# Window 数据类型

WINDOWS API 函数原型的参数都是使用的WINDOWS**数据类型**，搞windows**编程**，这个是必须要知道的。

        WINDOWS**数据类型**都是由C**数据类型**经过重新定义得到的所以看似陌生的**数据类型**其实实质还是C**数据类型**，只是经过了重新定义罢了，当然在VB中又会有相应的**数据类型**与之对应。

**例如：DWORD**的定义是 typedef unsigned long   DWORD;也就是它与标准C**数据类型**中的unsigned long **数据类型**。

**WORD** 定义是 typedef unsigned short     WORD; 也就是它与标准C**数据类型**中的unsigned short**数据类型**。而经常要用到的**HANDLE**类型实质上是无类型指针**void**

**HANDLE**定义为: typedof PVOID HANDLE;

**HANDLE**实际上就是一个**PVOID**，那**PVOID**就是什么呢？Typeof void \*PVOID;

**PVOID**其实就是一个指向**void**类型的指针。

**部分windows数据类型**  
  WORD：16位无符号整形数据

DWORD：32字节无符号整型数据（DWORD32）

DWORD64：64字节无符号整型数据

INT：32位有符号整型**数据类型**

INT\_PTR：指向INT**数据类型**的指针类型

INT32：32位符号整型

INT64：64位符号整型

UINT：无符号INT

LONG：32位符号整型（LONG32）

ULONG：无符号LONG

LONGLONG：64位符号整型（LONG64）

SHORT：无符号短整型（16位）

LPARAM：消息的L参数

WPARAM：消息的W参数

HANDLE：对象的句柄，最基本的句柄类型

HICON：图标的句柄

HINSTANCE：程序实例的句柄

HKEY：注册表键的句柄

HMODULE：模块的句柄

HWND：窗口的句柄

LPSTR：字符指针，也就是字符串变量

LPCSTR：字符串常量

LPCTSTR：根据环境配置，如果定义了UNICODE宏，则是LPCWSTR类型，否则则为LPCSTR类型

LPCWSTR：UNICODE字符串常量

LPDWORD：指向DWORD类型数据的指针

CHAR：8比特字节

TCHAR：如果定义了UNICODE，则为WCHAR，否则为CHAR

UCHAR：无符号CHAR

WCHAR：16位Unicode字符

BOOL：布尔型变量

BYTE：字节类型（8位）

CONST：常量

FLOAT：浮点**数据类型**

SIZE\_T：表示内存大小，以字节为单位，其最大值是CPU最大寻址范围

VOID：无类型，相当于标准C语言中的void

WINAPI：Windows API的函数调用方式，常见于SDK头文件中对API函数的声明中，相当于\_stdcall（更严格地说，这不是**数据类型**，而是一种函数调用约定

**windows数据类型命名规律**

基本**数据类型**包括：BYTE、CHAR、WORD、SHORT、INT等。

指针类型的命令方式一般是在其指向的**数据类型**前加“LP”或“P”，比如指向DWORD的指针类型为“LPDWORD”和“PDWORD”

各种句柄类型的命名方式一般都是在对象名前加“H”，比如位图（BITMAP）对应的句柄类型为“HBITMAP”。

无符号类型一般是以“U”开头，比如“INT”是符号类型，“UINT”是无符号类型。

其实要想知道WINDOWS**数据类型**到底是对应着标准C**数据类型**中的哪一个，可以在开发环境中选中相应的**数据类型**，按F12就可以转到**数据类型**定义的地方，这样就可以清楚的知道，它到底对应的是标准C**数据类型**中的哪一个。

# MFC 对话框添加菜单

VS2010的**MFC对话框**程序中**添加菜单**栏的过程，我大致分了这五步。

    一、将Menu加入Resource视图中

在WorkSpace中的Resource视图下,在任意一个文件夹图标上,右击选择Insert(插入)选项,在弹出的对话框中选择Menu以后,再点击,new按钮,菜单就会添加成功了,但是现在还没有完成,还不能进行编译,因为现在的菜单时空的,在编译的时候会被清除的。

二、用菜单编辑器**添加菜单**栏及菜单项

菜单添加成功后，菜单编辑器自动打开，可以在其中**添加菜单**栏及菜单项，比较简单。

三、将菜单加载到你的对话框

第一步中添加一个菜单项,这是菜单就真正的加到了你的工程中了,接下来就是如何将菜单加载到你的对话框中了。

打开你新加入的菜单项，单击“项目”-->“添加现有项”，在这里选择已有的类，就是你要为其**添加菜单**的对话框的类，例如，\*\*\*Dlg.cpp。然后确定。

