Description Syntax Remarks and examples Also see

Description

Stata provides user-definable business calendars. Business calendars are provided by StataCorp and by other users, and you can write your own. You can also create a business calendar automatically from the current dataset with the `bcal create` command; see [D] **bcal**. This entry concerns writing your own business calendars.

See [D] **Datetime business calendars** for an introduction to business calendars.

Syntax

Business calendar *calname* and corresponding display format `%tb`*calname* are defined by the text file *calname*`.stbcal`, which contains the following:

```
* comments
version version_of_stata
purpose "text"
dateformat { ymd | ydm | myd | mdy | dym | dmy }
range date date
centerdate date
[ from { date | . } to { date | . }: ] omit ... [ if ]
...
...
```

where

`omit` ... may be

```
omit date pdate [ and pmlist ]
omit dayofweek dowlist
omit dowinmonth pm# dow [ of monthlist ] [ and pmlist ]
```

[*if*] may be

```
if restriction [ & restriction ... ]
```

restriction is one of

```
dow(dowlist)
month(monthlist)
year(yearlist)
```

date is a date written with the *year*, *month*, and *day* in the order specified by `dateformat`. For instance, if `dateformat` is `dmy`, a *date* can be `12apr2013`, `12-4-2013`, or `12.4.2013`.

pdate is a *date* or it is a *date* with character * substituted where the year would usually appear. If `dateformat` is `dmy`, a *pdate* can be `12apr2013`, `12-4-2013`, or `12.4.2013`; or it can be `12apr*`, `12-4-*`, or `12.4.*`. `12apr*` means the 12th of April across all years.

dow is a day of the week, in English. It may be abbreviated to as few as 2 characters, and capitalization is irrelevant. Examples: `Sunday`, `Mo`, `tu`, `Wed`, `th`, `Friday`, `saturday`.

dowlist is a *dow*, or it is a space-separated list of one or more *dow*s enclosed in parentheses. Examples: `Sa`, `(Sa)`, `(Sa Su)`.

month is a month of the year, in English, or it is a month number. It may be abbreviated to the minimum possible, and capitalization is irrelevant. Examples: `January`, `2`, `Mar`, `ap`, `may`, `6`, `Jul`, `aug`, `9`, `Octob`, `nov`, `12`.

monthlist is a *month*, or it is a space-separated list of one or more *month*s enclosed in parentheses. Examples: `Nov`, `(Nov)`, `11`, `(11)`, `(Nov Dec)`, `(11 12)`.

year is a 4-digit calendar year. Examples: `1872`, `1992`, `2013`, `2050`.

yearlist is a *year*, or it is a space-separated list of one or more *year*s enclosed in parentheses. Examples: `2013`, `(2013)`, `(2013 2014)`.

pm# is a nonzero integer preceded by a plus or minus sign. Examples: `-2`, `-1`, `+1`. *pm#* appears in `omit dowinmonth` *pm# dow* `of` *monthlist*, where *pm#* specifies which *dow* in the month. `omit dowinmonth +1 Th` means the first Thursday of the month. `omit dowinmonth -1 Th` means the last Thursday of the month.

pmlist is a *pm#*, or it is a space-separated list of one or more *pm#*s enclosed in parentheses. Examples: `+1`, `(+1)`, `(+1 +2)`, `(-1 +1 +2)`. *pmlist* appears in the optional `and` *pmlist* allowed at the end of `omit` *date* and `omit dowinmonth`, and it specifies additional dates to be omitted. `and +1` means and the day after. `and -1` means and the day before.

Remarks and examples

Remarks are presented under the following headings:

Introduction

A business calendar is a regular calendar with some dates crossed out, such as

November 2011

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	X
X	7	8	9	10	11	X
X	14	15	16	17	18	X
X	21	22	23	X	X	X
X	28	29	30			

The purpose of the stbcal-file is to

1. Specify the range of dates covered by the calendar.
2. Specify the particular date that will be encoded as date 0.
3. Specify the dates from the regular calendar that are to be crossed out.

