

**net** — Install and manage community-contributed additions from the Internet

[Description](#)[Syntax](#)[Options](#)[Remarks and examples](#)[Also see](#)

## Description

`net` downloads and installs additions to Stata. The additions can be obtained from the Internet or from physical media. The additions can be ado-files (new commands), help files, or even datasets. Collections of files that may be installed as a group are bound together into a *package*.

`ado` manages the packages you have installed by using `net`. The `ado` command lets you list and uninstall previously installed packages.

You can also access the `net` and `ado` features by selecting **Help > SJ and community-contributed commands**; this is the recommended method to find and install additions to Stata.

## Syntax

*Set current location for net*

```
net from directory_or_url
```

*Change to a different net directory*

```
net cd path_or_url
```

*Change to a different net site*

```
net link linkname
```

*Search for installed packages*

```
net search (see [R] net search)
```

*Report current net location*

```
net
```

*Describe a package*

```
net describe pkgname [ , from(directory_or_url) ]
```

*Set location where packages will be installed*

```
net set ado dirname
```

Set location where ancillary files will be installed

```
net set other dirname
```

Report net ‘from’, ‘ado’, and ‘other’ settings

```
net query
```

Install ado-files and help files from a package

```
net install pkgname [ , all replace force from(directory_or_url) ]
```

Install ancillary files from a package

```
net get pkgname [ , all replace force from(directory_or_url) ]
```

Shortcut to access Stata Journal (SJ) net site

```
net sj vol-issue [insert]
```

Shortcut to access Stata Technical Bulletin (STB) net site

```
net stb issue [insert]
```

List installed packages

```
ado [ , find(string) from(dirname) ]
```

```
ado dir [pkgid] [ , find(string) from(dirname) ]
```

Describe installed packages

```
ado describe [pkgid] [ , find(string) from(dirname) ]
```

Update installed packages

```
ado update (see [R] ado update)
```

Uninstall an installed package

```
ado uninstall pkgid [ , from(dirname) ]
```

where

*pkgname* is name of a package

*pkgid* is name of a package

or a number in square brackets: [#]

*dirname* is a directory name

or PLUS (default)

or PERSONAL

or SITE

## Options

`all` is used with `net install` and `net get`. Typing it with either one makes the command equivalent to typing `net install` followed by `net get`.

`replace` is for use with `net install` and `net get`. It specifies that the downloaded files replace existing files if any of the files already exists.

`force` specifies that the downloaded files replace existing files if any of the files already exists, even if Stata thinks all the files are the same. `force` implies `replace`.

`find(string)` is for use with `ado`, `ado dir`, and `ado describe`. It specifies that the descriptions of the packages installed on your computer be searched, and that the package descriptions containing *string* be listed.

`from(dirname)`, when used with `ado`, specifies where the packages are installed. The default is `from(PLUS)`. PLUS is a code word that Stata understands to correspond to a particular directory on your computer that was set at installation time. On Windows computers, PLUS probably means the directory `c:\ado\plus`, but it might mean something else. You can find out what it means by typing `sysdir`, but doing so is irrelevant if you use the defaults.

`from(directory_or_url)`, when used with `net`, specifies the directory or URL where installable packages may be found. The directory or URL is the same as the one that would have been specified with `net from`.

## Remarks and examples

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

For an introduction to using `net` and `ado`, see [\[U\] 29 Using the Internet to keep up to date](#). The purpose of this documentation is

- to briefly, but accurately, describe `net` and `ado` and all their features and
- to provide documentation to those who wish to set up their own sites to distribute additions to Stata.

Remarks are presented under the following headings:

*Definition of a package*

*The purpose of the net and ado commands*

*Content pages*

*Package-description pages*

*Where packages are installed*

*A summary of the net command*

*A summary of the ado command*

*Relationship of net and ado to the point-and-click interface*

*Creating your own site*

*Format of content and package-description files*

*Example 1*

*Example 2*

*Additional package directives*

*SMCL in content and package-description files*

*Error-free file delivery*

## Definition of a package

A *package* is a collection of files—typically, `.ado` and `.sthlp` files—that together provide a new feature in Stata. Packages contain additions that you wish had been part of Stata at the outset. For instance, the package named `zz49` might add the `xyz` command to Stata. At a minimum, such a package would contain `xyz.ado`, the code to implement the new command, and `xyz.sthlp`, the system help to describe it. We write such additions, and so do other users.

