

Using T_EX to make Agendas and Calendars with Astronomical Events

Jordi Saludes

At the Cork T_EX meeting participants were presented with an agenda obviously typeset with T_EX. Calendars and agendas show a lot of computed information, like the number of day in the month, the name of day in the week. T_EX is well-suited to deal with these computations. Agendas also give relevant events for the user: 'Fixed' events occurring every year at fixed dates but also 'movable' events which depend on astronomical phenomena related to the sun and moon, such as religious feast days or moon phases.

It would be nice to have a package to typeset agendas including fixed events, even better, to have T_EX compute the date of movable events and include them in the agenda.* We describe here a macro package to make calendars and agendas for a given year, supporting

- Easy inclusion of custom events;
- Automatic inclusion of computed events.

This package is divided into several files devoted to computation of different type of events. File `calend0.tex` is the basic macro file; it provides tools to convert between and compute date formats. File `calend1.tex` gives macros to support computation of astronomical events which are calculated in `sun.tex` and `moon.tex` and file `feasts.tex` allows computation of movable feasts based on the date of Easter Sunday. We provide several example style files containing formatting instructions to typeset agendas and calendars.

Acknowledgments. The author is grateful to B. Beeton and the reviewer for suggesting several style formats.

Basic Mechanism

Events. To obtain an agenda for year $\langle my\ year \rangle$ you have to create a source file including custom events

```
\event<day>/<month>[<text>]
```

in order the agenda will show $\langle text \rangle$ in the cell corresponding to $\langle month \rangle$, $\langle day \rangle$.

The prototypical source file looks like this:

```
\input calend0
\setyear{\my year}
\settimezone{\my timezone}
\my definitions}
\beginevents
\input moon
\input sun
\input feasts
\event 6/6[Mom's anniversary]
\event 5/1[TUG Meeting]
\more events}
:
:
\includemoons
\includeseasons
\includefeasts
\endevents
```

And the procedure is as follows:

1. Run T_EX for the first time. This process reads the events and puts them into an auxiliary event file $\langle jobname \rangle.eve$. Each `\event` generates an entry in $\langle jobname \rangle.eve$ converting the calendar date of event to long date

```
\evententry{\long date}{\text}
```

For `\setyear{1993}` and `\settimezone{0}` the event file looks like

```
\evententry{34125}{Mom's anniversary}
\evententry{33973}{TUG Meeting}
```

```
:
:
```

```
\evententry{33969}{First quarter}
\evententry{33976}{Full Moon}
```

```
:
:
```

```
\evententry{34047}{Spring}
```

```
:
:
```

```
\evententry{34069}{Easter Sunday}
```

```
:
:
```

2. Sort the lines of the auxiliary file using some sorting utility. Now events are in chronological order.
3. Add a `\makeagenda{\style}` command just before `\beginevents` and run T_EX for the second time. This process reads in the auxiliary file and generates an agenda or calendar including the events in suitable days (see appendix A).

Converting dates.

- *The long date* is the Julian date referred to 1900 Jan 0, i.e. the number of days passed

* Like the `\primes` example of [K1], p. 218, T_EX not only writes, but provides the information too.

from 1899 Dec 31 up to current day. This package uses the long date as the base date to `\evententry` commands and to compute the day of the week.

- Precise astronomical computations require not only the date but also the fraction of day passed. We define *Julian date and time* (JDT) as the number of days passed from 1900 Jan 0.5 (i.e. 1899 Dec 31 at noon), including fractions of day.

Command `\dayno` performs the conversion between ordinary date contained in counters `\Day` and `\Month` and year in `\setyear` to long date. The value is returned in `\date`. This is used, for instance, when processing `\event` which is expanded to `\dayno` plus `\evday` which generates the actual entry in the event file. `\evday{<text>}` is like `\event` but reads the long date of the event from counter `\date`. Note that `<text>` is not expanded in `\event`, while it is in `\evday`; this is because `\evday` is intended for automatic generation of events whose description is, in general, the product of a macro expansion.

To convert between long dates and usual calendar dates, use `\caldate`. This will transform the value of `\date` to `\Day` and `\Month`; `\themonth` gives the name of month number `\Month`. The week day is given by the remainder of the integer division of long date by 7: `\weekday` does it for counter `\date`, returning the value to `\weekd` in such a way that Sunday is 0 and Saturday is 6. `\theweekday` gives the name of the day. `\theshortweekday` gives an abbreviation of the week day. Both are language-dependent. In the files of this package you can search for the string 'local' to find commands which depend on language or other local assumptions.

Command `\nextday{<number>}` advances counter `\date` to the next day with `\weekd={<number>}`; `\prevday` works like `\nextday` but going backwards. There are strict versions of these commands: `\snextday` and `\spreveday`. The difference between the strict and normal ones is the same as between '`<`' and '`≤`'; `\snextday` (`\spreveday`) always changes `\date` by at least 1 day, while `\nextday` (`\prevday`) does not. For instance, assume `\date` is the long date of a Monday then `\snextday1` advances `\date` by 7, but `\nextday1` leaves `\date` unchanged. These macros are useful when computing movable feasts which have to fall in a given week day.

