

```

function ChooseColor(var CC: TChooseColor): Bool;

type
  PFindReplace = ^TFindReplace;
  TFindReplace = record
    lStructSize: Longint;
    hWndOwner: HWND;
    hInstance: THandle;

    Flags: Longint;
    lpstrFindWhat: PChar;
    lpstrReplaceWith: PChar;
    wFindWhatLen: Word;
    wReplaceWithLen: Word;
    lCustData: Longint;
    lpfnHook: function (Wnd: HWND; Msg, wParam: Word; lParam: Longint): Word;

    lpTemplateName: PChar;
  end;
  { size of this struct $20 }
  { handle to owner's window }
  { instance handle of .EXE that contains cust. dlg. template }
  { one or more of the fr_?? }
  { ptr. to search string }
  { ptr. to replace string }
  { size of find buffer }
  { size of replace buffer }
  { data passed to hook fn. }
  { ptr. to hook fn. or nil }
  { custom template name }

function ReplaceText(var FindReplace: TFindReplace): HWND;

type
  PChooseFont = ^TChooseFont;
  TChooseFont = record
    lStructSize: Longint;
    hWndOwner: HWND;
    hDC: HDC;
    lpLogFont: PLogFont;
    iPointSize: Integer;
    Flags: Longint;
    rgbColors: Longint;
    lCustData: Longint;
    lpfnHook: function (Wnd: HWND; Msg, wParam: Word; lParam: Longint): Word;

    lpTemplateName: PChar;
    hInstance: THandle;

    lpszStyle: PChar;

    nFontType: Word;

    nSizeMin: Integer;
    nSizeMax: Integer;
  end;
  { caller's window handle }
  { printer DC/IC or nil }
  { ptr. to a LOGFONT struct }
  { 10 * size in points of selected font }
  { enum. type flags }
  { returned text color }
  { data passed to hook fn. }
  { ptr. to hook function }
  { custom template name }
  { instance handle of .EXE that contains cust. dlg. template }
  { return the style field here must be lf_FaceSize or bigger }
  { same value reported to the EnumFonts call back with the extra fonttype_bits added }
  { minimum pt size allowed & }
  { max pt size allowed if cf_LimitSize is used }

{ these are extra nFontType bits that are added to what is returned to the EnumFonts callback routine }

Italic_FontType = $0200;
Regular_FontType = $0400;

wm_ChooseFont_GetLogFont = wm_User + 1;

{ strings used to obtain unique window message for communication between dialog and caller }

SetRGBString = 'commdlg_SetRGBColor';
FindMsgString = 'commdlg_FindReplace';
HelpMsgString = 'commdlg_help';

{ HIWORD values for lParam of commdlg_LBSelChangeNotify message }

cferr_NoFonts = $2001;
cferr_MaxLessThanMin = $2002;

fnErr_FilenameCodes = $3000;
fnErr_SubclassFailure = $3001;
fnErr_InvalidFilename = $3002;
fnErr_BufferTooSmall = $3003;

frErr_FindReplaceCodes = $4000;
frErr_BufferLengthZero = $4001;

ccErr_ChooseColorCodes = $5000;

{

      Digital and the DIGITAL logo are registered trademarks of
      Digital Equipment Corporation.

}

```