

SCO OpenServer Product Family Release 5

mscreen(M)

mscreen -- serial multiscreens utility

Syntax

mscreen [**-s**] [**-n** *number*] [**-t**]

Description

mscreen allows a serial terminal to have multiple login screens similar to the [multiscreen\(M\)](#) console.

Note: For full **mscreen** support the terminal must have the ability to switch internal screen pages on command and it must retain a separate cursor position for each screen page.

The options are as follows:

- s** Silent mode. This flag suppresses the startup messages, and on ``dumb'' terminals it suppresses the screen switch messages.
- n** Selects the number of serial multiscreens desired up to the maximum defined for the terminal type. If **-n** is not specified, the default is the maximum defined for the terminal type.
- t** Disables the transparent tty checking. **mscreen** normally exits silently if the terminal device name starts with the characters ``ttyp''. Device names beginning with ``ttyp'' are used as slave devices for **mscreen**. The correct names for the master tty devices begin with ``ptyp''.

mscreen can be used on both ``smart'' and ``dumb'' terminals. Although it is optimized to take advantage of smart terminals with screen memory, **mscreen** also works on dumb terminals, although the screen images are not saved during screen changes. **mscreen** also supports terminals with two (or more) serial ports that are connected to different computers.

mscreen is designed to be invoked from the *.profile* or *.login* files so that serial multiscreens can be started when you log in (see ``Shell return codes and auto login/logout").

The ``stop'' and ``quit'' keys allow you to logout from all screens with a single keystroke.

mscreen pseudo terminal usage

The steps to use **mscreen** are:

- set up a serial terminal and enable it
- use the command **mkdev pty** to configure the pseudo terminal device files (*/dev/ttyp** and */dev/ptyp**).
- check that the capabilities for the terminal type are defined in the file */etc/mscreencap* or by the environment variable **MSCREENCAP** (see "Configuration" below)
- test **mscreen** by invoking it directly from the terminal
- add **mscreen** to your *.profile* (for **ksh(C)** and **sh(C)**) or *.login* (for **csch(C)**).

Configuration

mscreen determines the terminal type of the terminal it is invoked from by examining the environment variable **TERM**. **mscreen** looks in */etc/mscreencap* or in the filename contained in the environment variable **MSCREENCAP** to get the capabilities for the terminal type.

mscreencap format

mscreencap contains an entry for each terminal type supported. An entry may have several names if the support for several terminal types is the same. Within an entry are the key mappings for each potential pseudo terminal. Each pseudo terminal has a help key string, an input string (the sequence generated by the key that selects this screen), and an optional output string (the sequence to send to the terminal that will cause a page switch). The input and output strings are in a termcap like format: (the backslash and caret are special lead in (escape) characters).

\nnn an octal number, one to three digits are allowed

\n newline

\r carriage return

\t tab

\b backspace

\f form feed

\E escape (hex 1b, octal 33).

`\` enter backslash as a data character

`\^` enter caret as a data character

`^x` stands for `<Ctrl>x` where `x` can be one of: `@ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_` Effectively, the caret can generate hex 01 through hex 1f.

If a terminal type has no output strings then it is assumed to be a dumb terminal that does not have multiple internal memory pages.

There are five special entries that allow the user to define keys to support the other functions of **mscreen**. The entries are:

```help```  
prints a list of all of the keys that are currently available and their functions,

```who```  
prints the name of the current screen,

```stop```  
terminates **mscreen** and returns an exit value of zero,

```quit```  
terminates **mscreen** and returns a non-zero exit value,

the fifth entry is a dummy used for terminals with multiple ports.

The format is:

```
#this is a comment and may only appear between entries
entryname|alias1|alias1...|aliasn:
    :specialname,helpname,inputstring,pageselectstring:
...
    :specialname,helpname,inputstring,pageselectstring:
entryname|alias1|alias1...|aliasn:
    :specialname,helpname,inputstring,pageselectstring:
...
    :specialname,helpname,inputstring,pageselectstring:
```

The specialname is empty for real screen entries. See the provided `/etc/mscreencap` for examples.

.mscreenrc usage and format

mscreen automatically assigns pseudo terminal devices when it starts up. If necessary, you can assign pseudo terminals by creating a file named `.mscreenrc` in your home directory.

`.mscreenrc` contains a list of pseudo terminals to be used, for example:

```
ttyp0  
ttyp1  
...  
ttypn
```

Shell return codes and auto login/logout

mscreen exits with a bad (non-zero) return code if there is an error or when the ``quit" key is pressed. The ``stop" key causes **mscreen** to exit with a good (zero) return code. This allows users to place **mscreen** in the *.login* or *.profile* files. The *.login* or *.profile* files should set up an automatic logout if the **mscreen** return code is good (zero). The following is a **cs**h sample invocation of **mscreen** for a *.login* file:

```
mscreen -n 4  
if ($status == 0) logout
```

For **sh** and **ksh**, add the following line to *.profile*:

```
if mscreen -n 4 ; then exit ; fi
```

The single key logout feature of **mscreen** works as if a normal logout was entered on each pseudo terminal. A hangup signal is sent to all of the processes on all the pseudo terminals.

Multiple port option

mscreen provides a dummy entry type. It allows **mscreen** to be placed in an inactive state while the user uses his terminal to converse through another (physical) I/O port to another computer. See the provided */etc/mscreencap* for an example. To use it, you must take the example and configure it for your needs.

Warning

Do not enable the pseudo terminals used by **mscreen** for login; **mscreen** uses */etc/utmp_getty* to create its own [getty\(M\)](#) process on each pseudo terminal.

Do not switch screen pages in **mscreen** when output is occurring because if an escape sequence is cut in half it may leave the terminal in an indeterminate state and distort the screen image.

Terminals that save the cursor location for each screen often do not save states such as insert mode, inverse video, and others. For example, you should not change screens if you are in insert mode in **vi**, and you should be careful not to change screens during an inverse video output sequence.

For inactive screens (screens other than the current one) **mscreen** saves the last 2048 characters of data (2KB). Data older than this is lost. This limit occasionally results in errors for programs that require more data than this. Some applications have screen

redraw keys (often <Ctrl>R or <Ctrl>L) that can be used to restore the normal appearance of the screen.

Limitations

mscreen has a VTIM timeout of 1/5 second for input strings.

mscreen has a limit of twenty multiscreens per user.

mscreen depends on the pseudo terminal device names starting with ``ttyp" for the slave devices and ``ptyp" for the master devices. The number of trailing characters in the device name is not significant.

See also

[mkdev\(ADM\)](#), [multiscreen\(M\)](#)

``Using serial multiscreens with mscreen" in the *System Administration Guide*

Standards conformance

mscreen is not part of any currently supported standard; it is an extension of AT&T System V provided by The Santa Cruz Operation, Inc.

1 May 1995