

User's Guide to the `evweek` class*

Emanuele Vicentini
(`emanuelevicentini at yahoo dot it`)

2005/01/23

Abstract

This class, a simple spinn-off of the official `weekly`, incorporates small bug-fixes and some extensions. To use it you have to install the L^AT_EX Calendar Bundle and, optionally, the `evautofl` package.

Contents

1	Brief Introduction	1
2	The weekly calendar	2
2.1	Original Calendar Options	2
2.2	New Calendar Options	3
2.3	Fixes	4
2.4	New Class Options	4
3	The Class File	4
3.1	Preliminaries	4
3.2	Options	4
3.3	Option Postprocessing	5
3.4	Calendar Setup	5
3.5	Calendar Definition	7
	References	12

1 Brief Introduction

For the original documentation see the files `calguide.tex` and `weekly.dtx` distributed in the L^AT_EX Calendar Bundle.

This manual is typeset according to the conventions of the L^AT_EX DOCSTRIP utility which enables the automatic extraction of the L^AT_EX macro source files [GMS94].

*This file is version number 0.4; it was last revised on 2005/01/23.

2 The weekly calendar

The `evweek` class prints weekly calendars for use with a Filofax™ or other personal planner. Pages are framed with cut-lines, and punch-marks show where to cut holes at the edge of the page. For each date, the style initially attempts to set the event and date text as a single column. If the text overflows the box for that date, the style will attempt to recover by resetting the text as two columns.

It has the following special features and limitations:

- The text of the calendar is always rotated, using the `lscap` package from the `graphics` bundle. As a result, calendars generated using this package can only be printed using `pdflatex`, `dvips` or other PostScript™ print drivers supported by the `graphics` bundle.
- If the `graphics` bundle has not been configured for your system, you can specify the driver and other options for its use by giving L^AT_EX options to the class when it is loaded.
- While the paper size can be specified using options like `a4paper`, this will have no effect on the calendar size: it is governed by calendar options only.
- Documents created with this style should contain nothing other than a calendar environment.

2.1 Original Calendar Options

The following formatting options are supported by the `weekly` environment¹:

`firstday` This is the starting day of the calendar, expressed as an integer. Friday (the default) is 5.

`punchcluster` Most filofaxes have two or more clusters or groups of punches to hold the pages in place. This option sets the number of punches in each cluster. The default is three.

`intraspace` This governs the space between punchouts within a group. The default is 19.25mm.

`punchgroups` This option sets the number of groups of punches. The default is two groups.

`interspace` This option sets the distance between the groups of punches. The default is 51.25mm.

`pageheight` This fixes the height of a filofax page (not the physical paper on which it is printed). The default is 172mm.

`pagewidth` This fixes the width of an individual page. The physical printed area will be twice this figure. The default is 95mm, for a 190mm printed area.

`grip` This adjusts the distance from the edge of the page to the outer edge of the punchouts. Default is 5mm.

¹All but the first one are provided by the `autofilo` package (see the L^AT_EX Calendar Bundle) or the `evautofl` package.

punchmargin This adjusts the distance from the edge of the text to the inner edge of the punchouts. Default is 2mm.

punchpoints Size, in points, of punchouts². Defaults to 15.

topspace Gap between top of filofax page and top of text page.

bottomspace Gap between bottom of filofax page and bottom of text page.

jawsbread If set to a positive length, this places a set of rules the width of the punchmarks on either side of each punchhole, centered on its center and spread the distance specified. This can be useful as a guide with some one-hole punches that are designed to be used “blind”.

jawline Sets the width of the lines used to make jawmarks. Default value is 0.4 pt.

2.2 New Calendar Options

The following new formatting options are supported by the `weekly` environment as provided by this class only:

titletype This set the default typeface for the titlebox of each entry; default is `\bfseries`.

daynumbertype This set the typeface for the day number, allowing a finer control over the typesetting of each box; default is empty.

daynametype This set the typeface for the day name, allowing a finer control over the typesetting of each box; default is empty.

monthtype This set the typeface for month name, allowing a finer control over the typesetting of each box; default is empty.

yeartype This set the typeface for year number, allowing a finer control over the typesetting of each box; default is empty.

intraleskip This set the distance between lines that appears in every day’s box; set it to negative values to disable line printing; default is `-1 pt`.

bottomlineskip This set a small gap between the last line that will eventually appear in every day’s box and the bottom of that same box; default is `0 pt`.

ruleheight This set the lines’ height; default is `0.1 pt`.

