

The Perfect Christmas Gift

A Friends and Family Calendar

Overview

This short course is designed to help you create a **special calendar** for one or more special people in your life.

We will need some way for you to create the input files for your calendars and get them to me so I can use my program (aka app) to produce your calendars. Then I will create them and I will e-mail them to you.

This is an iterative process and I expect you to check the products carefully (get another pair of eyes to help), make necessary corrections to the input files, and send them to me again.

Skills needed

- ▶ Familiarity with files and folders
- ▶ Ability to create a text file with an editor like MS Notepad

Project background

The special calendar is personal project started for my late mother-in-law, then for my wife, then for the public.

- ▶ A personalized calendar
 - ▶ Programmed in Perl 4..5 (and now Raku)
 - ▶ Started in 1999, continuously improved since
 - ▶ Done annually
 - ▶ Plan to make most code public
 - ▶ Code is still ugly, but getting better
 - ▶ early days: get in fast, do the job, and get the heck out!
 - ▶ now, with maturity, I've settled in with Raku

Dynamite features

The following DRY (don't repeat yourself) features make my calendar special:

- ▶ Flat file (special ASCII format) database of special events (birthdays, anniversaries, etc.)
 - ▶ number of years since original date shown on annual calendar
- ▶ Easy personalization for gifts

Pass around

Multi-color calendar

I take the final pdf file to a local print shop (I have a recommendation) and get it spiral bound with clear cover and black end sheet, and printed in color for about \$20.

Of course you can use your own color printer, too.

Note I used to get a hole punched for hanging, but that wasn't easy to do cleanly and accurately. Last year we tried using a bulldog clip on a nail and it works fine!

Current Features

Current features:

- ▶ Federal holidays: name in red, actual date cell bordered in red
- ▶ other holidays: name in red
- ▶ birthdays: name in blue, date cell bordered in blue
- ▶ anniversaries: names in green
- ▶ other events: name in red (**not yet customizable**)
- ▶ **phases of the moon and time**
- ▶ **sunrise/sunset**
- ▶ season changes
- ▶ Daylight Saving Time start/stop

Note features in bold require local geo-position data (the default is the center of the Gulf Breeze City Hall).

Planned features

Planned features:

- ▶ customizable special events
- ▶ data on facing page:
 - ▶ three-month view
 - ▶ **sunrise/sunset**
 - ▶ **moonrise/moonset**
 - ▶ **astronomical events**
 - ▶ pictures
 - ▶ **tide charts**

Note features in bold require local geo-position data.

Requirements

Until I can go public with the code, I can produce your calendar in the current format, but I require from you:

- ▶ a single word, unique title for each calendar, e.g., 'NAME'
- ▶ specially formatted and named ASCII text files:
 1. a list of birthdays
 - ▶ "birthdays.txt"
 2. a list of anniversaries
 - ▶ "anniversaries.txt"
 3. an optional list of other, special events
 - ▶ "other-events.txt"
 4. a list of lines for the cover [default is available]
 - ▶ "front-cover-words.txt"
 5. an optional list of monthly sayings [default is available]
 - ▶ "monthly-quotes.txt"

ASCII text files

An ASCII (American Standard Code for Information Interchange) text file contains only characters from the ASCII set of 128 7-bit characters. We will be using the printed characters, mostly A-Z, a-z, 0-9, and the special and punctuation characters found on most US keyboards.

See the ASCII character set on a Linux system with the command 'man ascii' or search for ASCII on the internet.

On a Microsoft (MS) Windows computer a suitable app is Microsoft's **Notepad** (but freely available **Notepad++** is a much better app). MS's **Wordpad** is also usable, but you have to be careful how you save the file after changes.

Note MS **Word** is **NOT** suitable because it produces files that add formatting codes in addition to the text you enter. Such files may be ASCII text, but not the exact text you entered.

Example inputs

Let's look at some files and formats so you can start working on your inputs. But first, how can you get your inputs to me?

The preferred way is to send me files via e-mail attachment or arrange a mutual time to meet at the library and copy files to or from a thumb drive.

