Recently I needed to choose lots of different colors for a project. I'm not on Unix so I can't run xcolors. Also, I didn't find [color values available for syntactic use in Tcl scripts](http://wiki.tcl.tk/3538) with Jeff Hobb's utility until later.

Be that as it may, I went and wrote a little utility to display all of the colors that Tcl/Tk knows the name for. Moving your mouse over a color will display its name, RGB value and its numeric name. Double clicking will copy the color name to the clipboard.

It will also display the Windows-specific colors like *SystemButtonFace* and *SystemHightlightText*.

[LV](http://wiki.tcl.tk/472) 2003-03-25: When I saw this page title, I thought the page was going to be about the idea of creating the equivalent of named fonts - in this case, the ability for a user to associate a symbolic name to a color setting, use the symbolic name through the application, then change the colors of all items using the symbolic name via one command.

[KPV](http://wiki.tcl.tk/3500) responds: See the original wiki page for this [Named Colors](http://wiki.tcl.tk/8606).

[LV](http://wiki.tcl.tk/472) continues: My second thought, having read the page, is how sad it is that Tk doesn't provide these names via some command.

It's also possible to go the other way programmatically. That's the job of [closest color name](http://wiki.tcl.tk/9221).

[KPV](http://wiki.tcl.tk/3500) 2006-07-20: fixed a reformatting bug that allowed newlines into some color names. Also leveraged ::tk::Darken for prettier #xxyyzz display.

[Zipguy](http://wiki.tcl.tk/13606) 2013-02-09: I did copy, paste into a new file in Context (a nice editor for some purposes), commented out all these lines, added the two lines to the top, saved it as colornames.tcl (in the proper directory, re-opened it in [ML - Heavily Modified & Improved](http://wiki.tcl.tk/13591), and it looked good. So I hit the gear in ML (to save and run it), and it did work. It looked like this:

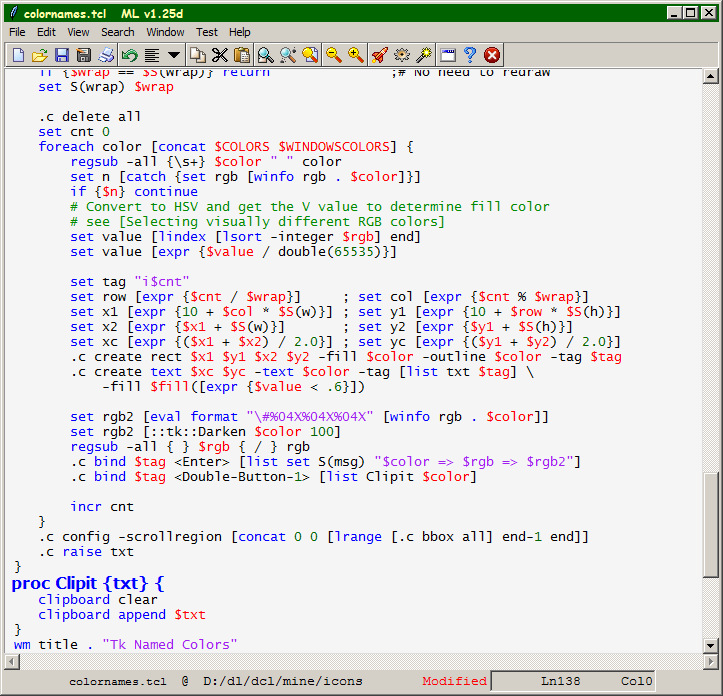


which looked wonderful! You can see the rest of the screens at [Color Names, running, all screens](http://wiki.tcl.tk/37701).

You did try real hard to hide that little, tiny button but I did find it:



So, what I plan on doing is to make a slightly modified version that will also display a second version of the colors, sorted by name, and add a button somewhere to toggle them back and forth (Sorted vs By Color). This is what the code looks like in ML (colorized for syntax):



You don't see the "Procedures" window, which is now separate, but joined with the main window a la [How do you make two windows act as one?](http://wiki.tcl.tk/37535). There, you can see the "Procedure" window slightly obscured (but you can see the first eight characters of it). Anyway, what I'm doing is trying to understand all of the code, first.

