



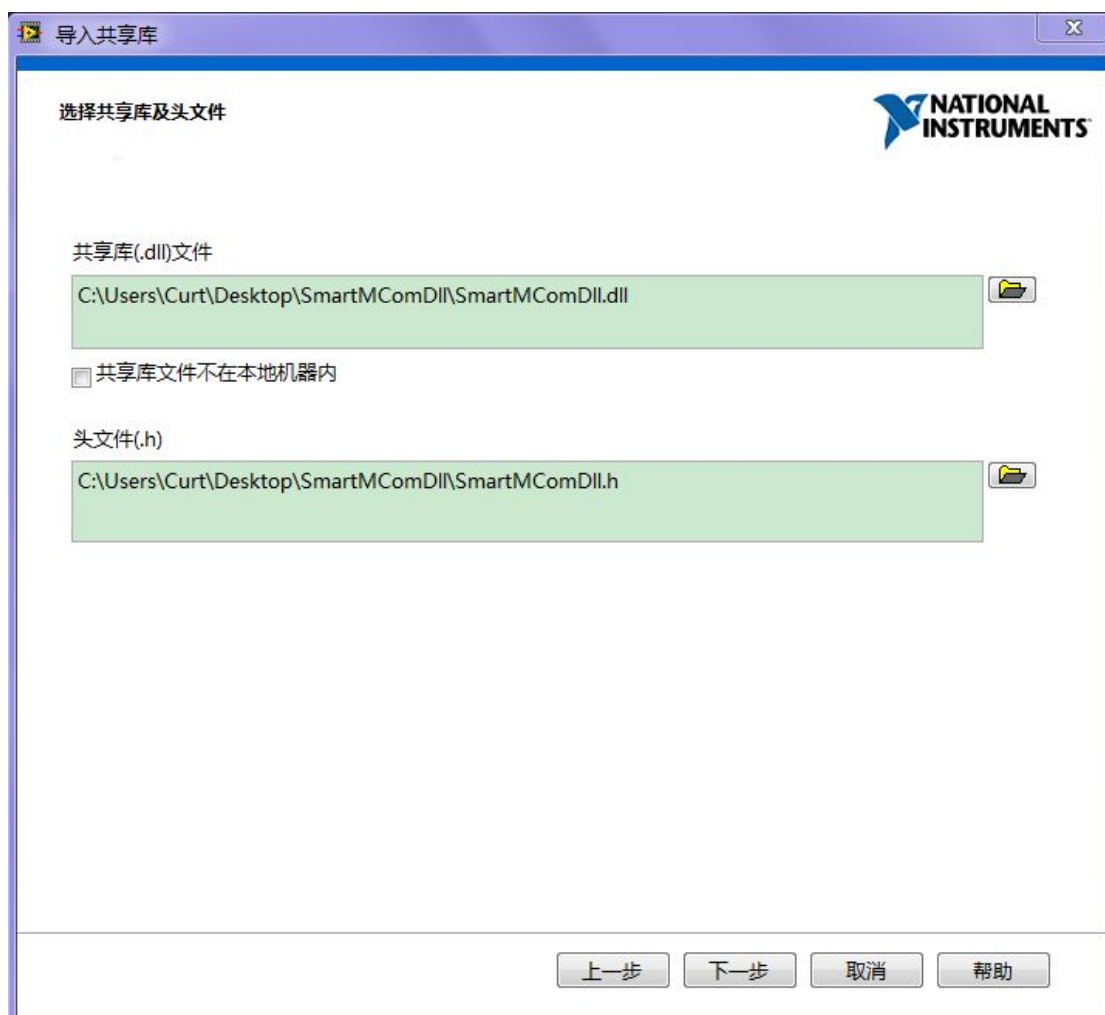
## LabVIEW 编程调用 DLL 时遇见的问题

对于使用 LabVIEW 编程的人来说，都知道 LabVIEW 功能强大，但也会发现似乎缺少某些功能，而像其他编程语言的 DLL、ActiveX 组件则能提供。

在使用 DLL( 动态链接库 )时 ,最大的困难就是把函数参数的数据类型映射为相应的 LabVIEW 中的数据类型。LabVIEW 提示：

未定义符号可能会造成函数和参数无法被识别。如要解决该问题，检查头文件并确定是否必须添加预定义符号。单击上一步按钮返回至向导的前一页并添加预定义符号（例如，"NIAPI\_stdcall = \_stdcall"或"NIAPIDefined = 1"）

