

tsappend — Add observations to a time-series dataset

[Description](#)
[Options](#)

[Quick start](#)
[Remarks and examples](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Also see](#)

Description

`tsappend` appends observations to a time-series dataset or to a panel dataset. `tsappend` uses and updates the information set by `tsset` or `xtset`. Any gaps in the dataset are removed.

Quick start

Add 10 time periods to `tsset` data

```
tsappend, add(10)
```

Incorporate additional months to data up to the third month of 1999

```
tsappend, last(1999m3) tsfmt(tm)
```

Add 2 time periods to the panel identified by `pvar = 333` after `xtset pvar tvar`

```
tsappend, add(2) panel(333)
```

Menu

Statistics > Time series > Setup and utilities > Add observations to time-series dataset

Syntax

```
tsappend, { add(#) | last(date | clock) tsfmt(string) } [options]
```

<i>options</i>	Description
* <code>add(#)</code>	add # observations
* <code>last(<i>date</i> <i>clock</i>)</code>	add observations at <i>date</i> or <i>clock</i>
* <code>tsfmt(<i>string</i>)</code>	use time-series function <i>string</i> with <code>last(<i>date</i> <i>clock</i>)</code>
<code>panel(<i>panel_id</i>)</code>	add observations to panel <i>panel_id</i>

* Either `add(#)` is required, or `last(date | clock)` and `tsfmt(string)` are required.

You must `tsset` or `xtset` your data before using `tsappend`; see [\[TS\] tsset](#) and [\[XT\] xtset](#).

Options

`add(#)` specifies the number of observations to add.

`last(date | clock)` and `tsfmt(string)` must be specified together and are an alternative to `add()`.

`last(date | clock)` specifies the date or the date and time of the last observation to add.

`tsfmt(string)` specifies the name of the Stata time-series function to use in converting the date specified in `last()` to an integer. The function names are `tc` (clock), `tC` (Clock), `td` (daily), `tw` (weekly), `tm` (monthly), `tq` (quarterly), and `th` (half-yearly).

For clock times, the last time added (if any) will be earlier than the time requested in `last(date | clock)` if `last()` is not a multiple of delta units from the last time in the data.

For instance, you might specify `last(17may2007) tsfmt(td)`, `last(2001m1) tsfmt(tm)`, or `last(17may2007 15:30:00) tsfmt(tc)`.

`panel(panel_id)` specifies that observations be added only to panels with the ID specified in `panel()`.

Remarks and examples

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

Remarks are presented under the following headings:

[Introduction](#)

[Using tsappend with time-series data](#)

[Using tsappend with panel data](#)

Introduction

`tsappend` adds observations to a time-series dataset or to a panel dataset. You must `tsset` or `xtset` your data before using `tsappend`. `tsappend` simultaneously removes any gaps from the dataset.

There are two ways to use `tsappend`: you can specify the `add(#)` option to request that # observations be added, or you can specify the `last(date | clock)` option to request that observations be appended until the date specified is reached. If you specify `last()`, you must also specify `tsfmt()`. `tsfmt()` specifies the Stata time-series date function that converts the date held in `last()` to an integer.

`tsappend` works with time series of panel data. With panel data, `tsappend` adds the requested observations to all the panels, unless the `panel()` option is also specified.

Using tsappend with time-series data

tsappend can be useful for appending observations when dynamically predicting a time series. Consider an example in which tsappend adds the extra observations before dynamically predicting from an AR(1) regression:

```
. use http://www.stata-press.com/data/r14/tsappend1
. regress y l.y
```

Source	SS	df	MS	Number of obs	=	479
Model	115.349555	1	115.349555	F(1, 477)	=	119.29
Residual	461.241577	477	.966963473	Prob > F	=	0.0000
				R-squared	=	0.2001
				Adj R-squared	=	0.1984
Total	576.591132	478	1.2062576	Root MSE	=	.98334

y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
y					
L1.	.4493507	.0411417	10.92	0.000	.3685093 .5301921
_cons	11.11877	.8314581	13.37	0.000	9.484993 12.75254

```
. matrix b = e(b)
. matrix colnames b = L.xb one
. tsset
      time variable:  t2, 1960m2 to 2000m1
      delta: 1 month
. tsappend, add(12)
. tsset
      time variable:  t2, 1960m2 to 2001m1
      delta: 1 month
. predict xb if t2<=tm(2000m2)
(option xb assumed; fitted values)
(12 missing values generated)
. generate one=1
. matrix score xb=b if t2>=tm(2000m2), replace
```

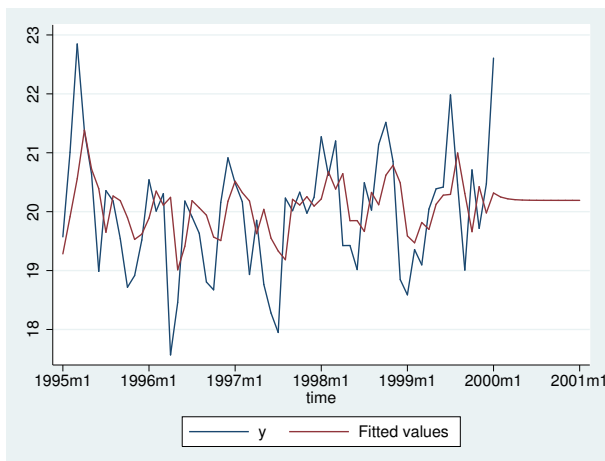
The calls to `tsset` before and after `tsappend` were made without a time variable; thus both commands display how the data are currently `tsset`. The results from the first `tsset` command show that we have monthly data and that our time variable, `t2`, starts at 1960m2 and ends at 2000m1.

