

# Termcal.sty—printing a class calendar\*

Bill Mitchell

December 19, 2008

## Abstract

This package is intended to print a term calendar for use in planning a class. It has a flexible mechanism for specifying which days of the week are to be included and for inserting text either regularly on the same day each week, or on selected days, or for a series of consecutive days. It also has a flexible mechanism for specifying class and nonclass days. Text may be inserted into consecutive days so that it automatically flows around nonclass days.

## 1 Description

`calendar` The main command in this package is the `calendar` environment. Figure 1 shows an example of its use, which might be suitable for a Monday-Wednesday-Friday class with a Thursday recitation in a mercifully short term. The  $\LaTeX$  input to generate it is given in figure 2.

The two arguments to the `calendar` environment are the starting date, in the format `m/d/y`, and the number of weeks to be printed. The contents of the environment describe what is to appear in the calendar. The commands fall into two classes: those which specify what is to be printed on a particular day in each week, and those which specify what is to be printed on specific days during the term.

`\skipday` The commands which specify what each week is to look like are `\calday` and `\skipday`. These commands specify the days of the week in order; thus there should be seven of them (although ten may be preferred for a course covering the French revolution). The macro `\skipday` simply declares that

---

\*Version: 1.8. Date: 1997/01/11. Documentation: 1996/01/10

MONDAY		TUESDAY		THURSDAY		FRIDAY	
Sep 26th	<b>1</b>	28th	<b>2</b>	29th		30th	<b>3</b>
Oct 3rd	<b>4</b>	5th	<b>5</b>	6th		7th	<b>6</b>
10th	<b>7</b>	12th	<b>8</b>	13th		14th	<b>9</b>
		Midterm Exam				§1.2,1.3 Sorting and disporting	
17th	<b>10</b>	19th	<b>11</b>	20th		21st	<b>12</b>
§1.3,1,4 Assembling and dissembling						§9.9 Tending and rending	
24th	<b>13</b>	26th	<b>14</b>	27th		28th	<b>15</b>
31st No Class Halloween		Nov 2nd	<b>16</b>	3rd		4th	<b>17</b>

Figure 1: Example of the use of the `calendar` environment

the corresponding day should not be printed in the exam; thus the sample calendar has boxes only for Monday, Wednesday, Thursday, and Friday. The macro `\calday` `\calday[optional head]{option list}` is used to specify a day which is to be printed. The required argument *option list* is a (possibly empty) list of  $\TeX$  commands which are executed before the text for that day is printed. It can (at least in principal) include any  $\LaTeX$  command, but several commands, described below, are provided specifically for this purpose. The optional argument is the text of a heading for that day which will at the top of each page of the calendar.

`\classday` **Available options.** The macros `\classday` and `\noclassday` declare that the specified day is, or is not, a class day. Days specified as class days

```

\begin{calendar}{9/26/94}{6}
\setlength{\calboxdepth}{.3in}

        % Description of the Week.
\calday[Monday]{\classday}      % Monday
\skipday                % Tuesday (no class)
\calday[Tuesday]{\classday}    % Wednesday
\calday[Thursday]{\noclassday} % Thursday (unnumbered)
\calday[Friday]{\classday}     % Friday
\skipday\skipday           % weekend (no class)

        % Holidays
\options{10/31/94}{\noclassday}
\caltext{10/31/94}{No Class\\Halloween}

        % Exams
\caltext{10/12/94}{Midterm Exam}

        % Text on consecutive days
                % the subject of the 9th lecture
\caltexton{9}{\S1.2,1.3\\Sorting and disporting}
                % the subject of the 10th lecture
\caltextnext{\S1.3,1,4\\Assembling and dissembling}
\caltextnext{}
                % the subject of the 12th lecture
\caltextnext{\S9.9\\Tending and rending}
\end{calendar}

