

**window manage** — Manage window characteristics[Syntax](#)[Description](#)[Remarks and examples](#)[Also see](#)

## Syntax

*Minimize or restore the main Stata window*

```
window manage minimize
```

```
window manage restore
```

*Manage window preferences*

```
window manage prefs { load "prefname" | save "prefname" | default }
```

*Restore file associations (Windows only)*

```
window manage associate
```

*Reset main window title (Unix and Windows only)*

```
window manage maintitle { reset | "title" }
```

*Set Dock icon's label (Mac only)*

```
window manage docklabel [ "label" ]
```

*Bring windows forward*

```
window manage forward window-name
```

where *window-name* can be `command`, `doeditor`, `graph`, `help`, `results`, `review`, `variables`, or `viewer`.

Commands to manage Graph windows

```
window manage print graph
```

```
window manage forward graph ["graphname"]
```

```
window manage close graph [{"graphname" | _all}]
```

```
window manage rename graph oldgraphname newgraphname
```

Commands to manage Viewer windows

```
window manage print viewer ["viewername"]
```

```
window manage forward viewer ["viewername"]
```

```
window manage close viewer [{"viewername" | _all}]
```

## Description

`window manage` allows Stata programs to invoke features from Stata's main menu.

## Remarks and examples

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

`window manage` accesses various parts of Stata's windowed interface that would otherwise be available only interactively. For instance, say that a programmer wanted to ensure that the Graph window was brought to the front. An interactive user would do that by selecting **Graph** from the **Window** menu. A Stata program could be made to do the same thing by coding `window manage forward graph`.

Remarks are presented under the following headings:

- Minimizing or restoring the main window*
- Windowing preferences*
- Restoring file associations (Windows only)*
- Resetting the main window title*
- Setting Dock icon's label (Mac only)*
- Bringing windows forward*
- Commands to manage Graph windows*
- Commands to manage Viewer windows*

## Minimizing or restoring the main window

`window manage minimize` minimizes (hides) the Stata window. With Stata for Windows and Stata for Unix, this has the same effect as clicking on the minimize button on Stata's title bar. With Stata for Mac, this has the same effect as selecting **Hide Stata** from the **Stata** menu.

For example,

```
window manage minimize
```

minimizes the overall Stata window if you are using Stata for Windows or Stata for Unix and hides Stata's windows if you are using Stata for Mac.

`window manage restore` restores the Stata window if necessary.

With Stata for Windows, this command has the same effect as clicking the Stata button on the taskbar. With Stata for Mac, this command has the same effect as clicking on the Stata icon on the Dock. With Stata for Unix, this command has the same effect as clicking on the Stata icon in the Window Manager.

For example,

```
window manage restore
```

restores Stata's overall window to its normal, nonminimized state.

## Windowing preferences

`window manage prefs {load "prefname" | save "prefname" | default}` loads, saves, and restores windowing preferences.

`window manage prefs load "prefname"` is equivalent to selecting **Edit > Preferences > Manage Preferences > Load Preferences** and loading a named preference set. `window manage prefs save "prefname"` is equivalent to selecting **Edit > Preferences > Manage Preferences > Save Preferences** and naming a new preference set. `window manage prefs default` is equivalent to selecting **Edit > Preferences > Manage Preferences > Load Preferences > Factory Settings**. In Stata for Mac, the **Preferences** menu is located in the **Stata** menu.

For example,

```
window manage prefs default
```

restores Stata's windows to their "factory" appearance.

## Restoring file associations (Windows only)

In Stata for Windows, `window manage associate` restores the default actions for Stata file types. For example, another application could take over the `.dta` extension so that double-clicking on a Stata dataset would no longer launch Stata. `window manage associate` restores the association between all Stata file extensions (such as `.dta`) and Stata. This is equivalent to selecting **Edit > Preferences > Restore File Associations**.

## Resetting the main window title

In Stata for Unix and Stata for Windows, `window manage maintitle "title"` changes the title of the main Stata Window. The title may be reset to the default with `window manage maintitle reset`.

## Setting Dock icon's label (Mac only)

In Stata for Mac, `window manage docklabel "label"` sets the label to be displayed in the badge area of Stata's application icon in the Dock. To clear the badge label, enter the command with no label. You should limit the label to 6 characters or fewer; otherwise, the label will be truncated.

`window manage docklabel` can be useful for displaying the progress of a do-file.

For example,

```
----- begin test.do -----  
do test1.do  
window manage docklabel "25%"  
do test2.do  
window manage docklabel "50%"  
do test3.do  
window manage docklabel "75%"  
do test4.do  
window manage docklabel  
----- end test.do -----
```

## Bringing windows forward

`window manage forward window-name` brings the specified window to the top of all other Stata windows. This command is equivalent to selecting one of the available windows from the **Window** menu. The following table lists the *window-names* that `window manage forward` understands:

<i>window-name</i>	Stata window
<code>command</code>	Command window
<code>doeditor</code>	Do-file editor window
<code>graph</code>	Graph window
<code>help</code>	Help/search window
<code>results</code>	Results window
<code>review</code>	Review window
<code>variables</code>	Variables window
<code>viewer</code>	Viewer window

If a window had not been available on Stata's **Window** menu (if it had been grayed out), specifying *window-name* after `window manage forward` would do nothing. For example, if there is no current graph, `window manage forward graph` will do nothing; it is not an error.

For example,

```
window manage forward results
```

brings the Results window to the top of the other Stata windows.

Under Stata for Mac and Stata for Unix, specifying the Command, Results, Review, or Variables windows will bring the main Stata window forward because these windows are all contained within one window.

## Commands to manage Graph windows

### Printing

`window manage print graph` invokes the action of the **File > Print > Graph (Graph)** menu item. If there is no current graph, `window manage print` does nothing; it does not return an error.

For example,

```
window manage print graph
```

displays the print dialog box just as if you pulled down **File > Print > Graph (Graph)**.

### Bringing forward

`window manage forward graph` [*graphname*] brings the Graph window named *graphname*, if it exists, to the top of other windows. If *graphname* is not specified and there are multiple graph windows open, `window manage forward graph` brings the topmost Graph window to the top of other windows.

### Closing

`window manage close graph` [*graphname* | `_all`] closes the Graph windows named *graphname*, if it exists. If `_all` is specified, all Graph windows are closed. If *graphname* is not specified and an unnamed Graph window exists, the unnamed Graph window will be closed.

### Renaming

`window manage rename` *oldgraphname newgraphname* renames the Graph window named *oldgraphname*, if it exists, to *newgraphname*. Note that this is not intended for an end user to use; it is a utility to be used by `graph rename`. End users should use `graph rename` to rename graphs.

## Commands to manage Viewer windows

### Printing

`window manage print viewer` [*viewername*] prints the Viewer window named *viewername*, if it exists. If *viewername* is not specified and there are multiple Viewer windows open, `window manage print` prints the topmost Viewer window. If there is no current Viewer window, `window manage print` does nothing; it does not return an error.

### Bringing forward

`window manage forward viewer` [*viewername*] brings the Viewer window named *viewername*, if it exists, to the top of other windows. If *viewername* is not specified and there are multiple Viewer windows open, `window manage` brings the topmost Viewer window to the top of other windows.

### Closing

`window manage close viewer [viewername|,|_all]` closes the Viewer window named *viewername*, if it exists. If `_all` is specified, all Viewer windows are closed. If *viewername* is not specified and an unnamed Viewer window exists, the unnamed Viewer window will be closed.

### Also see

[P] [window programming](#) — Programming menus and windows