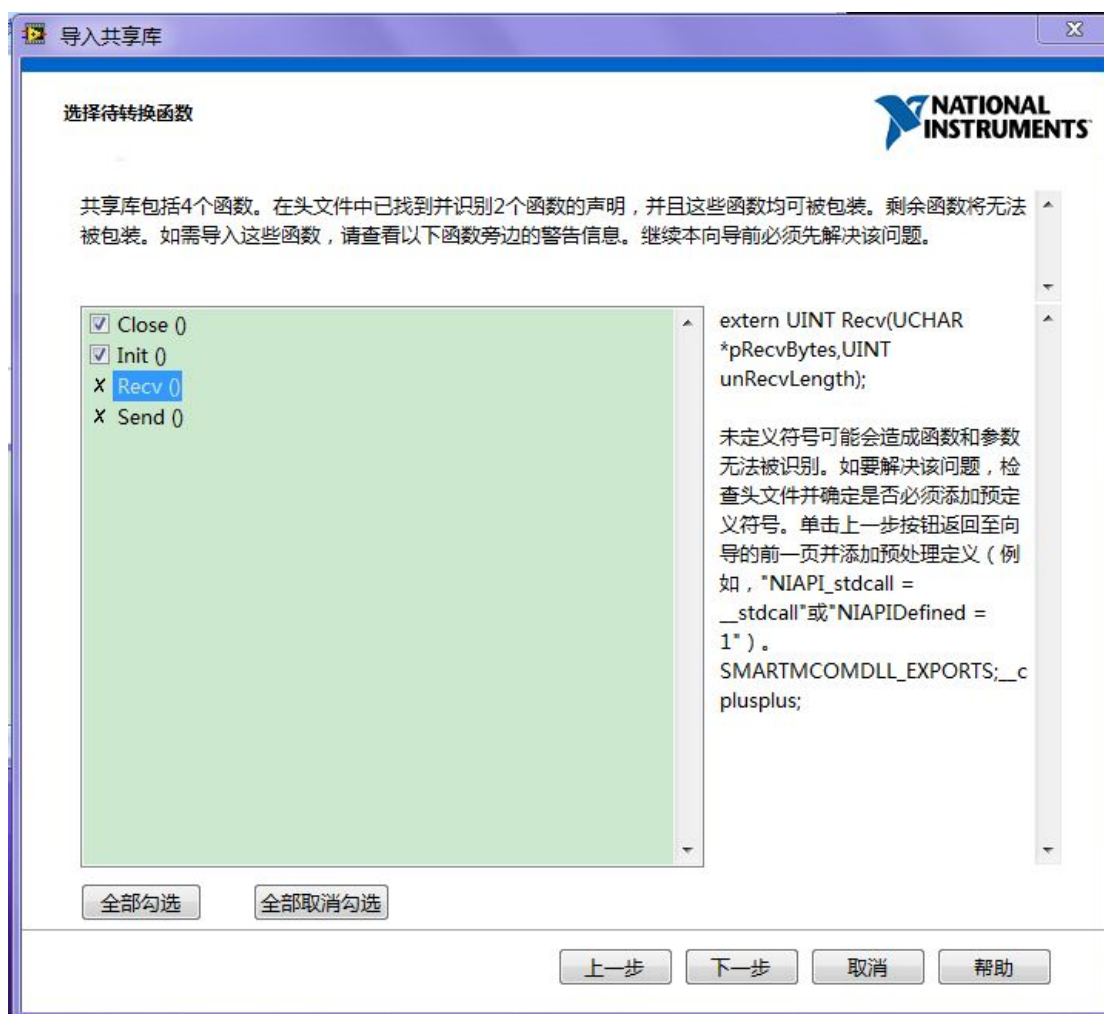
在一次进行串口调试软件的编写过程中，要求使用动态链接库（其实使用 VISA 同样可以实现），使用导入共享库工具以自动生成配置 CLN（Call Library Function）节点，工具在“工具—导入—共享库（.dll）”菜单项下，专门用于将 DLL 中函数包装成 VI。



下一步下一步点击解析头文件后，依然遇见（如下图）

extern UINT Recv(UCHAR \*pRecvBytes,UINT unRecvLength); ( 函数原型 )

未定义符号可能会造成函数和参数无法被识别。如要解决该问题，检查头文件并确定是否必须添加预定义符号。单击上一步按钮返回至向导的前一页并添加预定义符号（例如，"NIAPI\_stdcall = \_stdcall"或"NIAPIDefined = 1"）



