datetime.sty v2.60: Formatting Current Date and Time

Nicola L. C. Talbot

Dickimaw Books
http://www.dickimaw-books.com/

2015-03-20

This is the last version of this package. The replacement package is called datetime2 and will be released shortly after the this final version.

Contents

1 Introduction

The datetime package is a LaTeX2ε 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 Predefined Date Formats

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

As from version 2.43, the numerical date formats (such as `\ddmmyyyydate`) use the command

`\dateseparator`

```
\dateseparator
```

to separate the numbers. So, for example, if you want to hyphens instead of slashes, you can do:

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

2.1 ISO format

`\yyyymmdddate`

```
\yyyymmdddate
```

This declaration will redefine `\today` to produce the current date displayed in the form 2000/03/08. (You can redefine `\dateseparator` to `-` to change to 2000-03-08.)

2.2 ⟨***Day***⟩ ⟨***Month***⟩ ⟨***Year***⟩ formats

`\longdate`

```
\longdate
```

This declaration will redefine `\today` to produce the current date displayed in the form Wednesday 8$^{\text{th}}$ March, 2000 if the package option dayofweek is used, or 8$^{\text{th}}$ March, 2000 if the package option nodayofweek is used.

`\shortdate`

```
\shortdate
```

This declaration will redefine `\today` to produce the current date displayed in the form Wed 8$^{\text{th}}$ Mar, 2000 if the package option dayofweek is used, or 8$^{\text{th}}$ Mar, 2000 if the package option nodayofweek is used.

`\ddmmyyyydate`

```
\ddmmyyyydate
```

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

`\dmyyyydate`

```
\dmyyyydate
```

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

`\ddmmyydate`

```
\ddmmyydate
```

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

`\dmyydate`

```
\dmyydate
```

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

`\textdate`

```
\textdate
```

This declaration 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. Note that `\textdate` is defined for use with English, it won't look right if it is used when another language has been selected[1]. If you want to define a similar command for another language, you will first need to check that the fmtcount package supports that language.

2.3 ⟨*Month*⟩ ⟨*Day*⟩ ⟨*Year*⟩ formats

`\usdate`

```
\usdate
```

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

`\mmddyyyydate`

```
\mmddyyyydate
```

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

`\mdyyyydate`

```
\mdyyyydate
```

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

`\mmddyydate`

```
\mmddyydate
```

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

`\mdyydate`

```
\mdyydate
```

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

2.4 Dates defined by babel

In addition to the above, 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.

---

[1] in fact, you may get an error from the fmtcount package if you are using a language that it doesn't support.

3 Time Commands

The current time is displayed using the command

`\currenttime`

```
\currenttime
```

A specific time can be displayed using the command

`\formattime`

```
\formattime{⟨hour⟩}{⟨minute⟩}{⟨second⟩}
```

where ⟨*hour*⟩ is a number from 0 to 23, and ⟨*minute*⟩ and ⟨*second*⟩ are numbers from 0 to 59.

The format can be changed using the declaration

`\settimeformat`

```
\settimeformat{⟨style⟩}
```

where ⟨*style*⟩ is the name of the format. Predefined formats are:

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

hhmmsstime Twenty-four hour time in the form 22:28:00

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.

New time formats can be defined using the command:

`\newtimeformat`

```
\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), SECOND (number of seconds) 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`, `\THESECOND`, `\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`

```
\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`

```
\pdfdate
```

This command[2] prints the date in the format required for PDF files, e.g. 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)
   /ModDate (D:\pdfdate)
}
```

Note that PDFTEX introduced the new primitive `\pdfcreationdate` in version 1.30.0, so `\pdfdate` isn't required with newer versions of PDFTEX. (`\pdfcreationdate` is better than `\pdfdate` as it also stores the time zone.)

There are two commands that print the name of the current month:

`\monthname`

```
\monthname[⟨num⟩]
```

prints the current month name in full, e.g. August, and

`\shortmonthname`

```
\shortmonthname[⟨num⟩]
```

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.

---

[2] thanks to Ulrich Dirr for asking about this

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`

`\ifshowdow`

This TEX conditional 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} }{}
```

The command

`\ordinaldate`

`\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.

The macro[3]

`\formatdate`

`\formatdate{⟨day⟩}{⟨month⟩}{⟨year⟩}`

---

[3] 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.

formats the specified date according to the current format of `\today`[4]. (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.

You can ensure that a number is displayed with at least two digits using the command

`\twodigit`

```
\twodigit{⟨num⟩}
```

5 Defining New Date Formats

New date formats can be defined using the command:

`\newdateformat`

```
\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.
```

---

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

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

It is possible to save a date for later use using the command: [5]

`\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:

---

[5] Thanks to Denis Bitouzé for asking about this

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:

long make `\today` produce long date

short make `\today` produce short date

yyyymmdd make `\today` produce YYYY/MM/DD 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 in UK style

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.

hhmmss make `\currenttime` produce `hhmmsstime` format

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)

iso as ddmmyyyy and hhmmss but also sets date separator to - and time separator to :

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

9 Multilingual Support

If you use the ngerman package, you must use datetime's ngerman package option if you want the date displayed in the same form as ngerman. You must also load ngerman *before* you load datetime. Similarly for the german package.

If you want to use the babel package, you must load it *before* you load the datetime package. This is because the babel `\date`⟨*lang*⟩ commands redefine `\today`, whereas the datetime package redefines `\today` to use `\formatdate`, and the date formatting commands (such as `\longdate`) redefine `\formatdate` rather than `\today`. This ensures consistent formatting of the dates whether you use `\today` or `\formatdate`. Therefore, the datetime package has to redefine all the `\date`⟨*lang*⟩ commands accordingly. Thus the multilingual date support is mostly limited to that provided by babel. Additional support, such as the day of week names and abbreviations, are only supplied for those languages that I know, or that other people have been able to supply for me.

As from v2.56, the package options supplied to datetime override babel's date format. For example:

```
\usepackage[french,spanish]{babel}
\usepackage[ddmmyyyy]{datetime}
```

will use ddmmyyyy date format regardless of language. Note that if you use a declaration instead of a package option, for example:

```
\usepackage[french,spanish]{babel}
\usepackage{datetime}
\ddmmyyyydate
```

the date format will be switched back to `\datefrench` or `\datespanish` whenever the language is set. You can use

`\setdefaultdate`

`\setdefaultdate{`⟨*declaration*⟩`}`

to always use `{`⟨*declaration*⟩`}` whenever the language is set. For example:

```
\setdefaultdate{\ddmmyyyydate}
```

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. Note that `\textdate` is formatted for English dates, and won't look right if used with another language setting. If you want a textual date, the fmtcount package (which is loaded by datetime) defines some commands which display a number or ordinal as a word, but it only has very limited multilingual support. See the fmtcount documentation for further details.

10 Registers

TEX provides `\day`, `\month` and `\year` registers. In addition, datetime provides the registers: `\currenthour`, `\currentminute` and `\currentsecond`. Note that old distributions of TEX will always have `\currentsecond` set to zero.

11 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.

12 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. 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.

13 Troubleshooting

There is a datetime FAQ available at: `http://www.dickimaw-books.com/faqs/datetimefaq.html`.