

German holidays in the calendar software REMIND

Gunther Reissig

Abstract

REMIND is a calendar software running under UNIX/Linux and offering full control from either command line or graphical front-end. I present code, in the form of a package consisting of several input files for REMIND, which declares and handles official German holidays (“gesetzliche Feiertage”). The package distinguishes itself from previously known such code in that it permits to conveniently select holidays official only in parts of Germany or only in some interval of time, and globally and safely OMITs holidays. I also present and include additional code that may be of independent interest.

I. QUICK START

A. Prerequisites

Your computer is assumed to run under UNIX/Linux, and you yourself, to be familiar with working from the command line. Both REMIND [1], [2] and `subversion` [3] should be installed, which can be achieved, e.g. under GNU Linux by running the following command:

```
> sudo apt install remind subversion
```

Verify that the version of REMIND is 03.03.02 or newer, e.g. by running the command `remind` without arguments and inspecting the output. You could get a newer version from [2] and could install it.

B. Creating a working copy of the package

Create a working copy of the package in a local directory of your choice:

```
> svn checkout --username REMINDpublic \  
> https://subversion.unibw.de/lflagure/REMIND/ \  
> <full path to working copy of REMIND>
```

If asked for it, you would use the password `IuseREMIND`. You have only read access to the repository.

C. License

The package is free software, licensed under the terms of the MIT License, in one case with a slight modification. The file `include/lang/de.rem`, which is part of the software REMIND and has just been modified by me, is licensed under GNU GPL 2. For terms and conditions, inspect files in the working copy, and in the case of `include/lang/de.rem`, additionally consult [2].

D. Your first calendar sheet

Change into the working copy of the package and run the following command:

```
> remind -q@2m -pc test.rem nov 1 2022 | rem2pdf -l -mA4 >test.pdf
```

The produced file `test.pdf` should look similar to Fig. 1. You may also try

```
> remind -q@2m -c test.rem nov 1 2022
```

to produce analogous output on your terminal.

November 2022						
Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
	1 Allerheiligen	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16 Buß- und Betttag	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Figure 1. Your first calendar sheet, produced from file `test.rem`; see Section I.

What you see in Fig. 1 is two holidays, “Allerheiligen” and “Buß- und Betttag”, the former being printed in red color with shaded calendar day, and the latter printed in normal font with no shading. Indeed, “Allerheiligen” is an official holiday (“gesetzlicher Feiertag”) in mainly catholic parts of Bavaria, the region selected in line 17 of the file `test.rem`,

```
SET DE_BY_CATHOLIC "this is my place"
```

with red color being selected in line 18. In contrast, “Buß- und Betttag” is not an official holiday in the selected region, but it is so in the state of Saxony. The latter state is taken into account by REMIND because of lines 23 – 24 of the aforementioned file, where it is asked that all holidays official somewhere in Germany be printed in normal font, not repeating any holidays already printed.

You might want to experiment a little bit. If you change your region to Saxony, by replacing the variable name `DE_BY_CATHOLIC` with `DE_SN`, “Buß- und Betttag” will become official (red, shaded) and “Allerheiligen” official only somewhere else (normal font, no shading). Alternatively, if you stay with `DE_BY_CATHOLIC` but instead change the year 2022 in your `remind` command into 1994, both holidays will become official, simply because this is what Bavarian law stipulates for the year 1994.

To actually use the package, copy the contents of the file `test.rem` into your personal reminders file, before any of your reminders. Check comments in the file `libs/regions` to identify the name of the variable representing the region you live in, and replace the variable name `DE_BY_CATHOLIC` accordingly. You might also need to adjust directory names of `include` commands and to move the last three lines of the file `test.rem` toward the end of your personal reminders file.

This is basically it. You might want to return to the working copy of Section I-B, e.g. once per year, to run the command `svn up` which would download the latest update of the package.

II. OFFICIAL GERMAN HOLIDAYS

Official German holidays (“gesetzliche Feiertage”) are defined by state law. Therefore, the notion of “official holiday” not only varies with time, but also depends on the location within the country. The package takes into account that variation, for the time from 1990-10-3 onward. In this section, I discuss files 1 through 4 of the package, see Tab. I, which are strictly needed to handle official German holidays. Throughout this report, *iff* means “if and only if”.

Table I
FILES INCLUDED IN PACKAGE.