JDT is based on ephemeris time which is no more than 2 minutes from UT (Universal time). We

can then assume that $UT \approx JT + 0.5$, where JT is the fractional part of JDT. Command `\JDTtoL` converts JDT to local long date by adding the value of your timezone and discarding fractions of day. You can state your timezone using `\settimezone{<my timezone>}`, where `<my timezone>` is the difference in hours between your time and Greenwich time (West negative, East positive).

When a computed event requiring JDT falls near local midnight it is probable that, due to approximation errors, truncation would place the event in the adjacent day.

Style files

The `\makeagenda{<style>}` command in the second T_EX run reads the style file `<style>.sty` and accomodates the events following the formatting instructions contained in this file. We provide example files for two agenda styles and two calendars.

All these examples support marking of holidays by including `\holy` in the description part of an event. The style file can then use `\ifholy` to test this condition when making the cell day. In the example files, the expression `\ifholy\bf\fi` is used to typeset in bold the day number of a holyday.

A style file has to define the fonts being used, set the dimension parameters, and state the period of time covered by the agenda by setting counter `\currentdate` to the initial long date and `\lastdate` to the long date of the last day of the wanted period. This is how `\makeagenda` generates the cells:

- S1. The cell for the first day (actual value of `\currentdate`) is opened: the calendar date of `\currentdate` is placed into `\Day` and `\Month` and `\beginday` is executed. This control sequence is defined in the style file as instructions for making the day cell up to the moment when the events for this day have to appear.
- S2. Now an `\evententry` is read. If the date of the event d is not $\text{\currentdate} \leq d \leq \text{\lastdate}$, the entry is ignored.
- S3. If $d = \text{\currentdate}$ and the cell day for `\currentdate` is opened the event text is added as a paragraph (If the cell for `\currentdate` was not opened yet, it is done now). Back to S2.
- S4. If $d > \text{\currentdate}$ the open cell is closed by doing `\endday`. This control (symetric to `\beginday`) defines how to finish a cell day properly.
- S5. The value of `\currentdate` is incremented and a cell without events is generated. This is

repeated until $d = \backslash\text{currentdate}$. Then jump to S3.

When all the events have been read, cells without events are created if necessary to reach $\backslash\text{lastdate}$. Now T_EX expands $\backslash\text{epilog}$ which contains instructions to be executed after the last day cell was typeset. $\backslash\text{epilog}$ must always be defined though it can have an empty expansion.

The control sequence $\backslash\text{makeempty}$ generates day cells for dates between $\backslash\text{currentdate}$ and $\backslash\text{lastdate}$ but without including any event.

A5 style. File `a5.sty` contains A5 two-page style that gives space (cells) for each day separated by rules. Each cell displays the $\backslash\text{Day}$ for the month number and the day of the week. Events (if any) are listed right down the rule. The cell is of fixed height, so you can run over into the following cell if you have too many events in a day. This style gives one page per week and each page begins with a monday. The footline displays the name of the current month. Note that the agenda begins at the first Monday of September and finishes the Sunday after the end of May, to give a sort of academic-year agenda.

For styles like this one, not in a single natural year, you have to state events twice, since $\backslash\text{includemoons}$, $\backslash\text{includeseasons}$ and $\backslash\text{includefeasts}$ generate events for one year. For example, the following will give moons for years 1992 and 1993

```
 $\backslash\text{setyear}\{1992\}$ 
 $\backslash\text{beginevents}$ 
 $\langle 92\text{-events}\rangle$ 
 $\backslash\text{includemoons}$ 
 $\backslash\text{nextyear}$ 
 $\langle 93\text{-events}\rangle$ 
 $\backslash\text{includemoons}$ 
 $\backslash\text{endevents}$ 
```

A6 style. This style gives a double-page per week agenda for a natural year. Saturday and Sunday get half size cells. The headline of the left page contains the number of the week in the year while the other displays the current month. Sundays and holidays show a bold day numer. The style file is `a6.sty`.

Wall style. Put $\backslash\text{Month} = \langle \text{my month number} \rangle$ before $\backslash\text{makeagenda}\{\text{wall}\}$ to obtain an A4-landscape wall calendar for this month. Days are distributed in a grid, each week (Sunday to Saturday) in a row.

If you forget to include the $\backslash\text{Month}$ assignment, the style typesets the current month if today belongs to the first half of the month, otherwise it gives the next month.

Block style. Desktop block calendar (1 page per day). Each page displays the month name, a big day number and the week day. Small calendar displays of the current and next months appear on the left and right side of the big number. Note the use of $\backslash\text{makeempty}$ in the style file `block.sty`.

Astronomical computations

Complex numerical routines are done without allocation of new counters; $\backslash\text{count0}$ to $\backslash\text{count8}$ are used instead. To prevent interference with the standard use of these registers in the output routine, calculations remain local by grouping them into braces. The returned value is extracted using a global assignment.

Arithmetic. There are two kinds of computations needed

- Integer arithmetic computations;
- Computations involving trigonometric functions.

Approximation errors and overflow limit the validity of computations to years between 1900 and 2100.

