Title

stata.com

which — Display location and version for an ado-file

Syntax Description Option Remarks and examples Also see

Syntax

`which` *fname*[`.`*ftype*] [`, all`]

Description

`which` looks for *fname*`.`*ftype* along the `S_ADO` path. If Stata finds the file, `which` displays the full path and filename, along with, if the file is text, all lines in the file that begin with "`*!`" in the first column. If Stata cannot find the file, `which` issues the message "file not found along ado-path" and sets the return code to 111. *ftype* must be a file type for which Stata usually looks along the ado-path to find. Allowable *ftype*s are

`.ado`, `.class`, `.dlg`, `.idlg`, `.sthlp`, `.ihlp`, `.hlp`, `.key`, `.maint`, `.mata`, `.mlib`, `.mo`, `.mnu`, `.plugin`, `.scheme`, `.stbcal`, and `.style`

If *ftype* is omitted, `which` assumes `.ado`. When searching for `.ado` files, if Stata cannot find the file, Stata then checks to see if *fname* is a built-in Stata command, allowing for valid abbreviations. If it is, the message "built-in command" is displayed; if not, the message "command not found as either built-in or ado-file" is displayed and the return code is set to 111.

For information about internal version control, see [P] **version**.

Option

`all` forces `which` to report the location of all files matching the *fname*`.`*ftype* found along the search path. The default is to report just the first one found.

Remarks and examples

stata.com

If you write programs, you know that you make changes to the programs over time. If you are like us, you also end up with multiple versions of the program stored on your disk, perhaps in different directories. You may even have given copies of your programs to other Stata users, and you may not remember which version of a program you or your friends are using. The `which` command helps you solve this problem.

▷ Example 1

The `which` command displays the path for *filename*`.ado` and any lines in the code that begin with "`*!`". For example, we might want information about the `test` command, described in [R] **test**, which is an ado-file written by StataCorp. Here is what happens when we type `which test`:

```
. which test
C:\Program Files\Stata13\ado\base\t\test.ado
*! version 2.2.2  07feb2012
```

`which` displays the path for the `test.ado` file and also a line beginning with "`*!`" that indicates the version of the file. This is how we, at StataCorp, do version control—see **[U] 18.11.1 Version** for an explanation of our version control numbers.

We do not need to be so formal. `which` will display anything typed after lines that begin with '`*!`'. For instance, we might write `myprog.ado`:

```
. which myprog
.\myprog.ado
*! first written 1/03/2013
*! bug fix on 1/05/2013 (no variance case)
*! updated 1/24/2013 to include noconstant option
*! still suspicious if variable takes on only two values
```

It does not matter where in the program the lines beginning with `*!` are—`which` will list them (in particular, our "still suspicious" comment was buried about 50 lines down in the code). All that is important is that the `*!` marker appear in the first two columns of a line.

◁

▷ Example 2

If we type `which` *command*, where *command* is a built-in command rather than an ado-file, Stata responds with

```
. which summarize
built-in command:  summarize
```

If *command* was neither a built-in command nor an ado-file, Stata would respond with

```
. which junk
command junk not found as either built-in or ado-file
r(111);
```

◁

Also see

[P] findfile — Find file in path

[P] version — Version control

[U] 17 Ado-files

[U] 18.11.1 Version