

**NAME**

wy60 – curses based emulator for the Wyse 60™ terminal

**SYNOPSIS**

```
wy60 [ -c | --command command ] [ -h | --help ] [ -j | --job-control { on | off } ] [ -l | --login ]
      [ -o | --option key=value ] [ -t | --term terminal ] [ -v | --version ] [ -- ] shell arguments
```

**DESCRIPTION**

The **wy60** utility program invokes a new shell or *command* and adjusts the **\$TERM** environment variable to have a value of *wyse60*. All commands executed within this shell or any of its child processes automatically have their terminal escape sequences translated to be processed by the terminal that **wy60** was invoked from.

Unless a different *command* was requested, the user's default shell as determined by the value of the **\$SHELL** environment variable is invoked. If the environment variable has not been set, then the default system shell */bin/sh* is used instead. This value can be overridden in the configuration file.

**OPTIONS**

Parameters are parsed until the first unknown parameter is encountered, or until the special parameter **--** is found. All the following parameters are passed on to the inferior shell or to the user specified *command*. Check the manual page for **sh**(1) for a detailed list of options.

The following parameters are directly understood by **wy60** and will not be forwarded to any child process unless they appear after the **--** marker or after another set of unknown parameters:

**-c | --command *command***

Rather than invoking the user's shell, run the specified *command*, instead.

**-h | --help**

Display a brief usage message showing the valid command line parameters.

**-j | --job-control { on | off }**

Enable or disable job control. If the parent process is a shell that supports job control then **wy60** can enable this capability for its children by forking an additional process monitoring the children. By default, this option is enabled.

**-l | --login**

Invoke the inferior shell as a login shell. This is done by prepending a minus character to the application name. This convention is supported by all shells but probably gets ignored by most other applications. The **--login** parameter is ignored if the **--command** parameter has also been given.

As a special case, it is possible to make **wy60** a user's default login shell. To do so, **wy60** must be entered into the **shells**(5) file and you must call **chsh**(1) to change the user's default shell. If the emulator detects that it is invoked as a wrapper for a shell, it ignores all command line parameters. It then resets **\$SHELL** to the value specified in the configuration file or to */bin/sh* and launches that shell passing all the parameters that were given to **wy60**.

**-o | --option *key=value***

All of the configuration variables listed later in this document can also be overridden on the command line. The syntax is identical to the one used in the configuration files.

When using this option, it might be necessary to quote the key/value pair to prevent the shell from expanding escape sequences.

**-t | --term *terminal***

Override the value of the **\$TERM** environment variable. If this parameter is not present, then a default of *wyse60* will be used for the value that is passed to the inferior shell.

**-v | --version**

Display the version number of the program and the date when it was compiled.

-- All parameters following this argument will be literally passed to the inferior shell.

## EXAMPLES

### **wy60**

If invoked without parameters, an interactive shell is invoked.

### **wy60 -c emacs -nw README**

The **-c** parameter can be used to directly invoke an interactive application that should run in an emulated environment. The emulator will terminate when the invoked application has quit.

### **wy60 -c infocmp**

use **infocmp**(1M) to verify that the **terminfo**(5) database has appropriate entries for the emulated *wyse60* terminal.

## CONFIGURATION FILES

Some of the execution parameters can be controlled through configuration files. Every user can have a private configuration file in *\$HOME/.wy60rc* that overrides any system-wide settings. The system administrator can also set global parameters in */usr/local/etc/wy60.rc* or in */etc/wy60.rc*.

All configuration files are in plain ASCII format. Comments are introduced by the hash mark **#** character and extend to the end of the line. Blank lines are ignored. All lines are of the form **variable = value**. Variable names are case insensitive. Leading and trailing white space for both the variables and the values is stripped prior to processing. Non ASCII characters have to be escaped:

**\ooo** Up to three octal digits can be used to specify a character in the range **\001** through **\177**.

**\xXX** Up to two hexadecimal digits can be used to specify a character in the range **\x01** through **\xFF**.

**\a** The **BEL** character (ASCII 7).

**\b** The **BS** character (ASCII 8).

**\e** The **ESC** character (ASCII 27).

**\f** The **FF** character (ASCII 12).

**\n** The **NL** character (ASCII 10).

**\r** The **CR** character (ASCII 13).

**\t** The **TAB** character (ASCII 9).

**\v** The **VT** character (ASCII 11).

Please note that continuation lines are not supported and that is not possible to input a **NUL** character (ASCII 0).

