

Viết hoá một số Dialog thông dụng

Văn Chí Nam – Nguyễn Đức Hoàng Hạ

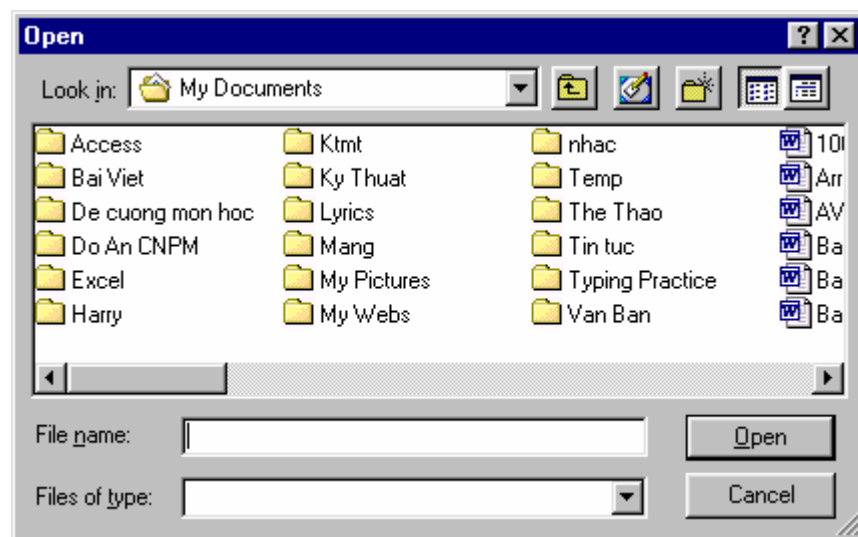
Khoa Công nghệ Thông tin, Trường ĐH KHTN TP.HCM

(vcnam@fit.hcmuns.edu.vn, ndhha@fit.hcmuns.edu.vn)

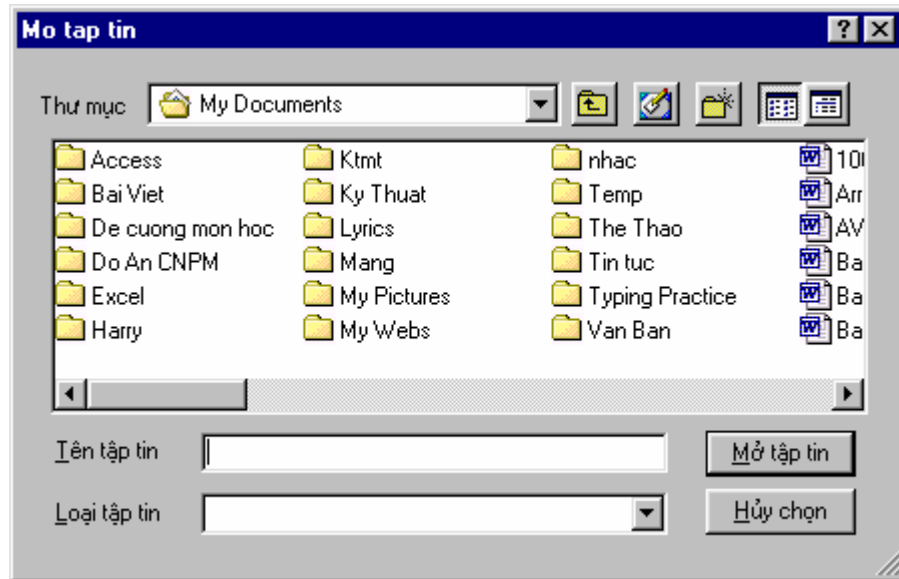
Phiên bản cập nhật ngày 16/05/2005

Chúng ta đã quá quen với một số dialog thông dụng : Open Dialog, Color Dialog, SaveAs Dialog,... Đây là những dialog chuẩn đã được MFC hỗ trợ tốt thông qua các lớp CFileDialog, CColorDialog,... Chúng ta cũng đều biết những dialog này đều sử dụng tiếng Anh làm ngôn ngữ giao tiếp, câu hỏi đặt ra ở đây, liệu chúng ta có thể Việt hoá những dialog này hay không ? Điều đó có nghĩa là vẫn giữ nguyên những thành phần trên các dialog chuẩn này, chỉ thay thế những phần hiển thị bằng ngôn ngữ Anh thành ngôn ngữ Việt.

Trong bài viết này, tôi sẽ giới thiệu cách thức viết hoá một dialog quen thuộc: Open Dialog. Các dialog khác có cách thức hiện tương tự.