Integer arithmetic is the natural one in T_EX. Calculations can be done using the primitive commands $\backslash\text{advance}$, $\backslash\text{multiply}$ and $\backslash\text{divide}$. Remember that (as stated in [K1], p. 118) the last operation means integer division (discarding the remainder) for positive values; for negative numbers this operation is equivalent to division of the absolute value and changing of sign. Conversions between dates are done in integer arithmetic.

The second class of computations is handled by a fixed-point approach and approximate calculation. A fractional value a is stored in a counter as \tilde{a} such that $a = \tilde{a} \cdot 10^{-n}$ and \tilde{a} is an integer with $|\tilde{a}| < 2^{31}$. We say that a is *scaled* 10^{-n} ; for example, JDT is usually stored in counter $\backslash\text{date}$ as $\text{JDT} = \backslash\text{date} \cdot 10^{-3}$. Thus, we can express JDT with an accuracy of $10^{-3} \approx 1.44$ minutes.

The astronomical formulae we use require products of high precision numbers (more than 6 digits) but we can't multiply two such numbers using single T_EX counters. In this case we store a in blocks a_0, \dots, a_k of m -digit integers such that $a = (a_0 + a_1 10^m + a_2 10^{2m} + \dots + a_k 10^{km}) \cdot 10^{-n}$. For instance $\backslash\text{lin}\langle a_2 \rangle \cdot \langle a_1 \rangle \cdot \langle a_0 \rangle + \langle b \rangle$ computes $\theta = aT + b$, where $T = \backslash\text{count0} \cdot 10^{-6}$ is Julian centuries and a is a 9-digit precision factor divided in 3 blocks (a_2 , a_1 and a_0) to avoid overflow during multiplication; b and θ are scaled 10^{-3} .

Accuracy conditions on the formulae imply that trigonometric functions have to be computed

with a maximum error of 0.01. To evaluate $\sin \theta$ the argument is reduced modulo 2π and the value is computed by linear interpolation on a table with step 0.1 in the interval $[0, \pi]$ (following [H], this is sufficient to fit in the above error). For $\cos \theta$ we use the equivalent form $\sin(\pi/2 - \theta)$.

The moon. File `moon.tex` allows generation of events for full, new moon and quarters in the given year. Formulae [M], pp. 159–161, imply computation of a sum of trigonometric functions evaluated with arguments which are linear transformations of T (the Julian century of the moment). We have adapted these formulae by discarding high order terms keeping an accuracy of 45 minutes.

Counter `\moonno` stands for the number of quarters of moon cycle passed from the beginning of 1900. `\moonno` modulo 4 gives the type of quarter as follows

| Phase | <code>\moonno</code> (4) |
|---------------|--------------------------|
| New moon | 0 |
| First quarter | 1 |
| Full moon | 2 |
| Last quarter | 3 |

The command `\includemoons` generates events for all quarters in the given year. First, the approximate value of `\moonno` is computed for the first phase in the year, then `\moondate` gives the exact long date and generates an `\evday` entry. This is performed repeatedly until the end of year is reached. Text for `\evday` comes from `\phase` (language dependent).

The sun. The beginning of seasons is determined by the dates of March and September equinoxes and June and December solstices. We use formulae [M], p. 90 (neglecting quadratic terms), that give values in JDT with error generally less than $1/2$ hour. These events are generated by `\includeseasons` in file `sun.tex`.

... **and all together.** Finally the file `feasts.tex` allows the inclusion of movable feasts based on the date of Easter Sunday, which is intended to be the first Sunday after the first full moon of spring. However the ecclesiastical computation does not always coincide with that definition since it assumes that the moon moves uniformly, so we take the Clavius-Lilius algorithm (XVI Century) used by most Western churches (see [K2]), instead of deriving the date from astronomical data.

File `feasts.tex` gives examples using `\easter` and `\snextday` in the computation of some movable religious feasts as in [E]. For instance, note that (in some countries) Epiphany is defined as the first

Sunday after New Year's day. Use `\includefeasts` to put feast dates into the events file.

References

- [K1] D. E. Knuth, *The T_EXbook, Computers & Typesetting vol. A*. Addison-Wesley (1984).
- [K2] D. E. Knuth, *The Art of Computer Programming: Fundamental Algorithms*. Addison-Wesley (1978).
- [M] J. Meeus, *Astronomical Formulæ for calculators*. Willmann-Bell, Richmond (1985).
- [H] P. Henrici, *Elements of Numerical Analysis*. John Wiley & Sons, New York (1984).
- [E] *Éphémérides 1984*. Annuaire du Bureau des Longitudes, Gauthier-Villars, Paris (1983).

◇ Jordi Saludes
 ETSEIT, Universitat
 Politècnica de Catalunya
 Colom, 11
 08222 Terrassa, Spain
 saludes@ma2.upc.es

Appendix A. Example pages

Examples are given of the following styles:

- Block style
- A5 style
- Wall style
- A6 style

Except for the wall style, a faint rule has been drawn around the example pages to indicate their borders.