`tsappend` with the `add(12)` option used these results to add 12 months to the dataset. The results of the second `tsset` command show that this new year of data has been added, as shown by the end year now being 2001m1. We could have skipped these calls to `tsset`, but they are shown here to illustrate how `tsappend` uses and updates time-series settings of the dataset.

We then used `predict` and `matrix score` to obtain the dynamic predictions, which allows us to produce the following graph:

4 tsappend — Add observations to a time-series dataset

```
. line y xb t2 if t2>=tm(1995m1), ytitle("") xtitle("time")
```



In the call to `tsappend`, instead of saying that we wanted to add 12 observations, we could have specified that we wanted to fill in observations through the first month of 2001:

```
. use http://www.stata-press.com/data/r14/tsappend1, clear
. tsset
    time variable:  t2, 1960m2 to 2000m1
      delta: 1 month
. tsappend, last(2001m1) tsfmt(tm)
. tsset
    time variable:  t2, 1960m2 to 2001m1
      delta: 1 month
```

We specified the `tm()` function in the `tsfmt()` option. [\[FN\] Date and time functions](#) contains a list of time-series functions for translating date literals to integers. Because we have monthly data, and since [\[FN\] Date and time functions](#) tells us that we want to use the `tm()` function, we specified the `tsfmt(tm)` option. The following table shows the most common types of time-series data, their formats, the appropriate translation functions, and the corresponding options for `tsappend`:

Description	Format	Function	Option
time	%tc	tc()	tsfmt(tc)
time	%tC	tC()	tsfmt(tC)
daily	%td	td()	tsfmt(td)
weekly	%tw	tw()	tsfmt(tw)
monthly	%tm	tm()	tsfmt(tm)
quarterly	%tq	tq()	tsfmt(tq)
half-yearly	%th	th()	tsfmt(th)
yearly	%ty	ty()	tsfmt(ty)

Using tsappend with panel data

tsappend's actions on panel data are similar to its action on time-series data, except that tsappend performs those actions on each time series within the panels. To work within panels, a panel variable must have been specified with tsset or xtset. It does not matter which command you use; the two are equivalent.

If the end dates vary over panels, last() and add() will produce different results. add(#) always adds # observations to each panel. If the data end at different periods before tsappend, add() is used, the data will still end at different periods after tsappend, add(). In contrast, tsappend, last() tsfmt() will cause all the panels to end on the specified last date. If the beginning dates differ across panels, using tsappend, last() tsfmt() to provide a uniform ending date will not create balanced panels because the number of observations per panel will still differ.

Consider the panel data summarized in the output below:

```
. use http://www.stata-press.com/data/r14/tsappend3, clear
. xtdescribe
      id:  1, 2, ..., 3                      n =          3
      t2: 1998m1, 1998m2, ..., 2000m1      T =         25
      Delta(t2) = 1 month
      Span(t2) = 25 periods
      (id*t2 uniquely identifies each observation)
Distribution of T_i:  min      5%      25%      50%      75%      95%      max
                   13      13      13      20      24      24      24
  Freq.  Percent  Cum. | Pattern
-----+-----
      1   33.33   33.33 | .....111111111111
      1   33.33   66.67 | 1111.111111111111111111
      1   33.33  100.00 | 11111111111111111111.....
-----+-----
      3  100.00                | XXXXXXXXXXXXXXXXXXXXXXXX
. by id: summarize t2
```

```
-> id = 1
  Variable |      Obs      Mean   Std. Dev.   Min      Max
-----+-----
      t2 |      13      474    3.89444    468    480
```

```
-> id = 2
  Variable |      Obs      Mean   Std. Dev.   Min      Max
-----+-----
      t2 |      20    465.5    5.91608    456    475
```

```
-> id = 3
  Variable |      Obs      Mean   Std. Dev.   Min      Max
-----+-----
      t2 |      24    468.3333    7.322786    456    480
```

The output from xtdescribe and summarize on these data tells us that one panel starts later than the other, that another panel ends before the other two, and that the remaining panel has a gap in the time variable but otherwise spans the entire time frame.


```

. use http://www.stata-press.com/data/r14/tsappend2, clear
. tsappend, last(2000m7) tsfmt(tm)
. xtdescribe
      id: 1, 2, ..., 3                n =          3
      t2: 1998m1, 1998m2, ..., 2000m7  T =          31
      Delta(t2) = 1 month
      Span(t2) = 31 periods
      (id*t2 uniquely identifies each observation)
Distribution of T_i:  min      5%    25%    50%    75%    95%    max
                   19      19     19     31     31     31     31
      Freq.  Percent  Cum. | Pattern
      -----|-----
      2      66.67  66.67 | 11111111111111111111111111111111
      1      33.33  100.00 | .....11111111111111111111
      -----|-----
      3      100.00                | XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
. by id: summarize t2

```

```

-> id = 1

```

Variable	Obs	Mean	Std. Dev.	Min	Max
t2	19	477	5.627314	468	486

```

-> id = 2

```

Variable	Obs	Mean	Std. Dev.	Min	Max
t2	31	471	9.092121	456	486

```

-> id = 3

```

Variable	Obs	Mean	Std. Dev.	Min	Max
t2	31	471	9.092121	456	486

Stored results

tsappend stores the following in r():

```

Scalars
      r(add)          number of observations added

```

Also see

- [TS] [tsset](#) — Declare data to be time-series data
- [XT] [xtset](#) — Declare data to be panel data