

SƠ NÉT VỀ VISUAL C++ 6.0 VÀ THƯ VIỆN MFC

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 12/10/2004

GIỚI THIỆU SƠ NÉT

Visual C++ 6.0 và thư viện lập trình MFC

Visual C++ 6.0 nằm trong bộ Microsoft Visual Studio 6.0. Đây là một môi trường lập trình đa năng dành cho ngôn ngữ C/C++ và vì là một môi trường lập trình trên hệ điều hành Windows nên Visual C++ 6.0 cho phép lập trình viên thực hiện rất nhiều công việc, hỗ trợ lập trình viên việc coding, thiết kế giao diện... Trong VC++ 6.0 chúng ta có thể tạo được : các ứng dụng trên Windows, ActiveX, hay thư viện liên kết động DLL... VC++ 6.0 có nhiều công cụ giúp việc thiết kế giao diện cho chương trình, kiểm lỗi và sửa lỗi.

Thư viện MFC (Microsoft Foundation Class) là thư viện tập hợp các lớp hướng đối tượng đóng gói các hàm API của Windows và các kiểu dữ liệu đặc biệt. Bên cạnh sử dụng thư viện MFC trong Visual C++ 6.0, chúng ta có thể sử dụng các hàm C chuẩn.

Các công cụ hỗ trợ lập trình

Visual Assist

Đây là một công cụ Add-in vào môi trường lập trình. Nó giúp cho việc hiển thị các hàm, các biến, các đoạn chương trình một cách rõ ràng (thông qua màu sắc của các biến, kiểu dữ liệu, hàm, từ khoá...). Một đoạn code trong Visual C++ có sử dụng Visual Assist được thể hiện dưới đây :

```
void CListCtrlDemoDlg::OnPaint()  
{  
    if (IsIconic())  
    {  
        CPaintDC dc(this); // device context for painting  
        SendMessage(WM_ICONERASEBKGD, (WPARAM) dc.GetSafeHdc(), 0);  
        // Center icon in client rectangle  
        int cxIcon = GetSystemMetrics(SM_CXICON);  
        int cyIcon = GetSystemMetrics(SM_CYICON);  
        CRect rect;  
        GetClientRect(&rect);  
        int x = (rect.Width() - cxIcon + 1) / 2;  
        int y = (rect.Height() - cyIcon + 1) / 2;  
        // Draw the icon
```

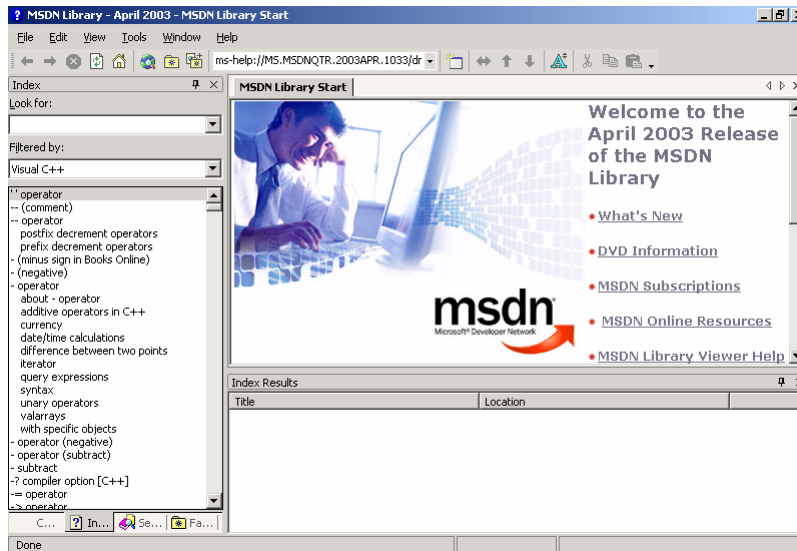
```
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}
```

Visual Assist cung cấp một bộ phận “nhắc tuồng” hoạt động rất hiệu quả. Các chức năng của bộ phận bao gồm : tự động điền tên biến, tên hàm chỉ sau khi gõ 1 vài ký tự, hiển thị các prototype của một hàm (có sẵn trong thư viện hay chỉ mới được tạo ở 1 lớp nào đó)...

Sau khi cài đặt Visual Assist, trong môi trường lập trình sẽ xuất hiện một toolbar như hình dưới đây :



MSDN (Microsoft Developer Network)



Lập trình viên trên hệ điều hành Windows nói chung rất quen thuộc với bộ thư viện này. Có thể xem nó như một tập “bách khoa” cho những ai lập trình trên các ngôn ngữ từ C/C++, Visual Basic cho đến C#, VB.Net.

Trong MSDN, ngoài các tài liệu giới thiệu chi tiết về việc sử dụng những hàm, lớp..., còn có các quyển sách, bài báo kỹ thuật, hay các mẫu code có thể được tận dụng trong việc lập trình.

