stty

Command

set or display terminal options

SYNOPSIS

SYNOPSIS

stty [**-a**] [**-**] [*mode*...]

DESCRIPTION

stty sets or reports console mode settings on Windows systems. To set specific modes to `off`, add a **-** (dash) before each *mode*. Specifying a *mode* without a leading dash sets the mode to `on`.

Options

-a

displays all current console modes. Each mode is displayed with a leading **-** (dash) if the mode is currently set to `off`; otherwise, only the mode name appears.

The **-a** option does not display current settings for the **hide** or **sane** modes. A description of these special modes appears later in this reference page.

Control Modes

columns

Defines the column width of your console window. This mode is system dependant, and returns an error if you specify a column width that is not supported by your operating system.

Optionally, you can specify a second argument (separated by a comma) that specifies the buffer size for the window. If you specify a column width with a comma but no buffer size, the current buffer size is used. If you specify a comma and a buffer size, but no column width, the current width is used. The following examples show how this can be used:

`stty columns 80`	sets window and buffer to 80 columns.
`stty columns 80,`	sets window to 80 columns without changing buffer.
`stty columns 80,1000`	sets window to 80 columns and buffer to 1000.
`stty columns ,1000`	sets buffer to 1000 column without changing window.
`stty columns ,`	does nothing, silently.

If you set your window size too large to fit on your screen, it is truncated to a size that fits. If you set the buffer size smaller than the window size, the buffer size is reset to the window size.

cp

Sets both the input and output code pages for the console window. This is equivalent to specifying both **icp** and **ocp**. The **stty -a** command displays **cp** when both **icp** and **ocp** are set to the same code page. It displays both **icp** and **ocp** when they differ.

cp is only valid on Vista/7/2008/8/2012/10.

display

Changes how the console window is displayed. There are four possible settings.

hide	hides the console window
max	maximizes the console window
min	minimizes the console window
restore	restores the console window

One use of this mode is in a KornShell script that performs graphical actions (for example a script with **dlg** commands), and you do not want the console visible while it is running. For example, you could start the shell script with

```
wstart -m sh.exe script.ksh
```

to minimize the window upon startup, and then have a **stty display hide** command near the beginning of the script to hide the minimized window, effectively allowing the shell script to run without any visible indication that a console has been created.

echo

Corresponds to the Win32 ***ENABLE_ECHO_INPUT*** mode. If **echo** is set, characters are echoed as they are read. This mode is only valid when **line** mode is also on. If **line** mode is turned off, **echo** mode is automatically turned off by the **stty** command.

erase

Set the erase character. You can set this character to either the backspace or delete character, by specifying `^H` or `^?`, respectively.

If you do not specify a character to use as the erase character, the current erase character is displayed.

Note:

> When using **secsh**, **xterm**, or **telnet**, the character set by **stty erase** only works as the erase character if you are running **sh.exe** on the remote Windows machine.

hide

Does not correspond to any standard Win32 console mode, but allows hiding and restoring of the console window under Windows 95/98/Me. **stty hide** hides the console window, and **stty -hide** brings it back.

This mode is deprecated and should no longer be used. Use **stty display hide** instead.

icp

Sets the input code page for the console window. The **stty -a** command displays **cp** when both **icp** and **ocp** are set to the same code page. It displays both **icp** and **ocp** when they differ.

pos

Moves the top left corner of the console window to the screen co-ordinates specified. For example

```
stty pos 100,300
```

moves the top left corner to a position 100 pixels in from the left side of your screen and 300 pixels down from the top.

The screen co-ordinates can be negative. That is, you can move the console window past the left side or above the top of the screen. You can also move it as far down or to the right as you like, but be careful, as it is possible to move the console window beyond the visible screen.

iproc

Corresponds to the Win32 ***ENABLE_PROCESSED_INPUT*** mode. This mode enables process input, meaning that any `^C` is processed by the system as an interrupt and not passed to the program as input. Also, if **line** mode is set, **iproc** mode tells the system to process backspace, carriage return, and linefeed characters.

line

Corresponds to the Win32 ***ENABLE_LINE_INPUT*** mode. If **line** mode is set, programs reading from the console only receive input (one line at a time) once the ENTER key is pressed. If this mode is not set, programs reading the console receive the input as each key is pressed.

minput

Corresponds to the Win32 ***ENABLE_MOUSE_INPUT*** mode. Allows a console program to receive mouse related events.

ocp

Sets the output code page for the console window. The **stty -a** command displays **cp** when both **icp** and **ocp** are set to the same code page. It displays both **icp** and **ocp** when they differ.

oproc

Corresponds to the Win32 ***ENABLE_PROCESSED_OUTPUT*** mode. Allows ASCII control sequences and backspace, tab, bell, carriage return, and linefeed to be processed by the system. If this mode is not set, then these special ASCII characters simply appear on the screen as special characters, rather than the appropriate motion (or sound, in the case of the bell character).

owrap

Corresponds to the Win32 ***ENABLE_WRAP_AT_EOL_OUTPUT*** mode. Allows output to wrap when it passes the rightmost column of the console. If this mode is not set, each line is truncated at the rightmost column of the console.

rows

Define the row height of your console window. This mode is system dependant, and returns an error if you specify a row height that is not supported by your operating system.

Optionally, you can specify a second argument (separated by a comma) that specifies the buffer size for the window. If you specify a row height with a comma but no buffer size, the current buffer size is used. If you specify a comma and a buffer size, but no row height, the current height is used. The following examples show how this can be used:

`stty rows 25`	sets both window and buffer to 25 rows.
`stty rows 25,`	sets window to 25 rows without changing buffer.
`stty rows 25,1000`	sets window to 25 rows and buffer to 1000 rows.
`stty rows ,1000`	sets buffer to 1000 rows without changing window.
`stty rows ,`	does nothing, silently.

If you set your window size too large to fit on your screen, it is truncated to a size that fits. If you set the buffer size smaller than the window size, the buffer size is reset to the window size.

`sane`

Turns on all of the above input and output modes, which is the default (and typical) console mode. Using the **`stty -sane`** command is not recommended, since it turns off all console modes.

`winput`

Corresponds to the Win32 ***ENABLE_WINDOW_INPUT*** mode. Allows a program to read events related to the resizing of a console window.

Notes

Turning off console mode bits can be dangerous. For example, if you do a **`stty -iproc`**, processed input is turned off and carriage return, linefeed, backspace, and `^C` no longer have any special meaning to the console, and programs may not receive their expected input.

Certain mode bits are automatically forced to `on` by **`cmd.exe`** and the MKS KornShell (in **`vi`** or **`emacs`** editing mode). Most console applications, however, do not affect mode bits unless editing modes are active, and **`stty`** actions have full effect.

ENVIRONMENT VARIABLES

TK_USE_CTRLD_AS_CONSOLE_EOF

When set, allows the use of CTRL-D as the EOF (end-of-file) character for the MKS KornShell (including **`sh`**, **`bash`**, **`ksh`** and **`resh`**) and MKS C Shell as well as for any utility that can accept input from the console.

PORTABILITY

Windows Vista. Windows 7. Windows Server 2008. Windows 8. Windows Server 2012. Windows 10.

The functionality of this command is a subset of the UNIX functionality.

AVAILABILITY

PTC MKS Toolkit for Power Users
PTC MKS Toolkit for System Administrators
PTC MKS Toolkit for Developers
PTC MKS Toolkit for Interoperability
PTC MKS Toolkit for Professional Developers
PTC MKS Toolkit for Enterprise Developers
PTC MKS Toolkit for Enterprise Developers 64-Bit Edition