One source of these additions is the *Stata Journal*, a printed and electronic journal with corresponding software. If you want the journal, you must subscribe, but the software is available for free from our website.

## The purpose of the net and ado commands

The `net` command makes it easy to distribute and install packages. The goal is to get you quickly to a package-description page that summarizes the addition, for example,

---

```
. net describe rte_stat, from(http://www.wemakeitupaswego.edu/faculty/sgazer/)
```

---

```
package rte_stat from http://www.wemakeitupaswego.edu/faculty/sgazer/
```

---

### TITLE

`rte_stat`. The robust-to-everything statistic; update.

### DESCRIPTION/AUTHOR(S)

S. Gazer, Dept. of Applied Theoretical Mathematics, WMIUAWG Univ. Aleph-0 100% confidence intervals proved too conservative for some applications; Aleph-1 confidence intervals have been substituted. The new robust-to-everything supplants the previous robust-to-everything-conceivable statistic. See "Inference in the absence of data" (forthcoming). After installation, see help `rte`.

### INSTALLATION FILES

(type `net install rte_stat`)

```
rte.ado
rte.sthlp
nullset.ado
random.ado
```

---

If you decide that the addition might prove useful, `net` makes the installation easy:

```
. net install rte_stat
checking rte_stat consistency and verifying not already installed...
installing into c:\ado\plus\ ...
installation complete.
```

The `ado` command helps you manage packages installed with `net`. Perhaps you remember that you installed a package that calculates the robust-to-everything statistic, but you cannot remember the command's name. You could use `ado` to search what you have previously installed for the `rte` command,

```
. ado
[1] package sg145 from https://www.stata.com/stb/stb56
STB-56 sg145. Scalar measures of fit for regression models.
(output omitted)
[15] package rte_stat from http://www.wemakeitupaswego.edu/faculty/sgazer
rte_stat. The robust-to-everything statistic; update.
(output omitted)
[21] package st0119 from https://www.stata-journal.com/software/sj7-1
SJ7-1 st0119. Rasch analysis
```

or you might type

```
. ado, find("robust-to-everything")
[15] package rte_stat from http://www.wemakeitupaswego.edu/faculty/sgazer
rte_stat. The robust-to-everything statistic; update.
```

Perhaps you decide that `rte`, despite the author's claims, is not worth the disk space it occupies. You can use `ado` to erase it:

```
. ado uninstall rte_stat
package rte_stat from http://www.wemakeitupaswego.edu/faculty/sgazer
rte_stat. The robust-to-everything statistic; update.
(package uninstalled)
```

`ado uninstall` is easier than erasing the files by hand because `ado uninstall` erases every file associated with the package, and, moreover, `ado` knows where on your computer `rte_stat` is installed; you would have to hunt for these files.

## Content pages

There are two types of pages displayed by `net`: content pages and package-description pages. When you type `net from`, `net cd`, `net link`, or `net` without arguments, Stata goes to the specified place and displays the content page:

```
. net from https://www.stata.com
```

---

```
https://www.stata.com/
StataCorp
```

---

Welcome to StataCorp.

Below we provide links to sites providing additions to Stata, including the Stata Journal, STB, and Statalist. These are NOT THE OFFICIAL UPDATES; you fetch and install the official updates by typing `-update-`.

PLACES you could `-net link-` to:

sj	The Stata Journal
----	-------------------

DIRECTORIES you could `-net cd-` to:

stb	materials published in the Stata Technical Bulletin
users	materials written by various people, including StataCorp employees
meetings	software packages from Stata Users Group meetings
links	links to other locations providing additions to Stata

---

A content page tells you about other content pages and package-description pages. The example above lists other content pages only. Below, we follow one of the links for the *Stata Journal*:

```
. net link sj
```

---

```
https://www.stata-journal.com/  
The Stata Journal
```

---

The Stata Journal is a refereed, quarterly journal containing articles of interest to Stata users. For more details and subscription information, visit the Stata Journal website at <https://www.stata-journal.com>.

