

# SỬ DỤNG RICHEDIT CONTROL

Văn Chí Nam – Nguyễn Đức Hoàng Hạ

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

([vcnam@fit.hcmuns.edu.vn](mailto:vcnam@fit.hcmuns.edu.vn), [ndhha@fit.hcmuns.edu.vn](mailto:ndhha@fit.hcmuns.edu.vn))

*Phiên bản cập nhật ngày 16/05/2005*

## RichEdit control là gì ?

RichEdit control là một loại control chuẩn của Windows, dùng cho việc nhập liệu hay hiển thị các đoạn văn bản lên dialog. RichEdit control khá giống với Edit control.

Điểm đặc biệt của RichEdit control so với Edit control là : RichEdit control cho phép thiết lập từng định dạng riêng biệt trên đơn vị từng ký tự trong control (tức là đối với mỗi ký tự trong RichEdit control, chúng ta có thể thiết lập các định dạng màu, cỡ chữ, kiểu chữ... ) trong khi đối với Edit control chúng ta chỉ có thể có một thiết lập duy nhất đối với toàn control (không thể thiết lập định dạng cho từng ký tự riêng lẻ).

## Lưu ý khi sử dụng RichEdit control

Khi sử dụng RichEdit control trong một chương trình nào đó, có một hàm **quan trọng** nếu chúng ta quên sử dụng hay đặt hàm đó sai vị trí thì sẽ dẫn đến việc chương trình không thể thực thi được. Do đó :

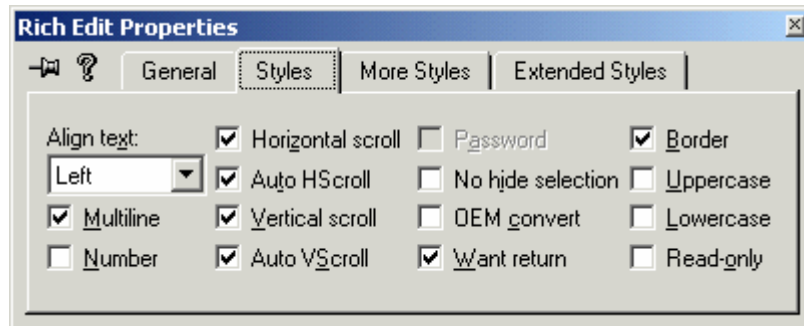
**NHỚ DÙNG HÀM `AfxInitRichEdit()` KHI SỬ DỤNG RICHEDIT CONTROL VÀ ĐẶT NÓ TRONG HÀM `InitInstance()` TRONG `C..App()`**

Ví dụ :

```
BOOL CRECDemoApp::InitInstance()
{
    AfxInitRichEdit();//Quan trọng
    CRECDemoDlg dlg;
    m_pMainWnd = &dlg;
    int nResponse = dlg.DoModal();
    if (nResponse == IDOK)
    {
    }
    else if (nResponse == IDCANCEL)
    {
        // TODO: Place code here to handle when the dialog is
        // dismissed with Cancel
    }
    return FALSE;
}
```

## Một số lưu ý khác

- Hiện thị được nhiều dòng trong RichEdit control, check **Styles\Multiline**.
- Hiện thị scroll, check vào **Styles\Horizontal scroll**,...
- Sử dụng được phím Enter trong RichEdit control, check vào **Styles\Want return**.



- Không bị giật màn hình khi định dạng lượng lớn ký tự trong RichEdit control, dùng hai hàm : **LockWindowUpdate** và **UnlockWindowUpdate**

```
void CRECDemoDlg::OnDinhdang()
{
    // TODO: Add your control notification handler code here
    CRichEditCtrl* pRE;
    int i;
    pRE = (CRichEditCtrl*)GetDlgItem(IDC_RICHEDIT1);
    LockWindowUpdate();
    ...
    for (i=0;i<tam.GetLength();i++)
    {
        ...
    }
    UnlockWindowUpdate();
}
```

## MỘT SỐ CẤU TRÚC, HÀM THƯỜNG GẶP

### Cấu trúc CHARFORMAT

```
typedef struct _charformat {
    UINT    cbSize;
    DWORD   dwMask;
    DWORD   dwEffects;
    LONG    yHeight;
    LONG    yOffset;
    COLORREF crTextColor;
```

```

    BYTE    bCharSet;
    BYTE    bPitchAndFamily;
    TCHAR    szFaceName[LF_FACESIZE];
} CHARFORMAT;

```

Đối với cấu trúc **CHARFORMAT** này, tham khảo thêm trong MSDN phần CHARFORMAT. Ở đây, chúng tôi chỉ đề cập một số vấn đề có liên quan xung quanh cấu trúc này:

- Cấu trúc này dùng cho định dạng các ký tự trong RichEdit control sau khi đã được chọn.
- Thành phần **cbSize** luôn được sử dụng và nhớ đặt :  

```
CHARFORMAT cf;
cf.cbSize = sizeof(CHARFORMAT);
```
- Thành phần **dwMask** cho biết những thuộc tính (định dạng) nào có sự thay đổi:

VD :

```
cf.dwMask=CFM_ITALIC|CFM_COLOR|CFM_BOLD;
```

Cho biết 3 thuộc tính về độ nghiêng (Italic), màu sắc (Color), thuộc tính đậm (Bold) sẽ có sự thay đổi.

- Đi liền với thành phần **dwMask** là thành phần **dwEffects**. Thành phần này dùng để gán (hoặc loại bỏ) thuộc tính định dạng đã được đưa ra trong thành phần **dwMask**. Các thuộc tính có ảnh hưởng từ **dwEffects** là Bold, Italic, Underlined, hay StrikeOut... Nếu muốn gán một thuộc tính nào thêm ta dùng toán tử | và loại bỏ thuộc tính nào ta dùng toán tử & và toán tử ~.