File	Type	Comments
1 <code>libs/holidays_official_DE</code>	subroutine	Provides and handles official German holidays.
2 <code>libs/holidays_process</code>	subroutine	Uniform interface for declaring holidays.
3 <code>libs/regions</code>	library	Selection of regions.
4 <code>libs/first</code>	library	Basics.
5 <code>test.rem</code>	subroutine	Simple use case of <code>libs/holidays_official_DE</code> .
6 <code>libs/holidays_unofficial_DE</code>	subroutine	Some unofficial holidays in Germany.
7 <code>include/lang/de.rem</code>	library	Support for the German language.
8 <code>libs/daylightST_DE</code>	subroutine	Daylight saving time in Germany.
9 <code>libs/earthseasons</code>	subroutine	Earth seasons.
10 <code>libs/colors</code>	subroutine	X color definitions.

A. Libraries `libs/first` and `libs/regions`

The file `libs/first` provides basic functionality including Boolean constants `true` and `false`, constants defining priorities, and constants related to the types `DATE` and `DATETIME`. In particular, `minDATE` and `maxDATE` is the minimal and maximal, respectively, feasible `DATE` in `REMIND`. The file also provides functions for generating error messages and warnings.

The file `libs/regions` provides functionality for selecting regions within Germany, in which selection is by definition of suitable variables. The user function `_regions_Intersects(region)` returns `true` iff the enabled region intersects the region corresponding to the `STRING` `region`.

There are three hierarchy levels of variables, corresponding to holidays that are official everywhere in Germany (`DE`), everywhere in a certain state (`DE_??`), and in a certain place (`DE_??_?*`). As for states, variable names `DE_??` coincide with *country codes* used by other software [4], and there are four variables `DE_BY_AUGSBURG`, `DE_BY_CATHOLIC`, `DE_SN_SORBIAN` and `DE_TH_CATHOLIC` corresponding to certain places. Variable names work in a hierarchical manner, but there is no hierarchy among places even if geometrically or politically, one place is a subset of another. Hence, e.g. if you live in the city of Augsburg in the state of Bavaria, you would need to define both variables `DE_BY_AUGSBURG` and `DE_BY_CATHOLIC`. Indeed, if you merely define `SET DE_BY_AUGSBURG "sth"`, then `_regions_Intersects("DE_BY_AUGSBURG")`, `_regions_Intersects("DE_BY")`, as well as `_regions_Intersects("DE")` will all evaluate to `true`, respecting the aforementioned hierarchy, while `_regions_Intersects("DE_BY_CATHOLIC")` would still evaluate to `false`.

Generally, defining multiple variables works as logical disjunction (union of geometrical or political regions). There is one exception to the logic described above, in that defining the variable `DE_SOMEWHERE` selects holidays that are official *somewhere* in Germany.

Both files are to be considered libraries in that they are parsed only once to make their functionality available. For more details, consult the files themselves.

B. Subroutine `libs/holidays_process`

The file `libs/holidays_process` provides a uniform interface for declaring holidays and allows you to mark holidays in your calendar and to `OMIT` them, taking into account time intervals of validity of holidays. `OMITs` are globally known within ± 174 days from the occurrence of the holiday. For simple use cases, see comments in the file itself as well as code in file `libs/holidays_official_DE`.

The file is to be considered a subroutine, meaning that it has to be `included` each time its functionality is needed. The most important parameters are given in Tab. II. The date specification of a holiday consists of the two functions `_holidays_process_actDateSpec` and `_holidays_process_dateCond`, and the `DATE` interval `[holidays_process_min_actDate, holidays_process_max_actDate]`. A holiday is considered to occur on `DATE` `d` iff the following three conditions hold:

Table II
PARAMETERS OF SUBROUTINE `libs/holidays_process`.