四、在相应的文件中添加代码

打开对话框头文件\*\*\*Dlg.h，声明CMenu 变量,例如m\_Menu;

打开\*\*\*Dlg.cpp 文件，在\*\*\*Dlg::OnInitDlg()中加入如下语句：

m\_Menu.LoadMenu(IDR\_MENU1);  //  IDR\_MENU1为你加入的菜单的ID，在Resource视图的Menu文件夹下可以找到

SetMenu(&m\_Menu);

这样就OK了，调试一下，菜单就已经出现在对应的对话框中。

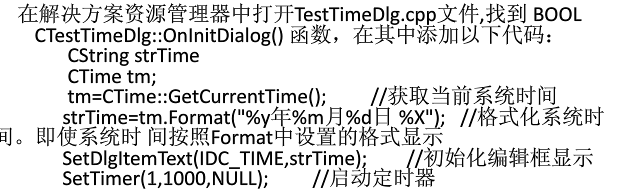
五、利用向导生成菜单消息响应函数

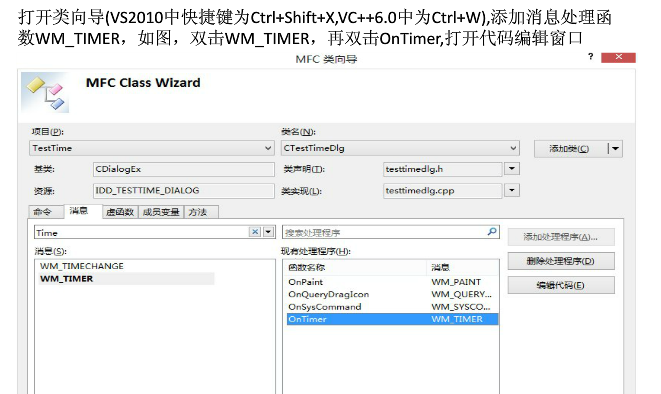
选择项目下的类向导，选择类名为”\*\*\*Dlg.cpp”，在“命令”选项卡下的“对象ID(B)”中，选择各子菜单项的对应的ID，然后在“消息(S)”中双击COMMAND消息。

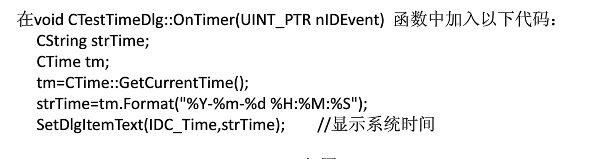
这时会弹出对话框类，你可以更改对应子菜单项的函数处理名，然后点击确定。菜单消息响应函数生成完毕，加入你需要的代码

# 对话框添加时间









# MFC中的ID命名规则及消息前缀 .

IDC\_：控件的ID命名前缀（Control)；

IDM\_：菜单的ID命名前缀（Menu)；

IDD\_：对话框的ID命名前缀（Dialog)；

IDR\_：资源的ID命名前缀（Resource)；

IDS\_：字符串的ID命名前缀（String）；

IDB\_：位图资源的ID命名前缀（Bitmap）；

二、MFC系统消息前缀小集

ABM 应用程序桌面工具条application desktop toolbar

BM 按钮 button control

CB 组合框 combo box control

CBEM 扩展组合框 extended combo box control

CDM 通用对话框common dialog box

DBT 设备devicae

DL 下拉列表 drag list box

DM Default Push button control

DTM Date and time picker control

EM 编辑框edit control

HDM Header control

HXM 热键hot key control

IPM IP控件IP address control

LB 列表框list box control

LVM 列表视图list view control

MCM 日历控件month calendar control

PBM 进度条progress bar

PGM Pager control

PSM 属性property control

RB 伸缩条rebar control

SB 状态条status bar window

SBM 滚动条scrol bar control

STM 静态条static control

TB 工具条toolbar

TBM 跟踪条trackbar

TCM Tab control

TTM Tooltip control

TVM 树视图tree\_view control

UDM Up\_down control

WM 一般窗口general window

此处需要注意：

BM/NM等中间的M表示向控件发送的消息，也就是命令控件做事情；

BN等中间的N，表示控件向外部（父窗口）发出的通知，表示自己的状态，不要混淆这两个不同方向的东西！

# 常用函数

Setdlgitemtext 该函数设置对话框中控件的文本和标题

BOOL[SetDlgltemText](http://baike.baidu.com/view/1080007.htm)(HWND hDlg,int nlDDlgltem,LPCTSTR IpString);

