Title

intro — Introduction to programming manual

Description

This entry describes this manual and what has changed since Stata 9.

Remarks

In this manual, you will find

1. matrix-manipulation commands, which are available from the Stata command line and for ado-programming; for advanced matrix functions and a complete matrix programming language, see the [M] *Mata Reference Manual*.
2. commands for programming Stata, and
3. commands and discussions of interest to programmers.

This manual is referred to as [P] in cross-references and is organized alphabetically.

If you are new to Stata's programming commands, we recommend that you first read the chapter about programming Stata in the *User's Guide*; see [U] **18 Programming Stata**. After you read this chapter, we recommend that you read the following sections from this manual:

[P] **program**	Define and manipulate programs
[P] **sortpreserve**	Sorting within programs
[P] **byable**	Making programs byable
[P] **macro**	Macro definition and manipulation

You may also find the subject table of contents, which immediately follows the table of contents, helpful.

We also recommend the Stata NetCourses™. At the time that this introduction was written, our current offerings of Stata programming NetCourses included

NC-151 An introduction to Stata programming
NC-152 Advanced Stata programming

To learn more about NetCourses and to view the current offerings of NetCourses, visit http://www.stata.com/netcourse.

To learn about writing your own maximum-likelihood estimation commands, read the book *Maximum Likelihood Estimation with Stata*.

What's new

1. First, a warning for time-series programmers: Stata's new date/time values, which contain the number of milliseconds since 01jan1960 00:00:00, result in large numbers. 21apr2007 12:14:07.123 corresponds to 1,492,776,847,123. Date/time values must be stored as `doubles`. Programmers should use scalars to store these values whenever possible. If you must use a macro, exercise caution that the value is not rounded. It would not do at all for 1,492,776,847,123 to be recorded as "`1.493e+12`" (which would be 24apr2007 02:13:20). If you must use macros, our recommendations are

a. If a date/time value is stored in one macro and you need it in another, code

```
local new `old'
```

b. If a date/time value is the result of an expression, and you must store it as a macro, code

```
local new = string(exp, "%21x")
```

or

```
local new : display %21x (exp)
```

Now we will continue with What's new.

2. Stata for Windows now supports Automation, formerly known as OLE Automation, which means that programmers can control Stata from other applications and retrieve results. See [P] **automation**.

3. New command `confirm {numeric | string | date} format` verifies that the format is of the specified type; see [P] **confirm**.

4. New function `fmtwidth(`*s*`)` returns the display width of a *%fmt* string, including date formats; see *Programming functions* in [D] **functions**.

5. Expression limits have been increased in Stata/MP, Stata/SE, and Stata/IC. The limit on the number of dyadic operators has increased from 500 to 800, and the limit on the number of numeric literals has increased from 150 to 300. See `help limits`.

6. Intercooled Stata has been renamed to Stata/IC. `c(flavor)` now contains "`IC`" rather than "`Intercooled`" if version $\geq$ 10. Backward-compatibility old global macro `$S_FLAVOR` continues to contain "`Intercooled`". See [P] **creturn** and [P] **macro**.

7. `c()` now contains values associated with Stata/MP: `c(MP)` (1 or 0 depending on whether you are running Stata/MP), `c(processors)` (the number of processors Stata will use), `c(processors_mach)` (the number of processors on the computer), `c(processors_lic)` (the maximum number of processors the license will allow you to use), `c(processors_max)` (the maximum number of processors that could be used on this computer with this license).

8. New command `include` is a variation on `do` and `run`. `include` executes the commands stored in a file just as if they were entered from the keyboard or the current do-file. It is most often used in advanced Mata situations to create the equivalent of `#include` files. See [P] **include**.

9. New commands `signestimationsample` and `checkestimationsample` are useful in writing estimation/postestimation commands that need to identify the estimation sample; see [P] **signestimationsample**.

10. New command `_datasignature` is the building block for Stata's `datasignature` command and the programming commands `signestimationsample` and `checkestimationsample`. In advanced situations, you may wish to call it directly. See [P] **_datasignature**.