Trang web tham khảo

Có thể tham khảo các chủ đề của C/C++, VB.NET, C#, ASP.NET ... trên trang web CodeProject (www.codeproject.com). Trên trang web này có rất nhiều bài viết hay về các chủ đề từ cơ bản đến nâng cao. Gần như có thể tìm thấy được tất cả những vấn đề mong muốn ở đây.

SỬ DỤNG MÔI TRƯỜNG LẬP TRÌNH

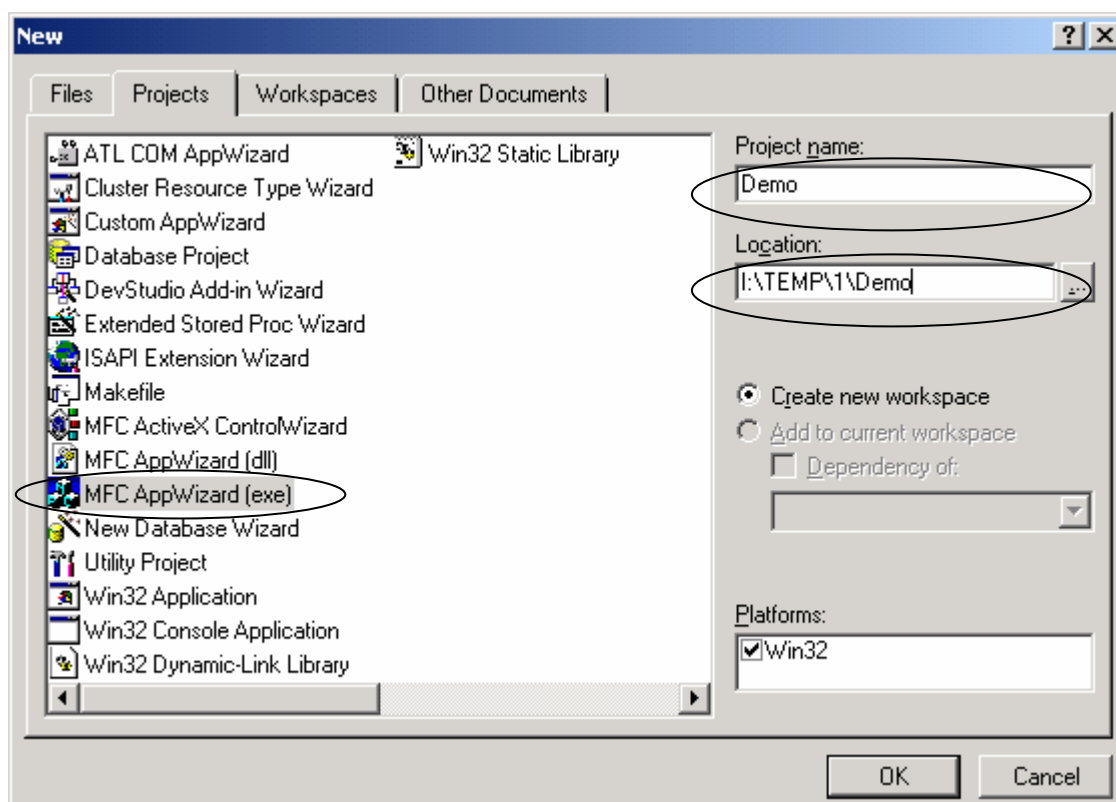
Tạo project sử dụng thư viện MFC

Để tạo một project mới sử dụng thư viện lập trình MFC, chúng ta thực hiện một số bước sau đây :