```

Figure 2: L<sup>A</sup>T<sub>E</sub>X input for Figure 1

are numbered and can be referred to by their numbers. The command `\weeklytext` `\weeklytext{text}` specifies that the indicated text should appear every week in the box for that day. The *text* may include T<sub>E</sub>X commands; for example the following could be used to number regular Monday quizzes:

```
\weeklytext{\stepcounter{quiznumber} Quiz~\arabic{quiznumber}}
```

**Options for a specific day** are specified by the command `\options{day}{option list}`. The *option list* argument is exactly the same as for `\calday`. The options added by `\options` are executed after those for `\calday`, and thus may be used to modify or nullify for a specific day the general instructions specified for a particular day of the week. If there no quiz is planned for Halloween then `\options{10/31/94}{\weeklytext{}}` could be used to suppress printing the usual weekly message.

There are two ways of specifying the *day* argument of `\options`. First, the date may be specified, in the format `m/d/y`; for example

```
\options{11/11/94}{\noclass}
```

specifies, that November 11th, Veterans's day, is not a class day. The second method of specifying the date is by its number:

```
\options{C6}{\weeklytext{}}
```

would suppress the normal text on the sixth day of class. Of course, only days specified as class days can be addressed in this style.

**Inserting Text.** There is a similar provision for text to be printed on a specific day, using the command `\caltext`. The command

```
\caltext{10/31/94}{Halloween\No Quiz!}
```

will print the indicated message on October 31, and `\caltext{C6}{Hour Exam}` will print the indicated message on the sixth class day.

Two extra commands are provided to simplify the job of entering text for consecutive class days, as in specifying the lecture topic or homework for each day. The command `\caltexton{class day}{text}` specifies a starting day and inserts the indicated text on that day. The command `\caltextnext{text}` can then be used to print text on successive class days. The command `\caltextnext{}`, with an empty argument, may be used to skip days.

## 1.1 Modifying the style of the calendar

Several parameters and commands will allow some modification of the style of the entire calendar. The size of the calendar is specified by `\calboxdepth`, which specifies the minimum height of the box for each day, and `\calwidth`

which specifies the width of the calendar. The defaults are 1 inch for `\calboxdepth` and `\textwidth` for `\calboxwidth`.

The printing of the date and classnumber in each box is done by the commands `\calprintdate` and `\calprintclass`. The default definitions of these macros are as follows:

`\calprintdate`

```
1 \newcommand{\calprintdate}{%
2   \ifnewmonth\framebox{\monthname\ \ordinaldate}%
3   \else \ordinaldate\fi
4 }
```

`\calprintclass`

```
5 \newcommand{\calprintclass}{\textbf{\small\theclassnum}}
```

They may be changed with `\renewcommand`.

By default the calendar will automatically be split over several pages. This can be avoided by putting the entire calendar in a `\vbox`. It will normally also be necessary to change `\calboxdepth` so that the calendar will fit on one page.

## 2 The Code

```
6 \ProvidesPackage{termcal}[\filedate\space\fileversion\space
7   Latex2e package to print a Term calendar]
8 \NeedsTeXFormat{LaTeX2e}
```

We use the package `longtable` so that the calendar can be split over several pages if desired.

```
9 \RequirePackage{longtable}
10 \RequirePackage{ifthen}
```

Parameters determining the size of the calendar.

```
11 \newlength{\calboxdepth}\setlength{\calboxdepth}{1in}
12 \newlength{\calwidth}\setlength{\calwidth}{\textwidth}
13 \newlength{\ca@boxwidth} %% set by \endcalendar
```

The following parameters are used to control the construction of the calendar.

`\ca@dpw`

The number of days used in a week.

```
14 \newcounter{ca@dpw}
```

`\ca@numwks` The number of weeks in the calendar.  
15 `\newcounter{ca@numwks} %% weeks in calendar`

`\ca@wknum` Used to keep track of the current week.  
16 `\newcounter{ca@wknum}`

`\ca@doaweek` Two token boxes used to build up the basic contents of the calendar. `\ca@doaweek` is built up by `\calday`, and prints a typical week. It will be filled up by the macros `\calday` and `\skipday`.  
17 `\newtoks\ca@doaweek`

