bcal — Business calendar file manipulation[+]

[+]This command includes features that are part of StataNow.

Description

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

`bcal check` lists the business calendars used by the data in memory, if any.

`bcal dir` *pattern* lists filenames and directories of all available business calendars matching *pattern*, or all business calendars if *pattern* is not specified.

`bcal describe` *calname* presents a description of the specified business calendar.

`bcal load` *calname* loads the specified business calendar. Business calendars load automatically when needed, and thus use of `bcal load` is never required. `bcal load` is used by programmers writing their own business calendars. `bcal load` *calname* forces immediate loading of a business calendar and displays output, including any error messages due to improper calendar construction.

`bcal webcopy` *webcalname* is part of StataNow. It copies a business calendar from https://www.stata.com/businesscals/r19/files/ to the working directory.

`bcal create` *filename*`, from(`*varname*`)` creates a business calendar file based on dates in *varname*. Business holidays are inferred from gaps in *varname*. The qualifiers `if` and `in`, as well as the option `excludemissing()`, can also be used to exclude dates from the new business calendar.

Quick start

Create business calendar file `mycal.stbcal` from date variable `tvar` in the dataset in memory

```
bcal create mycal, from(tvar)
```

Same as above, and generate business date variable `newt` formatted as `%tbmycal`

```
bcal create mycal, from(tvar) generate(newt)
```

List directories and filenames of available business calendars

```
bcal dir
```

Describe range, center date, and number of omitted days in business calendar `mycal.stbcal`

```
bcal describe mycal
```

Report any `%tb` formats applied to the variables in memory

```
bcal check
```

Obtain business calendar for the S&P 500 from https://www.stata.com/businesscals/r19/files/

```
bcal webcopy sp500
```

Same as above, but replace business calendar `_sp500.stbcal` if it already exists in the working directory

```
bcal webcopy sp500, replace
```

Menu

Data > Other utilities > Create a business calendar

Data > Other utilities > Manage business calendars

Data > Variables Manager

Syntax

List business calendars used by the data in memory

```
bcal check [varlist] [, rc0]
```

List filenames and directories of available business calendars

```
bcal dir [pattern]
```

Describe the specified business calendar

```
bcal describe calname
```

Load the specified business calendar

```
bcal load calname
```

Copy business calendar webcalname from https://www.stata.com/businesscals/r19/files/

```
bcal webcopy [webcalname] [, bcal_webcopy_options]
```

Create a business calendar from the current dataset

```
bcal create filename [if] [in], from(varname) [bcal_create_options]
```

varlist is a list of variable names to be checked for whether they use business calendars. If not specified, all variables are checked.

pattern is the name of a business calendar possibly containing wildcards * and ?. If *pattern* is not specified, all available business calendar names are listed.

calname is the name of a business calendar either as a name or as a datetime format; for example, *calname* could be `simple` or `%tbsimple`.

webcalname is the name of a business calendar that is available from https://www.stata.com/businesscals/r19/files/. *webcalname* can be one of: `lse`, for London Stock Exchange; `nasdaq`, for NASDAQ; `nyse`, for New York Stock Exchange; `sse`, for Shanghai Stock Exchange; `tosho`, for Tokyo Stock Exchange; `sp500` for the S&P 500; or `stock`, dates used in examples of [TS] **mgarch**.

filename is the name of the business calendar file created by `bcal create`.

bcal_webcopy_options	Description
Main	
`public`	make calendar *webcalname* readable by all
`replace`	may overwrite *webcalname*
`dir`	list calendars available in https://www.stata.com/businesscals/r19/files/

bcal_create_options	Description
Main	
* `from(`*varname*`)`	specify date variable for calendar
`generate(`*newvar*`)`	generate *newvar* containing business dates
`excludemissing(`*varlist* [`, any`]`)`	exclude observations with missing values in *varlist*
`personal`	save calendar file in your `PERSONAL` directory
`replace`	replace file if it already exists
Advanced	
`purpose(`*text*`)`	describe purpose of calendar
`dateformat(ymd | ydm | myd | mdy | dym | dmy)`	specify date format in calendar file
`range(`*fromdate todate*`)`	specify range of calendar
`centerdate(`*date*`)`	specify center date of calendar
`maxgap(`#`)`	specify maximum gap allowed; default is 10 days

*`from(`*varname*`)` is required.