Bước 1 : Chọn **File \ New \ Projects**

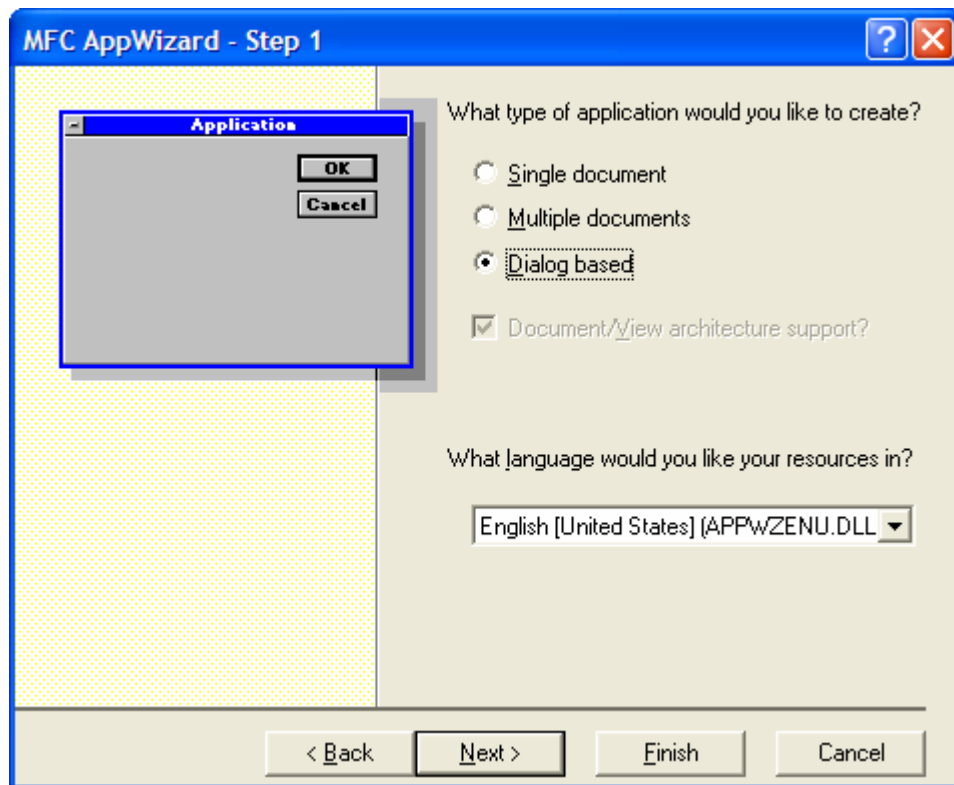
Bước 2 : Chọn **MFC AppWizard (exe)**

Bước 3 : Điền tên project mong muốn vào trong hộp **Project name**, và chọn thư mục / ổ đĩa chỉ đặt project trong hộp **Location**. Nhấn **OK** để tiếp tục.



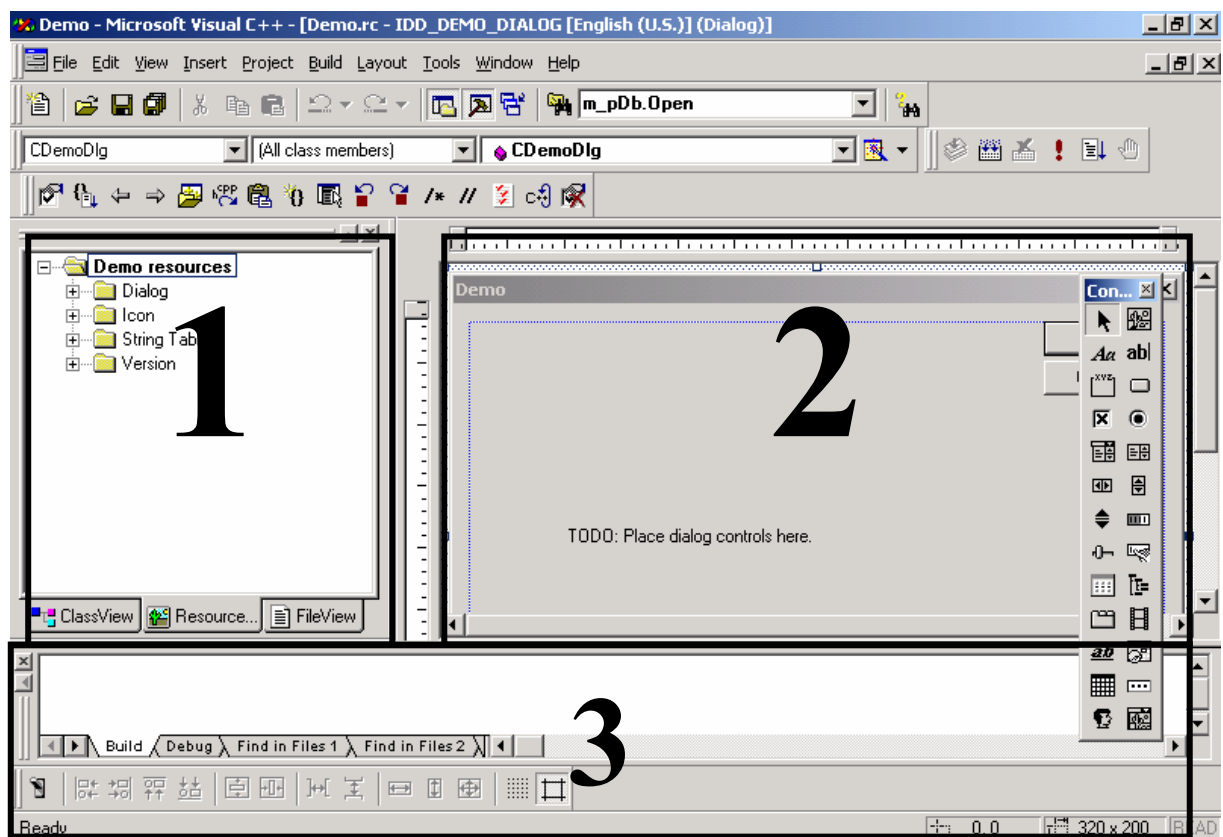
Bước 4 : Trong hộp thoại **MFC AppWizard – Step 1**, có 3 chọn lựa loại ứng dụng cần phát triển¹. Chọn loại ứng dụng thích hợp nhất theo yêu cầu và nhấn **Next** để tiếp tục. (Có thể nhấn **Finish** bất cứ lúc nào để tiết kiệm thời gian nếu như những thông số còn lại để ở dạng mặc định).