`\ca@doweeks` This will be essentially `ca@numwks` copys of `\ca@doaweek`.  
18 `\newtoks\ca@doweeks`

`\ca@colhead` The column headings which appear at the top of every page is collected in the token box `\ca@colhead` by `\calday`. The switch `\ifca@thead` is set to true if any such column heads are specified.  
19 `\newtoks\ca@colhead`  
20 `\newif\ifca@thead`

`calendar` This is the basic environment. The `\calendar` command only saves the parameters and initializes some counters.  
21 `\newenvironment{calendar}[2]%`  
22 `{%`  
23 `\setcounter{ca@numwks}{#2}`  
24 `\setdate{#1}`  
25 `\setcounter{ca@dpw}{0}`  
26 `\setcounter{classnum}{1}`  
27 `}`

`\endcalendar` The calendar is actually created in the code for `\endcalendar`. It will be printed as a longtable.  
28 `{`  
Since the `longtable` and `tabular` environments don't work well with loops in their body, we will build up the body in a token box, `\ca@doweeks`.

`\ca@doweeks`  
29 `\ifca@thead\ca@doweeks{\the\ca@colhead\endhead\hline\hline}\fi`  
30 `\setcounter{ca@wknum}{0}`

```

31 \whiledo{\value{ca@wknum}<\value{ca@numwks}}%
32     {\stepcounter{ca@wknum}%
33     \addtotoks{\ca@doweeks}{\the\ca@doaweek\\hline}}

```

Now we calculate the widths of the boxes, using a formula from the Latex Companion.

`\ca@boxwidth`

```

34 \ca@boxwidth=\calwidth
35 \divide\ca@boxwidth by \c@ca@dpw\relax
36 \advance\ca@boxwidth by -2\tabcolsep\relax
37 \setlength\@tempdima\arrayrulewidth\relax
38 \multiply\@tempdima\c@ca@dpw\relax
39 \advance\@tempdima\arrayrulewidth\relax
40 \divide\@tempdima\c@ca@dpw\relax
41 \advance\ca@boxwidth by -\@tempdima\relax

```

Now we use the `longtable` environment to print out the calendar.

```

42 \begin{longtable}[1]
43 {|\*{\theca@dpw}{p{\ca@boxwidth}}|@{}}%
44     \hline
45     \the\ca@doweeks
46 \end{longtable}
47 }

```

`\addtotoks` The command `\addtotoks{token box}{new tokens}` is used to add new tokens at the end of *token box*.

```
48 \newcommand\addtotoks[2]{#1\expandafter{\the#1#2}}
```

`\calday` The command `\calday` works by adding commands to the token box `\ca@doaweek`, a copy of which is inserted into `\ca@doweeks` for each week in the calendar. inserted If column headings are specified, then they are put into the token box `\ca@colhead`.

```

49 \ca@doaweek={\stepcounter{ca@wknum}\ignorespaces}
50 \newcommand\calday[2] []{\stepcounter{ca@dpw}%
51     \ifca@fday\addtotoks\ca@doaweek{&}\addtotoks\ca@colhead{&}\fi
52     \addtotoks\ca@doaweek{\ca@doaday{#2}}
53     \def\@tempa{#1}\ifx\@tempa\@empty
54         \else\addtotoks\ca@colhead{\strut\scshape\centering #1}\ca@theadtrue\fi
55     \ca@fdaytrue
56 }

```

`\ifca@fday` True if this is not the first day of the week. This determines whether `&` needs to be added as a separator.

```
57 \newif\ifca@fday
```

`\skipday` The command `\skipday` simply advances the date without printing anything in the calendar.

```
58 \newcommand\skipday{\addtotoks\ca@doaweek{\advancedate}}
```

After a couple of preliminaries, we will define the command `\ca@doaday` which actually prints out the text for each day of the calendar.