`collect` is allowed with all `bcal` commands; see **[U] 11.1.10 Prefix commands**.

Options

Options are presented under the following headings:

Option for bcal check
Options for bcal webcopy (StataNow)
Options for bcal create

Option for bcal check

Main

`rc0` specifies that `bcal check` is to exit without error (return 0) even if some calendars do not exist or have errors. Programmers can then access the results `bcal check` stores in `r()` to get even more details about the problems. If you wish to suppress `bcal dir`, precede the `bcal check` command with `capture` and specify the `rc0` option if you wish to access the `r()` results.

Options for bcal webcopy (StataNow)

Main

`public` specifies that *webcalname* is made readable for all.

`replace` will replace business calendars in the working directory with *_calendar.stbcal* if the calendar specified in *webcalname* already exists.

dir lists the business calendars available at https://www.stata.com/businesscals/r19/files/. This option cannot be specified with *webcalname*.

Options for bcal create

Main

from(*varname*) specifies the date variable used to create the business calendar. Gaps between dates in *varname* define business holidays. The longest gap allowed can be set with the maxgap() option. from() is required.

generate(*newvar*) specifies that *newvar* be created. *newvar* is a date variable in %tb*calname* format, where *calname* is the name of the business calendar derived from *filename*.

excludemissing(*varlist* [, any]) specifies that the dates of observations with missing values in *varlist* are business holidays. By default, the dates of observations with missing values in all variables in *varlist* are holidays. The any suboption specifies that the dates of observations with missing values in any variable in *varlist* are holidays.

personal specifies that the calendar file be saved in the PERSONAL directory. This option cannot be used if *filename* contains the pathname of the directory where the file is to be saved.

replace specifies that the business calendar file be replaced if it already exists.

Advanced

purpose(*text*) specifies the purpose of the business calendar being created. *text* cannot exceed 63 characters.

dateformat(ymd | ydm | myd | mdy | dym | dmy) specifies the date format in the new business calendar. The default is dateformat(ymd). dateformat() has nothing to do with how dates will look when variables are formatted with %tb*calname*; it specifies how dates are typed in the calendar file.

range(*fromdate todate*) defines the date range of the calendar being created. *fromdate* and *todate* should be in the format specified by the dateformat() option; if not specified, the default ymd format is assumed.

centerdate(*date*) defines the center date of the new business calendar. If not specified, the earliest date in the calendar is assumed. *date* should be in the format specified by the dateformat() option; if not specified, the default ymd format is assumed.

maxgap(#) specifies the maximum number of consecutive business holidays allowed by bcal create. The default is maxgap(10).

Remarks and examples

▷ Example 1: A sample workflow

bcal check reports on any %tb formats used by the data in memory:

```
. bcal check
                %tbsimple:  defined, used by variable
                            mydate
```

bcal dir reports on business calendars available:

```
. bcal dir
  1 calendar file found:
          simple:  C:\Program Files\Stata19\ado\base\s\simple.stbcal
```

bcal describe reports on an individual calendar.

```
. bcal describe simple
  Business calendar simple (format %tbsimple):
    Purpose: Example for manual
      Range: 01nov2011  30nov2011
                 18932      18961   in %td units
                     0         19   in %tbsimple units
     Center: 01nov2011
                 18932              in %td units
                     0              in %tbsimple units
    Omitted:        10              days
                 121.8              approx. days/year
   Included:        20              days
                243.5              approx. days/year
```

bcal load is used by programmers writing new stbcal-files. See [D] **Datetime business calendars creation**.