¹ 3 loại ứng dụng là : (1) Single Document, (2), Multiple Documents, (3) Dialog-based. Ví dụ, ứng dụng kiểu single document là *Notepad*, *MS Paint*, kiểu Multiple Documents là *Microsoft Word* (*Photoshop*, *EditPlus*), kiểu Dialog-based là *Calculator*...



Bước 5 : Tùy vào loại ứng dụng được chọn thì những bước còn lại có những chọn lựa khác nhau (tham khảo thêm ở những bài viết sau).

Màn hình làm việc



Một cách tương đối, có thể chia màn hình làm việc thành ba vùng khác nhau :

Vùng 1 – Workspace

Đây là không gian tương tác giúp lập trình viên nhanh chóng chuyển qua lại giữa các lớp, các tập tin hay các tài nguyên trong project.

(Tắt/ Mở workspace : Alt + 0 hoặc View/Workspace)

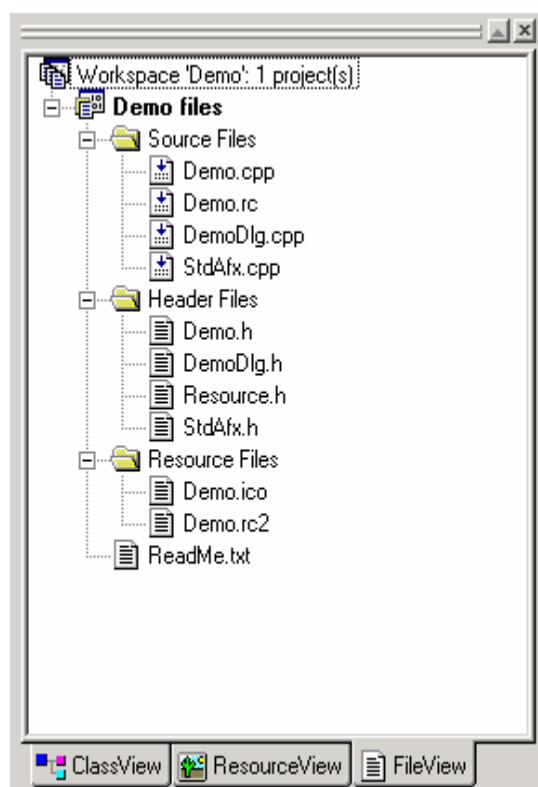
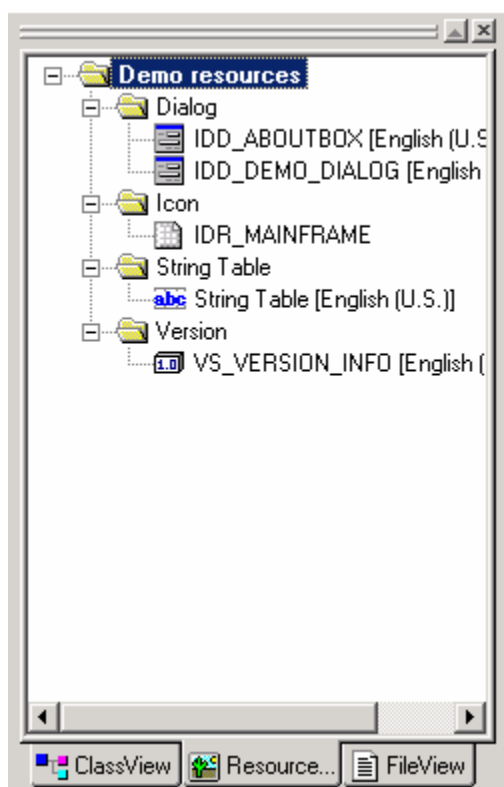
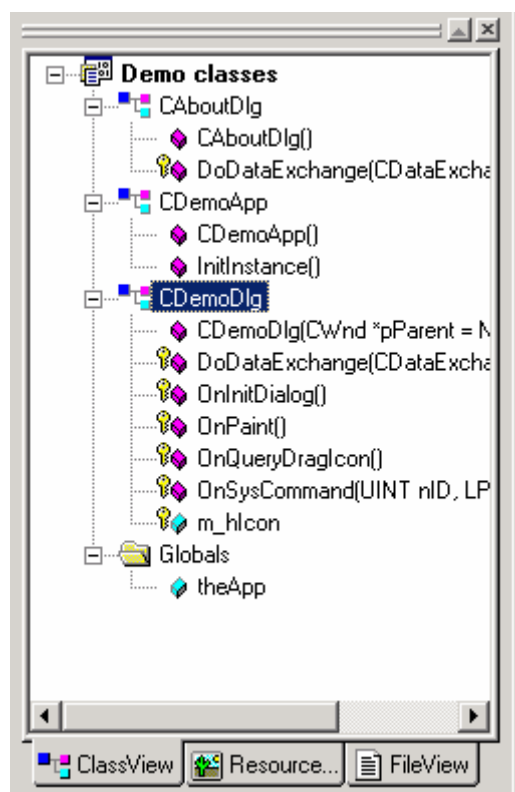
Workspace gồm có 3 tab chính (có thể có các tab phụ add-in): ClassView, ResourceView, FileView. Nội dung các tab này được thể hiện ở dạng cây có thể mở rộng, thu hẹp thông qua các nút +, -.

