## NAME

cmdtool – run a shell (or other program) in an OpenWindows enhanced terminal window

## SYNOPSIS

**cmdtool** [ –**C** ] [ –**M** *bytes* ] [ –**P** *count* ] [ –**B** *boldstyle* ] [ –**I** *command* ]
[ *generic-tool-arguments* ] [ *program* [ *program-arguments* ] ]

## DESCRIPTION

**Cmdtool** is the standard OpenWindows support facility for shells and other programs. When invoked, **cmdtool** runs a program (usually a shell) in a text-based command window. Characters typed on the keyboard are inserted at the caret. If the program is a shell, that shell accepts and runs commands in the usual manner. **Cmdtool** also supports programs that perform cursor motions directly, such as **vi**(1).

The text of the current command line can be edited using normal **textedit**(1) functions. The command window displays a log of the session, which can be scrolled through using the scrollbar (unless the escape command mode is in effect). This log can be edited, and saved by choosing the '**Store as New File**' item in the text facility's pop-up menu.

**Cmdtool's** log file has a default wrap-around value of 100,000 bytes. When the size of the log file exceeds this value, text at the top of the file is lost as new text is added at the insertion point. The default value can be changed with the –**M** option or by placing the following line in the user's **~/.Xdefaults** file:

**cmdtool.maxLogFileSize:** *max_size*

## OPTIONS

–**C** Console **cmdtool.** Redirect system console output to this cmdtool. Display console messages in this **cmdtool**, which might otherwise appear in unexpected places on the workstation screen. Since a **cmdtool** window can be scrolled, console error messages can be recorded for later examination.

–**M** *bytes* Set the log file **/tmp/tty.txt.pid** to wrap-around after the indicated number of *bytes* where the default value is 100,000 bytes. When the size of the log file exceeds this value, text at the beginning of the file is lost as new text is added at the insertion point. The log file contains editing characters as well as text so the number of characters in the cmdtool log itself may be less than the size of the log file. Also, the process size of the cmdtool is not limited to the indicated number of *bytes* .

–**P** *count* Checkpoint the log after every set of *count* editing operations.

–**B** *boldstyle* Set the style for displaying bold text to *boldstyle*. *boldstyle* can be a string specifying one of the choices for the **term.boldstyle** default, or it may be a numerical value for one of those choices, from 0 to 8, corresponding to the placement of the choice in the list.

–**I** *command* Pass *command* to the shell. SPACE characters within the command must be escaped.

*generic-tool-arguments*
**cmdtool** accepts the generic tool arguments listed in **xview**(7).

*program* [ *program-arguments* ]
If a *program* argument is present, **cmdtool** runs it and passes any remaining arguments to that *program*. If no *program* is given, **cmdtool** runs the program indicated by the **SHELL** environment variable, or **/bin/sh** by default.

## USAGE

### .Xdefaults File Options

You can specify a number of defaults using the options in the **.Xdefaults** file that effect the behavior of **cmdtool**. The ones of interest are those that begin with **text**,**term**, or **keyboard**. See **xview**(7) for more detailed information.

### cmdtool Windows

The window created by **cmdtool** is based on the text facility that is described in the **textedit** man page. The user is given a prompt at which to type commands and pop-up menus from which to select command options.

**cmdtool** windows support cursor motions, using an **/etc/terminfo** entry called **sun-cmd .** Command windows automatically set the **TERM** environment variable to **sun-cmd**. So, if you **rlogin**(1) to a machine that does not have an entry for **sun-cmd** in its **/etc/terminfo** file, the error message '**Type sun-cmd unknown**' results. To rectify this, type the command **set TERM=sun.** Programs written using the **curses**(3X) library packages will work in a command window, but programs hard-coded for **sun**-type terminals may not. When supporting a program that performs cursor motions, the command window automatically takes on the characteristics of a tty window (as with **shelltool**(1)). When that program terminates or sleeps, the full command window functionality is restored.