`c@classnum` This is the counter used to keep track of class days. It is initialized to 1 in the beginning of the `calendar` environment.

```
59 \newcounter{classnum}
```

`\ca@normbs` The meaning of `\` is changed by the `longtable` environment. We save its standard meaning so that it can be used in the text to be printed in the calendar boxes.

```
60 \let\ca@normbs=\
```

`\ca@doaday` The command `\ca@doaday` does the actual printing of the contents of the box for each day. First the options are read, in the following order: options specified in the argument to `\calday`, then options specified by date, and finally options specified by classday.

```
61 \newcommand\ca@doaday[1]{
62   \hbox{\vrule depth \calboxdepth height Opt width Opt\vtop{
63     #1%                               %options specified by |\calday|
64     \csname\curdate options\endcsname% % options specified by date
65     \ifclassday\csname C\theclassnum options\endcsname\fi %by classnumber
```

Then the heading is printed.

```
66   \hbox to \hsize{\calprintdate\hfill\ifclassday\calprintclass\fi}
67   \vspace{2pt}
```

Now we are ready to print the text. We do it inside a group in which the normal meaning of `\` is restored.

```
68     \begingroup
69     \let\=\ca@normbs
70     \raggedright
71     \sloppy
72
```



```

73     \the\weeklytext\par
74     \csname\curdate text\endcsname
75     \ifclasday\csname C\theclassnum text\endcsname
76         \stepcounter{classnum}\fi
77 \endgroup
78 }}

```

Finally we advance the date. The command `\advancedate` will set `\newmonthtrue` if appropriate.

```

79 \global\newmonthfalse
80 \advancedate
81 }

```

## 2.1 Options

The options and text for the individual days are stored in macros, with names of the form `\dateoptions` or `\datetext`.

`\ca@addmacro` `\ca@addmacro` is like `\ca@addtoks`, except it works on macros instead of token boxes. The first argument is the name of a macro (without the backslash) and the second is a sequence of tokens to be added to its definition. The definition is taken from the *T<sub>E</sub>Xbook*, exercise 20.15. Note that the spaces in the last line are essential.

```

82 \long\def\ca@addmacro#1#2{
83 \expandafter\ifx\csname#1\endcsname\relax%
84     \expandafter\def\csname#1\endcsname{#2}
85 \else
86 \toks0=
87     \expandafter\expandafter\expandafter{\csname#1\endcsname}
88 \toks2={#2}
89 \expandafter
90 \edef\csname#1\endcsname{\the\toks0 \the\toks2 }\fi}

```

`\options` `\options#1#2` adds the tokens in the second argument to the macro with the name `\csname #1options\endcsname`.

```

91 \newcommand\options[1]{\ca@addmacro{#1options}}

```

Now the code for the various options.

`\ifclasday` A switch determines which days are class days.  
`\clasday`  
`\noclasday`

```

92 \newif\ifclassday
93 \newcommand{\classday}{\classdaytrue}
94 \newcommand{\noclassday}{\classdayfalse}

```

`\weeklytext` This token box holds the standard text for each day of the week. It is filled and immediately used by `\ca@doaday`, inside the hbox containing each calendar day.

```

95 \newtoks\weeklytext

```

`\caltext` This macro works like `\options` to save the text for a specific day, saving the text in a macro with the name `\csname #1text\endcsname`. We add `\par` after the text, so successive texts for the same day start on separate lines. We use `\par` instead of `\\` since it is harmless if it is unneeded (but requires that `\ca@addmacro` be long).

```

96 \newcommand\caltext[2]{\ca@addmacro{#1text}{#2\par}}

```

`\c@textdaycount` The commands `\caltexton` and `\caltextnext` use `\caltext` with the day determined by the counter `textdaycount`.

`\caltexton`

`\caltextnext`

```

97 \newcounter{textdaycount}\setcounter{textdaycount}1
98 \newcommand\caltexton[2]{\setcounter{textdaycount}{#1}
99     \caltext{C#1}{#2}}
100 \newcommand\caltextnext[1]{\advance\c@textdaycount by 1
101     \caltext{C\thetextdaycount}{#1}}

```

## 2.2 Macros concerned with date calculations

Now we have a selection of macros which are concerned with calculating the dates in the calendar.