ClassView cho thấy hình ảnh tổng thể project dưới dạng các lớp. Trong classview, lập trình viên có thể biết được cấu tạo của một lớp (có những hàm, biến nào; tầm vực hoạt động của các hàm, các biến – public, protected, private).

ResourceView cho thấy được những tài nguyên có trong project (dialog, menu, toolbar, bitmap, icon, ...). Thực chất đây là phần hiện thực hoá những gì thể hiện trong tập tin *resource.h* và *.rc*.

FileView cho thấy những tập tin có trong

project :
source file,
header file,
resource file
và những
tập tin
được thêm
vào sau.



Vùng 2

Đó là vùng dùng để thiết kế giao diện (các dialog, menu, toolbar) hay viết code cho project.

Vùng 3 - Output

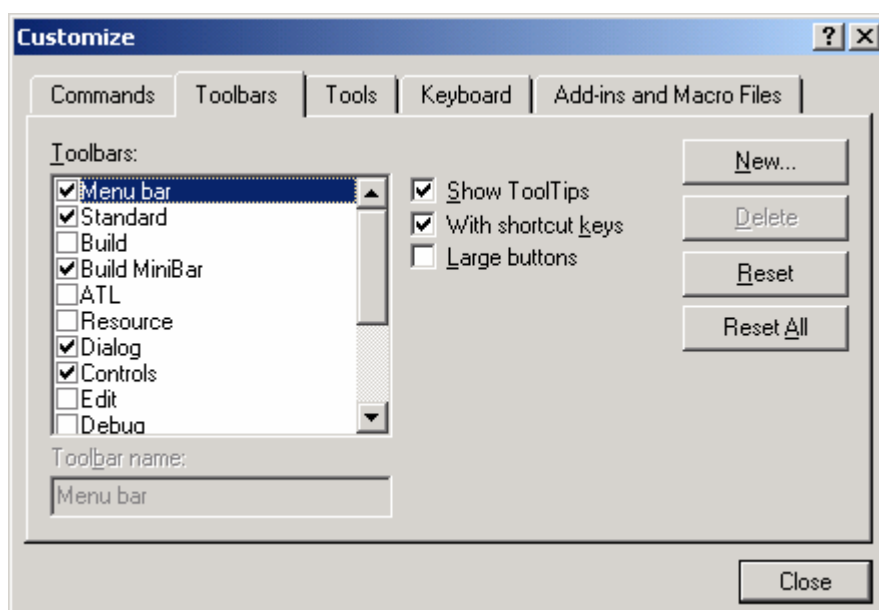
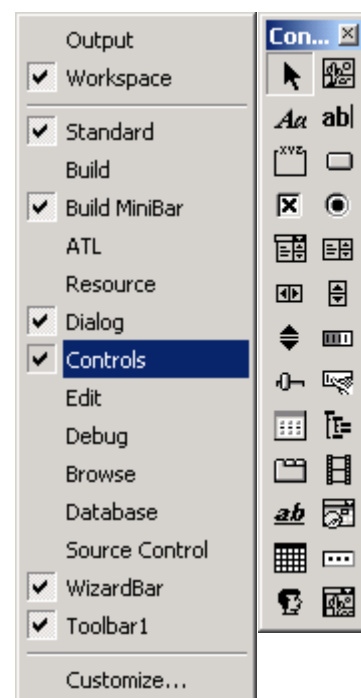
Vùng thể hiện các thông báo lỗi (nếu có), các kết quả thực thi (khi có sử dụng Macro), nội dung của biến (ở chế độ Debug) ...

Các control chuẩn

Hộp Control

Trong môi trường lập trình VC++ 6.0 có thể thực hiện việc kéo thả các control chuẩn trong khi thiết kế giao diện.