bcal create creates a business calendar file from the current dataset and describes the new calendar. For example, sp500.dta is a dataset installed with Stata that has daily records on the S&P 500 stock market index in 2001. The dataset has observations only for days when trading took place. A business calendar for stock trading in 2001 can be automatically created from this dataset as follows:

```
. sysuse sp500, clear
(S&P 500)
. bcal create sp500, from(date) purpose(S&P 500 for 2001) generate(bizdate)
  Business calendar sp500 (format %tbsp500):
    Purpose: S&P 500 for 2001
      Range: 02jan2001  31dec2001
                 14977      15340   in %td units
                     0        247   in %tbsp500 units
     Center: 02jan2001
                 14977              in %td units
                     0              in %tbsp500 units
    Omitted:       116              days
                 116.4              approx. days/year
   Included:       248              days
                248.9              approx. days/year
  Notes:
    Business calendar file sp500.stbcal saved.
    Variable bizdate created; it contains business dates in %tbsp500 format.
```

The business calendar file created:

begin sp500.stbcal

```
* Business calendar "sp500" created by -bcal create-
* Created/replaced on 02 Apr 2021

version 19
purpose "S&P 500 for 2001"
dateformat ymd

range 2001jan02 2001dec31
centerdate 2001jan02

omit dayofweek (Sa Su)
omit date 2001jan15
omit date 2001feb19
omit date 2001apr13
omit date 2001may28
omit date 2001jul04
omit date 2001sep03
omit date 2001sep11
omit date 2001sep12
omit date 2001sep13
omit date 2001sep14
omit date 2001nov22
omit date 2001dec25
```

end sp500.stbcal

`bcal create` *filename*`, from()` can save the calendar file anywhere in your directory system by specifying a path in *filename*. It is assumed that the directory where the file is to be saved already exists. The pattern of *filename* should be [*path*]*calname*[`.stbcal`]. Here *calname* should be without the `%tb` prefix; *calname* has to be a valid Stata name but limited to 10 characters. If *path* is not specified, the file is saved in the current working directory. If the `.stbcal` extension is not specified, it is added.

Save the file in a directory where Stata can find it. 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`). The `personal` option specifies that the calendar file be saved in your `PERSONAL` directory, which ensures that the created calendar can be easily found in future work.

◁

▷ Example 2: Obtaining a preedited calendar using bcal webcopy (StataNow)

We revisit the setup of the examples in [TS] **mgarch**. We start by obtaining the stocks dataset used in the example.

```
. use https://www.stata-press.com/data/r19/stocks, clear
(Data from Yahoo! Finance)
. describe
Contains data from https://www.stata-press.com/data/r19/stocks.dta
 Observations:         2,015                  Data from Yahoo! Finance
    Variables:             5                  9 May 2024 11:48
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
date            int     %td                   Date
t               int     %9.0g                 Generic time variable
toyota          float   %9.0g                 Daily return on Toyota stock
nissan          float   %9.0g                 Daily return on Nissan stock
honda           float   %9.0g                 Daily return on Honda stock
-------------------------------------------------------------------------------
Sorted by: t
```

Notice that the data are sorted by `t`. This is a numeric variable that has consecutive numbers from 1 to 2015. We also have a date variable `date`. Now, if we `tsset` the dataset, we obtain the following:

```
. tsset
Time variable: t, 1 to 2015
        Delta: 1 unit
```

We have a reasonable date variable we could use, but instead we use consecutive numbers. We do this to avoid gaps in the time series that would generate problems when using time-series operators using the date in our dataset. Let's generate the lag of `toyota` to illustrate this.

```
. generate ltoyota = l.toyota
(1 missing value generated)
. list date t toyota ltoyota in 1/10, separator(0) noobs
```

date	t	toyota	ltoyota
02jan2003	1	.0151675	.
03jan2003	2	.0048201	.0151675
06jan2003	3	.0199587	.0048201
07jan2003	4	-.0133226	.0199587
08jan2003	5	-.0270011	-.0133226
09jan2003	6	.0116346	-.0270011
10jan2003	7	-.0093722	.0116346
13jan2003	8	.0016935	-.0093722
14jan2003	9	-.0062234	.0016935
15jan2003	10	-.0039806	-.0062234