Hình ảnh Open dialog thông dụng



Hình ảnh Open dialog sau khi được Việt hoá

Bây giờ, chúng ta sẽ làm quen với một số thao tác cơ bản đầu tiên.

Việt hoá một control

Công việc này lúc mới nghe qua tưởng chừng là khó khăn, nhưng thực chất không có gì là gay go cả, đó chỉ là công việc thay đổi thuộc tính font của các control thành font tiếng Việt (trong trường hợp này, tôi sử dụng font tiếng Việt *VK Sans Serif*). Để thay đổi thuộc tính font của control, chúng ta có thể có các cách sau đây :

- + Thứ nhất, thay đổi thuộc tính font cho toàn bộ dialog, bằng cách thay đổi chọn font trong bảng property của dialog.

- + Thứ hai, thiết lập font cho từng control có lớp dẫn xuất từ CWnd.

```

CFont *font;
LOGFONT logFont;
CWnd* pWnd;
    pWnd = GetParent()->GetDlgItem(IDOK);
//lấy con trỏ CWnd của đối tượng có ID là IDOK
    font = pWnd->GetFont();
//lấy font của đối tượng
    font->GetLogFont(&logFont);
//chuyển đổi CFont về dạng cấu trúc LOGFONT
    strcpy(logFont.lfFaceName,"VK Sans Serif");//thay đổi tên
font ở đây
//thay đổi thuộc tính tên font của cấu trúc LOGFONT
    FontTemp = new CFont;
//FontTemp biến member thuộc tính private của lớp
    FontTemp->CreateFontIndirect(&logFont);

```

```

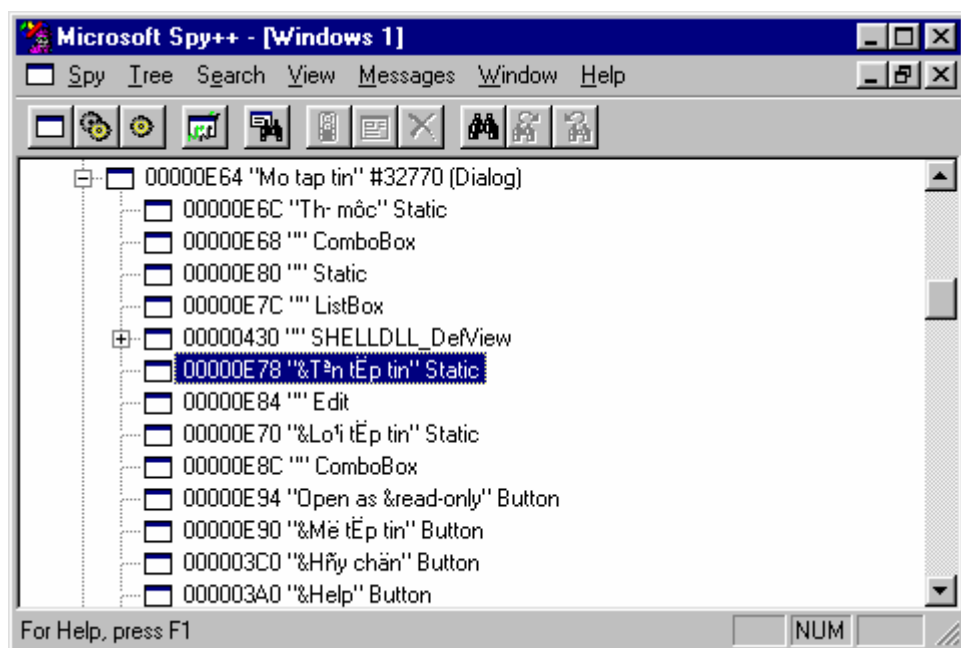
pWnd->SetFont (FontTemp) ;
//gán font mới cho đối tượng
    GetParent () ->GetDlgItem (IDOK) ->SetWindowText ("&Mở tệp
tin"); //Mở tệp tin
//thay đổi nội dung text của đối tượng

```

Lấy ID của một control trên một dialog

Thông thường, khi chúng ta sử dụng control, chúng ta đều đặt cho chúng một cái tên để dễ dàng sử dụng sau này. Các tên này đôi khi không cần thiết vì mỗi cái tên được tương ứng bằng một con số duy nhất do trình biên dịch gán cho và được đặt trong tập tin *Resource.h*. Chúng ta làm việc với các control thông qua các cái tên thực chất làm việc với các con số.