PLACES you could **-net link-** to:

stata	StataCorp website
-------	-------------------

DIRECTORIES you could **-net cd-** to:

production	Files for authors of the Stata Journal
software	Software associated with Stata Journal articles

---

```
. net cd software
```

---

```
https://www.stata-journal.com/software/  
The Stata Journal
```

---

PLACES you could **-net link-** to:

stata	StataCorp website
stb	Stata Technical Bulletin (STB) software archive

DIRECTORIES you could **-net cd-** to:

<i>(output omitted)</i>	
sj7-1	volume 7, issue 1
<i>(output omitted)</i>	
sj1-1	volume 1, issue 1

---

```
. net cd sj7-1
```

---

```
https://www.stata-journal.com/software/sj7-1/  
Stata Journal volume 7, issue 1
```

---

DIRECTORIES you could **-net cd-** to:

..	Other Stata Journals
----	----------------------

PACKAGES you could **-net describe-**:

dm0027	File filtering in Stata: handling complex data formats and navigating log files efficiently
st0119	Rasch analysis
st0120	Multivariable regression spline models
st0121	mhbounds - Sensitivity Analysis for Average Treatment Effects

---

dm0027, st0119, ..., st0121 are links to package-description pages.

1. When you type `net from`, you follow that with a location to display the location's content page.
  - a. The location could be a URL, such as <https://www.stata.com>. The content page at that location would then be listed.
  - b. The location could be `e:` on a Windows computer or a mounted volume on a Mac computer. The content page on that source would be listed. That would work if you had special media obtained from StataCorp or special media prepared by another user.
  - c. The location could even be a directory on your computer, but that would work only if that directory contained the right kind of files.

2. Once you have specified a location, typing `net cd` will take you into subdirectories of that location, if there are any. Typing

```
. net from https://www.stata-journal.com
. net cd software
```

is equivalent to typing

```
. net from https://www.stata-journal.com/software
```

Typing `net cd` displays the content page from that location.

3. Typing `net` without arguments redisplay the current content page, which is the content page last displayed.
4. `net link` is similar to `net cd` in that the result is to change the location, but rather than changing to subdirectories of the current location, `net link` jumps to another location:

```
. net from https://www.stata-journal.com
```

---

```
https://www.stata-journal.com/
```

```
The Stata Journal
```

---

```
The Stata Journal is a refereed, quarterly journal containing articles
of interest to Stata users. For more details and subscription information,
visit the Stata Journal website at
https://www.stata-journal.com.
```

```
PLACES you could -net link- to:
```

```
  stata           StataCorp website
```

```
DIRECTORIES you could -net cd- to:
```

```
  production      Files for authors of the Stata Journal
  software         Software associated with Stata Journal articles
```

---

Typing `net link stata` would jump to <https://www.stata.com>:

```
. net link stata
```

---

```
https://www.stata.com/
```

```
StataCorp
```

---

```
Welcome to StataCorp.
```

```
(output omitted)
```

---

## Package-description pages

Package-description pages describe what could be installed:

```
. net from https://www.stata-journal.com/software/sj7-1
```

---

```
https://www.stata-journal.com/software/sj7-1/
(output omitted)
```

```
. net describe st0119
```

---

```
package st0119 from https://www.stata-journal.com/software/sj7-1
```

---

### TITLE

SJ7-1 st0119. Rasch analysis

### DESCRIPTION/AUTHOR(S)

Rasch analysis

by Jean-Benoit Hardouin, University of Nantes, France

Support: jean-benoit.hardouin@univ-nantes.fr

After installation, type help **gammasym**, **gausshermite**,  
**geekel2d**, **raschtest**, and **raschtestv7**