A.1 Block style

| JANUARY | | | | | | | | | | | | | | | | | |
|---------------|----|----|----|----|----|----|---|----------|--|----|----|----|----|----|----|----|---|
| | | | | | | 1 | 2 | | | | | 1 | 2 | 3 | 4 | 5 | 6 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | | 1 | | 7 | 8 | 9 | 10 | 11 | 12 | 13 | |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | | | | 14 | 15 | 16 | 17 | 18 | 19 | 20 | |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | | | | 21 | 22 | 23 | 24 | 25 | 26 | 27 | |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | | | | 28 | | | | | | | |
| 31 | | | | | | | | | | | | | | | | | |
| Friday | | | | | | | | | | | | | | | | | |
| First quarter | | | | | | | | | | | | | | | | | |

A.2 A5 style

| | |
|--------------|-------------|
| | Monday 6 |
| | Tuesday 7 |
| | Wednesday 8 |
| Last quarter | Thursday 9 |
| | Friday 10 |
| | Saturday 11 |
| | Sunday 12 |

September

A.3 Wall style

January

| Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | S |
|---------------|--------|------------------|-----------|----------|--------------------|-------------|
| 27 | 28 | 29 | 30 | 31 | 1 First quarter | 2 |
| 3 Epiphany | 4 | 5 TUG Meeting | 6 | 7 | 8 Full Moon | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 Last quarter | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 New moon | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 First |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 Full |

A.4 A6 style

Week #2

January

| | | | | | | |
|-------|-------|-------|-------|-------|-------|--------|
| 4 Mon | 5 Tue | 6 Wed | 7 Thu | 8 Fri | 9 Sat | 10 Sun |
|-------|-------|-------|-------|-------|-------|--------|

Appendix B. The macros

B.1 calend0.tex

```

%%
%% FILE calend0.tex
%% Modificat 9/12/92
\catcode'\@=11
\newif\ifleapyear
\def\loadadvanced{%
  \input calend1.tex\relax}
\newcount\date
\newcount\weekd
\newcount\Year
\newcount\yearbase
\newcount\Month
\newcount\Day
\newcount\@catch
\newcount\timezone\timezone=0
\def\setyear#1{Year=#1
  \advance\Year by-1900\calcuyearbase}
\def\nextyear{\advance\Year by1
  \calcuyearbase}
\def\settimezone#1{timezone=#1
  \multiply\timezone by 1000
  \divide\timezone by24}
% \ifleapyear is set;
% \yearbase is the number of
% days passed from 1900, Jan 0
% to New year's date;
\def\calcuyearbase{%
  \yearbase=-1
  {\count0=\Year\divide\count0 by4
  \multiply\count0 by4
  \ifnum\Year=\count0
  \global\@catch=0\else\global\@catch=1\fi}%
  \ifcase\@catch \leapyeartrue\or
  \leapyearfalse\fi
  {\count0=\Year\multiply\count0
  by1461 \advance\count0 by3
  \divide\count0 by4
  \global\@catch=\count0}%
  \advance\yearbase by\@catch
  \ifnum\Year=0\yearbase=0\leapyearfalse\fi}
% Gives the number of days passed
% at the end of each month.
% Value returned in \Day
\def\monthdays{\global
  \@catch=\ifcase\Month 0\or31\or
  59\or90\or120\or151\or181\or212\or
  243\or273\or304\or334\or365\fi
  {\ifleapyear\ifnum\Month>1
  \global\advance\@catch by1\fi\fi}%
  \Day=\@catch}
% Long date of \Month, \Day
% in the year \Year.
% Value returned in \date
\def\dayno{\date=\Day{%
```

```

  \advance\Month by-1\monthdays
  \advance\date by\Day
  \global\advance\date by\yearbase}}
% Long date MOD 7 gives the week day.
% Sunday is 0 and Saturday is 6.
\def\weekday{{\count0=\date\relax
  \count1=\count0\divide\count0 by 7
  \multiply\count0 by 7
  \advance\count1 by -\count0
  \global\weekd=\count1}}
% Name of week day (Local).
\def\theweekday{\weekday
  \ifcase\weekd
  Sunday\or Monday\or Tuesday\or
  Wednesday\or Thursday\or Friday\or
  Saturday\fi}
% (Local)
\def\theshortweekday{\weekday
  \ifcase\weekd Sun\or Mon\or Tue\or
  Wed\or Thu\or Fri\or Sat\fi}
% Gives the usual calendar date for
% a long date in counter \date.
% Returned in \Day, \Month and \Year
\def\caldate{\Year=\date
  \multiply\Year by4\divide\Year by1461
  \calcuyearbase
  {\advance\date by-\yearbase\Month=0
  \loop\monthdays\ifnum\Day<\date
  \global\advance\Month by1\repeat
  {\advance\Month by-1\monthdays
  \advance\date by-\Day
  \global\Day=\date}}}}
% Name of month \Month. (Local)
\def\themoth{\ifcase\Month
  \or January\or February\or March\or
  April\or May\or June\or July\or
  August\or September\or October\or
  November\or December\fi}
% Find the next (previous) day after
% (before) \date with \weekday=#1.
\def\nextday#1{{\count0=#1\weekday
  \advance\count0 by-\weekd
  \ifnum\count0<0\advance\count0 by7\fi
  \global\advance\date by\count0}}
\def\prevday#1{\snextday#1
  \advance\date by-7}
\def\snextday#1{\advance\date by1
  \nextday#1}
\def\sprevday#1{\advance\date by-1
  \prevday#1}
% \beginevents... \endevents
% contains control sequences like
% \event or \evday or sequences
% generating these commands.
\newwrite\evefile
\def\beginevents{%
  \immediate\openout
  \evefile=\jobname.eve\relax}
\def\endevents{\immediate\closeout
```

```

\evetext\end}
\newtoks\evetext
\def\event#1/#2[#3]{\evetext={#3}%
\Day=#1\Month=#2\dayno
\evday[\the\evetext]}
\def\evday[#1]{\immediate\write\evetext{%
\string\evententry{\the\date}{#1}}
\def\thecaldate{\the\Day/\the\Month}
\def\mute{\def\thecaldate{}}
%
% 2nd run commands
%
\newcount\currentdate
\newcount\lastdate
\newcount\nextdate
\newif\ifdoing@day
\newif\ifholy
\def\holy{\global\holytrue}
\def\upto#1{\nextdate=#1
\loop\advance\currentdate by1\relax
\ifnum\currentdate<\nextdate
\begin@day\end@day\repeat}
\def\evententry#1#2{\unskip
\ifnum#1>\lastdate\else
\ifnum\currentdate<#1
\ifdoing@day\end@day\fi
\upto{#1}\fi
\ifnum\currentdate=#1
\ifdoing@day\else\begin@day\fi
#2\par\fi\fi}
\def\begin@day{\doing@daytrue
\date=\currentdate\caldate
\message{<\thecaldate}\beginday}
\def\end@day{\endday\doing@dayfalse
\global\holyfalse\message{>}}
\def\makeagenda#1{\input #1.sty\relax
\begin@day\input\jobname.eve\relax
\evententry{\the\lastdate}{}\end@day
\epilog\end}
% Do not read events
\def\makeempty{\begin@day
\evententry{\the\lastdate}{}\end@day}
\catcode'\@=12