Each of these parameters can be set independently, but the `titletype` option has an overall effect. For example:

```
titletype=\itshape, monthtype=\bfseries
```

will typeset the whole titlebox with an italic shape font and *only* the month name using a bold expanded, italic font (if available).

²Please, note that the original documentation reads `punchsize` but the option name is really `punchpoints`. Also, take care to write no dimension specification after the number.

2.3 Fixes

I should not call them *real* bug-fixes, but I have changed some bits of code that handle the layout of the titlebox of each day.

The original implementation just put the titlebox's elements in a `\TeX` box and then built a little sandwich with two `\hrule`; this has the unpleasent effect that the titleboxes can be of different height, so I've put the titleboxes in a `\vbox` of `\baselineskip` height.

2.4 New Class Options

traditional The `evweek` class needs `autofilo` or `evautofl` package to draw the page frame; the default behaviour is to use `evautofl` but if the `traditional` option is passed to `evweek` or `evautofl` is not available, then it will revert to use the old `autofilo` package.

compact With this option, `evweek` puts a whole week on a single framed page, without any empty scratching box, starting from Monday (starting day can be customized using `firstday`); the original input for this option came from Luis Rivera.

ferial With this option, `evweek` puts Monday to Wednesday on the left hand pages and Thursday to Sunday on the right hand ones, with Saturday and Sunday taking half the space of the other days and no empty scratching box³; the original request for this option came from Geert Kloosterman.

3 The Class File

3.1 Preliminaries

Use `LATEX 2ε`, and tell the user who we are.

```
1 (*class)
2 \NeedsTeXFormat{LaTeX2e}[1995/06/01]
3 \ProvidesClass{evweek}[2005/01/23 v0.4 evWeekly Calendar]
```

Define some more variables and switches for use in the body of the style.

```
4 \newif\ifwk@overthetop
5 \newcount\wk@firstday
6 \newcount\wk@tempcount
7 \newlength\wk@itemheight
8 \newif\ifwk@traditional
9 \newif\ifwk@compact
10 \newif\ifwk@ferial
11 \newskip\wk@intralineskip
12 \newskip\wk@bottomlineskip
13 \newlength\wk@ruleheight
```

3.2 Options

Things like page and type size go to the article class.

³Customization of starting day via `firstday` is not taken into account.

```

14 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
15 \DeclareOption{traditional}{\wk@traditionaltrue}
16 \DeclareOption{compact}{\wk@compacttrue\wk@ferialfalse}
17 \DeclareOption{ferial}{\wk@ferialtrue\wk@compactfalse}

```

We use the generic language options, the same as other parts of the system.

```

18 \input{calopts.cfg}
19 \InputIfFileExists{dates.cfg}{}{}
20 \ProcessOptions

```

3.3 Option Postprocessing

Now that the options lists are available to us, we can start loading packages. The following packages are always loaded.

```

21 \LoadClass{article}
22 \RequirePackage{calendar}
23 \RequirePackage{lscapex}
24 \RequirePackage{multicol}

```

If `evautofl` is not available, we simulate the effect of `traditional` to keep going.

```

25 \IfFileExists{evautofl.sty}{\relax}{\wk@traditionaltrue}
26 \ifwk@traditional
27 \RequirePackage{autofilo}
28 \else
29 \RequirePackage{evautofl}
30 \fi

```

It is now time to do the final page formatting.

```

31 \AtBeginDocument{%
Eliminate page numbers and indenting.
32 \pagestyle{empty}
33 \parindent=0pt
34 \multicolsep=0pt}

```

3.4 Calendar Setup

Define some key and their default values for use by the calendar environment.

```

35 \define@key{opt}{firstday}{%
36 \wk@firstday=#1}
37 \ifwk@compact
38 \wk@firstday=1
39 \else\ifwk@ferial
40 \wk@firstday=4
41 \else
42 \wk@firstday=5
43 \fi\fi
44 \define@key{opt}{titletype}{%
45 \def\wk@titletype{#1}}
46 \def\wk@titletype{\bfseries}
47 \define@key{opt}{daynumbertype}{%
48 \def\wk@daynumbertype{#1}}
49 \def\wk@daynumbertype{}
50 \define@key{opt}{daynametype}{%
51 \def\wk@daynametype{#1}}

