datetime.sty v2.55: Formatting Current Date and Time

Nicola L. C. Talbot

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

20 Aug 2007

Contents

1 Introduction 1
2 Date Declarations 2
3 Time Commands 3
4 Formating Dates 4
5 Defining New Date Formats 5
6 Saving Dates 6
7 Predefined Names 6
8 Package Options 6
9 Multilingual Support 7
10 Configuration File 8
11 LaTeX2HTML styles 9
12 Troubleshooting 9

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 \TeX{} 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 8\textsuperscript{th} March, 2000 if the package option dayofweek is used, or 8\textsuperscript{th} 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 8\textsuperscript{th} Mar, 2000 if the package option dayofweek is used, or 8\textsuperscript{th} 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 08/03/00

\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 08/03/00

\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. Note that \textdate is defined for use with English, it won’t look right if it is used when another language has been selected\textsuperscript{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.

⟨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.)

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

\footnote{\textsuperscript{1}In fact, you may get an error from the fmtcount package if you are using a language that it doesn’t support.}
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
\mddyydate The declaration \mddyydate 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\langle lang\rangle 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 \dmmyyyydate) 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 The current time is displayed using the command \currenttime. The format can be changed using the declaration \settimeformat{\langle style\rangle}, where \langle style\rangle is the name of the format\(^2\). Available formats are:
\begin{itemize}
  \item \xxivtime Twenty-four hour time in the form 22:28 (Default)
  \item \ampmtime Twelve hour time in the form 10:28pm
  \item \oclock Displays the current time as a string, e.g. Twenty-Eight minutes past Ten in the afternoon.
\end{itemize}
\settimeformat New time formats can be defined using the command:
\begin{verbatim}
\newtimeformat\langle name\rangle\langle format\rangle
\end{verbatim}
where \langle name\rangle is the name of the new format (used in \settimeformat), and \langle format\rangle is how to format the time. Within \langle format\rangle 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:
\begin{verbatim}
\newtimeformat{dottime}{\twodigit\THEHOUR.\twodigit\THEMINUTE}
\end{verbatim}
You then need to switch to this new format before you can use it:
\begin{verbatim}
\settimeformat{dottime}
\currenttime
\end{verbatim}

\(^2\)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 \langle style\rangle. So \settimeformat{\xxivtime} sets \currenttime to \xxivtime and so on.
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

The command \pdfdate 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)  
}

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

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}}{}

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 fmcntcount 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 \formatdate{⟨day⟩}{⟨month⟩}{⟨year⟩} formats the specified

\footnote{3 thanks to Ulrich Dirr for asking about this}
date according to the current format of \today. (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 the command \twodigit{⟨num⟩}. 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 8th, 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}{-}
\ddmmyyydate
\footnote{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.}
\footnote{To be more precise, \today is defined to be \formatdate{\day}{\month}{\year} where \longdate etc change the definition of \formatdate}
Another note: in the above code, \texttt{\textbackslash ordinal} was used to illustrate the use of the \texttt{DAY} counter. It is better to use \texttt{\textbackslash ordinaldate} instead:

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

\section{Saving Dates}

It is possible to save a date for later use using the command: \footnote{6}{Thanks to Denis Bitouzé for asking about this}

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

This date can later be displayed using the same format as that used by \texttt{\textbackslash formatdate} using the command:

\displaydate\{⟨name⟩\}

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

\getdateday\{⟨name⟩\}
\getdatemonth\{⟨name⟩\}
\getdateyear\{⟨name⟩\}

\section{Predefined Names}

The following commands are defined by the \texttt{datetime} package:

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>\dateseparator</td>
<td>/</td>
</tr>
<tr>
<td>\timeseparator</td>
<td>:</td>
</tr>
<tr>
<td>\amname</td>
<td>am</td>
</tr>
<tr>
<td>\pmname</td>
<td>pm</td>
</tr>
<tr>
<td>\amorpmname</td>
<td>\amname if morning, otherwise \pmname</td>
</tr>
<tr>
<td>\amstring</td>
<td>in the morning</td>
</tr>
<tr>
<td>\pmstring</td>
<td>in the afternoon</td>
</tr>
<tr>
<td>\amorpmstring</td>
<td>\amstring if morning, otherwise \pmstring</td>
</tr>
<tr>
<td>\halfpast</td>
<td>Half past</td>
</tr>
<tr>
<td>\quarterpast</td>
<td>Quarter past</td>
</tr>
<tr>
<td>\quarterto</td>
<td>Quarter to</td>
</tr>
<tr>
<td>\noon</td>
<td>Noon</td>
</tr>
<tr>
<td>\midnight</td>
<td>Midnight</td>
</tr>
<tr>
<td>\oclockstring</td>
<td>O’Clock</td>
</tr>
</tbody>
</table>

\section{Package Options}

The following options may be passed to this package:
long
make \today produce long date
short
make \today produce short date
ddmyyyy
make \today produce DD/MM/YYYY date
dmyyy
make \today produce D/M/YYYY date
ddmyy
make \today produce DD/MM/YY date
dmy
make \today produce D/M/YY date
text
make \today produce text date
us
make \today produce US style date
mmdyyyy
make \today produce MM/DD/YYYY date
mdyyyy
make \today produce M/D/YYYY date
mmdyy
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 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.

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.

There is currently only limited multilingual support for \dayofweekname and \shortdayofweekname (just English, French, Portuguese, Spanish and German7). You can add support for other languages by defining the commands

7 thanks to Uwe Bieling for supplying the German names
\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*{\shortdayofweeknameidenglish}[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 email 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 \texttt{datetime} bundle supplies the LaTeX2HTML style file \texttt{datetime.perl}. This file should be placed in a directory searched by LaTeX2HTML. The following limitations apply to the LaTeX2HTML styles:

- The configuration file \texttt{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:

\begin{verbatim}
\usepackage{datetime}
\html{\input{datetime.cfg}}
\end{verbatim}

This, I agree, is an unpleasant cludge.

- The commands \texttt{\monthname(language)} are not implemented.
- Some of the languages are not implemented.
- The package option \texttt{nodate} is not implemented.

12 Troubleshooting

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