datetime.sty v2.49: Formatting Current Date and Time

Nicola L.C. Talbot

5 Dec 2006

Contents

1 Introduction

The datetime package is a LaTeX 2_ε package that provides various different formats for `\today`, and provides commands for displaying the current time. If you only want the time commands but not the date changing commands, you can pass the option `nodate` to the package.

Since version 2.4, the datetime package has been separated into two packages: datetime and fmtcount. When I originally created this package, I defined the commands, `\ordinal` etc which could be used in the definition of `\today`. Since then, I have extended the number of commands available that can be used to

display the value of a LaTeX counter, however it seems more appropriate to define all these counter-related commands in a separate package. The fmtcount package is now distributed separately from the datetime package, and will also need to be installed.

As from version 2.42, the datetime package is now compatible with babel, however you must load the datetime package *after* the babel package. For example:

```
\usepackage[francais]{babel}
\usepackage{datetime}
```

2 Date Declarations

There are various declarations that change the effect of `\today`. The change can be localised by placing the declaration within a group.

⟨*Day*⟩ ⟨*Month*⟩ ⟨*Year*⟩ formats:

`\longdate` The declaration `\longdate` will redefine `\today` to produce the current date displayed in the form Wednesday 8th March, 2000 if the package option `dayofweek` is used, or 8th March, 2000 if the package option `nodayofweek` is used.

`\shortdate` The declaration `\shortdate` will redefine `\today` to produce the current date displayed in the form Wed 8th Mar, 2000 if the package option `dayofweek` is used, or 8th Mar, 2000 if the package option `nodayofweek` is used.

`\ddmmyyyydate` The declaration `\ddmmyyyydate` will redefine `\today` to produce the current date displayed in the form 08/03/2000

`\dmyyyydate` The declaration `\dmyyyydate` will redefine `\today` to produce the current date displayed in the form 8/3/2000

`\ddmmyydate` The declaration `\ddmmyydate` will redefine `\today` to produce the current date displayed in the form 08/03/00

`\dmyydate` The declaration `\dmyydate` will redefine `\today` to produce the current date displayed in the form 8/3/00

`\textdate` The declaration `\textdate` will redefine `\today` to produce the current date displayed in the form: Wednesday the Eighth of March, Two Thousand if the package option `dayofweek` is used, or Eighth of March, Two Thousand if the package option `nodayofweek` is used.

⟨*Month*⟩ ⟨*Day*⟩ ⟨*Year*⟩ formats:

`\usdate` The declaration `\usdate` will redefine `\today` to produce the current date displayed in the form March 8, 2000. (As TeX and LaTeX do by default.)

`\mmddyyyydate` The declaration `\mmddyyyydate` will redefine `\today` to produce the current date displayed in the form 03/08/2000

`\mdyyyydate` The declaration `\mdyyyydate` will redefine `\today` to produce the current date displayed in the form 3/8/2000

`\mmddyydate` The declaration `\mmddyydate` will redefine `\today` to produce the current date displayed in the form 03/08/00

`\mdyydate` The declaration `\mdyydate` will redefine `\today` to produce the current date displayed in the form 3/8/00

In addition, the declarations `\date`⟨*lang*⟩ are available for all languages defined either by calling babel prior to datetime or by passing the language name as an

option to datetime. See section 5 if you want to define your own customised date format.

As from version 2.43, the numerical date formats (such as `\ddmmyyyydate`) use the command `\dateseparator` to separate the numbers. So, for example, if you want to hyphens instead of slashes, you can do:

```
\renewcommand{\dateseparator}{-}
```

3 Time Commands

`\currenttime` `\settimeformat`

The current time is displayed using the command `\currenttime`. The format can be changed using the declaration `\settimeformat{`⟨*style*⟩`}`, where ⟨*style*⟩ is the name of the format[1]. Available formats are:

xxivtime Twenty-four hour time in the form 22:28 (Default)

ampmtime Twelve hour time in the form 10:28pm

oclock Displays the current time as a string, e.g. Twenty-Eight minutes past Ten in the afternoon.[2]

`\newtimeformat`

New time formats can be defined using the command:

`\newtimeformat{`⟨*name*⟩`}{`⟨*format*⟩`}`

where ⟨*name*⟩ is the name of the new format (used in `\settimeformat`), and ⟨*format*⟩ is how to format the time. Within ⟨*format*⟩ you can use the counters `HOUR` (number of hours after midnight), `MINUTE` (number of minutes past the hour), `HOURXII` (number of hours after midnight/midday), `TOHOUR` (the next hour) and `TOMINUTE` (number of minutes to the next hour), and the corresponding commands: `\THEHOUR`, `\THEMINUTE`, `\THEHOURXII`, `\THETOHOUR` and `\THETOMINUTE`.