The configuration file supports the following parameters:

### **IDENTIFIER**

The terminal identifier string that is reported when an *ENQ* (ASCII 5) is received, can be overridden by setting the **IDENTIFIER** configuration variable. If this option is not set, then the emulator replies with an *ACK* (ASCII 6).

### **PRINTCOMMAND**

Programs can print to a local printer by sending escape codes to **wy60**. If this variable is set to "auto" then **wy60** will try to print to either **lp**(1) or to **lpr**(1). Otherwise, this variable should contain the name of a script that can accept data on its standard input.

**RESIZE** If you want to use an external script to resize the console, then you can specify the absolute path to this script by setting the **RESIZE** configuration variable. For example, if you have **SVGATextMode**(8) installed and you want to run **wy60** from the Linux text console, then you might try using a script such as this one:

```
#!/bin/sh
PATH=/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin
```

```

egrep '.+' /proc/fb >&2 && exit 0
(tty <&3 | egrep '^/dev/tty[0-9]*$' >&2) 3<&1
if test "$?" -eq 0; then
    SVGATextMode "$1x$2" <&1 >&2 ||
    SVGATextMode "$1x$2x9" <&1 >&2 ||
    SVGATextMode "$1x$2x8" <&1 >&2
fi
exit 0

```

Make sure that **SVGATextMode**(8) understands these modes: 80x24, 80x25, 80x42, 80x43, 132x24, 132x25, 132x42, 132x43.

**SHELL** If neither a *command* has been passed on the command line, nor the user's **\$SHELL** environment variable was set, then use this variable to determine the default shell. If the variable is not set, then a value of */bin/sh* is used instead.

**TERM** If no *terminal* type has been specified on the command line, the program defaults to *wyse60*. This default value can be overridden by setting the **TERM** configuration variable.

#### WRITEPROTECT

The default attribute for indicating write-protect mode is *REVERSE*. This value can be overridden by setting the **WRITEPROTECT** configuration variable to one of these values: *NORMAL*, *BLANK*, *BLINK*, *REVERSE*, *UNDERSCORE*, *DIM*. The attributes can be combined with each other by listing multiple in the same line.

All the other variables specify character sequences that should be sent whenever the user presses a function key. The following table lists the variable name, the default value, and the **terminfo**(5) name for the function key.

Variable Name	Default Key Sequence	Terminfo Name
A1		ka1
A3		ka3
B2		kb2
C1		kc1
C3		kc2
Backspace	\b	kbs
Backtab	\eI	kcbt
Begin		kbeg
Cancel		kcan
Clear		kclr
Clear All Tabs		ktbc
Clear Tab		kctab
Close		kclo
Command		kcmd
Copy		kcpy
Create		kcrt
Delete	\eW	kdch1
Delete Line	\eR	kd11
Down	\n	kcud1
End	\eT	kend
End Of Line	\eY	kel
End Of Screen	\eT	ked
Enter	\e7	kent
Exit		kext
Exit Insert Mode		krmir
Find		kfnd
Help		khlp

Home	\x1E	khome
Insert	\eE	kich1
Insert Line	\eQ	kill
Left	\b	kcub1
Lower Left		kll
Mark		kmrk
Message		kmsg
Move		kmov
Next	\eK	knxt
Open		kopn
Options		kopt
Page Down	\eK	knp
Page Up	\eJ	kpp
Previous	\eJ	kprv
Print	\eP	kprt
Redo		krdo
Reference		kref
Refresh		krfr
Replace	\er	krpl
Restart		krst
Resume		kres
Right	\f	kcuf1
Save		ksav
Scroll Down		kind
Scroll Up		kri
Select		kslt
Set Tab		khts
Suspend	\x1A	kspd
Undo		kund
Up	\v	kcuu1
Shift Begin		kBEG
Shift Cancel		kCAN
Shift Command		kCMD
Shift Copy		kCPY
Shift Create		kCRT
Shift Delete	\eW	kDC
Shift Delete Line	\eR	kDL
Shift End	\eT	kEND
Shift End Of Line	\eT	kEOL
Shift Exit		kEXT
Shift Find		kFND
Shift Help		kHLP
Shift Home	\e{	kHOM
Shift Insert	\eQ	kIC
Shift Left	\b	kLFT
Shift Message		kMSG
Shift Move		kMOV
Shift Next	\eK	kNXT
Shift Options		kOPT
Shift Previous	\eJ	kPRV
Shift Print	\eP	kPRT
Shift Redo		krdo
Shift Replace	\er	krpl