### INSTALLATION FILES

(type **net install st0119**)

```
st0119/raschtest.ado
st0119/raschtest.hlp
st0119/raschtestv7.ado
st0119/raschtestv7.hlp
st0119/gammasym.ado
st0119/gammasym.hlp
st0119/gausshermite.ado
st0119/gausshermite.hlp
st0119/geekel2d.ado
st0119/geekel2d.hlp
```

### ANCILLARY FILES

(type **net get st0119**)

```
st0119/data.dta
st0119/outrasch.do
```

---

A package-description page describes the package and tells you how to install the component files. Package-description pages potentially describe two types of files:

1. Installation files: files that you type **net install** to install and that are required to make the addition work.
2. Ancillary files: additional files that you might want to install—you type **net get** to install them—but that you can ignore. Ancillary files are typically datasets that are useful for demonstration purposes. Ancillary files are not really installed in the sense of being copied to an official place for use by Stata itself. They are merely copied into the current directory so that you may use them if you wish.

You install the official files by typing **net install** followed by the package name. For example, to install **st0119**, you would type

```
. net install st0119
checking st0119 consistency and verifying not already installed...
installing into c:\ado\plus\ ...
installation complete.
```

You get the ancillary files—if there are any and if you want them—by typing **net get** followed by the package name:



```
. net get st0119
checking st0119 consistency and verifying not already installed...
copying into current directory...
  copying data.dta
  copying outrasch.do
ancillary files successfully copied.
```

Most users ignore the ancillary files.

Once you have installed a package—by typing `net install`—use `ado` to redisplay the package-description page whenever you wish:

```
. ado describe st0119
```

---

```
[1] package st0119 from https://www.stata-journal.com/software/sj7-1
```

---

**TITLE**

```
SJ7-1 st0119. Rasch analysis
```

**DESCRIPTION/AUTHOR(S)**

```
Rasch analysis
by Jean-Benoit Hardouin, University of Nantes, France
Support: jean-benoit.hardouin@univ-nantes.fr
After installation, type help gammasym, gausshermite,
geekel2d, raschtest, and raschtestv7
```

**INSTALLATION FILES**

```
r/raschtest.ado
r/raschtest.hlp
r/raschtestv7.ado
r/raschtestv7.hlp
g/gammasym.ado
g/gammasym.hlp
g/gausshermite.ado
g/gausshermite.hlp
g/geekel2d.ado
g/geekel2d.hlp
```

**INSTALLED ON**

```
24 Feb 2019
```

---

The package-description page shown by `ado` includes the location from which we got the package and when we installed it. It does not mention the ancillary files that were originally part of this package because they are not tracked by `ado`.

## Where packages are installed

Packages should be installed in `PLUS` or `SITE`, which are code words that Stata understands and that correspond to some real directories on your computer. Typing `sysdir` will tell you where these are, if you care.

```
. sysdir
STATA: C:\Program Files\Stata16\
BASE: C:\Program Files\Stata16\ado\base\
SITE: C:\Program Files\Stata16\ado\site\
PLUS: c:\ado\plus\
PERSONAL: c:\ado\personal\
OLDPLACE: c:\ado\
```

If you type `sysdir`, you may obtain different results.

By default, `net` installs in the PLUS directory, and `ado` tells you about what is installed there. If you are on a multiple-user system, you may wish to install some packages in the SITE directory. This way, they will be available to other Stata users. To do that, before using `net install`, type

```
. net set ado SITE
```

and when reviewing what is installed or removing packages, redirect `ado` to that directory:

```
. ado ..., from(SITE)
```

In both cases, you type `SITE` because Stata will understand that `SITE` means the site `ado`-directory as defined by `sysdir`. To install into `SITE`, you must have write access to that directory.

If you reset where `net` installs and then, in the same session, wish to install into your private `ado`-directory, type

```
. net set ado PLUS
```

That is how things were originally. If you are confused as to where you are, type `net query`.

## A summary of the net command

The `net` command displays content pages and package-description pages. Such pages are provided over the Internet, and most users get them there. We recommend that you start at <https://www.stata.com> and work out from there. We also recommend using `net search` to find packages of interest to you; see [R] [net search](#).