My e-mail:

<tbrowder@acm.org>

Input files

Assume you want to modify all you can, and you select a name of 'sally', you need to create four files in a directory (folder) named 'sally':

```
sally/  
  anniversaries.txt  
  birthdays.txt  
  front-cover-words.txt  
  monthly-quotes.txt
```

Note on MS Windows you will have to change the default directory (folder) view in order to see the ".txt" extension (file suffix).

You can then zip up the 'sally' directory and send the zip archive to me.

File contents and format

Each file has some common format rules for comments, i.e., text that is ignored by the parser (part of my program).

1. In all files, text on a line from a pound sign ('#', formally known as 'octothorpe') to the end of the line is ignored.
2. Blank lines are ignored.

Those rules are very common in program input files for many applications.

Additionally, an event entry can have a user-selected color prefacing the line by text as in this example:

```
color:RED 22 feb Jane
```

where the 'color' value is a color name and is in upper-case for bold and lower-case for normal font for the colored text.

Current colors

Default colors

- ▶ red - holidays
- ▶ BLUE - birthdays
- ▶ green - anniversaries
- ▶ black - other events

Other colors

I can fairly easily add more color names if desired. They should be standard HTML names found in this list:

<https://www.w3schools.com/colors/colors_names.asp>

Then you can use them as custom colors for any user-entered event as shown in the following example:

color:PURPLE 14 jun Jason graduation

Front cover words

File: front-cover-words.txt

```
# This file contains the front cover personalization
# for the personalized wall calendar.
@
# The following two lines will be typeset on the front
# cover of the wall calendar AFTER the identifying
# title:
#
#     The Year 2016
#
# The first line will be in bold and the second line
# will be typeset in a regular and smaller typeface.
```

```
A Special Calendar for a Special Person
To Missy with love, from Tom
```

Events files

1. File: birthdays.txt
2. File: anniversaries.txt
3. File: other-events.txt

Year conversions:

```
# {1999} <= year number in curly braces results in a
#         cardinal number (1, 2, etc.)
# [1999] <= year number in square brackets results
#         in an ordinal number (1st, 2nd, etc.)
```


Events files (2)

```
# The following examples assume a calendar year of 2022:
#
# This entry
11 may Kim {1999}
# will be formatted as:
#   'Kim b'day, 23'
#
# This entry
11 may Kim [1999]
# will be formatted as:
#   'Kim b'day, 23rd'
```

Anniversaries file

File: anniversaries.txt

Same input format as birthdays, but they get formatted on the calendar with different wording and color

Other-events file

File: other-events.txt

Same input format as birthdays, but they get formatted on the calendar with different wording and color

Monthly sayings file

File: monthly-quotes.txt

```
# This file contains monthly quotations for the  
# personalized wall calendar.
```

```
# Note that the quotations are one per line and  
# the longest one (122 characters) is as long as  
# my calendar format can handle (one printed  
# line). Note also that they are printed on each  
# sheet in top to bottom order from January  
# through December.
```

```
# The following (default) Bible verses are from  
# the King James Version (KJV):
```

```
For by me [the Lord] thy days shall be multiplied...
```

Calendar production

I take your inputs and run them with my calendar program.
Understand that I have not run any other person's files through my program, so there may be **bugs** in my program or your input files.
As the old saying goes, “garbage in, garbage out” or **GIGO**.

Data flow

1. You produce input files and send them to me
2. I input the files to my program **make-calendar.raku**
3. Output is a **PostScript** file
4. If there any problems with the input files I correct the problem and send them back to you for approval.
5. Repeat steps 1 through 4 until there no problems.
6. The file from successful step 3 is then input to program **ps2pdf**
7. Output from is a **PDF** file which I send to you
8. If you are happy with that view, then print the **PDF** file or send it to a local print shop for printing and binding

Summary

We've looked at one personal project done on Linux with Raku that would be somewhat harder to do on Windows (IMHO):

I'm happy to help with any and all who want a personalized version. Contact me via e-mail (please mention Friends and Family in the subject):

<**tbrowder@acm.org**>

If there are many takers, I can arrange to be at the library during set times to help.

Shameless plug

If you are interested, or know anyone who is interested in any of the following topics, please let me know:

- ▶ Computer programming
- ▶ Linux
- ▶ Database programming
- ▶ Other tech subjects?

I would love to present one or more sessions on any or all of them.

Finally

That's all **Doc!**