**Ví dụ :**

Để thiết lập các thuộc tính : chữ in nghiêng, không đậm, và có màu xanh lá cây (Green), font chữ là Times New Roman ta gán các thành phần của cấu trúc CHARFORMAT như sau :

```

CHARFORMAT cf;
cf.cbSize = sizeof(CHARFORMAT);
cf.dwMask = CFM_ITALIC | CFM_COLOR | CFM_BOLD | CFM_FACE;
cf.dwEffects = CFE_ITALIC;
cf.dwEffects &= CFE_BOLD;
cf.dwEffects &= ~CFE_AUTOCOLOR;
cf.crTextColor=RGB(0,255,0);
strcpy(cf.szFaceName,"Times New Roman");

```

## HÀM SetSel VÀ HÀM SetSelectionCharFormat

Để thiết lập định dạng cho một đoạn văn bản ta thực hiện các bước sau :

- Chọn đoạn văn bản đó (dùng hàm **SetSel**). Lưu ý, nếu muốn chọn toàn bộ nội dung của control thì dùng **SetSel(0,-1)**.
- Thiết lập các thuộc tính cần thay đổi của đoạn văn bản thông qua cấu trúc **CHARFORMAT**.

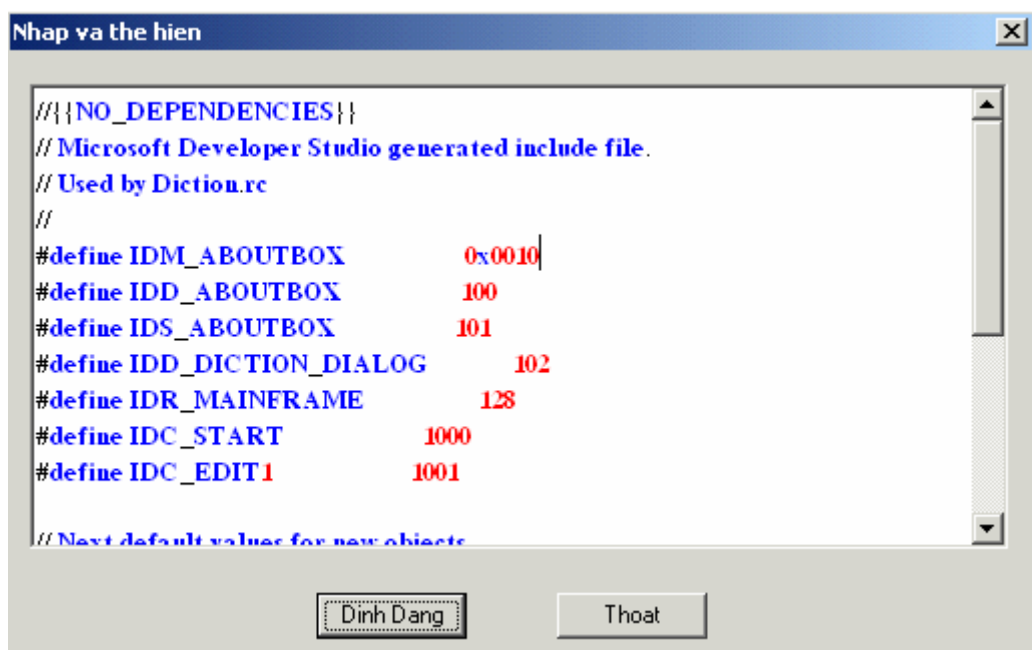
Thay đổi định dạng cho đoạn văn bản thông hàm **SetSelectionCharFormat**.

## VÍ DỤ MINH HOẠ

Thiết kế dialog như hình dưới đây.

Cho phép

- Người sử dụng nhập những dòng văn bản vào hộp RichEditCtrl.
- Khi người sử dụng bấm nút *Định Dạng*, các ký tự trong hộp Rich Edit Ctrl được định dạng theo một tiêu chuẩn nào đó. Chẳng hạn : các ký tự từ A đến Z : màu xanh, các ký số (0-9) : màu đỏ, các ký tự đặc biệt khác : màu đen...



## Viết hàm định dạng

```
void CRECDemoDlg::OnDinhdang()  
{  
    // TODO: Add your control notification handler code here  
    CRichEditCtrl* pRE;  
    int i;  
    pRE = (CRichEditCtrl*)GetDlgItem(IDC_RICHEDIT1);  
    CString tam;  
    pRE->GetWindowText(tam);  
}
```

```
pRE->SetSel(0,-1);
ChangeChar(pRE);

for (i=0;i<tam.GetLength();i++)
{
    if (tam[i]!=' ' && (tam[i]<'A' || tam[i]>'Z') &&
(tam[i]<'a' || tam[i]>'z'))
    {
        pRE->SetSel(i,i+1);
        if ('0'<=tam[i]&& tam[i]<='9')
            ChangeNumber(pRE);
        else
            ChangeOther(pRE);
    }
}
}
```

### Hàm ChangeChar :

```
void CRECDemoDlg::ChangeChar(CRichEditCtrl *pRE)
{
    CHARFORMAT cf;
    cf.cbSize = sizeof(CHARFORMAT);
    cf.dwMask=CFM_BOLD|CFM_COLOR;
    cf.dwEffects = CFE_BOLD;
    cf.crTextColor=RGB(0,0,255);
    pRE->SetSelectionCharFormat(cf);
}
```

Các hàm còn lại viết tương tự.

Tham khảo các hàm, các cấu trúc có liên quan trong MSDN.