The stbcal-file for the above calendar could be as simple as

begin example_1.stbcal

```
version 19.5      // (or version 19 if you do not have StataNow)
range 01nov2011 30nov2011
centerdate 01nov2011
omit date  5nov2011
omit date  6nov2011
omit date 12nov2011
omit date 13nov2011
omit date 19nov2011
omit date 20nov2011
omit date 24nov2011
omit date 25nov2011
omit date 26nov2011
omit date 27nov2011
```

end example_1.stbcal

In fact, this calendar can be written more compactly because we can specify to omit all Saturdays and Sundays:

begin example_2.stbcal

```
version 19.5      // (or version 19 if you do not have StataNow)
range 01nov2011 30nov2011
centerdate 01nov2011
omit dayofweek (Sa Su)
omit date 24nov2011
omit date 25nov2011
```

end example_2.stbcal

In this particular calendar, we are omitting 24nov2011 and 25nov2011 because of the American Thanksgiving holiday. Thanksgiving is celebrated on the fourth Thursday of November, and many businesses close on the following Friday as well. It is possible to specify rules like that in stbcal-files:

begin example_3.stbcal
```
version 19.5      // (or version 19 if you do not have StataNow)
range 01nov2011 30nov2011
centerdate 01nov2011
omit dayofweek (Sa Su)
omit dowinmonth +4 Th of Nov and +1
```
end example_3.stbcal

Understand that this calendar is an artificial example, and it is made all the more artificial because it covers so brief a period. Real stbcal-files cover at least decades, and some cover centuries.

Concepts

You are required to specify four things in an stbcal-file:

1. the version of Stata being used,
2. the range of the calendar,
3. the center date of the calendar, and
4. the dates to be omitted.

Version.

You specify the version of Stata to ensure forward compatibility with future versions of Stata. If your calendar starts with the line `version 19.5` or, if you do not have StataNow, `version 19.0`, future versions of Stata will know how to interpret the file even if the definition of the stbcal-file language has greatly changed.

Range.

A calendar is defined over a specific range of dates, and you must explicitly state what that range is. When you or others use your calendar, dates outside the range will be considered invalid, which usually means that they will be treated as missing values.

Center date.

Stata stores dates as integers. In a calendar, 57 might stand for a particular date. If it did, then $57 - 1 = 56$ stands for the day before, and $57 + 1 = 58$ stands for the day after. The previous statement works just as well if we substitute $-12{,}739$ for 57, and thus the particular values do not matter except that we must agree upon what values we wish to standardize because we will be storing these values in our datasets.

The standard is called the center date, and here center does not mean the date that corresponds to the middle of your calendar. It means the date that corresponds to the center of integers, which is to say, 0. You must choose a date within the range as the standard. The particular date you choose does not matter, but most authors choose easily remembered ones. Stata's built-in `%td` calendar uses 01jan1960, but that date will probably not be available to you because the center date must be a date on the business calendars, and most businesses were closed on 01jan1960.

It will sometimes happen that you will want to expand the range of your calendar in the future. Today, you make a calendar that covers, say 1990 to 2020, which is good enough for your purposes. Later, you need to expand the range, say back to 1970 or forward to 2030, or both. When you update your calendar, do not change the center date. This way, your new calendar will be backward compatible with your previous one.

Omitted dates.

Obviously you will need to specify the dates to be omitted. You can specify the exact dates to be omitted when need be, but whenever possible, specify the rules instead of the outcome of the rules. Rules change, so learn about the `from`/`to` prefix that can be used in front of `omit` commands. You can code things like

```
from 01jan1960 to 31dec1968: omit ...
from 01jan1979 to .: omit ...
```

When specifying `from`/`to`, `.` for the first date is synonymous with the opening date of the range. `.` for the second date is synonymous with the closing date.

The preliminary commands

Stbcal-files should begin with these commands:

```
version version_of_stata
purpose "text"
dateformat { ymd | ydm | myd | mdy | dym | dmy }
range date date
centerdate date
```

`version` *version_of_stata*

You could specify `version 19.5` or, if you do not have StataNow, `version 19.0`. Better still, type command `version` in Stata to discover the version of Stata you are currently using. Specify that version, and be sure to look at the documentation so that you use the modern syntax correctly.

`purpose "`*text*`"`

This command is optional. The purpose of `purpose` is not to make comments in your file. If you want comments, include those with a `*` in front. The `purpose` sets the text that `bcal describe` *calname* will display.

`dateformat` { `ymd` | `ydm` | `myd` | `mdy` | `dym` | `dmy` }

This command is optional. `dateformat ymd` is assumed if not specified. This command has nothing to do with how dates will look when variables are formatted with `%tb`*calname*. This command specifies how you are typing dates in this stbcal-file on the subsequent commands. Specify the format that you find convenient.

`range` *date date*

The date range was discussed in *Concepts*. You must specify it.

`centerdate` *date*

The centering date was discussed in *Concepts*. You must specify it.

The omit commands: from/to and if

An stbcal-file usually contains multiple `omit` commands. The `omit` commands have the syntax

[`from` { *date* | `.` } `to` { *date* | `.` }`:`] `omit` ... [*if*]

That is, an `omit` command may optionally be preceded by `from`/`to` and may optionally contain an `if` at the end.