Bật/tắt hộp control khi thiết kế thực hiện như sau: nhấn phím chuột phải trên thanh toolbar, một menu như bên tay phải xuất hiện, chọn (không chọn) chức năng **Controls**. Ngoài ra, cũng có thể vào **Tools\ Customize**, check vào toolbars ở vị trí **Controls**.



Các control chuẩn

Visual C++ 6.0 cho phép chèn các control chuẩn sau đây vào trong ứng dụng : PictureBox, Static Text (Label), Edit box (Textbox), GroupBox, Button, Checkbox, RadioButton, ComboBox, ListBox, Scrollbar, Spin, Progressbar, Slider, TreeControl, Tab Control, Animate, RichEdit, DateTime, Month Calendar, IP Address

Thiết lập thuộc tính cho control

Các thuộc tính của control trên Visual C++ 6.0 không nhiều. Tuy nhiên, có những thuộc tính đặc biệt (mặc định) của control bắt buộc lập trình viên phải nhớ để có thể thao tác nhanh chóng.

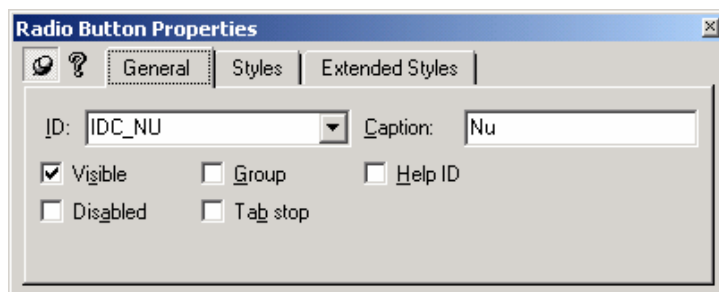
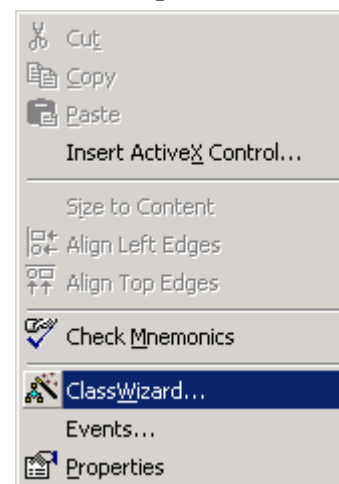
Để thiết lập thuộc tính cho control, thực hiện theo các bước sau :

Bước 1 : chọn đối tượng control cần thao tác.

Bước 2 : nhấn chuột phải trên đối tượng

Bước 3 : Chọn **Properties**.

Bước 4 : Hiệu chỉnh các thông số trên hộp thoại **Properties** theo mong muốn.



Lưu ý

Mỗi control có một định danh (ID) riêng. ID chính một số nguyên duy nhất giúp xác định duy nhất một control. Tuy nhiên, để dễ nhớ trong quá trình lập trình, lập trình viên xác định một control bằng một tên duy nhất (tốt nhất thể hiện BẢNG CHỮ IN HOA).

Có thể xem định nghĩa của các ID gắn với một tên của control trong tập tin *resource.h*.