##+#################################################################

#

# Named Colors -- lists and displays all of tcl's named colors

# by Keith Vetter, March 19 2003

[package](http://wiki.tcl.tk/package) require Tk

[set](http://wiki.tcl.tk/set) COLORS { snow {ghost white} {white smoke} gainsboro {floral white}

{old lace} linen {antique white} {papaya whip} {blanched almond}

bisque {peach puff} {navajo white} moccasin cornsilk ivory

{lemon chiffon} seashell honeydew {mint cream} azure {alice blue}

lavender {lavender blush} {misty rose} white black {dark slate gray}

{dim gray} {slate gray} {light slate gray} gray {light grey}

{midnight blue} navy {cornflower blue} {dark slate blue} {slate blue}

{medium slate blue} {light slate blue} {medium blue} {royal blue}

blue {dodger blue} {deep sky blue} {sky blue} {light sky blue}

{steel blue} {light steel blue} {light blue} {powder blue}

{pale turquoise} {dark turquoise} {medium turquoise} turquoise

cyan {light cyan} {cadet blue} {medium aquamarine} aquamarine

{dark green} {dark olive green} {dark sea green} {sea green}

{medium sea green} {light sea green} {pale green} {spring green}

{lawn green} green chartreuse {medium spring green} {green yellow}

{lime green} {yellow green} {forest green} {olive drab} {dark khaki}

khaki {pale goldenrod} {light goldenrod yellow} {light yellow} yellow

gold {light goldenrod} goldenrod {dark goldenrod} {rosy brown}

{indian red} {saddle brown} sienna peru burlywood beige wheat

{sandy brown} tan chocolate firebrick brown {dark salmon} salmon

{light salmon} orange {dark orange} coral {light coral} tomato

{orange red} red {hot pink} {deep pink} pink {light pink}

{pale violet red} maroon {medium violet red} {violet red}

magenta violet plum orchid {medium orchid} {dark orchid} {dark violet}

{blue violet} purple {medium purple} thistle snow2 snow3

snow4 seashell2 seashell3 seashell4 AntiqueWhite1 AntiqueWhite2

AntiqueWhite3 AntiqueWhite4 bisque2 bisque3 bisque4 PeachPuff2

PeachPuff3 PeachPuff4 NavajoWhite2 NavajoWhite3 NavajoWhite4

LemonChiffon2 LemonChiffon3 LemonChiffon4 cornsilk2 cornsilk3

cornsilk4 ivory2 ivory3 ivory4 honeydew2 honeydew3 honeydew4

LavenderBlush2 LavenderBlush3 LavenderBlush4 MistyRose2 MistyRose3

MistyRose4 azure2 azure3 azure4 SlateBlue1 SlateBlue2 SlateBlue3

SlateBlue4 RoyalBlue1 RoyalBlue2 RoyalBlue3 RoyalBlue4 blue2 blue4

DodgerBlue2 DodgerBlue3 DodgerBlue4 SteelBlue1 SteelBlue2

SteelBlue3 SteelBlue4 DeepSkyBlue2 DeepSkyBlue3 DeepSkyBlue4

SkyBlue1 SkyBlue2 SkyBlue3 SkyBlue4 LightSkyBlue1 LightSkyBlue2

LightSkyBlue3 LightSkyBlue4 SlateGray1 SlateGray2 SlateGray3

SlateGray4 LightSteelBlue1 LightSteelBlue2 LightSteelBlue3

LightSteelBlue4 LightBlue1 LightBlue2 LightBlue3 LightBlue4

LightCyan2 LightCyan3 LightCyan4 PaleTurquoise1 PaleTurquoise2

PaleTurquoise3 PaleTurquoise4 CadetBlue1 CadetBlue2 CadetBlue3

CadetBlue4 turquoise1 turquoise2 turquoise3 turquoise4 cyan2 cyan3

cyan4 DarkSlateGray1 DarkSlateGray2 DarkSlateGray3 DarkSlateGray4

aquamarine2 aquamarine4 DarkSeaGreen1 DarkSeaGreen2 DarkSeaGreen3

DarkSeaGreen4 SeaGreen1 SeaGreen2 SeaGreen3 PaleGreen1 PaleGreen2

PaleGreen3 PaleGreen4 SpringGreen2 SpringGreen3 SpringGreen4

green2 green3 green4 chartreuse2 chartreuse3 chartreuse4

OliveDrab1 OliveDrab2 OliveDrab4 DarkOliveGreen1 DarkOliveGreen2

DarkOliveGreen3 DarkOliveGreen4 khaki1 khaki2 khaki3 khaki4

LightGoldenrod1 LightGoldenrod2 LightGoldenrod3 LightGoldenrod4

LightYellow2 LightYellow3 LightYellow4 yellow2 yellow3 yellow4

gold2 gold3 gold4 goldenrod1 goldenrod2 goldenrod3 goldenrod4