For example, to define a new time format that uses a dot instead of a colon:

```
\newtimeformat{dottime}{\twodigit{\THEHOUR}.\twodigit{\THEMINUTE}}
```

You then need to switch to this new format before you can use it:

```
\settimeformat{dottime}
\currenttime
```

As from version 2.43, if you only want to change the separator, you can simply redefine `\timeseparator` instead of defining a new time format. For example:

```
\renewcommand{\timeseparator}{.}
```

The `xxivtime` format will now work like the `dottime` format defined above.

4 Formating Dates

`\pdfdate`

The command `\pdfdate`[3] prints the date in the format required for PDF files, e.g.

[1] Note that the commands `\xxivtime`, `\ampmtime` and `\oclock` are still available, `\settimeformat` redefines `\currenttime` to the command given by placing a backslash in front of ⟨*style*⟩. So `\settimeformat{xxivtime}` sets `\currenttime` to `\xxivtime` and so on.

[2] Version 2.43 fixed bug which caused an infinite loop on the hour.

if the date is 1 May 2004 and time is 22:02, `\pdfdate` will print 20040501220200. The reason this date format is separate from all the others is because the other form doesn't get properly expanded by PDFTEX. (This command is defined regardless of whether the package option `nodate` is called.) Example:

```
\pdfinfo{
   /Author (Me)
   /Title (A Sample Document)
   /CreationDate (D:20040501215500)
   /ModificationDate (D:\pdfdate)
}
```

`\monthname` `\shortmonthname` There are two commands that print the name of the current month: `\monthname` prints the current month name in full, e.g. August, and `\shortmonthname` prints the abbreviated month name, e.g. Aug. Both `\monthname` and `\shortmonthname` take an optional argument (a number from 1 to 12) if the name of a specific month is required. For example, `\monthname[6]` will produced the output: June.

The day of the week is computed using the algorithm documented at http://userpages.wittenburg.edu/bshelburne/Comp150/DayOfWeek.htm. This algorithm works for any date between 1st Jan, 1901 and 31st Dec, 2099. The following macros display the day of week for a given date:

`\dayofweekname` `\dayofweekname{`⟨*day*⟩`}{`⟨*month*⟩`}{`⟨*year*⟩`}` prints the day of week for the specified date. For example, `\dayofweekname{31}{10}{2002}` will produce the output: Thursday.

`\shortdayofweekname` `\shortdayofweekname{`⟨*day*⟩`}{`⟨*month*⟩`}{`⟨*year*⟩`}` prints the abbreviated name for the day of week for the specified date. For example
`\shortdayofweekname{31}{10}{2002}`
will produce the output: Thu.

`\ifshowdow` The TEX conditional `\ifshowdow` can be used to determine whether or not the option `dayofweek` has been passed to the package. For example:

```
\ifshowdow\dayofweekname{31}{10}{2002} \fi
```

will only display the day of week if the `dayofweek` option was passed to datetime. Alternatively, you can use David Carlisle's ifthen package:

```
\ifthenelse{\boolean{showdow}}{\dayofweekname{31}{10}{2002} }{}
```

`\ordinaldate` The command `\ordinaldate{`⟨*number*⟩`}` displays ⟨*number*⟩ as a date-type ordinal. If the current language is English, this will simply pass the argument to `\ordinalnum` (defined in the fmtcount package), if the current language is Breton, Welsh or French, a superscript will only be added if ⟨*number*⟩ is 1, otherwise only ⟨*number*⟩ will be displayed.

`\formatdate` The macro `\formatdate{`⟨*day*⟩`}{`⟨*month*⟩`}{`⟨*year*⟩`}`[4] formats the specified date according to the current format of `\today`[5]. (Arguments must all be integers.) For example, in combination with `\longdate`, the command
`\formatdate{27}{9}{2004}`
will produce the output: Monday 27th September, 2004.