As you can see, there is only one missing value, as expected. But you can also see there is a jump in the dates between January 3, 2003, and January 6, 2003. Yes, t is consecutive but the dates are not. What if we had used date with tsset instead?

```
. tsset date
Time variable: date, 02jan2003 to 31dec2010, but with gaps
        Delta: 1 day
. generate ltoyota2 = l.toyota
(434 missing values generated)
. list date t toyota ltoyota2 in 1/10, separator(0) noobs
```

date	t	toyota	ltoyota2
02jan2003	1	.0151675	.
03jan2003	2	.0048201	.0151675
06jan2003	3	.0199587	.
07jan2003	4	-.0133226	.0199587
08jan2003	5	-.0270011	-.0133226
09jan2003	6	.0116346	-.0270011
10jan2003	7	-.0093722	.0116346
13jan2003	8	.0016935	.
14jan2003	9	-.0062234	.0016935
15jan2003	10	-.0039806	-.0062234

You see the problem. Using consecutive numbers is a way around this issue, but a more elegant solution would be to use a business calendar. We constructed a business calendar for these data that can be accessed using bcal webcopy:

```
. bcal webcopy stock
(business calendar _stock.stbcal copied to working directory)
. generate datebcal = bofd("_stock", date)
```

Above we copied _stock.stbcal to our working directory. We then used the date function bofd() to generate a date. We can now tsset our data and consistently use time-series operators. We first format the date to appear as a date instead of a number.

```
. format datebcal %tb_stock
. tsset datebcal
Time variable: datebcal, 02jan2003 to 31dec2010
        Delta: 1 day
. generate ltoyota3 = l.toyota
(1 missing value generated)
. list date datebcal toyota ltoyota3 in 1/10, separator(0) noobs
```

date	datebcal	toyota	ltoyota3
02jan2003	02jan2003	.0151675	.
03jan2003	03jan2003	.0048201	.0151675
06jan2003	06jan2003	.0199587	.0048201
07jan2003	07jan2003	-.0133226	.0199587
08jan2003	08jan2003	-.0270011	-.0133226
09jan2003	09jan2003	.0116346	-.0270011
10jan2003	10jan2003	-.0093722	.0116346
13jan2003	13jan2003	.0016935	-.0093722
14jan2003	14jan2003	-.0062234	.0016935
15jan2003	15jan2003	-.0039806	-.0062234

Note that when we tsset our data, there are no gaps. Also, the lag of toyota is adequately generated. Yet if you look at date and datebcal, they seem to provide the same information. Indeed, they are the same dates, but datebcal understands that 03jan2003 is a Friday and that 06jan2003 is a Monday.

◁

Stored results

bcal check stores the following in r():

Macros

r(defined)	business calendars used, stbcal-file exists, and file contains no errors
r(undefined)	business calendars used, but no stbcal-files exist for them
r(varlist_*calname*)	list of variable names that use business calendar *calname*

Warning to programmers: Specify the rc0 option to access these returned results. By default, bcal check returns code 459 if a business calendar does not exist or if a business calendar exists but has errors; in such cases, the results are not stored.

bcal dir stores the following in r():

Macros

r(calendars)	business calendars available
r(fn_*calname*)	stbcal-file for business calendar *calname*

bcal describe and bcal create store the following in r():

Scalars

r(min_date_td)	calendar's minimum date in %td units
r(max_date_td)	calendar's maximum date in %td units
r(ctr_date_td)	calendar's zero date in %td units
r(min_date_tb)	calendar's minimum date in %tb units
r(max_date_tb)	calendar's maximum date in %tb units
r(omitted)	total number of days omitted from calendar
r(included)	total number of days included in calendar
r(omitted_year)	approximate number of days omitted per year from calendar
r(included_year)	approximate number of days included per year in calendar

Macros

r(name)	pure calendar name (for example, nyse)
r(purpose)	short description of calendar's purpose
r(fn)	name of stbcal-file

bcal load stores the same results in r() as bcal describe, except it does not store r(omitted), r(included), r(omitted_year) and r(included_year).

bcal webcopy stores the following in r():

Macros

r(webfrom)	website from which calendar was copied
r(webcalfile)	name of business calendar file copied to working directory
r(webcalname)	*webcalname* specified by user

Reference

Rajbhandari, A. 2016. Handling gaps in time series using business calendars. *The Stata Blog: Not Elsewhere Classified.* https://blog.stata.com/2016/02/04/handling-gaps-in-time-series-using-business-calendars/.

Also see

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

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

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