```

B.2 a5.sty

```

%%
%% FILE a5.sty
%%
\newcount\daysinpage
\newdimen\cellheight
\vsz=17.5cm\voffset=-0.5in
\hsz=11cm\hoffset=-0.6in
\cellheight=\vsz
\divide\cellheight by7
\advance\cellheight by-0.4pt
\parindent=0pt
\font\Rm=cmr10 at 14pt

```

```

\font\Bf=cmbx10 at 14pt
\font\rm=cmr10
\footline={\Bf\ifodd\pageno\hfill\botmark
\else\botmark\hfill\fi}
\def\newpage{\vfill\penalty-10000}
\def\beginday{\ifnum\daysinpage>0\hrule\fi
\mark{\themonth} % Outside any box
\setbox2=\vbox\bgroup
\ifodd\pageno
\rightskip=3cm plus 1fill
\else
\leftskip=3cm plus 1fill
\fi\rm}
\def\endday{\egroup
\vbox to\cellheight{\vglue2pt
\setbox0=\hbox to\hsz{%
\ifodd\pageno
\hfill{\Rm\theweekday\ \ifholy\Bf\fi\the\Day}
\else
{\ifholy\Bf\fi\the\Day}\ \theweekday}%
\hfill\fi}
\copy0\kern-\ht0\box2\vfill}
\advance\daysinpage by1
\ifnum\daysinpage=7 \daysinpage=0
\newpage\fi}
%
% Local: from monday ...
\Day=1 \Month=9\dayno\nextday1\currentdate=\date
% Local: ... to sunday.
{\nextyear\Day=31 \Month=5\dayno\nextday0
\global\lastdate=\date}
\leftskip=0pt\rightskip=0pt
\daysinpage=0
\def\epilog{}

```

B.3 a6.sty

```

%%
%% FILE a6.sty
%%
\baselineskip=10pt
\newcount\daysinpage
\newdimen\cellwidth
\newdimen\cellheight
\vsz=5in\voffset=-0.2in
\hsz=3.5in\hoffset=-0.25in
\newdimen\sphsize
\sphsize=\hsz
\divide\sphsize by2\advance\sphsize by-0.4pt
\cellheight=\vsz
\divide\cellheight by3
\advance\cellheight by-0.4pt\parindent=0pt
\font\bf=cmbx12
\font\rm=cmr12
\font\small=cmr9
\headline={\hfill
\ifodd\pageno
\rm\xdef\ftmk{\firstmark}}%