`net from` moves you to a location and displays the content page.

`net cd` and `net link` change from your current location to other locations. `net cd` enters subdirectories of the original location. `net link` jumps from one location to another, depending on the code on the content page.

`net describe` lists a package-description page. Packages are named, and you type `net describe pkgname`.

`net install` installs a package into your copy of Stata. `net get` copies any additional files (ancillary files) to your current directory.

`net sj` and `net stb` simplify loading files from the *Stata Journal* and its predecessor, the *Stata Technical Bulletin*.

```
net sj vol-issue
```

is a synonym for typing

```
net from https://www.stata-journal.com/software/sjvol-issue
```

whereas

```
net sj vol-issue insert
```

is a synonym for typing

```
net from https://www.stata-journal.com/software/sjvol-issue
net describe insert
```

`net set` controls where `net` installs files. By default, `net` installs in the PLUS directory; see [P] [sysdir](#). `net set ado SITE` would cause subsequent `net` commands to install in the SITE directory. `net set other` sets where ancillary files, such as `.dta` files, are installed. The default is the current directory.

`net query` displays the current `net from`, `net set ado`, and `net set other` settings.

## A summary of the ado command

The `ado` command lists the package descriptions of previously installed packages.

Typing `ado` without arguments is the same as typing `ado dir`. Both list the names and titles of the packages you have installed.

`ado describe` lists full package-description pages.

`ado uninstall` removes packages from your computer.

Because you can install packages from a variety of sources, the package names may not always be unique. Thus the packages installed on your computer are numbered sequentially, and you may refer to them by name or by number. For instance, say that you wanted to get rid of the `robust-to-everything` statistic command you installed. Type

```
. ado, find("robust-to-everything")
[15] package rte_stat from http://www.wemakeitupaswego.edu/faculty/sgazer
rte_stat. The robust-to-everything statistic; update.
```

You could then type

```
. ado uninstall rte_stat
```

or

```
. ado uninstall [15]
```

Typing `ado uninstall rte_stat` would work only if the name `rte_stat` were unique; otherwise, `ado` would refuse, and you would have to type the number.

The `find()` option is allowed with `ado dir` and `ado describe`. It searches the package description for the word or phrase you specify, ignoring case (`alpha` matches `Alpha`). The complete package description is searched, including the author's name and the name of the files. Thus if `rte` was the name of a command that you wanted to eliminate, but you could not remember the name of the package, you could type

```
. ado, find(rte)
[15] package rte_stat from http://www.wemakeitupaswego.edu/faculty/sgazer
rte_stat. The robust-to-everything statistic; update.
```

## Relationship of net and ado to the point-and-click interface

Users may instead select **Help > SJ and community-contributed commands**. There are advantages and disadvantages:

1. Flipping through content and package-description pages is easier; it is much like a browser. See [GS] **19 Updating and extending Stata—Internet functionality** (*GSM*, *GSU*, or *GSW*).
2. When browsing a product-description page, note that the `.sthlp` files are highlighted. You may click on `.sthlp` files to review them before installing the package.
3. You may not redirect from where `ado` searches for files.

## Creating your own site

Users who wish to share content with the Stata community often do so via the Statistical Software Components (SSC) archive. See [R] [ssc](#) and also see <http://repec.org/bocode/s/sscsubmit.html>.

The rest of this entry concerns how to create your own site to distribute additions to Stata. The idea is that you have written additions for use with Stata—say, `xyz.ado` and `xyz.sthlp`—and you wish to put them out so that coworkers or researchers at other institutions can easily install them. Or, perhaps you just have a dataset that you and others want to share.

In any case, all you need is a webpage. You place the files that you want to distribute on your webpage (or in a subdirectory), and you add two more files—a content file and a package-description file—and you are done.