```

```

52 \def\wk@daynametype{}
53 \define@key{opt}{monthtype}{%
54   \def\wk@monthtype{#1}}
55 \def\wk@monthtype{}
56 \define@key{opt}{yeartype}{%
57   \def\wk@yeartype{#1}}
58 \def\wk@yeartype{}
59 \define@key{opt}{intralineskip}{%
60   \wk@intralineskip=#1}
61 \wk@intralineskip=-\p@
62 \define@key{opt}{bottomlineskip}{%
63   \wk@bottomlineskip=#1}
64 \wk@bottomlineskip=\z@
65 \define@key{opt}{ruleheight}{%
66   \wk@ruleheight=#1}
67 \wk@ruleheight=0.1pt

```

Internal re-definition of \hrulefill with variable height.

```

68 \def\wk@hrulefill{\leavevmode\leaders\hrule height\wk@ruleheight
69   \hfill\kern\z@}

```

We declare eight boxes (or just seven if compact or ferial options are in place), one for each day of the week plus an extra.

```

70 \wk@tempcount=0
71 \expandafter\newsavebox\csname wk@box\the\wk@tempcount\endcsname
72 \ifwk@compact
73   \loop
74     \advance\wk@tempcount by 1
75     \ifnum\wk@tempcount<7
76       \expandafter\newsavebox\csname wk@box\the\wk@tempcount\endcsname
77     \repeat
78 \else\ifwk@ferial
79   \loop
80     \advance\wk@tempcount by 1
81     \ifnum\wk@tempcount<7
82       \expandafter\newsavebox\csname wk@box\the\wk@tempcount\endcsname
83     \repeat
84 \else
85   \loop
86     \advance\wk@tempcount by 1
87     \ifnum\wk@tempcount<8
88       \expandafter\newsavebox\csname wk@box\the\wk@tempcount\endcsname
89     \repeat
90 \fi\fi

```

We need a macro to lay the boxes out on a page. We define this now, with parameterized measurements that will be determined by the settings given to the environment within which this expands.

```

91 \def\wk@spitoutboxes{%
92   \wk@tempcount=-1
93   \ifwk@compact
94     \loop
95       \advance\wk@tempcount by 1
96       \ifnum\wk@tempcount<7
97         \hbox to\linewidth{%

```

```

98     \expandafter\usebox\csname wk@box\the\wk@tempcount\endcsname}%
99     \repeat
100  \else\ifwk@ferial
101     \loop
102     \advance\wk@tempcount by 1
103     \ifnum\wk@tempcount<7
104     \hbox to\linewidth{%
105     \expandafter\usebox\csname wk@box\the\wk@tempcount\endcsname}%
106     \repeat
107  \else
108     \loop
109     \advance\wk@tempcount by 1
110     \ifnum\wk@tempcount<3
111     \hbox to\linewidth{%
112     \expandafter\usebox\csname wk@box\the\wk@tempcount\endcsname}%
113     \repeat
114     \hbox to\linewidth{%
115     \vbox to\wk@itemheight{%
116     \hrule width\linewidth height0.4pt depth0pt
117     \hbox to\linewidth{\hfil}%
118     \vfill}}}%
119     \wk@tempcount=2
120     \loop
121     \advance\wk@tempcount by 1
122     \ifnum\wk@tempcount<7
123     \hbox to\linewidth{%
124     \expandafter\usebox\csname wk@box\the\wk@tempcount\endcsname}%
125     \repeat
126  \fi\fi}

```

We define a macro that calculates the height of the date boxes. The value of `\normallineskip` sneaks in between stacked `\hboxes`, so we need to account for that.

```

127 \def\wk@getitemheight{%
128   \setlength{\wk@itemheight}{\textheight}
129   \addtolength{\wk@itemheight}{-4pt}
130   \addtolength{\wk@itemheight}{-5\normallineskip}
131   \ifwk@compact
132     \divide\wk@itemheight by 7
133   \else\ifwk@ferial
134     \divide\wk@itemheight by 3
135   \else
136     \divide\wk@itemheight by 4
137   \fi\fi}%

```

3.5 Calendar Definition

The calendar definition itself consists of a single command and its associated arguments.

```
138 \newcalendar%
```

We give the environment a name.

```
139 {weekly}
```

We are producing DVI, so we use the standard macro for dropping text into the hopper.

```
140 {\cal@insert}
```