```

```

\divide\pageno by2\advance\pageno by1
Week \#\the\pageno
\else
\bf\edef\btmk{\botmark}%
\ifx\ftmk\btmk\else\ftmk/\fi\btmk
\fi\hfill}
\footline={}
\def\newpage{\vfill\penalty-10000}
\def\beginaday{
\ifcase\daysinpage\or\hrule\or\hrule\or\or
\hrule\or\hrule\fi
\mark{\themonth} % Outside any box
\setbox1=\vbox to\cellheight\bgroup\vglue4pt
\ifnum\daysinpage>4\hsize=\sphsize\fi
\setbox3=\vbox\bgroup
\ifnum\daysinpage>2
\rightskip=1.45cm plus 1fill
\else
\leftskip=1.45cm plus 1fill\fi
\noindent\small}
\def\endday{\egroup
\setbox0=\hbox to\hsize{\rm\ifholly\bf\fi
\ifnum\daysinpage=6\bf\fi
\ifnum\daysinpage>2
\hfill{\rm\theshortweekday}
\ \the\Day\hskip0.5em}
\else
{\hskip0.5em\the\Day
\ {\rm\theshortweekday}}\hfill\fi}
\copy0\kern-\ht0\vskipOpt\box3
\vfill\egroup
\advance\daysinpage by1
\ifcase\daysinpage\or\box1\or\box1\or
\box1\newpage\or\box1\or\box1\or
\setbox2=\box1\or
\hbox{\box2\vrule\box1}\newpage
\daysinpage=0\fi}
%
% Local: from monday ...
\Day=1\Month=1\dayno\prevday1\currentdate=\date
% Local: ... to sunday.
\Day=31\Month=12\dayno\nextday0\lastdate=\date
\leftskip=3pt\rightskip=3pt
\daysinpage=0
\def\epilog{}

B.4 wall.sty

%%
%% FILE wall.sty
%%
\hsize=9.5in
\vsizer=7.5in\voffset=-0.65in
\baselineskip=12pt
\newcount\daysinrow
\newdimen\cellwidth
\newdimen\cellheight
\cellheight=1.13in

\parindent=0pt
\cellwidth=\hsize
\divide\cellwidth by7
\advance\cellwidth by-0.4pt
\font\BBf=cmbx12 at 24pt
\font\Bf=cmbx10 at 20pt
\font\Rm=cmr10 at 20pt
\font\rm=cmr12 at 14pt
\font\small=cmr9
\footline={}
\def\hstrut{\hrule height0pt depth0pt
width\cellwidth}
\def\beginaday{
\ifnum\daysinrow=0\setbox1=\hbox{\vrule}\fi
\setbox0=\vbox to\cellheight\bgroup
\hstrut\hsize=\cellwidth\vskip5pt
\setbox2=\vbox\bgroup\small}
\def\endday{\egroup
\hbox{\hskip0.5em\Rm\ifholly\Bf\fi\the\Day}
\vskip2pt\box2\vfill\egroup
\setbox1=\hbox{\unhbox1\box0\vrule}
\advance\daysinrow by1
\ifnum\daysinrow=7\box1\hrule\daysinrow=0\fi}
%
\def\advancemonth{\ifnum\Month=12
\nextyear\Month=1\else\advance\Month by1\fi}
% If no Month is given, take today's date.
\ifnum\Month=0 \Month=\month
\ifnum\day>15 \advancemonth\fi\fi
\Day=1\dayno
\topskip=0pt\hbox{}
\vfill
\centerline{\BBf\themonth}
\penalty10000
\vskip40pt minus32pt
% Local: from Sunday ...
\prevday0\currentdate=\date
\hbox{\rm\loop
\hbox to\cellwidth{\hss\theweekday\hss}
\unskip\ifnum\weekd<6\advance\date by1\repeat}
\penalty10000
\vskip3pt
% Local: ... to Saturday.
\Day=1 \advance\Month by1\dayno
\advance\date by-1\nextday6\lastdate=\date
\leftskip=3pt
\raggedright
\daysinrow=0
\vbox\bgroup\hrule
\def\epilog{\vfill\egroup\supereject}

```

B.5 block.sty

```

%%
%% FILE block.sty
%%
\newcount\daysinrow
\newcount\saveMonth
\font\bf=cmbx12 at 16pt
\font\Rm=cmr12 at 48pt
\font\Bf=cmbx12 at 48pt
\font\rm=cmr12
\font\small=cmr10
\footline={ }
\def\card{\mute
\saveMonth=\Month
\def\beginaday{
\ifnum\daysinrow=0\setbox1=\hbox{\fi
\setbox0=\hbox to1.3em{\hss
\ifnum\Month=\saveMonth\the\Day\fi}}
\def\endaday{\setbox1=\hbox{\unhbox1\box0}
\advance\daysinrow by1
\ifnum\daysinrow=7\box1\daysinrow=0\fi}
% Local: from sunday ...
\Day=1\dayno
\prevday0\currentdate=\date
\Day=1\advancemonth \dayno
% Local: ... to saturday.
\advance\date by-1 \nextday6 \lastdate=\date
\daysinrow=0
\baselineskip=9pt\small
\global\setbox3=\vbox to0.55in{\makeempty\vss}}
%
\def\cstrut{\vrule width0pt depth0.3in}
\baselineskip=14pt
\parindent=0pt
\vsiz=5in \voffset=-0.2in
\hsiz=4in \hoffset=-0.25in
\def\advancemonth{\ifnum\Month=12
\nextyear\Month=1\else
\advance\Month by1\fi}
\def\beginaday{\ifnum\Day=1
\setbox2=\box3{\advancemonth\card}\fi
\date=\currentdate\caldate
\centerline{\bf\uppercase
\expandafter{\themoth}}
\setbox0=\vbox\bgroup\noindent\rm}
\def\endaday{\egroup\vskip10pt
\centerline{\cstrut\copy2\quad
\hbox to1in{\Rm\weekday}
% Local: sundays in bf.
\ifnum\weekd=0\Bf\fi\ifholly\Bf\fi
\hss\the\Day\hss}\quad\copy3}
\centerline{\rm\theweekday}
\vskip20pt\box0\newpage}
\def\newpage{\vfill\penalty-10000}
%
\Day=31\Month=12\dayno\lastdate=\date
\Day=1\Month=1\dayno\currentdate=\date
\card

