



[All Topics](#), [MFC / C++](#) >> [Miscellaneous Controls](#) >> [Colour Selection Controls](#)
http://www.codeproject.com/miscctrl/colour_picker.asp

VC4, VC5, VC6, Win9x,
WinNT

Posted 20 Nov 1999

34,934 views

Office 97 style Colour Picker control

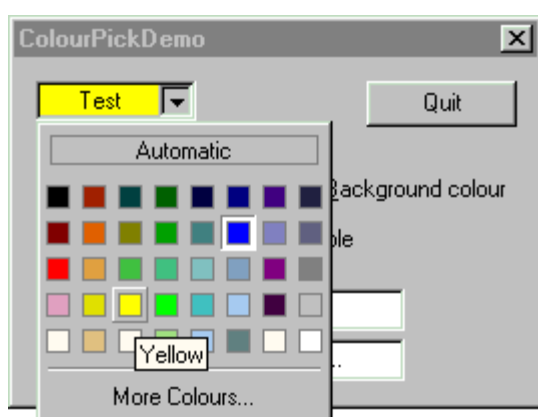
By [Chris Maunder](#)

A simple drop in color chooser control

30 members have rated this article. Result:

Popularity: 7.12. Rating: 4.82 out of 5.

[Download demo project - 41 Kb](#)
 [Download source - 13 Kb](#)



In an effort to have the latest and greatest wizz-bang features in my programs I unashamedly ripped off the colour picker from Office 97.

Initially I tried to modify an owner drawn combo box and combine that with a multicolumn combobox, but current multicolumn combo boxes are really just a single column with dividing lines drawn in. I then decided to write the whole thing from scratch based on a button, since it would at least give me a `BN_CLICKED` notification to get things started.

The colour picker is in two parts: an owner drawn button that reflects the currently selected colour, and a popup colour chooser window to select this colour. When the user clicks on the button the popup window appears and all mouse messages are captured until the left mouse button is clicked, or until the Enter or Escape keys are pressed. The popup window can be navigated using the mouse or the keyboard and includes tooltips explaining what each colour is.

The control can be incorporated into a project like any other `CButton` derived control. Either Create the control manually, subclass an existing `CButton` or `DDX_control` it. The control also comes with a `DDX_ColourPicker` routine to get/set the colour of the control using a variable of type `COLORREF`.

The Colour Picker is contained in the class `CColourPicker`. It uses the class `CColourPopup` for the popup window. These classes are contained in the file [colour_picker_src.zip](#), and a sample project is contained in [colour_picker_demo.zip](#).

`CColourPicker` only has the following public member functions:

```

void      SetColour(COLORREF crColour);
COLORREF  GetColour();

void      SetDefaultText(LPCTSTR szDefaultText);
void      SetCustomText(LPCTSTR szCustomText);

void      SetSelectionMode(UINT nMode); // Either CP_MODE_TEXT or CP_MODE_BK
UINT      GetSelectionMode();

void      SetBkColour(COLORREF crColourBk);
COLORREF  GetBkColour();

void      SetTextColour(COLORREF crColourText);
COLORREF  GetTextColour();

```

SetDefaultText allows you to set the text that will appear in the "Default" area of the colour chooser. If you pass NULL, then the Default area will not be available to the user. If this area is available and the user selects it, the value **CLR_DEFAULT** will be returned.

SetCustomText allows you to set the text that will appear in the "Custom" area of the colour chooser. If you pass NULL, then the Custom area will not be available to the user. The Custom area allows the user to select a custom colour using the standard windows colour selection dialog.

You can choose whether the colour chosen using the dropdown colour chooser will affect the text or the background colour using the function **SetSelectionMode(int nMode)**. Possible values for nMode are **CP_MODE_TEXT** to make colour changes affect the text colour, and **CP_MODE_BK** to make changes affect the background (default).

SetColour, **GetColour** and the the DDX-function will set and get the colour according to the current selection mode. To access the text colour and the background colour directly use the **Set/GetTextColour** and **Set/GetBkColour** functions.

There are also a number of user messages that may be handled to get more information from the control. These are:

Message	Description
CPN_SELCHANGE	Colour Picker Selection change
CPN_DROPDOWN	Colour Picker drop down
CPN_CLOSEUP	Colour Picker close up
CPN_SELENDOK	Colour Picker end OK
CPN_SELENDCANCEL	Colour Picker end (cancelled)

These messages can be handled using **ON_MESSAGE(< MESSAGE>, MessageFn)** in you message map entries, where **MessageFn** is of the form

```
afx_msg LONG MessageFn(UINT lParam, LONG wParam);
```

The demo program gives an example of how to do this.

Acknowledgments

Alexander Bischofberger kindly supplied the Selection mode modifications, as well as the background and text color methods. Paul Wilkerson fixed a focus related bug, and Geir Arne Trillhus also helped fix a few bugs.

About Chris Maunder



Chris is the Administrator, Designer, Chief Editor and Shameless VBScript Hack who wrote and runs CodeProject. He's been programming since 1988 while pretending to be, in various guises, an astrophysicist, mathematician, physicist, hydrologist, geomorphologist, defence intelligence researcher and then, when all that got a bit rough on the nerves, a web developer. He is a Microsoft Visual C++ MVP both globally and for Australia locally.

His programming experience includes C/C++, C#, MFC, ASP, ASP.NET, and far, far too much FORTRAN. He has worked on PocketPCs, AIX mainframes, Sun workstations, and a CRAY YMP C90 behemoth but finds PCs take up less desk space.



Site Admin



Editor



Site Builder

He dodges, he weaves, and he never gets enough sleep. He is kind to small animals.

Chris was born and bred in Australia but is currently based in Toronto enjoying the weather. For relaxation he is into snowboarding, rock climbing, mountain biking, and storm chasing.

Click [here](#) to view Chris Maunder's online profile.



Discussions and Feedback



18 comments have been posted for this article. Visit http://www.codeproject.com/miscctrl/colour_picker.asp to post and view comments on this article.

[All Topics](#), [MFC / C++](#) >> [Miscellaneous Controls](#) >> [Colour Selection Controls](#)

Updated: 20 Nov 1999

Article content copyright Chris Maunder, 1999
everything else Copyright © [CodeProject](#), 1999-2004.