[1. 语法 2](#_Toc430706311)

[1-1. Afx\_msg 声明是否多余 2](#_Toc430706312)

[1-2. C、C++、MFC宏 2](#_Toc430706313)

[1-3. 字符和字符串 2](#_Toc430706314)

[1-3-1. C++ 定义字符串时，用L标示是ANSI字符还是Unicode字符 2](#_Toc430706315)

[1-3-2. \_T的作用 2](#_Toc430706316)

[1-3-3. c++ 字符类型总结区别wchar\_t,char,WCHAR 2](#_Toc430706317)

[1-3-4. Visual Studio 2013已经不支持非Unicode的工程的编译 3](#_Toc430706318)

[2. Project 3](#_Toc430706319)

[2-1. ListCtrlDemo 3](#_Toc430706320)

# 语法

## Afx\_msg 声明是否多余

afx\_msg void OnPaint();

貌似没啥用。不过查了网上的说明还是留下算了。

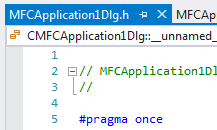
你可以在afxwin.h中找到如下的定义  
#ifndef afx\_msg  
#define afx\_msg         // intentional placeholder  
#endif  
就是什么也没有做  
看它的注释 intentinal placeholder 一个有意的占位符  
应该是为将来可能的升级所准备的  
所以 最好不要删除你现有程序中的afx\_msg

没什么意思.只是定义了这个符号而已. 这个对编译器来说,相当于什么都没有,对于人来说,我们可以看到这样的符号. 对于类向导来说.这个符号才是有意义的.它是一个消息处理函数的前缀. 类向导生成的消息函数,分发函数,事件响应函数都以这个为前缀. 如果去掉了,向导将不能识别。

## C、C++、MFC宏

[C、C++、MFC宏](Doc/C、C++、MFC宏.pdf)

### #pragma once



In the C and C++ programming languages, #pragma once is a non-standard but widely supported preprocessor directive designed to cause the current source file to be included only once in a single compilation.

## 字符和字符串

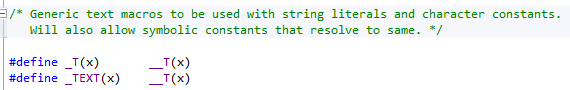
### C++ 定义字符串时，用L标示是ANSI字符还是Unicode字符

this->textBox1->Name = L"textBox1";

It's a wchar\_t literal, for extended character set. [Wikipedia](http://en.wikipedia.org/wiki/Wide_character) has a little discussion on this topic, and c++ examples.

### \_T的作用

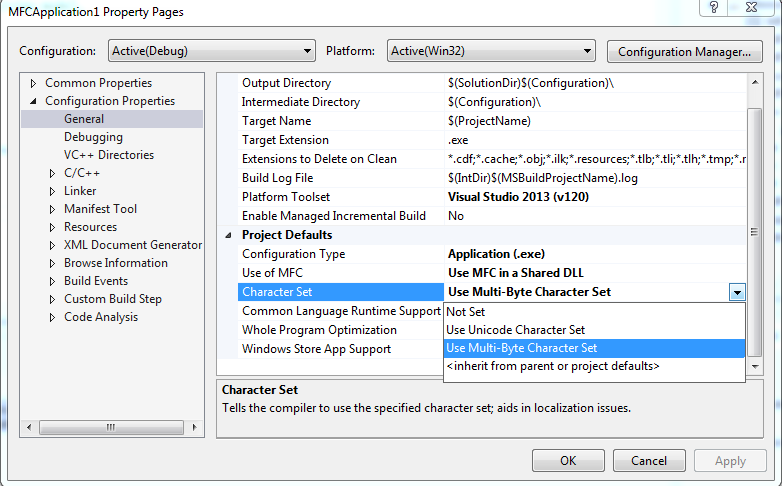
它其实是tchar.h里定义的一个宏



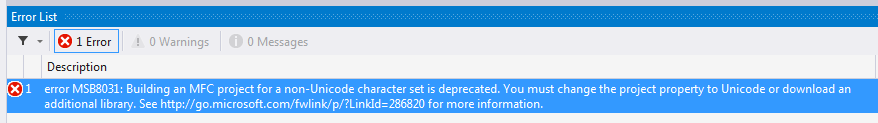
### c++ 字符类型总结区别wchar\_t,char,WCHAR

[c++ 字符类型总结区别wchar\_t,char,WCHAR](Doc/c++%20字符类型总结区别wchar_t,char,WCHAR.docx)

### Visual Studio 2013已经不支持非Unicode的工程的编译



Character Set选择 Use Multi-Byte Character Set时，编译提示出错



这也意味着今后的MFC工程，字符串最好定义成L开头的unicode或者用\_T(“fasdfad”) 之类的进行封装

# Project

## ListCtrlDemo

[CListCtrl](ListCtrlDemo/doc/CListCtrl.docx)