`\twodigit` You can ensure that a number is displayed with at least two digits by using

[3] thanks to Ulrich Dirr for asking about this

[4] Note the name change since version 1.1. The command name was changed from `\thedate` to `\formatdate` to avoid a name clash when using the seminar class file.

[5] To be more precise, `\today` is defined to be `\formatdate{\day}{\month}{\year}` where `\longdate` etc change the definition of `\formatdate`

the command \twodigit{⟨*num*⟩}[6]. This is of use if you want to define your own date or time formats.

5 Defining New Date Formats

\newdateformat

New date formats can be defined using the command:

\newdateformat{⟨*name*⟩}{⟨*format*⟩}

where ⟨*name*⟩ is the name of the new format, and ⟨*format*⟩ is how to format the date. Within the argument ⟨*format*⟩ you can use the commands \THEDAY, \THEMONTH and \THEYEAR to represent the relevant day, month and year, or you can use the counters DAY, MONTH and YEAR if you want to use \ordinal etc. Once you have defined the new date format, you can then switch to it using the declaration \⟨*name*⟩ (i.e. the name you specified preceded by a backslash), and subsequent calls to \today and \formatdate will use your new format.

For example, suppose you want to define a new date format called, say, mydate, that will typeset the date in the form: 8-3-2002, then you can do:

```
\newdateformat{mydate}{\THEDAY-\THEMONTH-\THEYEAR}
```

\newdateformat will then define the declaration \mydate which can be used to switch to your new format. In the following example, two new date formats are defined, and they are then selected to produce two different formats for the current date:

```
\newdateformat{dashdate}{%
\twodigit{\THEDAY}-\twodigit{\THEMONTH}-\THEYEAR}

\newdateformat{usvardate}{%
\monthname[\THEMONTH] \ordinal{DAY}, \THEYEAR}

Dash: \dashdate\today.
US: \usvardate\today.
```

If the current date is, say, 8th March, 2002, the above code will produce the following: Dash: 08-03-2002. US: March 8$^{\text{th}}$, 2002.

Note that \THEDAY etc and DAY etc have no real meaning outside \newdateformat (this is why they are in uppercase). Incidentally, the dashdate format is not really necessary, as you can achieve this format using:

```
\renewcommand{\dateseparator}{-}
\ddmmyyyydate
```

Another note: in the above code, \ordinal was used to illustrate the use of the DAY counter. It is better to use \ordinaldate instead:

```
\newdateformat{usvardate}{%
\monthname[\THEMONTH] \ordinaldate{\THEDAY}, \THEYEAR}
```

[6] New to version 2.2

6 Saving Dates

It is possible to save a date for later use using the command: [7]

\newdate `\newdate{`⟨*name*⟩`}{`⟨*day*⟩`}{` ⟨*month*⟩`}{`⟨*year*⟩`}`

This date can later be displayed using the same format as that used by `\formatdate` using the command:

\displaydate `\displaydate{`⟨*name*⟩`}`

Individual elements of the date can be extracted using the commands:

\getdateday `\getdateday{`⟨*name*⟩`}`

\getdatemonth `\getdatemonth{`⟨*name*⟩`}`

\getdateyear `\getdateyear{`⟨*name*⟩`}`

7 Predefined Names

The following commands are defined by the datetime package:

Command Name	**Default Value**
`\dateseparator`	`/`
`\timeseparator`	`:`
`\amname`	`am`
`\pmname`	`pm`
`\amorpmname`	`\amname` if morning, otherwise `\pmname`
`\amstring`	`in the morning`
`\pmstring`	`in the afternoon`
`\amorpmstring`	`\amstring` if morning, otherwise `\pmstring`
`\halfpast`	`Half past`
`\quarterpast`	`Quarter past`
`\quarterto`	`Quarter to`
`\noon`	`Noon`
`\midnight`	`Midnight`
`\oclockstring`	`O'Clock`

8 Package Options

The following options may be passed to this package:

