DLL:动态链接库

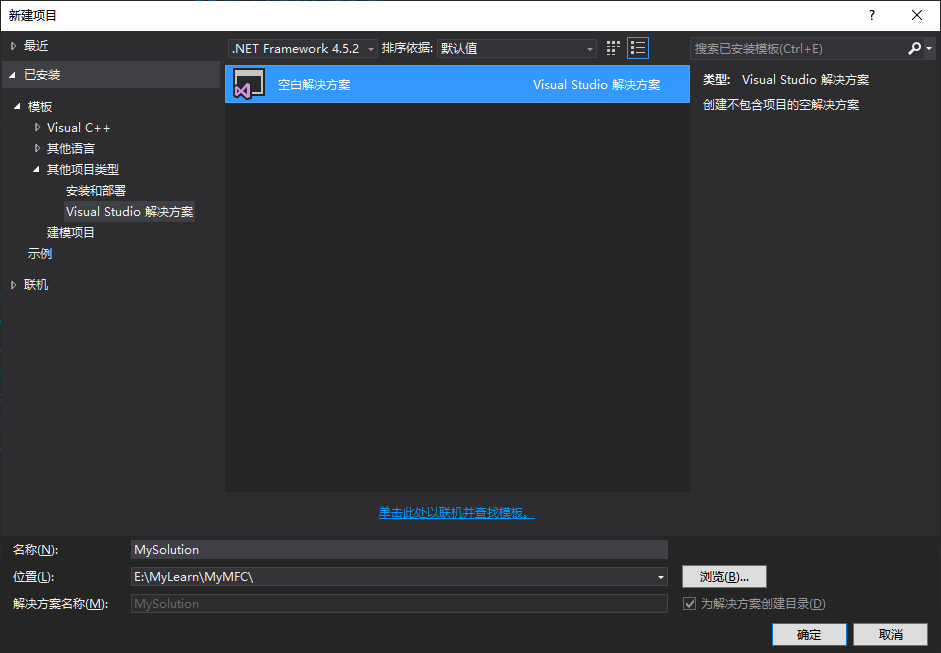
调用方式主要分为两种：

隐式（通过lib文件与头文件） 与 显式（只通过DLL文件）。

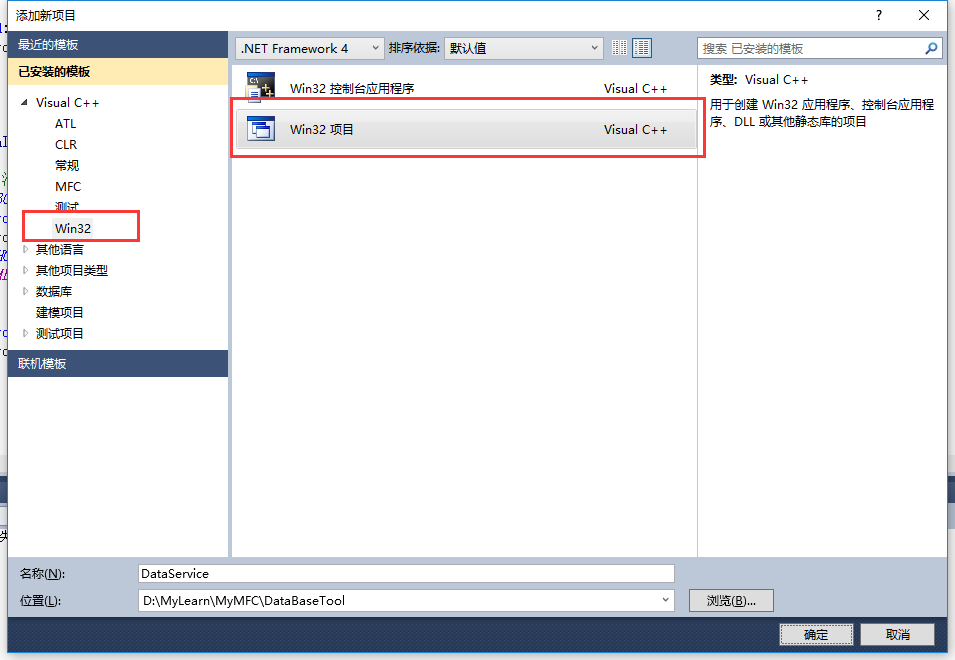
优势：

动态链接库DLL最大的好处就是节省内存

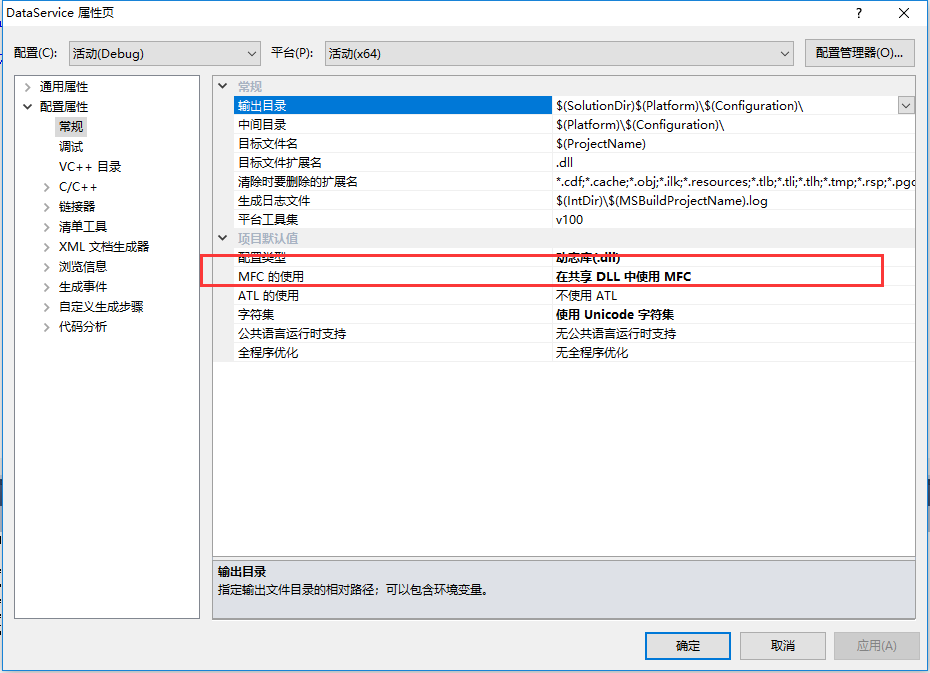
创建空白解决方案



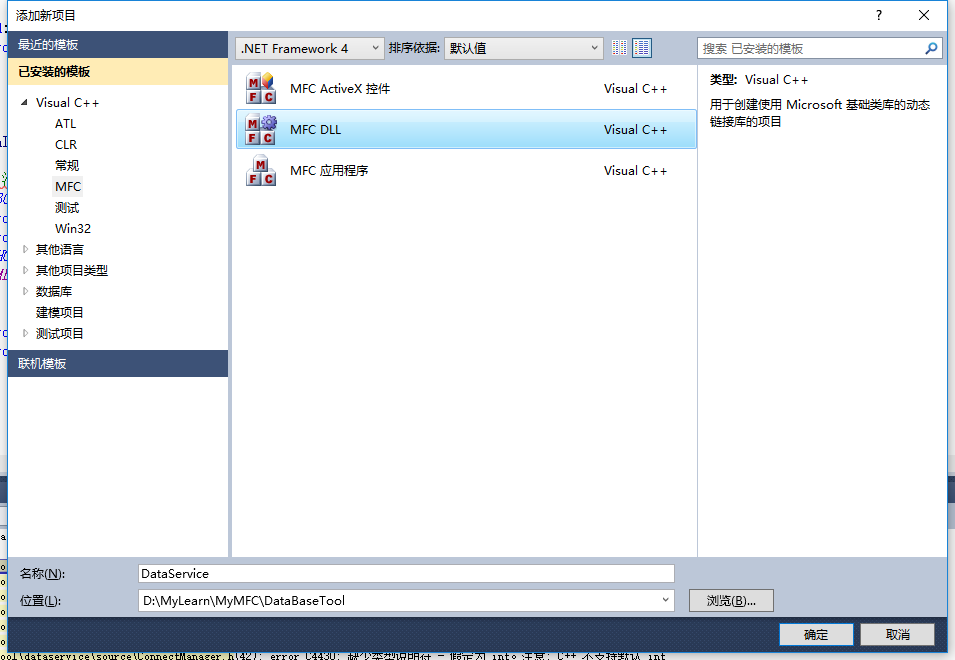
新建一个dll项目，用来导出一些公用的功能函数

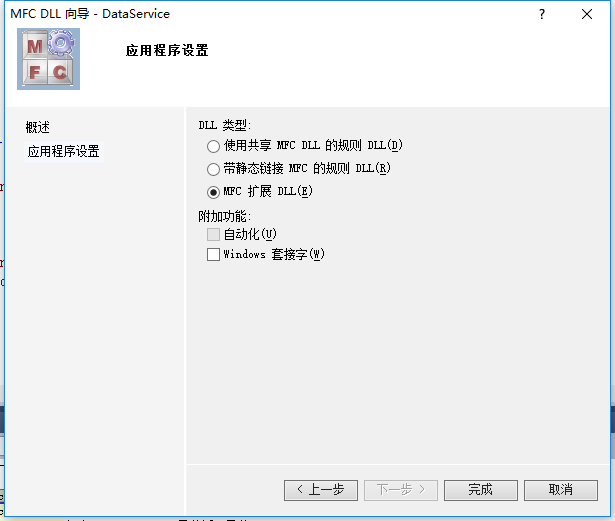






MFC下DLL使用





# 问题：

### 编译问题



解决：

配置项更改为x64位

### rror:预编译头文件来自编译器的早期版本，或者预编译头为 C++ 而在 C 中使用它(或相反)

这个问题是在VS项目中同时有.c和.cpp文件时，编译器会对它们采用不同的编译方式（主要因为函数声明的处理方式不同，C语言没有多态，函数名编译后比较正常，而C++有多态的特性，所以编译之后函数名面目全非）。

对于每个cpp文件，C++编译器是对其进行单独编译的。而对于文件中的#include包含的头文件等代码要进行文本替换，生成编译单元。而头文件可能被多个cpp文件包含，这样，每次编译cpp文件时，都要对头文件进行一次编译，也就是说，头文件被多次重复编译，浪费了编译时间。因此，C++编译器使用预编译（precompiled）的方式，预先把这些头文件进行编译，得到一个pch文件（precompiled header），当包含该头文件的cpp文件被编译时，只需使用该头文件对应的pch文件即可，无需再对头文件进行重复编译