Parameter	Type	Default	Comments
<code>holidays_omitactQ</code>	BOOL (in)	<code>true</code>	holiday will be OMITted iff <code>true</code>
<code>_holidays_process_actMSG(d)</code>	coercible to STRING (in)	-	name of holiday to be printed if it happens to occur on the DATE <code>d</code>
<code>_holidays_process_actDateSpec(y)</code>	coercible to STRING (in)	-	actual date specification of holiday
<code>_holidays_process_dateCond(d)</code>	BOOL (in)	-	condition on validity of holiday
<code>holidays_process_min_actDate</code>	DATE (in)	<code>minDATE</code>	condition on validity of holiday
<code>holidays_process_max_actDate</code>	DATE (in)	<code>maxDATE</code>	condition on validity of holiday
<code>holidays_process_actHolidayQ</code>	BOOL (in/out)	<code>false</code>	<code>true</code> on termination iff <code>true</code> initially or holiday occurs on <code>today()</code> and has been processed

- (i) `d` satisfies the date specification `_holidays_process_actDateSpec(year(d))`.
- (ii) `_holidays_process_dateCond(d)` evaluates to `true`.
- (iii) `holidays_process_min_actDate ≤ d ≤ holidays_process_max_actDate`.

The date specification must satisfy the following requirements.

- Either A) `_holidays_process_actDateSpec(y)` specifies a day of the week, or B) two consecutive occurrences of the holiday are at least 350 days apart.
- If case A) in one year `y`, then the functions `_holidays_process_actDateSpec` and `_holidays_process_dateCond` are constant.
- required for `_holidays_process_actDateSpec(y)`:
 - Case A): Pure date specification (no backward scanning, no advance warning, no time, no OMITFUNC nor any other keywords).
 - Case B): Pure date specification plus possibly backward scanning and advance warning parameters, plus possibly duration specified by `THROUGH`; no time, no OMITFUNC nor any other keywords.

This subroutine is not strictly needed. However, without it, achieving equivalent results would require tailoring `REMIND` code to the specific case of a holiday at hand, and would also require detailed knowledge of algorithms used by `REMIND`. Some of the difficulties are discussed in comments in the file `libs/holidays_process`. In contrast, the subroutine provides a uniform interface which applies to all kinds of holidays, does not require detailed knowledge about any algorithms, and allows you to focus on correctly specifying holiday dates and validity intervals.

C. Subroutine `libs/holidays_official_DE`

The file provides a database of official German holidays from 1990-103 and allows you to mark holidays in your calendar and to globally OMIT them, taking into account time intervals of validity as well as regions of validity. The file `test.rem` contains a simple use case; see Section I-D.

The file is to be considered a subroutine whose most important parameters are given in Tab. III. Only occurrences of holidays from `max(date(1990,10,3),holidays_minValidDate)` to `holidays_maxValidDate` are taken into account. The subroutine also respects variables selecting regions as detailed in Section II-A. For even more details, see comments in file.

III. ADDITIONAL FILES

Files 7 through 8 in Tab. I provide further support to German users of `REMIND`, while files 9 and 10 could be of use to any user. See the files themselves for more information about their functionality.

Table III
PARAMETERS OF SUBROUTINE `libs/holidays_official_DE`.

Parameter	Type	Default	Comments
<code>holidays_omitactQ</code>	BOOL (in/out)	<code>true</code>	holidays will be OMITted iff true
<code>holidays_color</code>	STRING (in)	<code>\$DefaultColor</code>	color used in printing holidays
<code>holidays_minValidDate</code>	DATE (in)	<code>minDATE</code>	see Section II-C
<code>holidays_maxValidDate</code>	DATE (in)	<code>maxDATE</code>	see Section II-C
<code>holidays_actHolidayQ</code>	BOOL (out)	-	<code>true</code> iff <code>today()</code> is a holiday in the selection region and has been processed

ACKNOWLEDGMENT

I thank Dianne Skoll for her endless patience with my questions.

REFERENCES

- [1] D. F. Skoll, "Remind: The UNIX geek's ultimate calendar," Talk, Linux-Ottawa meeting, 6 Nov. 2007.
- [2] D. Skoll, "Remind," <https://dianne.skoll.ca/projects/remind/> (Sep. 21, 2022).
- [3] B. Collins-Sussman, B. W. Fitzpatrick, and C. M. Pilato, *Version Control with Subversion*, for subversion 1.7 ed., 2011, <http://svnbook.red-bean.com/>.
- [4] T. Esken, *GNU gcal: An extended calendar program*, 4.1 ed., Jun. 2000, <https://www.gnu.org/software/gcal/>.
- [5] G. Reissig, "German holidays in the calendar software REMIND," Tech. Rep., 17 Oct. 2022. <https://www.reiszig.de/gunther/pubs/Reissig22b.html>