

**bcal** — Business calendar file manipulation
[Description](#)[Syntax](#)[Remarks and examples](#)[Also see](#)[Quick start](#)[Option for bcal check](#)[Stored results](#)[Menu](#)[Options for bcal create](#)[Reference](#)

## 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 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)
```

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
```

## 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
```

Create a business calendar from the current dataset

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

where

*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*.

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

<i>bcal_create_options</i>	Description
<hr/>	
Main	
* <b>from</b> ( <i>varname</i> )	specify date variable for calendar
<b>generate</b> ( <i>newvar</i> )	generate <i>newvar</i> containing business dates
<b>excludemissing</b> ( <i>varlist</i> [, any])	exclude observations with missing values in <i>varlist</i>
<b>personal</b>	save calendar file in your <b>PERSONAL</b> directory
<b>replace</b>	replace file if it already exists
<hr/>	
Advanced	
<b>purpose</b> ( <i>text</i> )	describe purpose of calendar
<b>dateformat</b> ( <i>ydm</i>   <i>ydm</i>   <i>myd</i>   <i>mdy</i>   <i>dym</i>   <i>dmy</i> )	specify date format in calendar file
<b>range</b> ( <i>fromdate</i> <i>todate</i> )	specify range of calendar
<b>centerdate</b> ( <i>date</i> )	specify center date of calendar
<b>maxgap</b> (#)	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.

## 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 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 `%tbcalname` 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 `%tbcalname`; 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

[stata.com](https://www.stata.com)

`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\Stata17\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
(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:

---

```

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

version 17
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

```

---

`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 option `personal` specifies that the calendar file be saved in your `PERSONAL` directory, which ensures that the created calendar can be easily found in future work.

## Stored results

`bcal check` stores the following in `r()`:

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

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	
<code>r(calendars)</code>	business calendars available
<code>r(fn_<i>calname</i>)</code>	<code>stbcal</code> -file for business calendar <i>calname</i>

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

Scalars

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

Macros

<code>r(name)</code>	pure calendar name (for example, <code>nyse</code> )
<code>r(purpose)</code>	short description of calendar's purpose
<code>r(fn)</code>	name of <code>stbcal</code> -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)`.

## Reference

Rajbhandari, A. 2016. Handling gaps in time series using business calendars. *The Stata Blog: Not Elsewhere Classified*. <http://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