## Format of content and package-description files

The content file describes the content page. It must be named `stata.toc`:

---

```

begin stata.toc
OFF                                     (to make site unavailable temporarily)
* lines starting with * are comments; they are ignored
* blank lines are ignored, too
* v indicates version—specify v 3, which is the current version of .toc files
v 3
* d lines display description text
* the first d line is the title, and the remaining ones are text
* blank d lines display a blank line
d title
d text
d text
d
...
* l lines display links
l word-to-show path-or-url [description]
l word-to-show path-or-url [description]
...
* t lines display other directories within the site
t path [description]
t path [description]
...
* p lines display packages
p pkgname [description]
p pkgname [description]
...
end stata.toc

```

---

Package files describe packages and are named `pkgname.pkg`:

```

-----begin pkgname.pkg-----
* lines starting with * are comments; they are ignored
* blank lines are ignored, too
* v indicates version—specify v 3, which is the current version of .toc files
v 3
* d lines display package description text
* the first d line is the title, and the remaining ones are text
* blank d lines display a blank line
d title
d text
d Distribution-Date: date
d text
d
...
* f identifies the component files
f [path/]filename [description]
f [path/]filename [description]
...
* e line is optional; it means stop reading
e
-----end pkgname.pkg-----

```

Note the Distribution-Date description line. This line is optional but recommended. Stata can look for updates to community-contributed programs with the `ado update` command if the package files from which those programs were installed contain a Distribution-Date description line.

## Example 1

Say that we want the user to see the following:

```

.net from http://www.university.edu/~me
-----
http://www.university.edu/~me
Chris Farrar, Uni University
-----
PACKAGES you could -net describe-:
    xyz          interval-truncated survival
.net describe xyz
-----
package xyz from http://www.university.edu/~me
-----
TITLE
    xyz. interval-truncated survival.
DESCRIPTION/AUTHOR(S)
    C. Farrar, Uni University.
INSTALLATION FILES                                (type net install xyz)
    xyz.ado
    xyz.sthlp
ANCILLARY FILES                                    (type net get xyz)
    sample.dta
-----

```

The files needed to do this would be

```

-----begin stata.toc-----
v 3
d Chris Farrar, Uni University
p xyz interval-truncated survival
-----end stata.toc-----

```

```
-----begin xyz.pkg-----  
v 3  
d xyz. interval-truncated survival.  
d C. Farrar, Uni University.  
f xyz.ado  
f xyz.sthlp  
f sample.dta  
-----end xyz.pkg-----
```

On his homepage, Chris would place the following files:

stata.toc	(shown above)
xyz.pkg	(shown above)
xyz.ado	file to be delivered (for use by <code>net install</code> )
xyz.sthlp	file to be delivered (for use by <code>net install</code> )
sample.dta	file to be delivered (for use by <code>net get</code> )

Chris does nothing to distinguish ancillary files from installation files.

## Example 2

S. Gazer wants to create a more complex site:

```
. net from http://www.wemakeitupaswego.edu/faculty/sgazer
```

---

```
http://www.wemakeitupaswego.edu/faculty/sgazer
```

```
Data-free inference materials
```

---

S. Gazer, Department of Applied Theoretical Mathematics

Also see my homepage for the preprint of "Irrefutable inference".

PLACES you could **-net link-** to:

stata                   StataCorp website

DIRECTORIES you could **-net cd-** to:

ir                      irrefutable inference programs (work in progress)

PACKAGES you could **-net describe-**:

rtec                   Robust-to-everything-conceivable statistic

rte                    Robust-to-everything statistic

```
. net describe rte
```

---

```
package rte from http://www.wemakeitupaswego.edu/faculty/sgazer/
```

---

**TITLE**

```
rte. The robust-to-everything statistic; update.
```

**DESCRIPTION/AUTHOR(S)**

```
S. Gazer, Dept. of Applied Theoretical Mathematics, WMIUAWG Univ.  
Aleph-0 100% confidence intervals proved too conservative for some  
applications; Aleph-1 confidence intervals have been substituted.  
The new robust-to-everything supplants the previous robust-to-  
everything-conceivable statistic. See "Inference in the absence  
of data" (forthcoming). After installation, see help rte.
```

```
Distribution-Date: 20190320
```

```
Support: email sgazer@wemakeitupaswego.edu
```