三种解决方案：

1.最简单粗暴的办法就是整个项目都不使用预编译头。不过强烈不推荐，预编译头可以使编译时间大大减少，因而保留预编译头是很重要的。

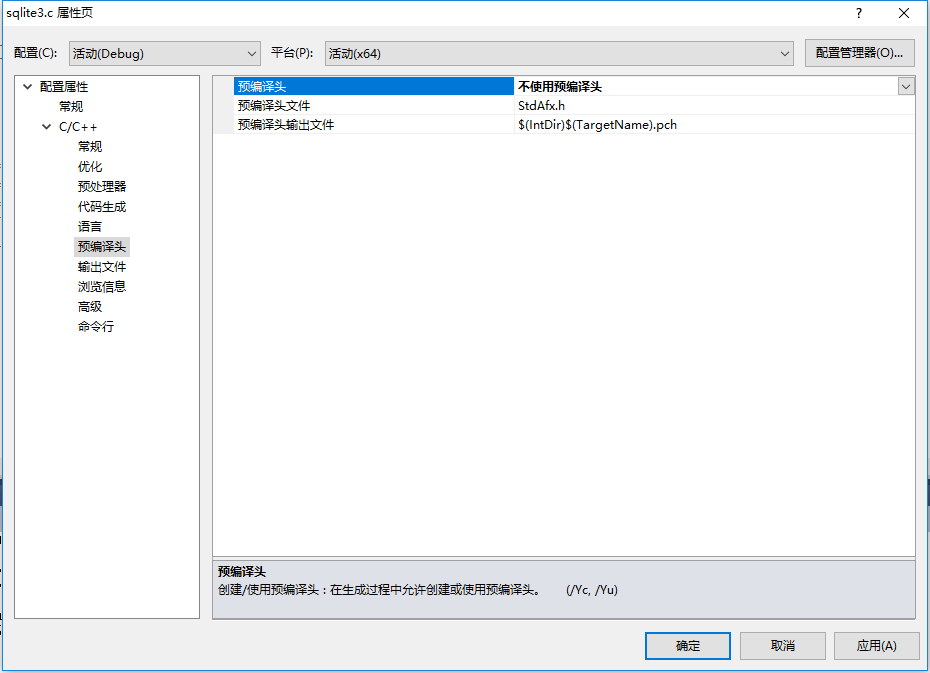
2.观察一下，看C语言还是C++的文件多，把少的那种文件的预编译头去掉。这种比较简单。