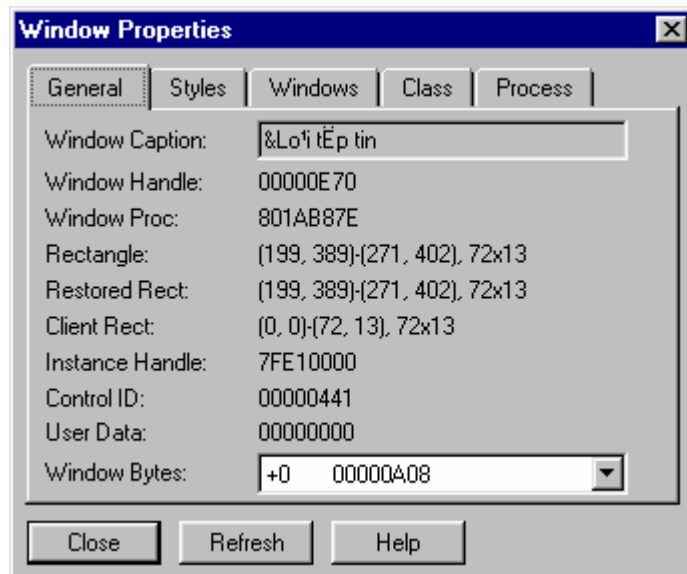
Với các dialog do chúng ta tạo ra, chúng ta dễ dàng xác định được các con số này, nhưng với các dialog do người khác viết thì làm sao chúng ta có thể xác định được các giá trị này ? Cách đơn giản là dùng chương trình *Spy++* có sẵn trong bộ tools khi chúng ta cài VC++ hoặc VB.



Hình ảnh của chương trình *Spy++* thực thi

Để lấy control ID của một control, chúng ta phải xác định control đó thuộc về dialog nào và nó có tên là gì. Khi chạy *Spy++* chúng ta sẽ thấy hình ảnh một Tree control, mỗi nhánh sẽ đại diện cho một chương trình (hoặc 1 dialog) hiện đang chạy trên máy. Cố gắng dò tìm dialog thích hợp. Mở rộng (expand) nhánh bằng cách bấm vào nút "+". Một loạt các control có trên chọn control đó sẽ xuất hiện. Chọn control

thích hợp cần lấy ID (dĩ nhiên là phải cố đoán đó là control nào trong cả đồng control), nhấn phím chuột phải, chọn chức năng *Property*. Ta sẽ thấy được hình ảnh sau :



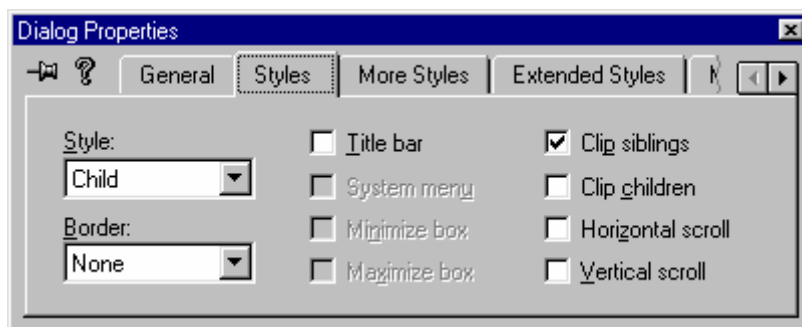
Trên *Window Properties* có rất nhiều thuộc tính, ở đây, chúng ta chỉ quan tâm đến ID của nó nên ta chỉ cần biết đến giá trị control ID (dạng Hexa) mà thôi. Lưu ý rằng giá trị ID này chỉ duy nhất trên dialog được chọn mà thôi.

Các bước thực hiện

Bước 1

Tạo một dialog mới.

Thiết lập cho nó các thuộc tính sau (trong Styles) : style = **child** ; border = **none**; check *clip siblings*



Thêm trên dialog mới này một *group box* control không có caption và ID của nó là $stc32 = 0x045F$ (Quan trọng !!!). (Trên dialog này có thể thêm các control khác tùy theo ý người thiết kế.)



Bước 2

Tạo một lớp mới cho dialog này, lớp mới này ban đầu dẫn xuất từ `CDialog` và được đặt tên là `CSpecialFileDialog`.