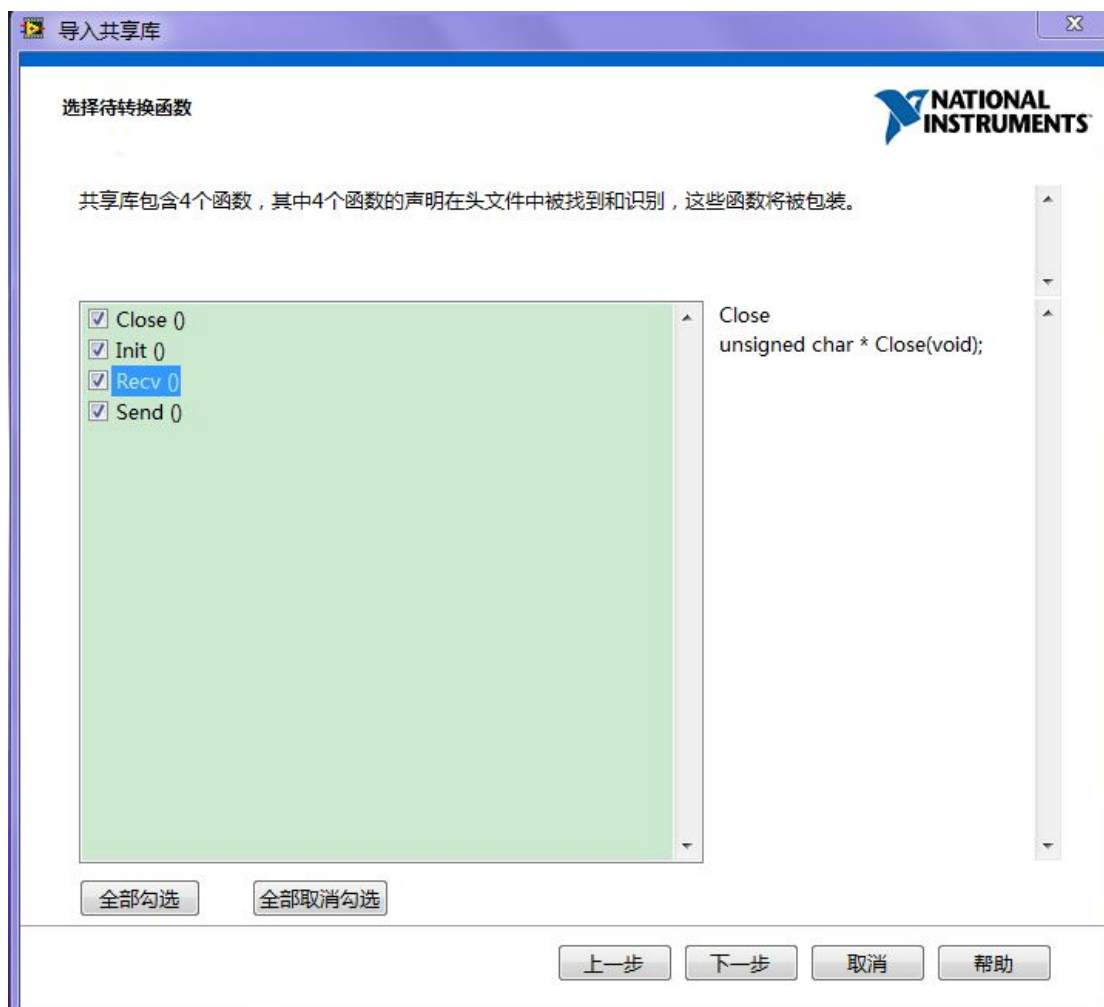
归咎原因就是头文件中的一些类型定义不符合标准 C 语法，而使解析器无法获得正确的 mPWIN32\_COMMAND 定义。DLL 函数的头文件中可能使用了某个系统定义的数据类型 数据类型的定义在 windows.h 中 ( windows.h 是 Windows SDK 的一个文件 ,VC 等开发环境中常常带有 Windows SDK ) ,要正确解析必须得到这些数据类型 ,也就是找到 windows.h 这个头文件，用户须把 windows.h 文件的全路径加在“包括路径”中。例如 Visual C++ 6.0 编译环境中头文件位于安装目录下 VC98 文件夹下的 Include 文件中。

而“预处理定义”中，当用户需要写一些宏定义，那么就写在这个位置。

在网络上看到在预处理定义中添加了如下代码

```
ULONG = unsigned long; VOID = void; LONG = long; UCHAR = unsigned char; PCHAR = unsigned char*;
PULONG = unsigned long*; WINAPI; BOOL = bool; USHORT = unsigned short; PUSHORT = unsigned short*;
```

点击下一步也不会遇到上述问题，导入工具会帮助你解析出全部导出函数并自动生成 VI。



在安装 LabVIEW 的 user.lib 文件夹下就会找到各个函数的 VI。

LabVIEW 自带一个用于演示的示例 ,D:\Measurement studio\LabVIEW 8.6\examples\dll\regexpr\Import Shared Library Tutorial GUI.vi ( 假设 LabVIEW 安装于 D 盘目录下 ) , 用户可以打开示例自行学习这个工具。