GetDlgItem(IDC\_BFFX\_DFY)->SetWindowText("123456.78");\*/

CWnd::SetWindowText 设置窗口的标题到指定的文本

void SetWindowText(

LPCTSTR lpszString

);

CEdit::ReplaceSel 调用此函数。lpszNewText指定的文本替换在编辑控件中当前选择

void ReplaceSel(

LPCTSTR lpszNewText,

BOOL bCanUndo = FALSE

);

lpszNewText

指向包含替换文本的一个Null终止的字符串。

bCanUndo

若要指定此功能可以取消，请将该参数的值设置为 TRUE。 默认值为 FALSE

## \_tcstoul

//将字符串转为16整形 1234 转成 4660 12转成18

int ieng = \_tcstoul(streng,0,16);

int x10=strtol(tmp,NULL,16);

## Mid

str=”quanshiwei”

tmp=str.Mid(2,6);

tmp=anshiw 有返回值

## format

str=“123456”；

tmp.Format("%x",str.Mid(2,4));

# 设置编辑框内容

/\*CWnd\* pWnd = GetDlgItem(IDC\_EDIT1);

pWnd->SetWindowText(\_T("Gerald Samper"));\*/

//GetDlgItem(IDC\_EDIT1)->SetWindowText(\_T("Gerald Samper"));

/\*CString str;

GetDlgItem(IDC\_EDIT1)->GetWindowText(str); //获取IDC\_EDIT1的值给str

MessageBox(str); \*/

SetDlgItemText(IDC\_EDIT1,\_T("Gerald Samper")); 将字符串显示到IDC\_EDIT1

# MFC中换行实现

在mfc中编辑框允许输入多行时，换行符被表示为<归位><换行>即“\r\n”，用ascii码表示为13 10

如果为编辑框中想要输入换行，就请将编辑框的属性：

Auto HScroll 设置为 False

MultiLine 设置为 True

Want Return 设置为 True

txt文本框中的换行实现

换行符被表示为<换行>即“\n”，用ascii码表示为10

# This function or variable may be unsafe

使用VS2005以上版本（VS2005、VS2008、VS2010）编译在其他编译器下正常通过的C语言程序，你可能会遇到类似如下的警告提示：

引用内容

warning C4996: 'strcpy': This function or variable may be unsafe. Consider using strcpy\_s instead. To disable deprecation, use \_CRT\_SECURE\_NO\_WARNINGS. See online help for details.

1> c:/program files/microsoft visual studio 10.0/vc/include/string.h(105) : 参见“strcpy”的声明

warning C4996: 'fopen': This function or variable may be unsafe. Consider using fopen\_s instead. To disable deprecation, use \_CRT\_SECURE\_NO\_WARNINGS. See online help for details.

1> c:/program files/microsoft visual studio 10.0/vc/include/stdio.h(234) : 参见“fopen”的声明

原因解释

这种微软的警告，主要因为那些C库的函数，很多函数内部是不进行参数检测的（包括越界类的），微软担心使用这些会造成内存异常，所以就改写了同样功能的函数，改写了的函数进行了参数的检测，使用这些新的函数会更安全和便捷。关于这些改写的函数你不用专门去记忆，因为编译器对于每个函数在给出警告时，都会告诉你相应的安全函数，查看警告信息就可以获知，在使用时也再查看一下MSDN详细了解。库函数改写例子：

mkdir改写为 \_mkdir

fopen”改写为 fopen\_s

stricmp改写为 stricmp\_s

strcpy改写为strcpy\_s

解决方案：

1> 根据下面的warning提示：参见“fopen”的声明

消息:“This function or variable may be unsafe. Consider using fopen\_s instead. To disable deprecation, use \_CRT\_SECURE\_NO\_DEPRECATE. See online help for details.”