Trong file Header thay đổi :

```
class CSpecialFileDialog : public CFileDialog
//dẫn xuất từ CFileDialog thay vì từ CDialog

CSpecialFileDialog(BOOL bOpenFileDialog, LPCTSTR lpszDefExt =
NULL,
    LPCTSTR lpszFileName = NULL,
    DWORD dwFlags = OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT,
    LPCTSTR lpszFilter = NULL,
    CWnd* pParentWnd = NULL);
//constructor
```

Trong file thực thi.CPP :

```
CSpecialFileDialog::CSpecialFileDialog(BOOL bOpenFileDialog,
    LPCTSTR lpszDefExt, LPCTSTR lpszFileName, DWORD dwFlags,
    LPCTSTR lpszFilter, CWnd* pParentWnd)
: CFileDialog(bOpenFileDialog, lpszDefExt, lpszFileName,
    dwFlags, lpszFilter, pParentWnd)
{
    //{{AFX_DATA_INIT(CSpecialFileDialog)
    // NOTE: the ClassWizard will add member initialization here
    //}}AFX_DATA_INIT
    m_ofn.Flags |= OFN_ENABLETEMPLATE;
    m_ofn.lpTemplateName = MAKEINTRESOURCE(IDD_DIALOG1);
    m_ofn.lpstrTitle = "Mô tap tin";//thay đổi nội dung caption bar
của dialog
}
```

Thêm vào lớp `CSpecialDialog` một biến private là một đối tượng con trỏ `FontTemp` thuộc lớp `CFont`, con trỏ này có nhiệm vụ lưu giữ thuộc tính font của các control trong Dialog, cần phải hủy bỏ con trỏ khi không sử dụng dialog nữa.

Bước 3

Thêm vào lớp *CSpecialDialog* hàm *OnInitDialog* có nội dung như sau :

```
BOOL CSpecialFileDialog::OnInitDialog()
{
    BOOL bRet = CFileDialog::OnInitDialog();
    CFont *font;
    LOGFONT logFont;
    if (bRet == TRUE)
    {
        //xác định và thay đổi font tiếng Việt cho các control cần thiết.
        CWnd* pWnd;
        pWnd = GetParent()->GetDlgItem(IDOK);
        font = pWnd->GetFont();
        font->GetLogFont(&logFont);
        strcpy(logFont.lfFaceName,"VK Sans Serif");
        FontTemp = new CFont;
        FontTemp->CreateFontIndirect(&logFont);
        pWnd->SetFont(FontTemp);
        GetParent()->GetDlgItem(IDOK)->SetWindowText("&Mở tệp tin");/*Mở tệp tin*/
        GetParent()->GetDlgItem(0x442)->SetFont(FontTemp);
        GetParent()->GetDlgItem(0x442)->SetWindowText("&T^n tệp tin");/*Tên tệp tin*/
        GetParent()->GetDlgItem(0x443)->SetFont(FontTemp);
        CRect rect;
        pWnd = GetParent()->GetDlgItem(0x443);
        //---
        //đoạn chương trình này dùng để điều chỉnh vị trí của control.
        pWnd->GetClientRect(&rect);
        rect.right += 30;
        pWnd-> SetWindowPos
        (&wndTop,rect.left,rect.top+10,rect.Width(),rect.Height(), SWP_SHOWWINDOW);
        //--
        GetParent()->GetDlgItem(0x443)->SetWindowText("Th- mục");/*Thư mục*/
        GetParent()->GetDlgItem(0x441)->SetFont(FontTemp);
        GetParent()->GetDlgItem(0x441)->SetWindowText("&Lo^i tệp tin");/*Loại tệp tin*/
        GetParent()->GetDlgItem(IDCANCEL)->SetFont(FontTemp);
        GetParent()->GetDlgItem(IDCANCEL)->SetWindowText("&Hủy ch^n");/*Hủy chọn*/
    }
    return bRet;
}
```

Ghi chú : các giá trị ID **0x441,0x442,0x443** được xác định bằng *Spy++* theo cách đã trình bày ở trên.

Bước 4

Xoá bỏ con trỏ *FontTemp* khi không còn sử dụng dialog nữa.

```
BOOL CSpecialFileDialog::DestroyWindow()
{
    // TODO: Add your specialized code here and/or call the base
class
    if (FontTemp) delete FontTemp;

    return CDialog::DestroyWindow();
}
```

Do *CSpecialDialog* dẫn xuất từ *CFileDialog* nên ta có thể thêm cho nó một số hàm cần thiết của riêng chúng ta như :

```
CString CSpecialFileDialog::Layduongdan()
{
    return CFileDialog::GetPathName();
}
```

CÁCH SỬ DỤNG

Cách sử dụng không khác gì so với *CFileDialog*, khai báo một đối tượng thuộc lớp *CSpecialDialog* và sử dụng hàm *DoModal* để kích hoạt dialog hoạt động

Ví dụ :

```
CSpecialDialog dlg(TRUE);
if (dlg.DoModal() == IDOK)
{
}
```

Các dialog dẫn xuất từ các lớp còn lại có cách thực hiện không khác lắm.

Tài Liệu Tham Khảo :

- Programming VC++6.0 Fifth Edition
- MSDN Library