```

```

\leftskip=1em plusifill\rightskip=1em plusifill
\def\epilog{}

```

B.6 calend1.tex

```

%%
%% FILE calend1.tex
%%
\def\loadadvanced{\relax}
% Convert from Julian date and time in
% \date to long date (in local time)
\def\JDttoL{\advance\date by500
\advance\date by\timezone
\divide\date by1000 }
%% Trigonometric functions
\def\sintable#1{\ifcase #1 0 \or100 \or199
\or296 \or389 \or479 \or565 \or644 \or717
\or783 \or841 \or891 \or932 \or964 \or985
\or997 \or1000 \or992 \or974 \or946 \or909
\or863 \or808 \or746 \or675 \or598 \or516
\or427 \or335 \or239 \or141 \or42 \or-58
\or-158\fi}
% Reduces modulo 2\pi (requires positive
% argument theta):
% theta := theta MOD 2\pi, where
% theta = count1*10^(-3)
\def\twopimod{\count2 =\count1
\divide\count2 by6284 \count3 =1853
\count4 =6283 \multiply\count3 by\count2
\multiply\count4 by\count2
\divide\count3 by10000
\advance\count3 by\count4
\advance\count1 by-\count3 }
\newif\ifsign
% v := sin(theta), where
% v = count4*10^(-3);
% theta = count1*10^(-3)
% theta is reduced MOD 2\pi to be
% 0<=theta<2\pi by \twopimod,
% then linear interpolation is performed
% using \sintable.
\def\Sin{
\ifnum\count1<0 \signtrue
\count1=-\count1 \else \signfalse\fi
\loop\ifnum\count1>6284 \twopimod\repeat
\ifnum\count1>3142
\advance\count1 by-3142
\ifsign\signfalse\else\signtrue\fi\fi
\multiply\count1 by10 \count3 =\count1
\divide\count3 by1000 \count2 =\count3
\multiply\count3 by1000
\advance\count3 by-\count1
\count5 =\sintable{\count2 }
\count4 =\count5\advance\count2 by1
\advance\count4 by -\sintable{\count2 }
\multiply\count4 by\count3
\divide\count4 by1000
\advance\count4 by\count5

```

```

\ifsign\count4 =-\count4 \fi
% v := cos(theta), where
% v = count4*10^(-3);
% theta = count1*10^(-3).
% \Sin is evaluated on pi/2-theta.
\def\Cos{\advance\count1 by-1571
\multiply\count1 by-1\Sin}
% Linear transformation of T giving
% theta := a*T+b, where
% T = count0*10^(-6);
% theta = count1*10^(-3);
% a = #1*10^3+#2+#3*10^(-3);
% b = #4*10^(-3)
\def\lin#1.#2.#3+#4.{\count1 =#3
\count2 =#2 \count3 =#1
\multiply\count1 by\count0
\multiply\count2 by\count0
\multiply\count3 by\count0
\divide\count1 by1000
\advance\count1 by\count2
\divide\count1 by1000
\advance\count1 by\count3
\advance\count1 by #4}
% Accumulate value returned by a
% trigonometric function, scaled by
% factor f, into count6:
% ac := ac+f*v, where
% ac = count6*10^(-7)
% v = value of Sin or Cos =count4*10^(-3)
% f = #1*10^(-4)
\def\fac#1{\multiply\count4 by #1
\advance\count6 by\count4 }
\def\id{\count4=\count1 } % Identity

```

B.7 moon.tex

```

%%
%% FILE moon.tex
%%
\loadadvanced
\newcount\moonno
\def\firstmoon{\moonno=\Year
\multiply\moonno by123685
\divide\moonno by10000
\multiply\moonno by4 \advance\moonno by-1
\loop\moondate\ifnum\date<\yearbase
\advance\moonno by1 \repeat}
% Compute date for cycle quarter MOONNO
\def\moondate{\count0=\moonno
\lin 202.126.369+0.\count0=\count1
\count6=0
\lin 0.2.319+2907.\Sin\fac{3}
\divide\count6 by1000
\lin 365.249.86+7593.\id\fac{1}
\divide\count6 by10 \count7 =\count6
\ifodd\moonno\quarters
\else\fullornew\fi\global\date=\count7}
\JDTtoL\message{\the\date}