3.将c文件整个设置为一个项目，编译成dll或者lib动态载入，整个项目使用相同的预编译头。

解决：

由于这里只有一个c文件，所以就直接把.c文件的属性修改成不使用预编译头了。

关于C/C++混合编程还要注意的一个地方，C++要使用C语言的函数，或者引用C语言的头文件时，需要在前面加上extern "C"关键字。



# PmDbTool要依赖的项

PmBaseClass.dll

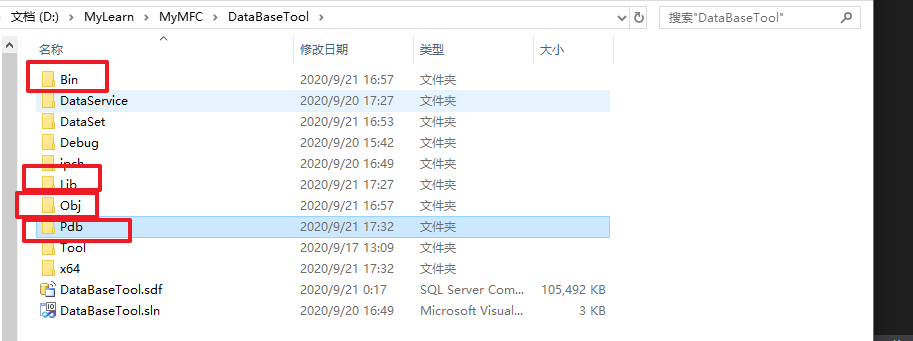
PmBaseOpr.dll

Dataset.dll

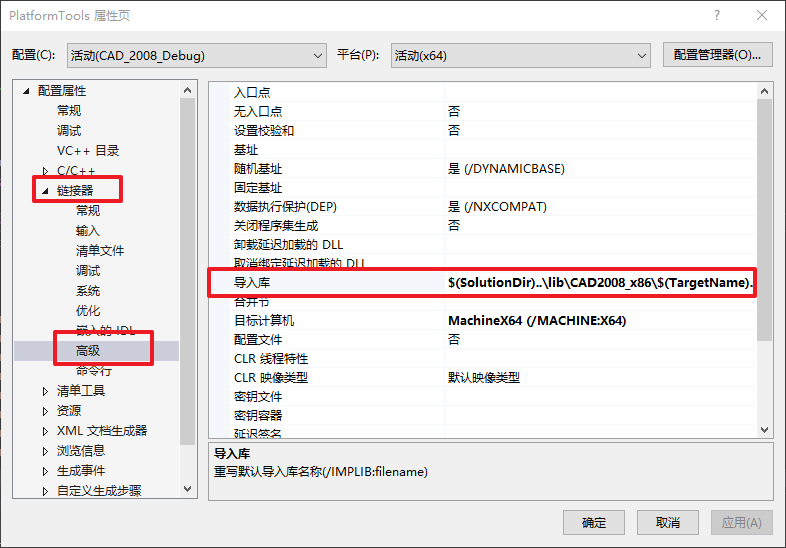
PM\_LogicDataOpr.dll