DarkGoldenrod1 DarkGoldenrod2 DarkGoldenrod3 DarkGoldenrod4

RosyBrown1 RosyBrown2 RosyBrown3 RosyBrown4 IndianRed1 IndianRed2

IndianRed3 IndianRed4 sienna1 sienna2 sienna3 sienna4 burlywood1

burlywood2 burlywood3 burlywood4 wheat1 wheat2 wheat3 wheat4 tan1

tan2 tan4 chocolate1 chocolate2 chocolate3 firebrick1 firebrick2

firebrick3 firebrick4 brown1 brown2 brown3 brown4 salmon1 salmon2

salmon3 salmon4 LightSalmon2 LightSalmon3 LightSalmon4 orange2

orange3 orange4 DarkOrange1 DarkOrange2 DarkOrange3 DarkOrange4

coral1 coral2 coral3 coral4 tomato2 tomato3 tomato4 OrangeRed2

OrangeRed3 OrangeRed4 red2 red3 red4 DeepPink2 DeepPink3 DeepPink4

HotPink1 HotPink2 HotPink3 HotPink4 pink1 pink2 pink3 pink4

LightPink1 LightPink2 LightPink3 LightPink4 PaleVioletRed1

PaleVioletRed2 PaleVioletRed3 PaleVioletRed4 maroon1 maroon2

maroon3 maroon4 VioletRed1 VioletRed2 VioletRed3 VioletRed4

magenta2 magenta3 magenta4 orchid1 orchid2 orchid3 orchid4 plum1

plum2 plum3 plum4 MediumOrchid1 MediumOrchid2 MediumOrchid3

MediumOrchid4 DarkOrchid1 DarkOrchid2 DarkOrchid3 DarkOrchid4

purple1 purple2 purple3 purple4 MediumPurple1 MediumPurple2

MediumPurple3 MediumPurple4 thistle1 thistle2 thistle3 thistle4

gray1 gray2 gray3 gray4 gray5 gray6 gray7 gray8 gray9 gray10

gray11 gray12 gray13 gray14 gray15 gray16 gray17 gray18 gray19

gray20 gray21 gray22 gray23 gray24 gray25 gray26 gray27 gray28

gray29 gray30 gray31 gray32 gray33 gray34 gray35 gray36 gray37

gray38 gray39 gray40 gray42 gray43 gray44 gray45 gray46 gray47

gray48 gray49 gray50 gray51 gray52 gray53 gray54 gray55 gray56

gray57 gray58 gray59 gray60 gray61 gray62 gray63 gray64 gray65

gray66 gray67 gray68 gray69 gray70 gray71 gray72 gray73 gray74

gray75 gray76 gray77 gray78 gray79 gray80 gray81 gray82 gray83

gray84 gray85 gray86 gray87 gray88 gray89 gray90 gray91 gray92

gray93 gray94 gray95 gray97 gray98 gray99

}

[set](http://wiki.tcl.tk/set) WINDOWSCOLORS {

SystemButtonFace SystemButtonText SystemDisabledText SystemHighlight

SystemHightlightText SystemMenu SystemMenuText SystemScrollbar

SystemWindow SystemWindowFrame SystemWindowText

}

[set](http://wiki.tcl.tk/set) S(w) 130

[set](http://wiki.tcl.tk/set) S(h) 20

[set](http://wiki.tcl.tk/set) S(wrap) -1

[proc](http://wiki.tcl.tk/proc) DrawColors {} {

[global](http://wiki.tcl.tk/global) S COLORS WINDOWSCOLORS

array [set](http://wiki.tcl.tk/set) fill {0 black 1 white}

[set](http://wiki.tcl.tk/set) cwidth [winfo width .c]

[set](http://wiki.tcl.tk/set) wrap [[expr](http://wiki.tcl.tk/expr) {($cwidth - 20) / $S(w)}]

[if](http://wiki.tcl.tk/if) {$wrap <= 0} {[set](http://wiki.tcl.tk/set) wrap 1}

[if](http://wiki.tcl.tk/if) {$wrap == $S(wrap)} [return](http://wiki.tcl.tk/return) ;# No need to redraw

[set](http://wiki.tcl.tk/set) S(wrap) $wrap

.c delete all

[set](http://wiki.tcl.tk/set) cnt 0

[foreach](http://wiki.tcl.tk/foreach) color [[concat](http://wiki.tcl.tk/concat) $COLORS $WINDOWSCOLORS] {

[regsub](http://wiki.tcl.tk/regsub) -all {\s+} $color " " color

[set](http://wiki.tcl.tk/set) n [[catch](http://wiki.tcl.tk/catch) {[set](http://wiki.tcl.tk/set) rgb [winfo rgb . $color]}]

[if](http://wiki.tcl.tk/if) {$n} [continue](http://wiki.tcl.tk/continue)

# Convert to HSV and get the V value to determine fill color

# see [Selecting visually different RGB colors]

[set](http://wiki.tcl.tk/set) value [[lindex](http://wiki.tcl.tk/lindex) [[lsort](http://wiki.tcl.tk/lsort) -integer $rgb] end]

[set](http://wiki.tcl.tk/set) value [[expr](http://wiki.tcl.tk/expr) {$value / double(65535)}]