```

```

% Correction for full and new moon
\def\fullornew{\count6=0
\lin -393.0.0+0.
\divide\count1 by100000000
\advance\count6 by\count1
\lin 0.628.300+6269.\Sin
\multiply\count6 by\count4
\lin -7.-700.-369+ 928.\Sin\fac{ -74}
\lin 0. 628. 300+ 6269.\Sin\fac{ 1734}
\lin 1. 256. 600+12539.\Sin\fac{ 21}
\lin 8. 328. 670+ 5341.\Sin\fac{-4068}
\lin 8. 538. 220+-4597.\Sin\fac{ 10}
\lin 8. 956. 970+11610.\Sin\fac{ -51}
\lin 16. 238. 589+-5526.\Sin\fac{ -4}
\lin 16. 657. 340+10682.\Sin\fac{ 161}
\lin 16. 866. 890+ 743.\Sin\fac{ 104}
\lin 17. 285. 640+16951.\Sin\fac{ 50}
\lin 17. 495. 190+ 7013.\Sin\fac{ 4}
\lin 24. 986. 10+16023.\Sin\fac{ -4}
\lin 25. 195. 560+ 6084.\Sin\fac{ -6}
\divide\count6 by10000
\advance\count7 by\count6 }
% Correction for quarters
\def\quarters{\lin -393.0.0+0.
\divide\count1 by100000000
\count6 =\count1
\lin 0.628.300+6269.
\Sin\multiply\count6 by\count4
\lin -16. -29. -40+-4413.\Sin\fac{ 40}
\lin -7.-700.-369+ 928.\Sin\fac{ -47}
\lin -7. -72. -69+ 7198.\Sin\fac{ -30}
\lin 0. 628. 300+ 6270.\Sin\fac{ 1721}
\lin 1. 256. 600+12539.\Sin\fac{ 21}
\lin 8. 328. 670+ 5341.\Sin\fac{-6280}
\lin 8. 538. 220+-4598.\Sin\fac{ 21}
\lin 8. 956. 970+11611.\Sin\fac{ -119}
\lin 16. 238. 589+-5526.\Sin\fac{ -4}
\lin 16. 657. 340+10682.\Sin\fac{ 89}
\lin 16. 866. 890+ 743.\Sin\fac{ 79}
\lin 17. 285. 640+16952.\Sin\fac{ 3}
\lin 17. 495. 190+ 7013.\Sin\fac{ 3}
\lin 24. 986. 10+16023.\Sin\fac{ -4}
\lin 25. 195. 560+ 6085.\Sin\fac{ -6}
\count8=\count6\count6=28000
\lin 628. 300. 373+ 6270.\Cos\fac{ -4}
\lin 8. 328. 670+ 5341.\Cos\fac{ 3}
\count2=\moonno
\advance\count2 by-1\divide\count2 by2
\ifodd\count2\multiply\count6 by-1 \fi
\advance\count6 by\count8
\divide\count6 by10000
\advance\count7 by\count6 }
% Events for phases.
% Uses PHASE (local).
\def\includemoons{
\message{Including moons}
{\firstmoon
\Month=12\Day=31\dayno\advance\date by1
\count1=\date

```

```

\loop\moondate\phase\ifnum\date<\count1
\evday[\themoon]\advance\moonno by1
\repeat}}
\def\phase{\count0=\moonno
\count1=\moonno
\divide\count0 by4 \multiply\count0 by4
\advance\count1 by-\count0 \relax
\global\edef\themoon{\ifcase\count1
New moon\or First quarter\or
Full Moon\or Last quarter\fi}}

```

B.8 sun.tex

```

%%
%% FILE sun.tex
%%
\loadadvanced
\def\linY#1.#2.#3+#4.{\count0=\Year
\multiply\count0 by1000 \count1=0
\lin #1.#2.#3+#4.
\global\date=\count1 }\JDTtoL}
% (local)
\def\includeseasons{%
\message{Including seasons}
\linY 365.242.365+ 79573.\evday[Spring]
\linY 365.241.628+172405.\evday[Summer]
\linY 365.242.045+266014.\evday[Fall]
\linY 365.242.756+355784.\evday[Winter]}

```

B.9 feasts.tex

```

%%
%% FILE feasts.tex
%%
% Date of Easter for year \Year.
% Value returned in \date
\def\easter{\count0=\Year\count1=\count0
\divide\count0 by19 \multiply\count0 by19
\advance\count1 by-\count0
\advance\count1 by 1 % Golden number
\count2=\count1 \multiply\count2 by11
\advance\count2 by18 \count0=\count2
\divide\count0 by30 \multiply\count0 by30
\advance\count2 by-\count0 % Epact
\ifnum\count2=25 \ifnum\count1>11
\count2=26 \fi\fi
\ifnum\count2=24 \count2=25\fi
\advance\count2 by-44
\multiply\count2 by-1
\ifnum\count2<21 \advance\count2 by30\fi
\Month=3 \Day=0 \dayno
\global\advance\date by\count2}\snextday0}
% (local)
\def\includefeasts{%
\message{Including feasts}
\advance\evday[Easter Sunday]
{\advance\date by49 \evday[Whitsun]}
{\advance\date by-7 \evday[Palm Sunday]}
{\advance\date by-46 \evday[Ash Wednesday]}
\advance\date by-3 \evday[Carnival]}
\Month=1 \Day=1 \dayno\snextday0
\evday[Epiphany]
\event 25/12[\holy Christmas]}

```