`\c@date` Counter for the day of the month.

```

102 \newcounter{date}

```

`\c@month` Counter for month (January = 1)

```

103 \newcounter{month}

```

`\c@year` Counter for year.

```

104 \newcounter{year}

```

`\curdate` Print out the date.

```

105 \newcommand\curdate{\arabic{month}/\arabic{date}/\arabic{year}}

```

```

\monthname Print the name of the Month.
106 \newcommand\monthname{\ifcase\c@month\or Jan\or Feb\or Mar\or Apr%
107     \or May\or June\or July\or Aug\or Sep\or Oct%
108     \or Nov\or Dec\fi}

\advancedate Move the date forward by one day.
109 \newcommand\advancedate{\stepcounter{date}
110     \ifnum\thedate>\monthlength\relax
111     \addtocounter{date}{-\monthlength}\advancemonth\fi}

\ifnewmonth True if no days have been printed during the current month.
112 \newif\ifnewmonth\newmonthtrue

\advancemonth
113 \newcommand\advancemonth{%
114     \global\newmonthtrue\stepcounter{month}
115     \ifnum\c@month>12
116     \stepcounter{year}\setleap\setcounter{month}1\fi}

\ifleap True if the year is a leap year.
117     \newif\ifleap

\setleap Determine whether the year is a leap year. This is valid until 2100, thanks
to the 400 year rule.
118 \newcommand\setleap{%
119     \@tempcnta=\c@year
120     \divide\@tempcnta by 4 \multiply\@tempcnta by 4
121     \ifnum\@tempcnta=\c@year\global\leaptrue
122     \else\global\leapfalse\fi}

\monthlength Determine the number of days in the current month.
123 \newcommand\monthlength{%
124     \ifcase\c@month\or31\or\ifleap29\else28\fi
125     \or31\or30\or31\or30\or31\or31\or30\or31\or30\or31\fi%
126     \relax}

\setdate Take argument in the form m/d/y and set the counters month, date and year.
127 \newcommand\setdate[1]{\setdate@#1!}
128 \def\setdate@#1/#2/#3!{
129     \setcounter{month}{#1}

```

```

130     \setcounter{date}{#2}
131     \setcounter{year}{#3}
132     \global\newmonthtrue\setleap}

```

`\ordinaldate` Print the day of the month as an ordinal.

```
133 \newcommand\ordinaldate{\ordinal{\c@date}}
```