11. New extended macro function `:copy` copies one macro to another and is faster when the macro being copied is long. That is, coding

```
local new : copy local old
```

is faster than

```
local new `old'
```

See [P] **macro**.

12. New command `timer` times sections of code; see [P] **timer**.

13. Existing command `matrix accum` is now faster when some observations contain zeros; see [P] **matrix accum**.

14. Existing command `ml display` has new option `showeqns` that requests that equation names be displayed in the coefficient table; see [R] **ml**.

15. Existing command `mkmat` has new options `rownames()`, `roweq()`, `rowprefix()`, `obs`, and `nchar()` that specify the row names of the created matrix; see [P] **matrix mkmat**.

16. Existing command `_rmdcoll`'s `nocollin` option has been renamed to `normcoll`. `nocollin` will continue to work as a synonym for `normcoll`. See [P] **_rmcoll**.

17. Existing command `describe`'s option `simple` no longer saves the names of the variables in `r(varlist)`; you must specify option `varlist` if you want that. Also, existing command `describe using` *filename* now allows options `simple` and `varlist`. See [D] **describe**.

18. New extended macro function `adosubdir` returns the subdirectory in which Stata would search for a file along the ado-path. Determining the subdirectory in which Stata stores files is now a function of the file's extension. Command `adosubdir` returns the subdirectory in which to look. See [P] **macro**.

19. Existing command `syntax` { [*optionname*`(real ...)`] } now returns the number specified in `%18.0g` format if `version` is set to 10.0 or higher. For `version` less than 10, the number is returned in `%9.0g` format. See [P] **syntax**.

20. New functions `_byn1()` and `_byn2()`, available within a `byable(recall)` program, return the beginning and ending observation numbers of the by-group currently being executed; see [P] **byable**.

21. Existing command `program drop` may now specify `program drop _allado` to drop programs that were automatically loaded from ado-files; see [P] **program**.

22. Concerning SMCL,

    a. Existing directive `{synoptset}` has new optional argument `notes` that specifies that some of the table entries will have footnotes and results in a larger indentation of the first column.

    b. Existing directive `{p}` now has an optional fourth argument specifying the paragraph's width.

    See [P] **smcl**.

23. Concerning classes, you can now define an `oncopy` member program to perform operations when a copy of an object is being created. See [P] **class**.

24. Concerning programmable menus, the maximum number of menu items that can be added to Stata has increased to 1,250 from 1,000; see `help window programming`.

25. Concerning programmable dialogs,

    a. Child dialogs can now be created.

    b. New control `TEXTBOX` allows displaying multiline text.

    c. In the dialog programming language, (1) `if` now allows `else` and (2) new command `close` closes the dialog programmatically.

    d. Messages can be passed to dialogs when they are launched; see `help db`.

    e. Dialogs can now be designated as modal, meaning that this dialog must be dealt with by the user before new dialogs (other than children) can be launched.

    f. Several controls have new options and new member programs. For instance, `FILE` and `LISTBOX` now have option `multiselect`, which lets the user pick more than one item.

    See `help dialog programming`.

26. Stata's help files are now named `*.sthlp` rather than `*.hlp`, meaning that user-written help files can be sent via email more easily. Many email filters flag `.hlp` files as potential virus carriers because Stata was not the only one to use the `.hlp` suffix. You need not rename your old help files. See [R] **help**.

27. Two new C functions have been exposed from Stata for use by plugins: `sstore()` and `sdata()`. `sstore()` stores string data in the Stata dataset and `sdata()` reads them. See http://www.stata.com/plugins/.

There are other new additions to Stata that will be of interest to programmers, but, because they are of interest to others, too, they are documented in [U] **1.3 What's new**.

Also See

[U] **18 Programming Stata**

[U] **1.3 What's new**

Maximum Likelihood Estimation with Stata

[R] **intro** — Introduction to base reference manual