解压SpeechSDK51.exe 、SpeechSDK51LangPack.exe，分别安装，记住SpeechSDK51.exe安装目录，博主安装在了默认路径下；安装msttss22L.exe

设置：   
1，属性–配置属性–C/C++–常规–附加包含目录：C:\Program Files\Microsoft SDKs\Speech\v11.0\Include（具体路径与安装路径有关）   
2，属性–配置属性–链接器–输入–附加依赖项：sapi.lib;

首先，属性，链接，预处理\_CRT\_SECURE\_NO\_WARNINGS

SAPI 包含sphelper.h编译错误解决方案 在使用Microsoft Speech SDK 5.1开发语音识别程序时，包含了头文件“sphelper.h”和库文件“sapi.lib”。编译时出错： 1>c:/program files/microsoft speech sdk 5.1/include/sphelper.h(769) : error C4430: missing type specifier - int assumed. Note: C++ does not support default-int 1>c:/program files/microsoft speech sdk 5.1/include/sphelper.h(1419) : error C4430: missing type specifier - int assumed. Note: C++ does not support default-int 1>c:/program files/microsoft speech sdk 5.1/include/sphelper.h(2373) : error C2065: 'psz' : undeclared identifier 1>c:/program files/microsoft speech sdk 5.1/include/sphelper.h(2559) : error C2440: 'initializing' : cannot convert from 'CSpDynamicString' to 'SPPHONEID \*' 1> No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called 1>c:/program files/microsoft speech sdk 5.1/include/sphelper.h(2633) : error C2664: 'wcslen' : cannot convert parameter 1 from 'SPPHONEID \*' to 'const wchar\_t \*' 1> Types pointed to are unrelated; conversion requires reinterpret\_cast, C-style cast or function-style cast 搜索了一圈，根据大家的经验汇总，应该是Speech代码编写时间太早，语法不严密。而VS2005对于语法检查非常严格，导致编译无法通过。修改头文件中的以下行即可正常编译：

 Ln 769 const ulLenVendorPreferred = wcslen(pszVendorPreferred);

             const unsigned long ulLenVendorPreferred = wcslen(pszVendorPreferred);

Ln 1418 static CoMemCopyWFEX(const WAVEFORMATEX \* pSrc, WAVEFORMATEX \*\* ppCoMemWFEX)

              static HRESULT CoMemCopyWFEX(const WAVEFORMATEX \* pSrc, WAVEFORMATEX \*\* ppCoMemWFEX)

Ln 2372 for (const WCHAR \* psz = (const WCHAR \*)lParam; \*psz; psz++) {}

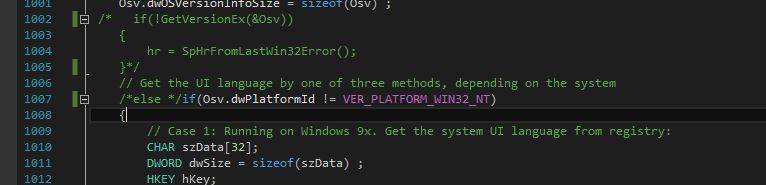
             const WCHAR \* psz; for (psz = (const WCHAR \*)lParam; \*psz; psz++) {}

Ln 2559 SPPHONEID\* pphoneId = dsPhoneId;

              SPPHONEID\* pphoneId = (SPPHONEID\*)((WCHAR \*)dsPhoneId);

Ln 2633 pphoneId += wcslen(pphoneId) + 1;

              pphoneId += wcslen((const wchar\_t \*)pphoneId) + 1; 好了，编译通过，下面可以正式编写程序逻辑了。

若不通过将图中sphelper.h头文件中的屏蔽掉：

# 例子程序：

#include <sphelper.h>//语音头文件

#include <iostream>//C++头文件，用来提示错误信息

int main()

{

::CoInitialize(NULL);//初始化语音环境

ISpVoice \* pSpVoice = NULL;//初始化语音变量

if (FAILED(CoCreateInstance(CLSID\_SpVoice, NULL, CLSCTX\_INPROC\_SERVER, IID\_ISpVoice, (void \*\*)&pSpVoice)))

//给语音变量创建环境，相当于创建语音变量，FAILED是个宏定义，就是来判断CoCreateInstance这个函数又没有成功创建语音变量，下面是不成功的提示信息。

{

std::cout << "Failed to create instance of ISpVoice!" << std::endl;

return -1;

}

pSpVoice->Speak(L"Hello,world!", SPF\_DEFAULT, NULL);//执行语音变量的Speek函数，这个函数用来读文字。

pSpVoice->Release(); //释放语音变量

::CoUninitialize();//释放语音环境

return 0;

}

#include <sapi.h>

#include <sphelper.h>

#include <spuihelp.h>

#pragma comment(lib,"ole32.lib") //CoInitialize CoCreateInstance需要调用ole32.dll

#pragma comment(lib,"sapi.lib") //sapi.lib在SDK的lib目录,必需正确配置

#include<iostream>

int main()

{

::CoInitialize(NULL);//初始化语音环境

ISpVoice \* pSpVoice = NULL;//初始化语音变量

CoCreateInstance(CLSID\_SpVoice, NULL, CLSCTX\_INPROC\_SERVER, IID\_ISpVoice, (void \*\*)&pSpVoice);

//给语音变量创建环境，相当于创建语音变量，FAILED是个宏定义，就是来判断CoCreateInstance这个函数又没有成功创建语音变量，下面是不成功的提示信息。

pSpVoice->Speak(L"speek english ", SPF\_DEFAULT, NULL);//执行语音变量的Speek函数，这个函数用来读文字。

pSpVoice->Release(); //释放语音变量

::CoUninitialize();//释放语音环境

return 0;

}