

Using the Korn Shell

Using the Korn Shell

This is an automatically-generated printer-friendly PDF version of the HTML web page at [Using the Korn Shell](#) ← CLICK HERE to get the standard HTML version of this document.

.profile file

.profile login script

To set up the shell automatically every time a user logs in, create a .profile file in the user's home directory. The commands in this file are executed when the shell starts up.

Testing changes to the .profile file

Changes to the .profile file can be tested by executing the script in the shell with the [.\(dot\) command](#):

```
./profile
```

Korn shell prompt

For most users, the default prompt in the Korn shell is a dollar sign:

```
$
```

For root, the prompt is a hash symbol:

```
#
```

Changing the Korn shell prompt

The prompt can be changed by setting the PS1 environment variable. For example:

```
export PS1='$PWD ! $ '
```

```
export PS1='${PWD#${PWD%/*/*}} ! $ '
```

```
export PS1="$LOGNAME@`uname -n` \ $PWD ! $ "
```

\$PWD shows the path for the current directory. \${PWD#\${PWD%/*/*}} uses Korn shell pattern matching to show only the last two directories in the current directory path. ! shows the number of the command in the command history.

When setting the prompt with a double-quoted ("...") string, unescaped variables, such as \$LOGNAME above, and commands, such as `uname -n` above, are interpreted when the environment variable is exported, and therefore will not change unless the prompt is set again.

Command Line Editing

Setting the Command Line Editor

In the Korn shell, use one of the following commands to set the option to be used for command line editing:

```
set -o emacs
```

```
set -o gmacs
```

```
set -o vi
```

The keys for specific commands will depend on which editor option is specified.

Command Line Editor Keys

Begin command line editing by entering the key sequence to recall a previous command from history. In vi mode, the initial command must be preceded by Esc, so command line editing is initiated with Esc k or Esc -. In emacs or gmacs mode, enter Ctrl-P to recall the previous command or Esc < to recall the first command from the history.

emacs / gmacs

Automatic file name completion

Esc Esc

Begin command line editing

Ctrl-P

Esc <

Previous line/command

Ctrl-P

Next line/command

Ctrl-N

First line/command in history

Esc <

Last line/command in history

Esc >

Search backward in history

Search forward in history

Repeat search in same direction

vi

Esc \

Esc k

Esc -

k

-

j

+

/string

?string

n

Using the Korn Shell

emacs / gmacs

vi

Repeat search in reverse direction		N
Execute and advance to next command in history	Ctrl-O	
Clear entire line	Ctrl-U	
Backwards one character	Ctrl-B	h Backspace
Forward one character	Ctrl-F	l
Backwards to first character of word	Esc b	b
Forward to first character of word		w
Forward to last character of word		e
Forward to end of word	Esc f	
Beginning of line	Ctrl-A	0 (zero)
First non-blank character in line		^
End of line	Ctrl-E	\$
Column <i>n</i>		<i>n</i> ("or" bar)
Find next character in line		<i>fchar</i>
Repeat find in same direction		;
Repeat find in reverse direction		,
Delete character at cursor	Ctrl-D	x
Delete character before cursor	Backspace	X
Delete <i>n</i> characters		<i>nx</i>
Delete <i>n</i> characters before cursor		<i>nX</i>
Delete word before cursor	Esc Backspace	Ctrl-W
Delete from cursor to end of line	Ctrl-K	D
Delete to end of word	Esc d	
Replace current character		r
Replace multiple character		R
Insert before current character		i
Insert after current character		a
Insert at beginning of line		I
Insert at end of line		A
Repeat most recent command		.
Undo last change		u
Undo all changes		U
Cancel emacs command (cancels Esc)	Ctrl-G	
Cancel UNIX command; new prompt	Ctrl-C	
Display version temporarily	Ctrl-V	
Transpose two characters	Ctrl-T	
emacs: transpose current character with following character		
gmacs: transpose previous two characters		

Command History

Turn on extended command history

To start recording the date and time that each command is executed, set the EXTENDED_HISTORY environment variable to ON:

```
export EXTENDED_HISTORY=ON
```

The CUT (Coordinated Universal Time) timestamp will be appended to each line of the ~/.sh_history file.

Display command history

Using the Korn Shell

The command history can be displayed with various history or fc (fix command) commands:

```
history
fc -l
history n
fc -l n
fc -t
history -t
fc -t n
history -t n
```

n specifies which command the listing is to start with. If it is negative, then the listing starts with the *n*th previous command; if it is non-negative, the listing starts with the specified command number. If *-t* is specified, the output will include the timestamps.

Repeat a previous command

r
without any parameters repeats the preceding command.

r searchtext
repeats the last command starting with the given search text.

r -n
repeats the *n*th previous command.

r n
repeats the specified command. The number of a command can be determined by displaying the command history or by [setting the prompt](#) to include the command number.

Setting the PATH Environment Variable

Adding directories to the PATH during login

Specific directories containing commands and/or scripts can be added to the PATH by appending the directories to the PATH in the [.profile file](#). For example, to automatically search a bin directory owned by the current user, add the following code to the `.profile` file in the user's home directory.

```
if [ ${OLDPATH:-0} = 0 ]; then
export OLDPATH=$PATH
fi
export PATH=$OLDPATH:~/bin
```

Special Built-in Commands

. (dot command)

Execute a command script

The dot command executes a script within the shell.

```
. filepath
```

The file does not need to have executable permissions.

export

Permanently set an environment variable

```
export var=value
export var='value ...'
export var="expression"
```

set

set

```
set -flag
set +flag
set -o option
set +o option
```

Using `-` turns on the flag or option; using `+` turns it off.

One of the most beneficial uses of this command is to set an option for [command line editing](#).

Regular Built-in Commands

Using the Korn Shell

cd Command

Change to specified directory

`cd dirpath`

Change to parent directory

`cd ..`

In addition to the functions of the [UNIX cd command](#) shown above, the Korn shell provides some additional functions:

Change to previously accessed directory

`cd -`

Change to similarly named directory

`cd string1 string2`

replaces "string1" in the current directory path with "string2" and changes to the resulting directory

echo Command

echo

`echo text ...`

print Command

print

`print`

This is an automatically-generated printer-friendly PDF version of the HTML web pages at [Using the Korn Shell](#) ← [CLICK HERE](#) to get the standard HTML version of this document.