所以可以将函数按warning提示的第二句，改为使用fopen\_s函数即可：

例如:FILE \*pFile=fopen("1.txt", "w");

改为：

FILE\* pFile;

fopen\_s(&pFile, "1.txt", "w");

2> 还是根据warning提示的地三句话:use \_CRT\_SECURE\_NO\_DEPRECATE

项目|属性|配置属性|C/C++|命令行|附加选项,加入【/D "\_CRT\_SECURE\_NO\_DEPRECATE" 】(注：加入中括号中完整的内容)

3> 降低警告级别：项目|属性|配置属性|C/C++|常规,自己根据情况降低警告级别（此法不推荐）

注意：高度重视警告：使用编译器的最高警告级别。应该要求构建是干净利落的（没有警告）。理解所有警告。通过 修改代码而不是降低警告级别来排除警告。

编译器是你的朋友。如果它对某个构造发出警告，这经常是说明你的代码中存在潜在的问题。成功的构建应该是无声无息的（没有警告的）。【《C++编程规 范》】

# char\* a= new char(10);

这个语句怎么解释？

在下只在学校的时候学过c++，工作中未用到过。

可是今天遇到这么个奇怪的语句。请高手指点一下。

------解决方案--------------------

等价于

char\* a= new char;

\*a=10;

------解决方案--------------------

生成一个字符，并初始化为10，如果要生成数组，应该为new char[10]

------解决方案--------------------

分配一个字符大小的内存,内存里放的是ANSI为10的字符

然后将内存地址给指针a

# 全局变量

\_\_argc 全局变量是传递给程序的命令行参数的数量计数。 \_\_argv 是一个指向包含程序参数的单字节字符或多字节字符字符串的数组的指针，\_\_wargv 是一个指向包含程序参数的宽字符字符串的数组的指针。这些全局变量提供了 main 或 wmain 参数。

Const char\*类型指针（指向常量的指针）其所指向的内容是常量，是不可以修改的，但其指针值是可以修改的。

char ch[5]=”list”;

const char\* pstr=ch;

\*pstr=”w”; error

pstr=”wangwu” //ok

435页

Char\* const类型指针（常量指针）来说，它的地址是一个常量，它的指针值是常量，但其所指向的内容可以修改。

char ch[5]=”list”;

char \* const pstr=ch;

pstr=”zhangsan” //error

\*pstr=”w”; //ok

# 指针

先定义数组，将指针指向数组首地址，防止数据溢出

避免如此定义 BYTE\* btNumber= new BYTE[8];要换成

BYTE btll[8];

btNumber=&btll[0];

# 字符集调试

/D "\_CRT\_SECURE\_NO\_DEPRECATE" /D "\_CRT\_NON\_CONFORMING\_SWPRINTFS"

# fstream.h

VS2010 C++编译时出现“ofstream”: 未声明的标识符

我提个建议哈，

#include <fstream>

using namespace std;

放到 stdafx.h 文件里面，然后其他地方就不要在写了

# EnableWindow(FALSE)

窗口禁用无效

# atof

将字符串转为浮点型 atoi可以用strtol替代， 而atof可以用strtod等替代

a=atof(str);

lin=(float)\_tstof(tmp);

b=atof(tmp);

c=a+b;

tmp.Format("%f",c);

SetDlgItemText(IDC\_EDIT2,tmp);

# ctype.

ctype.h是C标准函数库中的头文件，定义了一批C语言字符分类函数（C character classification functions），用于测试字符是否属于特定的字符类别，如字母字符、控制字符等等。既支持单字节（Byte)字符，也支持宽字符。

isalnum

# for 编辑框消息自动推送

for(i = 0;i < m;i++)

{ m\_yiducishu = i+1; UpdateData(FALSE);

搜索///////////////////////////

MSG msg;

while( PeekMessage( &msg, 0, 0, 0, PM\_REMOVE ) ) { TranslateMessage(&msg); DispatchMessage(&msg);}

///////////////////////////////}自己推动消息循环就ok了。

# 删除字符串空格

m=m.Trim(); //删除前后的空格

n=m.Replace(" ", "");//删除内部的空格

# 问题？

160308

1、数字编辑框怎么判断是否为空？ If( kongjian== 0)

2、手动创建控件？

3、“25600”字符串 怎么转化为 10进制 25600？

160309

1、串口提示中断 assert提示

2、进度条使用？

3、工具箱拉入一个控件，它的具体语句在哪里

160310

1、产生串口打开中断？ 没有关闭或者 打开失败

2、指针时产生 变量周围损坏？ 指针的地址，位数，清零的位数要一致，指针可重复初始化使用，也可以重新定义

3、生成带时间标志的文件，路径怎么设置？

出厂编程2只电表。DTSD631-VT,57.7V,1.A,0.2S,河南省统一编程