**INSTALLATION FILES**

```
(type net install rte_stat)
```

```
rte.ado  
rte.sthlp  
nullset.ado  
random.ado
```

**ANCILLARY FILES**

```
(type net get rte_stat)
```

```
empty.dta
```

---

The files needed to do this would be

---

```
-----begin stata.toc-----
```

```
v 3  
d Data-free inference materials  
d S. Gazer, Department of Applied Theoretical Mathematics  
d  
d Also see my homepage for the preprint of "Irrefutable inference".  
l stata https://www.stata.com  
t ir irrefutable inference programs (work in progress)  
p rtec Robust-to-everything-conceivable statistic  
p rte Robust-to-everything statistic
```

```
-----end stata.toc-----
```

---

```
-----begin rte.pkg-----
```

```
v 3  
d rte. The robust-to-everything statistic; update.  
d {bf:S. Gazer, Dept. of Applied Theoretical Mathematics, WMIUAWG Univ.}  
d Aleph-0 100% confidence intervals proved too conservative for some  
d applications; Aleph-1 confidence intervals have been substituted.  
d The new robust-to-everything supplants the previous robust-to-  
d everything-conceivable statistic. See "Inference in the absence  
d of data" (forthcoming). After installation, see help {bf:rte}.  
d  
d Distribution-Date: 20190320  
d  
d Support: email sgazer@wemakeitupaswego.edu  
f rte.ado  
f rte.sthlp  
f nullset.ado  
f random.ado  
f empty.dta
```

```
-----end rte.pkg-----
```

On his homepage, Mr. Gazer would place the following files:

```

stata.toc      (shown above)
rte.pkg        (shown above)
rte.ado        (file to be delivered)
rte.sthlp      (file to be delivered)
nullset.ado    (file to be delivered)
random.ado     (file to be delivered)
empty.dta      (file to be delivered)

rtec.pkg       the other package referred to in stata.toc
rtec.ado       the corresponding files to be delivered
rtec.sthlp

ir/stata.toc   the contents file for when the user types net cd ir
ir/...         whatever other .pkg files are referred to
ir/...         whatever other files are to be delivered

```

If Mr. Gazer later updated the `rte` package, he could change the Distribution-Date description line in his package. Then, if someone who had previously installed the `rte` packaged wanted to obtain the latest version, that person could use the `ado update` command; see [R] [ado update](#).

For complex sites, a different structure may prove more convenient:

```

stata.toc      (shown above)
rte.pkg        (shown above)
rtec.pkg       the other package referred to in stata.toc

rte/           directory containing rte files to be delivered:
rte/rte.ado    (file to be delivered)
rte/rte.sthlp  (file to be delivered)
rte/nullset.ado (file to be delivered)
rte/random.ado (file to be delivered)
rte/empty.dta  (file to be delivered)

rtec/         directory containing rtec files to be delivered:
rtec/...      (files to be delivered)

ir/stata.toc   the contents file for when the user types net cd ir
ir/*.*.pkg     whatever other package files are referred to
ir/*/*.*       whatever other files are to be delivered

```

If you prefer this structure, it is simply a matter of changing the bottom of the `rte.pkg` from

```

f rte.ado
f rte.sthlp
f nullset.ado
f random.ado
f empty.dta

```

to

```

f rte/rte.ado
f rte/rte.sthlp
f rte/nullset.ado
f rte/random.ado
f rte/empty.dta

```

In writing paths and files, the directory separator forward slash (/) is used, regardless of operating system, because this is what the Internet uses.

It does not matter whether the files you put out are in Windows, Mac, or Unix format (how lines end is recorded differently). When Stata reads the files over the Internet, it will figure out the file format on its own and will automatically translate the files to what is appropriate for the receiver.



## Additional package directives

**F** *filename* is similar to **f** *filename*, except that, when the file is installed, it will always be copied to the system directories (and not the current directory).

With **f** *filename*, the file is installed into a directory according to the file's suffix. For instance, `xyz.ado` would be installed in the system directories, whereas `xyz.dta` would be installed in the current directory.