在预处理afx.h文件中定义预处理宏

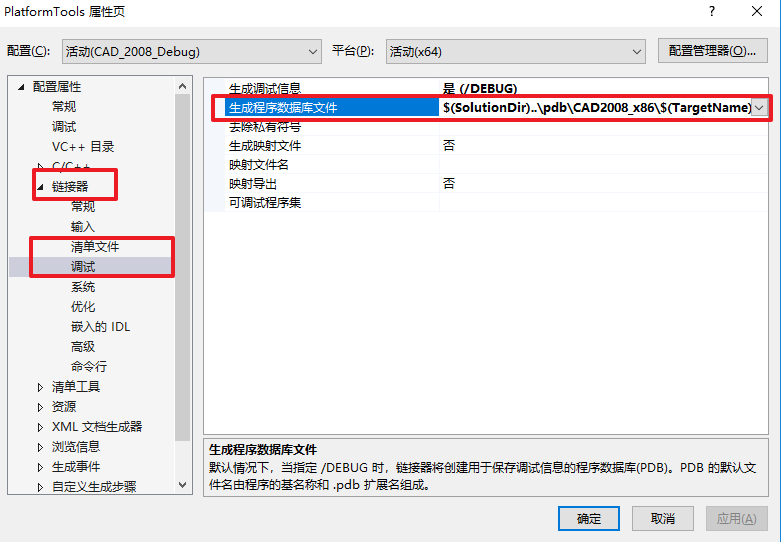
# 创建代码结构目录



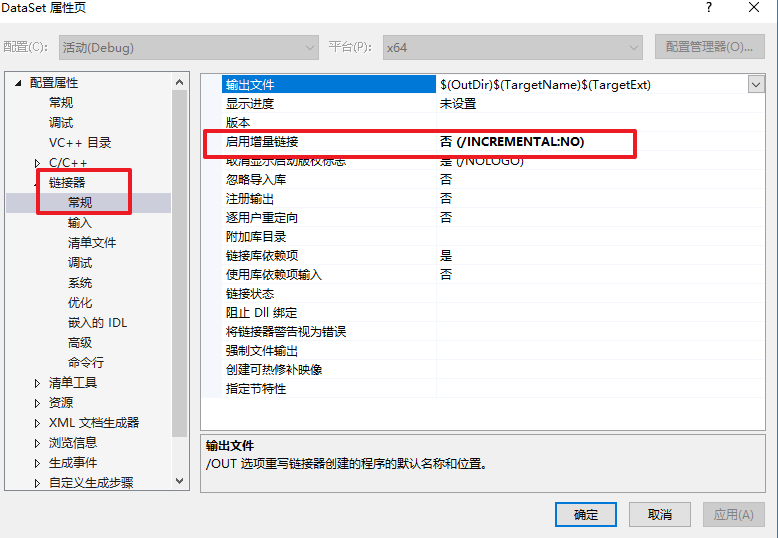
## VS DLL修改lib生成路径



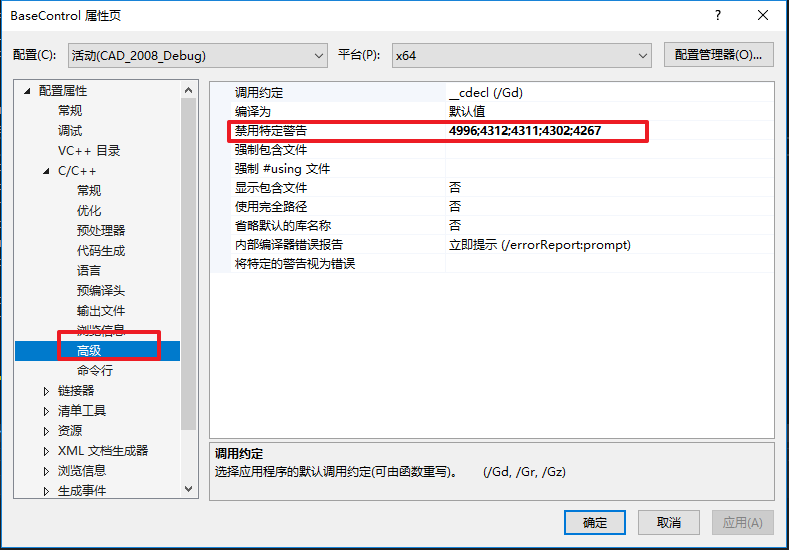
## 修改pdb导出路径

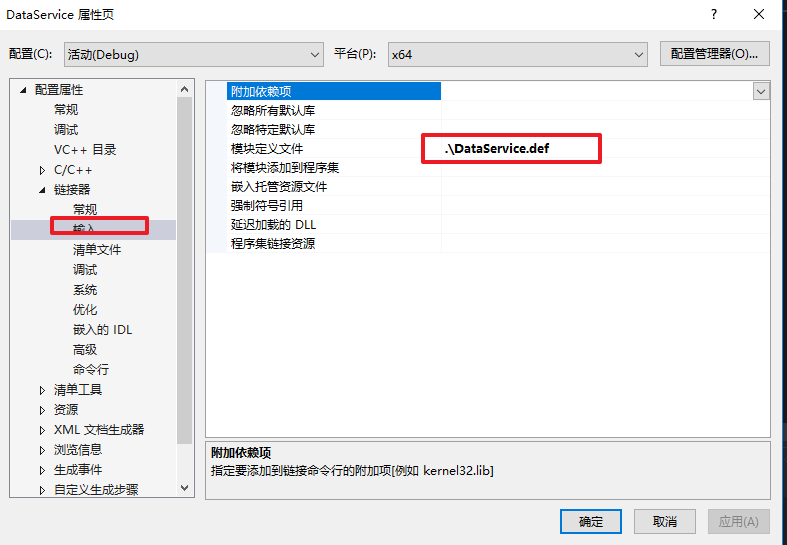


## 禁止生成ilk文件



## 禁用警告 #pragma disable 4996





默认库“MSVCRT”与其他库的使用冲突；请使用 /NODEFAULTLIB:library