Starting day for the calendar is not fixed here, because we do not pad with blank entries. Season to taste. Use integers 0 through 6. A value over 7 means no padding.

```
141 {8}
```

The length of a group is one week.

```
142 {7}
```

The length of a subgroup is one week.

```
143 {7}
```

We do not need delimiters between output items, since we're just building boxes.

```
144 {\relax}
```

Increment the date counter with every item.

```
145 {I}
```

Define the header.

```
146 {\expandafter\autofilo\expandafter{\cal@optionslist}\wk@getitemheight}
```

We define the end of the calendar. This spits out all of the boxes one last time and closes the page formatting environment.

```
147 {\wk@spitoutboxes\endautofilo}
```

We define the way to open a new group. This spits out all of the boxes built for a page of text.

```
148 {\wk@spitoutboxes}
```

We define a way to end a subgroup. In this case, we just relax.

```
149 {\relax}
```

We set up to require a month and day in the environment typed into the user's document.

```
150 {\dates@requiremonthtrue\dates@requiredaytrue}
```

The next section runs after the calendar dates have been scanned from the environment, but before the table is opened.

We step back, if necessary, from the designated first date to reach the starting day for pages. (Default is Friday.)

```
151 {\global\dates@date=\cal@range@start
```

```
152 \prevday\wk@firstday
```

```
153 \cal@range@start=\dates@date
```

```
154 \caldate
```

We tell the user where we are starting.

```
155 \message{^^JStart: \theshortweekday\space
```

```
156 \theday\space\theshortmonth\space\theyear}%
```

Then we jump forward, again if necessary, to the next day just before the starting day of the week.

```
157 \global\dates@date=\cal@range@end
```

```
158 \caldate
```

```
159 \advance\wk@firstday by -1
```

```
160 \ifnum\wk@firstday=7
```



```

161     \wk@firstday=0
162     \fi
163     \nextday\wk@firstday
164     \caldate
165     \cal@range@end=\dates@date

```

We let the user know where we plan to stop.

```

166     \message{^^JEnd: \theshortweekday\space
167         \theday\space\theshortmonth\space\theyear^^J}%

```

We jump back to the starting date and reset the human-side values.

```

168     \global\dates@date=\cal@range@start
169     \caldate}

```

Events are delimited by \par.

```

170     {\par}

```

The next argument is dropped on DVI for each item. We use it here to fetch the text of appointments and events to the standard list macros, using the standard text-fetching commands. Each generates a stack delimited by the phrase that was set above.

```

171     {\cal@get@appointments\cal@get@events}

```

This sets \cal@entry@text. It is executed for every event or appointment in the database that applies to a given item. We differentiate between whole-day events, pinpointed events and ranged events.

```

172     {\ifnum\dates@time@start>0
173         \ifnum\dates@time@start=\dates@time@end
174             \protected@edef\cal@entry@text{%
175                 \thetimestart\space
176                 \cal@textcolor{\calcolor}{\theshorttext}}
177         \else
178             \protected@edef\cal@entry@text{%
179                 \thetimestart--\thetimeend\space
180                 \cal@textcolor{\calcolor}{\theshorttext}}
181         \fi
182     \else
183         \protected@edef\cal@entry@text{%
184             \cal@textcolor{\calcolor}{\theshorttext}}
185     \fi}

```

Finally we tell the style how to create the text for each item. We stash the text in each of the numbered boxes, for extraction at the end of each group and at the end of the run. In the event of an overfull \vbox, we retry with two columns of text in a last-ditch attempt to squeeze everything in.

```

186     {\global\wk@overthetopfalse
187     \ifwk@ferial
188         \ifcase\cal@subgroup@count
189             \def\wk@itemheight@scalefactor{1}%
190         \or
191             \def\wk@itemheight@scalefactor{1}%
192         \or
193             \def\wk@itemheight@scalefactor{0.5}%
194         \or
195             \def\wk@itemheight@scalefactor{0.5}%
196         \else

