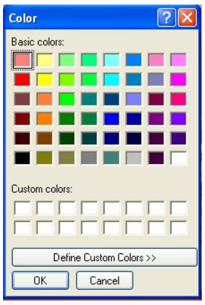
ColorDialog

-
Alyce

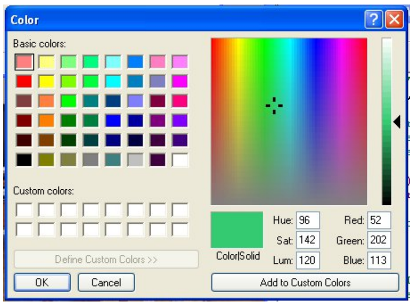
ColorDialog | What is a ColorDialog? | Syntax | RGB Range | Contents of the Returned String | ColorDialog Usage | Parsing the Returned String with Word$() | Demo

What is a ColorDialog?

The ColorDialog is a Windows dialog that allows users to select an RGB color from a visual display. Liberty BASIC's ColorDialog command opens the simple form, which looks like this:



When the user clicks the button to "define custom colors" the ColorDialog window opens another pane and looks like this:



Hue: 96
Sat: 142
Lum: 120
Red: 52
Green: 202
Blue: 113

Syntax

The syntax forthe ColorDialog command is as follows.

```
colordialog color$, chosen$
```

The parameter color$ can be an empty string, but it must be included. The color chosen by the user is contained in the variable "chosen$" after the dialog is closed. The input parameter may be in one of two forms. It can be a named Liberty BASIC color, or a string containing the red, green, blue values of the desired color with which to seed the colordialog. Remember that this is a string, and that the values are NOT separated by commas.

```
'named Liberty BASIC color as input string:
color$="red"
or
'RGB color as input string:
color$="255 0 0"
colordialog color$, chosen$
```

The color specification can be placed as a literal string into the colordialog:

```
colordialog "red", chosen$
or
colordialog "255 0 0", chosen$
```

RGB Range

Red, green and blue values must each be in the range of 0 to 255. 0 is the absence of a color, and 255 is total saturation. The RGB for red is 255 0 0, while blue is 0 0 255. Black has an RGB of 0 0 0, and white is 255 255 255. Any RGB where the values for red, green and blue are equal will be a shade of gray. Example: RGB 127 127 127.

Contents of the Returned String

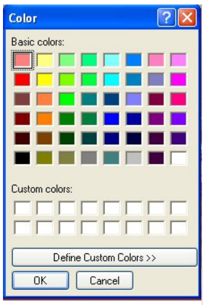
If the user chooses a named, Liberty BASIC color, then chosen$ will contain both the red, green, blue values and the name. If the chosen color is not a named color, then the return will contain only the red, green and blue values:

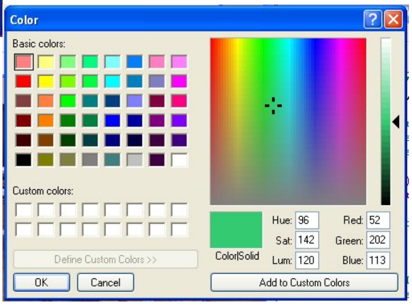
```
color$="255 0 0"
colordialog color$, chosen$
print chosen$

'will print RGB and name for a named color
'255 255 0 yellow
```

```
'will print RGB only for a non-named color:
'250 230 190
```

ColorDialog Usage

The colordialog opens to the simple view, allowing the user to choose a color from one of the standard color boxes. The user can click the button "Define Custom Colors" to open the panel that provides access to all possible RGB combinations.





Hue: 96
Sat: 142
Lum: 120
Red: 52
Green: 202
Blue: 113

Parsing the Returned String with Word$()

Here is a little demo to help you parse the returned string. The first word$() in the string will be the red value. The second word$() will be the green value and the third word$() will be the blue value. If there is a fourth word$(), it will be the Liberty BASIC named color.

```
color$="52 202 113"
colordialog color$, chosen$

print "Red is ";word$(chosen$,1)
print "Green is ";word$(chosen$,2)
print "Blue is ";word$(chosen$,3)

if word$(chosen$,4)<>"" then
    print "Name is ";word$(chosen$,4)
end if
```

If you are going to fill a graphicbox with the color chosen by the user, then check for the fourth word$(). If it exists, then the command will look like this:

```
print #1.gbox, "fill ";word$(chosen$,4)
```

If the user didn't choose a named color, then make a color out of the red, green and blue.

```
color$=word$(chosen$,1)+" "+word$(chosen$,2)+" "+word$(chosen$,3)
print #1.gbox, "fill ";color$
```

Demo

The following demo program allows the user to choose a color with the ColorDialog. The returned string is parsed. If it contains a fourth word, that is a named color and it is used. If there is not a fourth word, the first three words are used to build an RGB color, and that is used. A graphics window is then filled with the selected color and the color value is printed in the window.

```
color$="52 202 113"
colordialog color$, chosen$

red$ = word$(chosen$,1)
green$ = word$(chosen$,2)
blue$ = word$(chosen$,3)

if word$(chosen$,4)<>"" then
    selColor$ = word$(chosen$,4)
    else
    selColor$ = red$;" ";green$;" ";blue$
end if

nomainwin
open "Colordialog Demo" for graphics_nsb as #g
    #g "down; place 20 100; color black"
    #g "fill ";selColor$
    #g "backcolor ";selColor$
    #g "\Selected color is ";selColor$
    #g "flush"
    #g "trapclose [quit]"
    wait
[quit] close #g:end
```

ColorDialog | What is a ColorDialog? | Syntax | RGB Range | Contents of the Returned String | ColorDialog Usage | Parsing the Returned String with Word$() | Demo