When you do not specify from/to, results are the same as if you specified

```
from . to .:  omit ...
```

That is, the omit command applies to all dates from the beginning to the end of the range. In *Introduction*, we showed the command

```
omit dowinmonth +4 Th of Nov and +1
```

Our sample calendar covered only the month of November, but imagine that it covered a longer period and that the business was open on Fridays following Thanksgiving up until 1998. The Thanksgiving holidays could be coded

```
from . to 31dec1997: omit dowinmonth +4 Th of Nov
from 01jan1998 to .: omit dowinmonth +4 Th of Nov and +1
```

The same holidays could also be coded

```
omit dowinmonth +4 Th of Nov
from 01jan1998 to .: omit dowinmonth +4 Th of Nov and +1
```

We like the first style better, but understand that the same dates can be omitted from the calendars multiple times and for multiple reasons, and the result is still the same as if the dates were omitted only once.

The optional if also determines when the omit statement is operational. Let's think about the Christmas holidays. Let's say a business is closed on the 24th and 25th of December. That could be coded

```
omit date 24dec*
omit date 25dec*
```

although perhaps that would be more understandable if we coded

```
from . to .: omit date 24dec*
from . to .: omit date 25dec*
```

Remember, from . to . is implied when not specified. In any case, we are omitting 24dec and 25dec across all years.

Now consider a more complicated rule. The business is closed on the 24th and 25th of December if the 25th is on Tuesday, Wednesday, Thursday, or Friday. If the 25th is on Saturday or Sunday, the holidays are the preceding Friday and the following Monday. If the 25th is on Monday, the holidays are Monday and Tuesday. The rule could be coded

```
omit date 25dec* and -1      if dow(Tu We Th Fr)
omit date 25dec* and (-2 -1) if dow(Sa)
omit date 25dec* and (-3 -2) if dow(Su)
omit date 25dec* and +1      if dow(Mo)
```

The if clause specifies that the omit command is only to be executed when 25dec* is one of the specified days of the week. If 25dec* is not one of those days, the omit statement is ignored for that year. Our focus here is on the if clause. We will explain about the and clause in the next section.

Sometimes, you have a choice between using from/to or if. In such cases, use whichever is convenient. For instance, imagine that the Christmas holiday rule for Monday changed in 2011 and 2012. You could code

```
from . to 31dec2010: omit date 25dec* and +1 if dow(Mo)
from 01jan2011 to .: omit date ... if dow(Mo)
```

or

```
omit date 25dec* and +1 if dow(Mo) & year(2007 2008 2009 2010)
omit date ... if dow(Mo) & year(2011 2012)
```

Generally, we find `from`/`to` more convenient to code than `if year()`.

The omit commands: and

The other common piece of syntax that shows up on `omit` commands is `and` *pmlist*. We used it above in coding the Christmas holidays,

```
omit date 25dec* and -1      if dow(Tu We Th Fr)
omit date 25dec* and (-2 -1) if dow(Sa)
omit date 25dec* and (-3 -2) if dow(Su)
omit date 25dec* and +1      if dow(Mo)
```

`and` *pmlist* specifies a list of days also to be omitted if the date being referred to is omitted. The extra days are specified as how many days they are from the date being referred to. Please excuse the inelegant "date being referred to", but sometimes the date being referred to is implied rather than stated explicitly. For this problem, however, the date being referred to is `25dec` across a number of years. The line

```
omit date 25dec* and -1      if dow(Tu We Th Fr)
```

says to omit `25dec` and the day before if `25dec` is on a Tuesday, Wednesday, etc. The line

```
omit date 25dec* and (-2 -1) if dow(Sa)
```

says to omit `25dec` and two days before and one day before if `25dec` is Saturday. The line

```
omit date 25dec* and (-3 -2) if dow(Su)
```

says to omit `25dec` and three days before and two days before if `25dec` is Sunday. The line

```
omit date 25dec* and +1      if dow(Mo)
```

says to omit `25dec` and the day after if `25dec` is Monday.

Another `omit` command for solving a different problem reads

```
omit dowinmonth -1 We of (Nov Dec) and +1 if year(2009)
```

Please focus on the `and +1`. We are going to omit the date being referred to and the date after if the year is 2009. The date being referred to here is `-1 We of (Nov Dec)`, which is to say, the last Wednesday of November and December.

