**Syntax**

```
. adoupdate [pkglist] [ , options ]
```

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>update</td>
<td>perform update; default is to list packages that have updates, but not to update them</td>
</tr>
<tr>
<td>all</td>
<td>include packages that might have updates; default is to list or update only packages that are known to have updates</td>
</tr>
<tr>
<td>ssconly</td>
<td>check only packages obtained from SSC; default is to check all installed packages</td>
</tr>
<tr>
<td>dir(dir)</td>
<td>check packages installed in dir; default is to check those installed in PLUS</td>
</tr>
<tr>
<td>verbose</td>
<td>provide output to assist in debugging network problems</td>
</tr>
</tbody>
</table>

**Description**

User-written additions to Stata are called packages. These packages can add remarkable abilities to Stata. Packages are found and installed by using `ssc`, `search`, and `net`; see [R] `ssc`, [R] `search`, and [R] `net`.

User-written packages are updated by their developers, just as official Stata software is updated by StataCorp.

To determine whether your official Stata software is up to date, and to update it if it is not, you use `update`; see [R] `update`.

To determine whether your user-written additions are up to date, and to update them if they are not, you use `adoupdate`.

**Options**

`update` specifies that packages with updates be updated. The default is simply to list the packages that could be updated without actually performing the update.

The first time you `adoupdate`, do not specify this option. Once you see `adoupdate` work, you will be more comfortable with it. Then type

```
. adoupdate, update
```

The packages that can be updated will be listed and updated.

`all` is rarely specified. Sometimes, `adoupdate` cannot determine whether a package you previously installed has been updated. `adoupdate` can determine that the package is still available over the web but is unsure whether the package has changed. Usually, the package has not changed, but if you want to be certain that you are using the latest version, reinstall from the source.
Specifying all does this. Typing

```
    . adoupdate, all
```

adds such packages to the displayed list as needing updating but does not update them. Typing

```
    . adoupdate, update all
```

lists such packages and updates them.

`ssconly` is a popular option. Many packages are available from the Statistical Software Components (SSC) archive—often called the Boston College Archive—which is provided at [http://repec.org](http://repec.org). Many users find most of what they want there. See [R] ssc for more information on the SSC.

`ssconly` specifies that `adoupdate` check only packages obtained from that source. Specifying this option is popular because SSC always provides distribution dates, and so `adoupdate` can be certain whether an update exists.

`dir(dir)` specifies which installed packages be checked. The default is `dir(PLUS)`, and that is probably correct. If you are responsible for maintaining a large system, however, you may have previously installed packages in `dir(SITE)`, where they are shared across users. See [P] sysdir for an explanation of these directory codewords. You may also specify an actual directory name, such as `C:\mydir`.

`verbose` is specified when you suspect network problems. It provides more detailed output that may help you diagnose the problem.

### Remarks and examples

Do not confuse `adoupdate` with `update`. Use `adoupdate` to update user-written files. Use `update` to update the components (including ado-files) of the official Stata software. To use either command, you must be connected to the Internet.

Remarks are presented under the following headings:

- Using `adoupdate`
- Possible problem the first time you run `adoupdate` and the solution
- Notes for developers

#### Using `adoupdate`

The first time you try `adoupdate`, type

```
    . adoupdate
```

That is, do not specify the `update` option. `adoupdate` without `update` produces a report but does not update any files. The first time you run `adoupdate`, you may see messages such as

```
    . adoupdate
    (note: package utx was installed more than once; older copy removed)
    (remaining output omitted)
```

Having the same packages installed multiple times is common; `adoupdate` cleans that up.

The second time you run `adoupdate`, pick one package to update. Suppose that the report indicates that package `st0008` has an update available. Type

```
    . adoupdate st0008, update
```

You can specify one or many packages after the `adoupdate` command. You can even use wildcards such as `st*` to mean all packages that start with `st` or `st*8` to mean all packages that start with `st` and end with `8`. You can do that with or without the `update` option.
Finally, you can let adoupdate update all your user-written additions:

```
. adoupdate, update
```

**Possible problem the first time you run adoupdate and the solution**

The first time you run adoupdate, you might get many duplicate messages:

```
. adoupdate
   (note: package ___ installed more than once; older copy removed)
   (note: package ___ installed more than once; older copy removed)
   (note: package ___ installed more than once; older copy removed)
   ...
   (note: package ___ installed more than once; older copy removed)
   (remaining output omitted)
```

Some users have hundreds of duplicates. You might even see the same package name repeated more than once:

```
   (note: package stylus installed more than once; older copy removed)
   (note: package stylus installed more than once; older copy removed)
```

That means that the package was duplicated twice.

Stata tolerates duplicates, and you did nothing wrong when you previously installed and updated packages. adoupdate, however, needs the duplicates removed, mainly so that it does not keep checking the same files.

The solution is to just let adoupdate run. adoupdate will run faster next time, when there are no (or just a few) duplicates.

**Notes for developers**

adoupdate reports whether an installed package is up to date by comparing its distribution date with that of the package available over the web.

If you are distributing software, include the line

```
   d Distribution-Date: date
```

somewhere in your .pkg file. The capitalization of Distribution-Date does not matter, but include the hyphen and the colon as shown. Code the date in either of two formats:

- all numeric: `yyyymmdd`, for example, 20120701
- Stata standard: `ddMONyyyy`, for example, 01ju12012

**Stored results**

adoupdate stores the following in `r()`:

- **Macros**
  - `r(pkglist)` a space-separated list of package names that need updating (update not specified) or that were updated (update specified)
Also see

[R] net — Install and manage user-written additions from the Internet
[R] search — Search Stata documentation and other resources
[R] ssc — Install and uninstall packages from SSC
[R] update — Check for official updates