`\ordinal` Print the contents of a register as ordinal number.

```

134 \newcommand\ordinal[1]{%
135     \let\last@=\relax\let\last@@=\relax
136     \expandafter\@rd\the#1x}
137 \newcommand\@rd[1]{\ifx#1x\if\last@@@1th\else\@rdend{\last@}\fi\else
138     \let\last@@=\last@\def\last@{#1}#1\expandafter\@rd\fi}
139 \newcommand\@rdend[1]{\ifcase#1 th\or st\or nd\or rd\else th\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	C	
<code>\@empty</code> . . . . . 53	<code>\c@ca@dpw</code> .. 35, 38, 40	<code>\ca@normbs</code> . . . <u>60</u> , 69
<code>\@rd</code> . . . . . 136–138	<code>\c@classnum</code> . . . . . <u>59</u>	<code>\ca@numwks</code> . . . . . <u>15</u>
<code>\@rdend</code> . . . . . 137, 139	<code>\c@date</code> . . . . . <u>102</u> , 133	<code>\ca@wknum</code> . . . . . <u>16</u>
<code>\@tempdima</code> . . . . . 37–41	<code>\c@month</code> . . . . .	<code>\calboxdepth</code> 4, 11, 62
	<u>103</u> , 106, 115, 124	<code>\calday</code> . . . . . 2, <u>49</u> , 63
<code>\sq</code> . . . . . 2	<code>\c@textdaycount</code> .. <u>97</u>	calendar (environ- ment) . . . . . 1, <u>21</u>
	<code>\c@year</code> . <u>104</u> , 119, 121	<code>\calprintclass</code> . 5, 66
<b>A</b>	<code>\ca@addmacro</code> <u>82</u> , 91, 96	<code>\calprintdate</code> .. <u>1</u> , 66
<code>\addtocounter</code> . . . . . 111	<code>\ca@boxwidth</code> 13, <u>34</u> , 43	<code>\caltext</code> 4, <u>96</u> , 99, 101
<code>\addtotoks</code> . . . . . 33,	<code>\ca@theadtrue</code> . . . . . 54	<code>\caltextnext</code> .. 4, <u>97</u>
<u>48</u> , 51, 52, 54, 58	<code>\ca@colhead</code> <u>19</u> , 29, <u>49</u>	<code>\caltexton</code> . . . . . 4, <u>97</u>
<code>\advance</code> 36, 39, 41, 100	<code>\ca@doaday</code> . . . . . 52, <u>61</u>	<code>\calwidth</code> .. 4, 12, 34
<code>\advancedate</code> . . . . .	<code>\ca@doaweek</code> . . . . .	<code>\classday</code> . . . . . 2, <u>92</u>
. . . . . 58, 80, <u>109</u>	. . . . . <u>17</u> , 33, <u>49</u> , 58	<code>\classdayfalse</code> . . . . . 94
<code>\advancemonth</code> 111, <u>113</u>	<code>\ca@doweeks</code> <u>18</u> , <u>29</u> , 45	<code>\classdaytrue</code> . . . . . 93
<code>\arabic</code> . . . . . 105	<code>\ca@dpw</code> . . . . . <u>14</u>	<code>\curdate</code> .. 64, 74, <u>105</u>
<code>\arrayrulewidth</code> 37, 39	<code>\ca@fdaytrue</code> . . . . . 55	

<p><b>E</b></p> <p><code>\endcalendar</code> . 13, <u>28</u></p> <p>environments:</p> <p>  <code>calendar</code> . . . . . <u>21</u></p> <p>environments:calendar</p> <p>  <code>calendar</code> . . . . . <u>1</u></p> <p><b>I</b></p> <p><code>\ifca@thead</code> .. <u>19</u>, 29</p> <p><code>\ifca@fday</code> ... 51, <u>57</u></p> <p><code>\ifclassday</code> . . . . .</p> <p>  ... 65, 66, 75, <u>92</u></p> <p><code>\ifleap</code> . . . . . <u>117</u>, 124</p> <p><code>\ifnewmonth</code> .. 2, <u>112</u></p> <p><b>L</b></p> <p><code>\leapfalse</code> . . . . . 122</p>	<p><code>\leaptrue</code> . . . . . 121</p> <p><b>M</b></p> <p><code>\monthlength</code> . . . . .</p> <p>  . . . . . 110, 111, <u>123</u></p> <p><code>\monthname</code> ... 2, <u>106</u></p> <p><b>N</b></p> <p><code>\newmonthfalse</code> ... 79</p> <p><code>\newmonthtrue</code> . . . . .</p> <p>  . . . . . 112, 114, 132</p> <p><code>\noclassday</code> ... 2, <u>92</u></p> <p><b>O</b></p> <p><code>\options</code> . . . . . 4, <u>91</u></p> <p><code>\ordinal</code> . . . . . 133, <u>134</u></p> <p><code>\ordinaldate</code> 2, 3, <u>133</u></p>	<p><b>S</b></p> <p><code>\setdate</code> . . . . . 24, <u>127</u></p> <p><code>\setdate@</code> ... 127, 128</p> <p><code>\setleap</code> 116, <u>118</u>, 132</p> <p><code>\skipday</code> . . . . . <u>1</u>, <u>58</u></p> <p><b>T</b></p> <p><code>\theca@dpw</code> . . . . . 43</p> <p><code>\theclassnum</code> 5, 65, 75</p> <p><code>\thedata</code> . . . . . 110</p> <p><code>\thetextdaycount</code> 101</p> <p><b>W</b></p> <p><code>\weeklytext</code> 3, 73, <u>95</u></p>
--	--	---