**cmdtool** supports programs that use CBREAK and NO ECHO terminal modes. This support is normally invisible to the user. However, programs that use RAW mode, such as **rlogin**(1) and **script**(1), inhibit command-line editing with the mouse. In this case, however, tty-style ERASE, word-kill and line-kill characters can still be used to edit the current command line.

### The cmdtool Menu

The **cmdtool** window menu is called the **Term Pane** menu and contains the following options and their submenus:

**History** Creates a a list of commands used during the cmdtool session.

- **Mode**
  - **Editable** You can edit the contents of the window.
  - **Read Only** You can only read from the window.
- **Store Log as new file** Create a new file that contains the contents of the log.
- **Clear log** Clears all entries from the log.

**Edit** Provides a set of editing functions for this window.

- **Again** Repeats the last action.
- **Undo**
  - **Undo Last Edit** Undoes the last edit made in **cmdtool .**
  - **Undo All Edits**

Undoes all edits made during this session of **cmdtool .**

**Copy** Makes a copy of the selected text and places it on the clipboard.

**Paste** Pastes a copy of the text stored on the clipboard at the cursor location.

**Cut** Deletes the selected text.

**Find** Provides a set of find and replace functions.

**Find and Replace**
Brings up a pop-up menu containing text fields and command buttons that allow you to search forward and backward through the file being edited for specific text strings. Allows you to specify options for the replacement of text.

**Find Selection**
**Forward** Searches forward to find a selected text string.
**Backward** Searches backward to find a selected text string.

**Find Marked Text**
Brings up a pop-up menu that allows you to find text that is included between specified bracket pairs. Also allows you to insert or remove bracket pairs around selected text.

**Replace |>field<| >**
Allows you to replace selected text forward and backward throughout the file.

**Extras** A user-definable pull-right menu controlled by the **/usr/openwin/lib/locale/C/xview/.text_extras_menu** file. This can be overridden in two ways:
1) Change the value of the **.Xdefaults** parameter **text.extrasMenuFilename** to the correct file path.
2) Set the environment variable **EXTRASMENU** to the file desired.
Note that option 1 overrides option 2 if both are used. For more information see the OpenWindows user documentation.

**File Editor**
**Enable** Allows you to edit files from within **cmdtool.**
**Disable** Turns off the ability to edit files from within **cmdtool.**

**Scrolling**

**Enables Scrolling**
Enables scrolling within **cmdtool.**

**Disable Scrolling**
Turns off the ability to scroll within **cmdtool.** Once scrolling in **cmdtool** is disabled, its functionality is identical to **shelltool** and a more restricted menu appears. Selecting the **Enable Scrolling** option from the restricted menu restores the full menu and functionality of **cmdtool.**

**User Defined Keyboard Remapping**

The file **~/.textswrc** specifies filter programs that are assigned to (available) function keys. These filters are applied to the contents of the primary selection. Their output is entered at the caret.

**Accelerators**

Text facility accelerators that are especially useful in command windows are described here. See **textedit**(1) for more information.

CTRL-RETURN Position the caret at the bottom, and scroll it into view as determined by **Text.LowerContext**.

CAPS-lock
Toggle between all-upper-case keyboard input, and mixed-case.

## FILES

**/tmp/tty.txt.***pid* log file
**~/.textswrc**
**~/.ttyswrc**
**/usr/openwin/lib/locale/C/xview/.text_extras_menu**
**$HOME/.TextExtraMenu**
**/etc/terminfo**
**/usr/bin/sh**

## SEE ALSO

**rlogin**(1), **script**(1), **shelltool**(1), **textedit**(1), **vi**(1), **xview**(7), **curses**(3X)

*Solaris User's Guide*

## BUGS

Typing ahead while **cmdtool** changes between its scrollable and cursor motion modes will sometimes freeze **cmdtool**.

Full terminal emulation is not complete. Some manifestations of this deficiency are:

- File completion in the C shell does not work.
- Enhanced display of text is not supported.