[set](http://wiki.tcl.tk/set) tag "i$cnt"

[set](http://wiki.tcl.tk/set) row [[expr](http://wiki.tcl.tk/expr) {$cnt / $wrap}] ; [set](http://wiki.tcl.tk/set) col [[expr](http://wiki.tcl.tk/expr) {$cnt % $wrap}]

[set](http://wiki.tcl.tk/set) x1 [[expr](http://wiki.tcl.tk/expr) {10 + $col \* $S(w)}] ; [set](http://wiki.tcl.tk/set) y1 [[expr](http://wiki.tcl.tk/expr) {10 + $row \* $S(h)}]

[set](http://wiki.tcl.tk/set) x2 [[expr](http://wiki.tcl.tk/expr) {$x1 + $S(w)}] ; [set](http://wiki.tcl.tk/set) y2 [[expr](http://wiki.tcl.tk/expr) {$y1 + $S(h)}]

[set](http://wiki.tcl.tk/set) xc [[expr](http://wiki.tcl.tk/expr) {($x1 + $x2) / 2.0}] ; [set](http://wiki.tcl.tk/set) yc [[expr](http://wiki.tcl.tk/expr) {($y1 + $y2) / 2.0}]

.c create rect $x1 $y1 $x2 $y2 -fill $color -outline $color -tag $tag

.c create text $xc $yc -text $color -tag [[list](http://wiki.tcl.tk/list) txt $tag] \

-fill $fill([[expr](http://wiki.tcl.tk/expr) {$value < .6}])

[set](http://wiki.tcl.tk/set) rgb2 [[eval](http://wiki.tcl.tk/eval) [format](http://wiki.tcl.tk/format) "\#%04X%04X%04X" [winfo rgb . $color]]

[set](http://wiki.tcl.tk/set) rgb2 [::tk::Darken $color 100]

[regsub](http://wiki.tcl.tk/regsub) -all { } $rgb { / } rgb

.c bind $tag <Enter> [[list](http://wiki.tcl.tk/list) [set](http://wiki.tcl.tk/set) S(msg) "$color => $rgb => $rgb2"]

.c bind $tag <Double-Button-1> [[list](http://wiki.tcl.tk/list) Clipit $color]

[incr](http://wiki.tcl.tk/incr) cnt

}

.c config -scrollregion [[concat](http://wiki.tcl.tk/concat) 0 0 [[lrange](http://wiki.tcl.tk/lrange) [.c bbox all] end-1 end]]

.c raise txt

}

[proc](http://wiki.tcl.tk/proc) Clipit {txt} {

clipboard clear

clipboard [append](http://wiki.tcl.tk/append) $txt

}

wm title . "Tk Named Colors"

canvas .c -width 670 -height 600 -yscrollcommand {.sby [set](http://wiki.tcl.tk/set)} -highlightthick 0

scrollbar .sby -orient v -command {.c yview}

label .msg -textvariable S(msg) -relief sunken -bg white

image create photo ::img::blank -width 1 -height 1

button .about -image ::img::blank -highlightthickness 0 -command \

{tk\_messageBox -message "Tk Named Colors\nby Keith Vetter, March 2003"}

place .about -in .msg -relx 1 -rely 1 -anchor se

bind Canvas <Button-2> [bind Text <Button-2>]

bind Canvas <B2-Motion> [bind Text <B2-Motion>]

#bind Canvas <MouseWheel> [bind Text <MouseWheel>]

bind Canvas <MouseWheel> {%W yview scroll [[expr](http://wiki.tcl.tk/expr) {%D > 0 ? -2 : 2}] units}

bind .c <Alt-c> [[list](http://wiki.tcl.tk/list) console show]

focus .c

grid .c .sby -row 0 -sticky news

grid .msg - -sticky ew

grid rowconfigure . 0 -weight 1

grid columnconfigure . 0 -weight 1

bind .c <Configure> DrawColors

[return](http://wiki.tcl.tk/return)

[[](http://wiki.tcl.tk/40372)Grandclosing[]](http://wiki.tcl.tk/40372) - 2014-07-16 04:20:37:

This is a really great program and very useful - I use it as a reference when creating my own scripts! However there is an issue with the scrollbar keybinding. Whenever I use the mousewheel to scroll, I get the following error message:

bad argument "pixels": must be units or pages

bad argument "pixels": must be units or pages

[while](http://wiki.tcl.tk/while) executing

".c yview scroll [expr {(2--120)/3}] pixels"

invoked from within

"if {-120 >= 0} {

.c yview scroll [expr {--120/3}] pixels

} else {

.c yview scroll [expr {(2--120)/3}] pixels

}"

(command bound to event)

A fix for this would be really great.

[MG](http://wiki.tcl.tk/6939): has put a different mousewheel binding in, which works for him on Win7.