在导出头文件中加入定义， 并链接lib文件

#ifdef DataSet\_Moudle

# define DataSetAPI \_\_declspec(dllexport)

#else

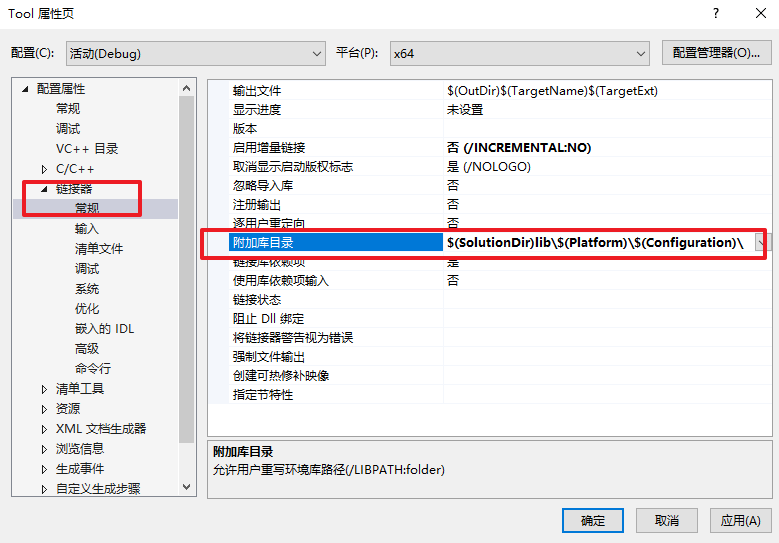
# define DataSetAPI \_\_declspec(dllimport)

#pragma comment(lib, "BaseControl.lib")

#endif

## Link链接Lib失败





# VS2010 VC++目录被否决

修改目录包含默认值方法：（区分x32, x64）