[7] Thanks to Denis Bitouzé for asking about this

`long`	make `\today` produce long date
`short`	make `\today` produce short date
`ddmmyyyy`	make `\today` produce DD/MM/YYYY date
`dmyyyy`	make `\today` produce D/M/YYYY date
`ddmmyy`	make `\today` produce DD/MM/YY date
`dmyy`	make `\today` produce D/M/YY date
`text`	make `\today` produce text date
`us`	make `\today` produce US style date
`mmddyyyy`	make `\today` produce MM/DD/YYYY date
`mdyyyy`	make `\today` produce M/D/YYYY date
`mmddyy`	make `\today` produce MM/DD/YY date
`mdyy`	make `\today` produce M/D/YY date
`raise`	make ordinal st,nd,rd,th appear as superscript
`level`	make ordinal st,nd,rd,th appear level with rest of text
`dayofweek`	make the day of week appear for `\longdate`, `\shortdate` or `\textdate`
`nodayofweek`	don't display the day of week.
`24hr`	make `\currenttime` produce `xxivtime` format
`12hr`	make `\currenttime` produce `ampmtime` format
`oclock`	make `\currenttime` produce `oclock` format
`nodate`	Don't redefine `\today` or define the month or day of week commands (useful if you only want the time commands or `\pdfdate`)

The default options are: `long`, `raise`, `dayofweek` and `24hr`.

9 Multilingual Support

If the babel package is called prior to datetime, `\date`⟨*lang*⟩ will be the default date format, where ⟨*lang*⟩ is the current language.

The commands `\monthname` and `\shortmonthname`, will produce the month name in the current language. If you want the month name in a specific language, you can use the command `\monthname`⟨*lang*⟩. For example, `\monthnamefrench[6]` will produce the output: juin.

There is currently only limited multilingual support for `\dayofweekname` and `\shortdayofweekname` (just English, French, Portuguese and Spanish). You can add support for other languages by defining the commands `\dayofweeknameid`⟨*lang*⟩ and `\shortdayofweeknameid`⟨*lang*⟩. Note that these commands only take *one* argument which should be a number from 1 to 7 indicating the day of the week.

You can use the following as templates. Replace `english` with the name of your language (as given by `\languagename`) and replace `Sunday` etc as appropriate:

```
\providecommand{\dayofweeknameidenglish}[1]{%
\ifcase#1\relax
\or Sunday%
\or Monday%
\or Tuesday%
\or Wednesday%
\or Thursday%
\or Friday%
\or Saturday%
\fi}
```

```
\providecommand{\shortdayofweekidnameenglish}[1]{%
\ifcase#1\relax
\or Sun%
\or Mon%
\or Tue%
\or Wed%
\or Thu%
\or Fri%
\or Sat%
\fi}
```

If you want them added to future versions of datetime, please e-mail me the code.

10 Configuration File

As from Version 2.4, the datetime package will read in settings from the configuration file `datetime.cfg`, if it exists, which will override the default package options. For example, suppose you prefer a short date without the day of week by default, you will need to create a file called `datetime.cfg` that contains the line:

```
\shortdate\showdowfalse
```

The file `datetime.cfg` should then go somewhere on the TEX path. Now all you need to do is:

```
\usepackage{datetime}
```

without having to specify the `short` and `nodayofweek` options.

You can also use this file to define and set your own date styles. For example, you could create a configuration file that has the following lines:

```
\newdateformat{dashdate}{\twodigit{\THEDAY}-\twodigit{\THEMONTH}-\THEYEAR}
\dashdate
```

Whenever you use the datetime package, it will now use this format by default.

11 LaTeX2HTML styles

Version 2.43 and above of the datetime bundle supplies the LaTeX2HTML style file `datetime.perl`. This file should be placed in a directory searched by LaTeX2HTML. The following limitations apply to the LaTeX2HTML styles:

- The configuration file `datetime.cfg` is currently ignored. (This is because I can't work out the correct code to do this. If you know how to do this, please let me know.) You can however do:

  ```
  \usepackage{datetime}
  \html{\input{datetime.cfg}}
  ```

  This, I agree, is an unpleasant cludge.

- The commands `\monthname`⟨*language*⟩ are not implemented.
- Some of the languages are not implemented.
- The package option `nodate` is not implemented.

12 Troubleshooting

There is a datetime FAQ available at: http://theoval.cmp.uea.ac.uk/~nlct/latex/packages/faq/

13 Contact Details

Dr Nicola Talbot
School of Computing Sciences
University of East Anglia
Norwich. NR4 7TJ.
United Kingdom.
http://theoval.cmp.uea.ac.uk/~nlct/