```

```

197     \def\wk@itemheight@scalefactor{1}%
198     \fi
199     \else
200     \def\wk@itemheight@scalefactor{1}%
201     \fi
202     \expandafter\sbox\csname wk@box\the\cal@subgroup@count\endcsname{%
203     \vbox to\wk@itemheight@scalefactor\wk@itemheight{%
204     \hbox{%
205     \parbox{\linewidth}{%
206     \vbox to\baselineskip{%
207     \hrule width\linewidth
208     \vskip 1pt
209     \ifwk@compact
210     {\wk@titletype{\wk@daynametype\theweekday}\hfill
211     {\wk@daynumbertype\theday}\space
212     {\wk@monthtype\themoth}\space
213     {\wk@yeartype\theyear}%
214     }\par
215     \else\ifwk@ferial
216     \ifnum\cal@subgroup@count<4
217     {\wk@titletype{\wk@daynametype\theweekday}\hfill
218     {\wk@daynumbertype\theday}\space
219     {\wk@monthtype\themoth}\space
220     {\wk@yeartype\theyear}%
221     }\par
222     \else
223     {\wk@titletype{\wk@daynumbertype\theday}\space
224     {\wk@monthtype\themoth}\space
225     {\wk@yeartype\theyear}\hfill
226     {\wk@daynametype\theweekday}%
227     }\par
228     \fi
229     \else
230     \ifnum\cal@subgroup@count<3
231     {\wk@titletype{\wk@daynametype\theweekday}\hfill
232     {\wk@daynumbertype\theday}\space
233     {\wk@monthtype\themoth}\space
234     {\wk@yeartype\theyear}%
235     }\par
236     \else
237     {\wk@titletype{\wk@daynumbertype\theday}\space
238     {\wk@monthtype\themoth}\space
239     {\wk@yeartype\theyear}\hfill
240     {\wk@daynametype\theweekday}%
241     }\par
242     \fi
243     \fi\fi
244     \vfill
245     }%         vbox
246     \hrule width\linewidth
247     \vskip 1pt
248     \raggedright
249     \small
250     \ifcat$\theevents$\%

```

```

251         \else
252             \theevents\par
253         \fi
254         \ifcat$\theappointments$\%
255         \else
256             \theappointments
257         \fi
258     }%           parbox
259     }%           hbox
260     \leaders\vbox{\vskip\wk@intralineskip\wk@hrulefill}\vfill
261     \vskip\wk@bottomlineskip
262 }%           vbox
263 \ifnum\badness=1000000
264     \global\wk@overthetoptrue
265 \fi
266 }%           sbox
267 \ifwk@overthetop
268     \message{Resetting this date in two-columns.}%
269     \expandafter\sbox\csname wk@box\the\cal@subgroup@count\endcsname{%
270         \vbox to\wk@itemheight@scalefactor\wk@itemheight{%
271             \hbox{%
272                 \parbox{\linewidth}{%
273                     \vbox to\baselineskip{%
274                         \hrule width\linewidth
275                         \vskip 1pt
276                         \ifwk@compact
277                             {\wk@titletype{\wk@daynametype\theweekday}\hfill
278                             {\wk@daynumbertype\theday}\space
279                             {\wk@monthtype\themonth}\space
280                             {\wk@yeartype\theyear}%
281                             }\par
282                         \else
283                             \ifnum\cal@subgroup@count<3
284                                 {\wk@titletype{\wk@daynametype\theweekday}\hfill
285                                 {\wk@daynumbertype\theday}\space
286                                 {\wk@monthtype\themonth}\space
287                                 {\wk@yeartype\theyear}%
288                                 }\par
289                             \else
290                                 {\wk@titletype{\wk@daynumbertype\theday}\space
291                                 {\wk@monthtype\themonth}\space
292                                 {\wk@yeartype\theyear}\hfill
293                                 {\wk@daynametype\theweekday}%
294                                 }\par
295                             \fi
296                         \fi
297                     \vfill
298                 }%           vbox
299                 \hrule width\linewidth
300                 \vskip 1pt
301                 \raggedright
302                 \small
303                 \begin{multicols}{2}
304                     \ifcat$\theevents$\%

```

```

305         \else%
306         \theevents\par
307         \fi
308         \ifcat$\theappointments$%
309         \else
310         \theappointments
311         \fi
312     \end{multicols}
313     }%         parbox
314     }%         hbox
315     \leaders\ vbox{\vskip\wk@intralineskip\wk@hrulefill}\vfill
316     \vskip\wk@bottomlineskip
317     }%         vbox
318     }%         sbox
319 \fi
320 \advance\wk@tempcount by 1
321 \ifnum\wk@tempcount>6
322     \wk@tempcount=0
323 \fi}
324 </class>

```

References

- [GMS94] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley Company, 1994.

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.