Shift Resume		kRES
Shift Right	\f	kRIT
Shift Save		kSAV
Shift Suspend		kSPD
Shift Undo		kUND
F0		kf0
F1	\001@\r	kf1
F2	\001A\r	kf2
F3	\001B\r	kf3
F4	\001C\r	kf4
F5	\001D\r	kf5
F6	\001E\r	kf6
F7	\001F\r	kf7
F8	\001G\r	kf8
F9	\001H\r	kf9
F10	\001I\r	kf10
F11	\001J\r	kf11
F12	\001K\r	kf12
F13	\001 \r	kf13
F14	\001a\r	kf14
F15	\001b\r	kf15
F16	\001c\r	kf16
F17	\001d\r	kf17
F18	\001e\r	kf18
F19	\001f\r	kf19
F20	\001g\r	kf20
F21	\001h\r	kf21
F22	\001i\r	kf22
F23	\001j\r	kf23
F24	\001k\r	kf24
F25	\001L\r	kf25
F26	\001M\r	kf26
F27	\001N\r	kf27
F28	\001O\r	kf28
F29	\001I\r	kf29
F30	\001m\r	kf30
F31	\001n\r	kf31
F32	\001o\r	kf32
F33		kf33
F34		kf34
F35		kf35
F36		kf36
F37		kf37
F38		kf38
F39		kf39
F40		kf40
F41		kf41
F42		kf42
F43		kf43
F44		kf44
F45		kf45
F46		kf46
F47		kf47

F48		kf48
F49		kf49
F50		kf50
F51		kf51
F52		kf52
F53		kf53
F54		kf54
F55		kf55
F56		kf56
F57		kf57
F58		kf58
F59		kf59
F60		kf60
F61		kf61
F62		kf62
F63		kf63
Alt a	\ea	
Alt b	\eb	
Alt c	\ec	
Alt d	\ed	
Alt e	\ee	
Alt f	\ef	
Alt g	\eg	
Alt h	\eh	
Alt i	\ei	
Alt j	\ej	
Alt k	\ek	
Alt l	\el	
Alt m	\em	
Alt n	\en	
Alt o	\eo	
Alt p	\ep	
Alt q	\eq	
Alt r	\er	
Alt s	\es	
Alt t	\et	
Alt u	\eu	
Alt v	\ev	
Alt w	\ew	
Alt x	\ex	
Alt y	\ey	
Alt z	\ez	
Alt A	\eA	
Alt B	\eB	
Alt C	\eC	
Alt D	\eD	
Alt E	\eE	
Alt F	\eF	
Alt G	\eG	
Alt H	\eH	
Alt I	\eI	
Alt J	\eJ	
Alt K	\eK	

Alt L	eL
Alt M	eM
Alt N	eN
Alt O	eO
Alt P	eP
Alt Q	eQ
Alt R	eR
Alt S	eS
Alt T	eT
Alt U	eU
Alt V	eV
Alt W	eW
Alt X	eX
Alt Y	eY
Alt Z	eZ
Alt 0	e0
Alt 1	e1
Alt 2	e2
Alt 3	e3
Alt 4	e4
Alt 5	e5
Alt 6	e6
Alt 7	e7
Alt 8	e8
Alt 9	e9
Alt Space	e x20
Alt Exclamation	e!
Alt Double Quote	e"
Alt Pound	e#
Alt Dollar	e\$
Alt Percent	e%
Alt Ampersand	e&
Alt Single Quote	e'
Alt Left Paren	e(
Alt Right Paren	e)
Alt Asterisk	e*
Alt Plus	e+
Alt Comma	e,
Alt Dash	e-
Alt Period	e.
Alt Slash	e/
Alt Colon	e:
Alt Semicolon	e;
Alt Less	e<
Alt Equals	e=
Alt Greater	e>
Alt Question	e?
Alt At	e@
Alt Left Bracket	e[
Alt Backslash	e\
Alt Right Bracket	e]
Alt Circumflex	e^
Alt Underscore	e_

Alt Backtick	e`
Alt Left Brace	e{
Alt Pipe	e
Alt Right Brace	e}
Alt Tilde	e~
Alt Backspace	e\x7F

Not all of these keys are necessarily available on all types of terminals. This is particularly true for the keys with the **Alt** modifier, which assume that the terminal precedes all characters with an **ESC** character if **Alt** is pressed.