The omit commands: omit date

The full syntax of `omit date` is

[`from` { *date* | . } `to` { *date* | . } :] `omit date` *pdate* [`and` *pmlist*] [*if*]

You may omit specific dates,

```
omit date 25dec2010
```

or you may omit the same date across years:

```
omit date 25dec*
```

The omit commands: omit dayofweek

The full syntax of `omit dayofweek` is

[`from` { *date* | . } `to` { *date* | . }:] `omit dayofweek` *dowlist* [*if*]

The specified days of week (Monday, Tuesday, ...) are omitted.

The omit commands: omit dowinmonth

The full syntax of `omit dowinmonth` is

[`from` { *date* | . } `to` { *date* | . }:] `omit` *pm# dow* [`of` *monthlist*] [`and` *pmlist*] [*if*]

`dowinmonth` stands for day of week in month and refers to days such as the first Monday, second Monday, ..., next-to-last Monday, and last Monday of a month. This is written as `+1 Mo`, `+2 Mo`, ..., `-2 Mo`, and `-1 Mo`.

Creating stbcal-files with bcal create

Business calendars can be obtained from your Stata installation or from other Stata users. You can also write your own business calendar files or use the `bcal create` command to automatically create a business calendar from the current dataset. With `bcal create`, business holidays are automatically inferred from gaps in the dataset, or they can be explicitly defined by specifying the `if` and `in` qualifiers, as well as the `excludemissing()` option. You can also edit business calendars created with `bcal create` or obtained from other sources. It is advisable to use `bcal load` or `bcal describe` to verify that a business calendar is well constructed and remains so after editing.

See [D] **bcal** for more information on `bcal create`.

Where to place stbcal-files

Stata automatically searches for stbcal-files in the same way it searches for ado-files. Stata looks for ado-files and stbcal-files in the official Stata directories, your site's directory (`SITE`), your current working directory (`.`), your personal directory (`PERSONAL`), and your directory for materials written by other users (`PLUS`). On this writer's computer, these directories happen to be

```
. sysdir
   STATA:  C:\Program Files\Stata19\
    BASE:  C:\Program Files\Stata12\ado\base\
    SITE:  C:\Program Files\Stata19\ado\site\
    PLUS:  C:\ado\plus\
PERSONAL:  C:\ado\personal\
OLDPLACE:  C:\ado\
```

Place calendars that you write into `.`, `PERSONAL`, or `SITE`. Calendars you obtain from others using `net` or `ssc` will be placed by those commands into `PLUS`. See [P] **sysdir**, [R] **net**, and [R] **ssc**.

How to debug stbcal-files

Stbcal-files are loaded automatically as they are needed, and because this can happen anytime, even at inopportune moments, no output is produced. If there are errors in the file, no mention is made of the problem, and thereafter Stata simply acts as if it had never found the file, which is to say, variables with `%tb`*calname* formats are displayed in `%g` format.

You can tell Stata to load a calendar file right now and to show you the output, including error messages. Type

```
. bcal load calname
```

It does not matter where *calname*`.stbcal` is stored, Stata will find it. It does not matter whether Stata has already loaded *calname*`.stbcal`, either secretly or because you previously instructed the file be loaded. It will be reloaded, you will see what you wrote, and you will see any error messages.

Ideas for calendars that may not occur to you

Business calendars obviously are not restricted to businesses, and neither do they have to be restricted to days.

Say you have weekly data and want to create a calendar that contains only Mondays. You could code

begin `mondays.stbcal`

```
version 19.5      // (or version 19 if you do not have StataNow)
purpose "Mondays only"
range 04jan1960 06jan2020
centerdate 04jan1960
omitdow (Tu We Th Fr Sa Su)
```

end `mondays.stbcal`

Say you have semimonthly data and want to include the 1st and 15th of every month. You could code

begin `smnth.stbcal`

```
version 19.5      // (or version 19 if you do not have StataNow)
purpose "Semimonthly"
range 01jan1960 15dec2020
centerdate 01jan1960
omit date 2jan*
omit date 3jan*
.
.
omit date 14jan*
omit date 16jan*
.
.
omit date 31jan*
omit date  2feb*
.
.
```

end `smnth.stbcal`

Forgive the ellipses, but this file will be long. Even so, you have to create it only once.

As a final example, say that you just want Stata's `%td` dates, but you wish they were centered on 01jan1970 rather than on 01jan1960. You could code

begin `rectr.stbcal`

```
version 19.5      // (or version 19 if you do not have StataNow)
Purpose "%td centered on 01jan1970"
range 01jan1800 31dec2999
centerdate 01jan1970
```

end `rectr.stbcal`

Also see

[D] **bcal** — Business calendar file manipulation

[D] **Datetime business calendars** — Business calendars

[D] **Datetime** — Date and time values and variables