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by DEMO.RC
//
#define IDR_MAINFRAME          128
#define IDM_ABOUTBOX          0x0010
#define IDD_ABOUTBOX          100
#define IDS_ABOUTBOX          101
#define IDD_DEMO_DIALOG        102
// Next default values for new objects
```

Một vài thuộc tính đáng quan tâm của các control

Control	Thuộc tính mặc định
Static Text	Align : canh chỉnh (trái, phải, giữa) dòng chữ trong static text

	<p>Center vertically : canh chỉnh đều theo chiều dọc</p> <p>No wrap : không nhảy dòng nếu chiều dài dòng chữ vượt khỏi chiều rộng của static text</p> <p>Sunken, Border : đường viền cho static text</p>
Edit box	<p>Align text : canh chỉnh (trái, phải, giữa) dòng chữ trong static text</p> <p>Multiline : cho phép hiển thị nhiều dòng</p> <p>Number : chỉ cho phép nhập vào số</p> <p>HScroll, VScroll : thanh cuộn theo chiều ngang (H), chiều dọc (V)</p> <p>AutoHScroll : tự động cuộn theo chiều ngang (nếu có)</p> <p>AutoVScroll : tự động cuộn theo chiều dọc (nếu có)</p> <p>Password : không hiển thị ký tự nhập vào (chỉ thấy ký tự *)</p> <p>Uppercase, Lowercase : chữ hoa, chữ thường</p> <p>Read-only : không cho phép gõ vào control</p>
Check box Radio Button	<p>Push-like : dạng nút nhấn (chìm/nổi)</p> <p>Horizontal Alignment : canh chỉnh theo chiều ngang</p> <p>Vertical Alignment : canh chỉnh theo chiều dọc</p>
Combobox Listbox	<p>Đối với ComboBox, có một lưu ý nhỏ trong thiết kế : kéo chiều cao của ComboBox đến vị trí tối đa mong muốn ComboBox có thể xổ xuống.</p> <p>Data : Vùng dữ liệu thêm vào trong Listbox, ComboBox khi thiết kế. Nhấn Ctrl-Enter để kết thúc một mục.</p> <p>Sort : Dữ liệu sẽ được sắp xếp theo thứ tự tăng giảm khi xuất hiện trong Listbox, ComboBox</p> <p>Multicolumn : xuất hiện thành nhiều cột (phụ thuộc vào chiều rộng).</p>

Các phím tắt chức năng

Chức năng	Phím tắt
Biên dịch (Compile)	Ctrl – F7
Dịch và liên kết (Build)	F7
Thực thi (Execute)	Ctrl – F5
Thực thi ở chế độ Debug	F5