Coding **F** `xyz.ado` would have the same result as coding **f** `xyz.ado`.

Coding **F** `xyz.dta`, however, would state that `xyz.dta` is to be installed in the system directories.

**g** *platformname filename* is also a variation on **f** *filename*. It specifies that the file be installed only if the user's operating system is of type *platformname*; otherwise, the file is ignored. The platform names are **WIN64** (64-bit x86-64) for Windows; **MACINTEL64** (64-bit Intel, GUI) and **OSX.X8664** (64-bit Intel, console) for Mac; and **LINUX64** (64-bit x86-64) and **LINUX64P** (64-bit x86-64, libpng v1.6) for Linux. For **LINUX64P**, the **LINUX64**-specific version of a file will be installed if no **LINUX64P**-specific one exists. **LINUX64** will not install a **LINUX64P**-specific version of a file because of possible changes in system libraries.

**G** *platformname filename* is a variation on **F** *filename*. The file, if not ignored, is to be installed in the system directories.

**g** *platformname filename1 filename2* is a more detailed version of **g** *platformname filename*. In this case, *filename1* is the name of the file on the server (the file to be copied), and *filename2* is to be the name of the file on the user's system; for example, you might code

```
g WIN64 mydll.forwin mydll.plugin
g LINUX64 mydll.forlinux mydll.plugin
```

When you specify one *filename*, the result is the same as specifying two identical *filenames*.

**G** *platformname filename1 filename2* is the install-in-system-directories version of **g** *platformname filename1 filename2*.

**h** *filename* asserts that *filename* must be loaded, or this package is not to be installed; for example, you might code

```
g WIN64 mydll.forwin mydll.plugin
g LINUX64 mydll.forlinux mydll.plugin
h mydll.plugin
```

if you were offering the plugin `mydll.plugin` for Windows and Linux only.

## SMCL in content and package-description files

The text listed on the second and subsequent **d** lines in both `stata.toc` and `pkgname.pkg` may contain SMCL as long as you include **v** 3; see [P] [smcl](#).

Thus, in `rte.pkg`, S. Gazer coded the third line as

```
d {bf:S. Gazer, Dept. of Applied Theoretical Mathematics, WMIUAWG Univ.}
```

## Error-free file delivery

Most people transport files over the Internet and never worry about the file being corrupted in the process because corruption rarely occurs. If, however, the files must be delivered perfectly or not at all, you can include checksum files in the directory.

For instance, say that `big.dta` is included in your package and that it must be sent perfectly. First, use Stata to make the checksum file for `big.dta`

```
. checksum big.dta, save
```

That command creates a small file called `big.sum`; see [\[D\] checksum](#). Then, copy both `big.dta` and `big.sum` to your homepage. If `set checksum` is on (the default is `off`), whenever Stata reads *filename.whatever* over the net, it also looks for *filename.sum*. If it finds such a file, it uses the information recorded in it to verify that what was copied was error free.

If you do this, be cautious. If you put `big.dta` and `big.sum` on your homepage and then later change `big.dta` without changing `big.sum`, people will think that there are transmission errors when they try to download `big.dta`.

## Also see

- [\[R\] ado update](#) — Update community-contributed packages
- [\[R\] net search](#) — Search the Internet for installable packages
- [\[R\] netio](#) — Control Internet connections
- [\[R\] search](#) — Search Stata documentation and other resources
- [\[R\] sj](#) — Stata Journal and STB installation instructions
- [\[R\] ssc](#) — Install and uninstall packages from SSC
- [\[R\] update](#) — Check for official updates
- [\[D\] checksum](#) — Calculate checksum of file
- [\[P\] smcl](#) — Stata Markup and Control Language
- [\[GSM\] 19 Updating and extending Stata—Internet functionality](#)
- [\[GSU\] 19 Updating and extending Stata—Internet functionality](#)
- [\[GSW\] 19 Updating and extending Stata—Internet functionality](#)
- [\[U\] 29 Using the Internet to keep up to date](#)