitions

`\weekday` Define `\weekday` to typeset the day of the week.

```
44 \newcommand\weekday{%
45   \ifcase\theeuk@date Saturday\or Sunday\or Monday\or
46     Tuesday\or Wednesday\or Thursday\or Friday\fi}
```

`\monthname` Define `\monthname` to typeset the month.

```
47 \newcommand\monthname{%
48   \ifcase\number\month\or January\or February\or March\or
49     April\or May\or June\or July\or August\or
50     September\or October\or November\or December\fi}
```

`\today` Redefine `\today`.

```
51 \def\today{%
52   \if@euk@noweekday\else\weekday,\space\fi
53   \if@euk@usdate
54     \monthname\space\number\day,%
55   \else
56     \number\day\space\monthname
57   \fi
58   \space\number\year}
```

An ‘undocumented’ feature.

```
59 \edef\euk@ordinal{%
60    \ifcase\number\day\or st\or nd\or rd\or th\or th\or
61      th\or th\or th\or th\or th\or th\or th\or th\or th\or
62      th\or th\or th\or th\or th\or th\or st\or nd\or rd\or
63      th\or th\or th\or th\or th\or th\or th\or st\fi}
```

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.

Symbols
\@euk@julianfalse . . 7
\@euk@juliantrue . . . 8
\@euk@noweekdaytrue . 4
\@euk@usdatefalse . . 5
\@euk@usdatetrue . . . 6

C
\c@euk@century . . 18, 38
\c@euk@month 27, 28
\c@euk@year 31

E
\euk@ordinal 59

I
\if@euk@julian . . . 3, 33
\if@euk@noweekday 1, 52
\if@euk@usdate . . . 2, 53

M
\monthname *2*, 47, 54, 56

T
\theeuk@century 20, 34, 37, 39
\theeuk@date 41, 45
\theeuk@month . . . 21, 29
\theeuk@year 30, 32
\today *1*, 51

W
\weekday *2*, 44, 52