Tạo breakpoint	F9 (chạy tiếp bằng cách nhấn F5)

Nhảy đến vị trí con trỏ (Debug)	Ctrl – F10
Nhảy qua một dòng thử thi (Debug)	F10
Nhảy vào trong 1 hàm (Debug)	F11
Nhảy ra khỏi 1 hàm (Debug)	Ctrl – F11
Hiển thị dialog xem thử ngay khi thiết kế	Ctrl – T
Đặt thứ tự tab giữa các control	Ctrl – D

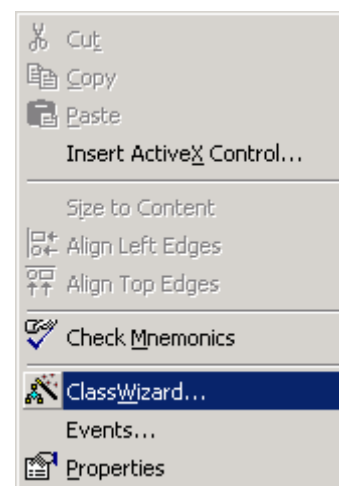
Sự kiện và xử lý sự kiện

Sự kiện

Trên Windows, các sự kiện xảy ra liên tục (khi có một tác động của người dùng, hay của chương trình lên hệ thống). Windows sẽ tiếp nhận các sự kiện rồi gửi đến cho các ứng dụng đang chạy.

Ví dụ : nhấn phím, nhấn chuột trái, nhấn chuột phải, vẽ lại màn hình, thay đổi chọn lựa, bắt đầu chương trình, thoát khỏi chương trình...

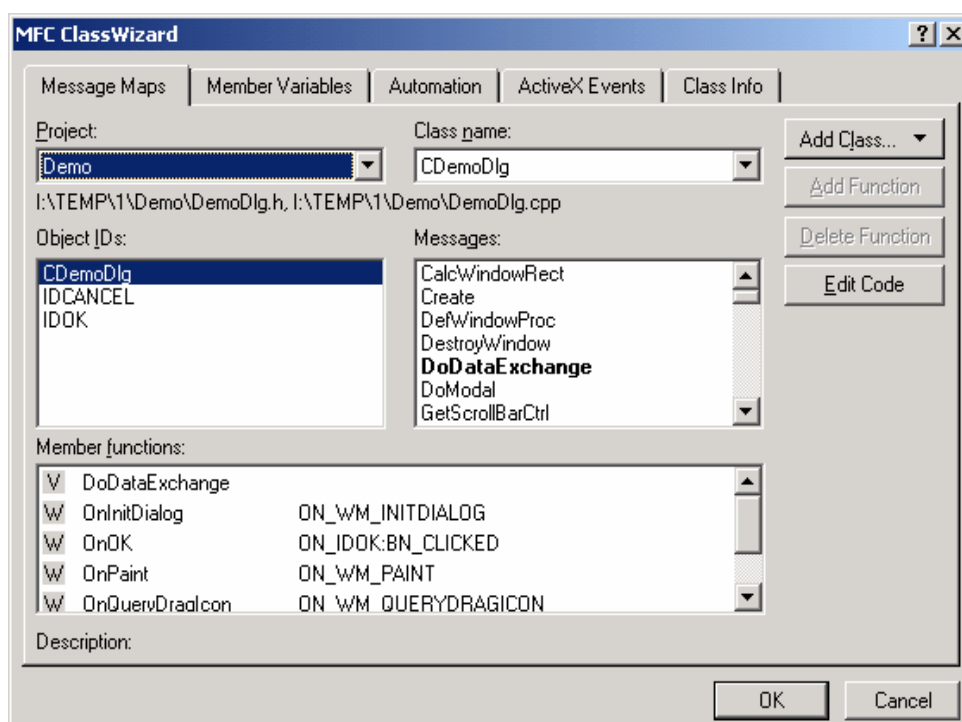
Các ứng dụng nhận được “thông báo” từ Windows rồi tùy theo “nhu cầu” có “tiếp nhận” hay không. Nếu “tiếp nhận” sự kiện thì phải “giải quyết” như thế nào. Việc giải quyết được đặt trong hàm xử lý sự kiện.



Tạo hàm xử lý sự kiện

Sử dụng **ClassWizard** để tạo hàm xử lý cho các sự kiện (lên chương trình hay lên một control cụ thể).

(Phím tắt : Ctrl – W. Ở chế độ thiết kế dialog có thể nhấn phím chuột phải và

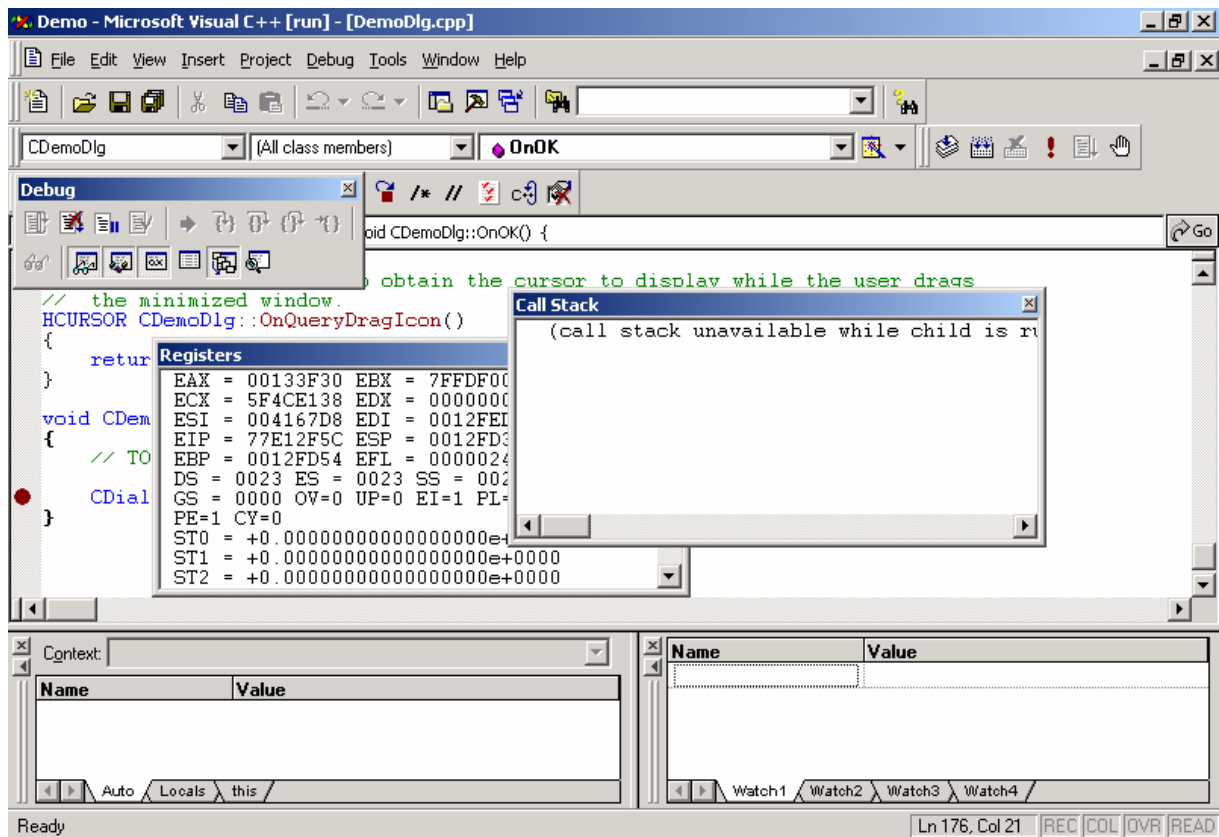


chọn mục **ClassWizard** trên menu.).

Trên hộp thoại **MFC ClassWizard**, có thể chọn lựa lớp (trong **Class name**), ID của đối tượng (**Object IDs**) cần thêm sự kiện vào (**Messages**).

Sau khi chọn **Add Functions**, môi trường lập trình sẽ tự động thêm khai báo hàm, cấu trúc hàm vào những vị trí thích hợp trên tập tin .CPP và .H của lớp.

Debug chương trình



Vùng màn hình **Context** cho biết giá trị của các biến có trong đoạn chương trình đang chạy (kiểm lỗi). Các biến này được đưa vào một cách tự động.

Vùng màn hình **Watch** cho biết giá trị của các biến do lập trình viên mong muốn.

Vùng màn hình **Call Stack** cho biết thứ tự gọi hàm của chương trình.

Vùng màn hình **Registers** cho biết giá trị các thanh ghi.