## EXIT STATUS

- 0** Successful program execution.
- 125** The inferior shell or *command* terminated abnormally.
- 126** The emulator has been terminated by an unexpected signal.
- 127** Some fatal error prevented or aborted the execution of **wy60**.
- Other* The inferior shell or *command* returned a non-zero exit status.

## DIAGNOSTICS

The most common cause for the emulator to fail is the **\$TERM** environment variable not being set or the terminal having insufficient capabilities. If this happens, try using a more powerful terminal type and also use **infocmp(1M)** to verify that your **terminfo(5)** database is configured correctly.

You might also see an error message saying that the terminal size could not be determined. **wy60** requires that the terminal settings have been configured properly prior to invoking the emulator. This usually happens at the time when you log into the system, but you can also adjust the values by calling **stty(1)**.

If the emulator complains that it cannot launch the child process then verify that **\$SHELL** is set and points to a valid shell.

## FILES

- \$HOME/.wy60rc* Per-user configuration settings.
- /usr/local/etc/wy60.rc* System-wide configuration settings.
- /etc/wy60.rc* System-wide configuration settings (this file must be created manually and does not get overwritten when upgrading **wy60**).
- /usr/share/terminfo/?/\** Files containing terminal descriptions.

## ENVIRONMENT

The **\$TERM** environment variable must be set for the emulator to work. If the **\$SHELL** environment variable is set, its value will be used for launching the inferior shell. Otherwise a default filename is taken from the configuration files or */bin/sh* is used.

The emulator sets the **\$TERM**, **\$COLUMNS**, and **\$LINES** environment variables for its child processes.

## SEE ALSO

**chsh(1)**, **infocmp(1M)**, **lp(1)**, **lpr(1)**, **sh(1)**, **shells(5)**, **stty(1)**, **SVGATextMode(8)**, **terminfo(5)**, **tput(1)**, **xterm(1)**.

## SECURITY

The program can be executed with normal user privileges. As such, the amount of damage that a malicious application could cause is limited to the user's account. To the best of the author's knowledge the current version of **wy60** does not have any security issues, but common sense should still be used whenever running any untrusted software within the emulator.

## AUTHOR

Copyright (C) 2001, 2002 by Markus Gutschke <markus+wy60@wy60.gutschke.com>.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General

Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

## MAILING LIST

There is a low-traffic mailing list for the discussion of this program at <http://gutschke.com/cgi-bin/mailman/listinfo/Wy60>.

## TRADEMARKS

*Wyse* and *Wyse 60* are trademarks or registered trademarks of *Wyse Technology, Inc.*

## BUGS

The emulator attempts to seamlessly switch to *wyse60* emulation. This requires the ability to query the host terminal for its cursor position at the time when the emulator is started. As **terminfo(5)** does not provide this capability, **wy60** can only support seamless switching on a small number of well known terminal types. If it does not recognize the terminal type, then the screen will be cleared at startup.

The escape sequences for setting the screen size can only be processed if the host terminal supports dynamic resizing under program control. Currently, **xterm(1)** appears to be the only program that has this capability.

Function keys work only if the host terminal generates distinct character sequences for each of them and if the **terminfo(5)** entry for the host terminal is correct. Frequently one or the other is not the case which limits the usability of the emulator. Testing has shown, that the most correct implementation appears to be available in very recent versions of **xterm(1)**.

The current version of the emulator supports only a small subset of the full features offered by a real *Wyse 60™* terminal. Several of the advanced features are missing. Most notably this means that there is no support for reprogrammable function keys (but you can use the configuration file to assign strings to individual keys) or for the status bar. Frequently, the supported set of escape sequences is sufficient to run an application within the emulated environment, but if that does not work, you should consider either filing a detailed bug report with the author or you should switch to using a hardware terminal.