A	
<code>\autofilo</code>	146
C	
<code>\cal@entry@text</code>	174, 178, 183
<code>\cal@get@appointments</code>	171
<code>\cal@get@events</code>	171
<code>\cal@insert</code>	140
<code>\cal@optionslist</code>	146
<code>\cal@range@end</code>	157, 165
<code>\cal@range@start</code>	151, 153, 168
<code>\cal@subgroup@count</code>	188, 202, 216, 230, 269, 283
<code>\cal@textcolor</code>	176, 180, 184
<code>\calcolor</code>	176, 180, 184
<code>\caldate</code>	154, 158, 164, 169
D	
<code>\dates@date</code> ...	151, 153, 157, 165, 168
<code>\dates@requiredaytrue</code>	150
<code>\dates@requiremonthtrue</code>	150
<code>\dates@time@end</code>	173
<code>\dates@time@start</code>	172, 173
E	
<code>\endautofilo</code>	147
I	
<code>\ifcase</code>	188
<code>\ifwk@compact</code> 9, 37, 72, 93, 131, 209, 276	
<code>\ifwk@ferial</code>	10, 39, 78, 100, 133, 187, 215
<code>\ifwk@overthetop</code>	4, 267
<code>\ifwk@traditional</code>	8, 26
N	
<code>\newcalendar</code>	138
<code>\nextday</code>	163
O	
<code>\or</code>	190, 192, 194
P	
<code>\prevday</code>	152
T	
<code>\theappointments</code> ..	254, 256, 308, 310
<code>\theday</code>	156, 167, 211, 218, 223, 232, 237, 278, 285, 290
<code>\theevents</code>	250, 252, 304, 306
<code>\themonth</code>	212, 219, 224, 233, 238, 279, 286, 291
<code>\theshortmonth</code>	156, 167
<code>\theshorttext</code>	176, 180, 184
<code>\theshortweekday</code>	155, 166
<code>\thetimeend</code>	179
<code>\thetimestart</code>	175, 179
<code>\theweekday</code>	210, 217, 226, 231, 240, 277, 284, 293
<code>\theyear</code>	156, 167, 213, 220, 225, 234, 239, 280, 287, 292
W	
<code>\wk@bottomlineskip</code>	12, 63, 64, 261, 316
<code>\wk@compactfalse</code>	17
<code>\wk@compacttrue</code>	16
<code>\wk@daynametype</code>	51, 52, 210, 217, 226, 231, 240, 277, 284, 293
<code>\wk@daynumbertype</code> ...	48, 49, 211, 218, 223, 232, 237, 278, 285, 290
<code>\wk@ferialfalse</code>	16
<code>\wk@ferialtrue</code>	17
<code>\wk@firstday</code>	5, 36, 38, 40, 42, 152, 159–161, 163
<code>\wk@getitemheight</code>	127, 146
<code>\wk@hrulefill</code>	68, 260, 315
<code>\wk@intralineskip</code>	11, 60, 61, 260, 315
<code>\wk@itemheight</code>	7, 115, 128–130, 132, 134, 136, 203, 270
<code>\wk@itemheight@scalefactor</code> .	189, 191, 193, 195, 197, 200, 203, 270
<code>\wk@monthtype</code>	54, 55, 212, 219, 224, 233, 238, 279, 286, 291
<code>\wk@overthetopfalse</code>	186
<code>\wk@overthetoptrue</code>	264
<code>\wk@ruleheight</code>	13, 66–68
<code>\wk@spitoutboxes</code>	91, 147, 148
<code>\wk@tempcount</code>	6, 70, 71, 74–76, 80–82, 86–88, 92, 95, 96, 98, 102, 103, 105, 109, 110, 112, 119, 121, 122, 124, 320–322
<code>\wk@titletype</code>	45, 46, 210, 217, 223, 231, 237, 277, 284, 290
<code>\wk@traditionaltrue</code>	15, 25
<code>\wk@yeartype</code>	57, 58, 213, 220, 225, 234, 239, 280, 287, 292

Change History

v0.1	General: First public release incorporating small bug-fixes and extensions	1	dar options to enable printing of lines in every day's box	3	
v0.2	General: Switched to evautofl package by default; provided an option to keep using autofilo	4	v0.4	General: Added a control on evautofl availability	4
v0.3	General: Added three new calen-			Added a little, stupid example of use	1
				Added the compact option	4
				Added the ferial option	4