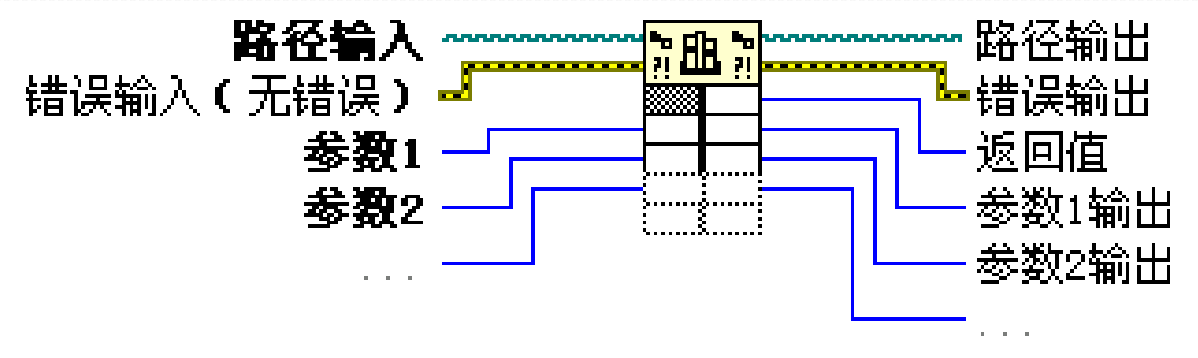
第10章 LabVIEW常用外部接口

10.1 调用库函数

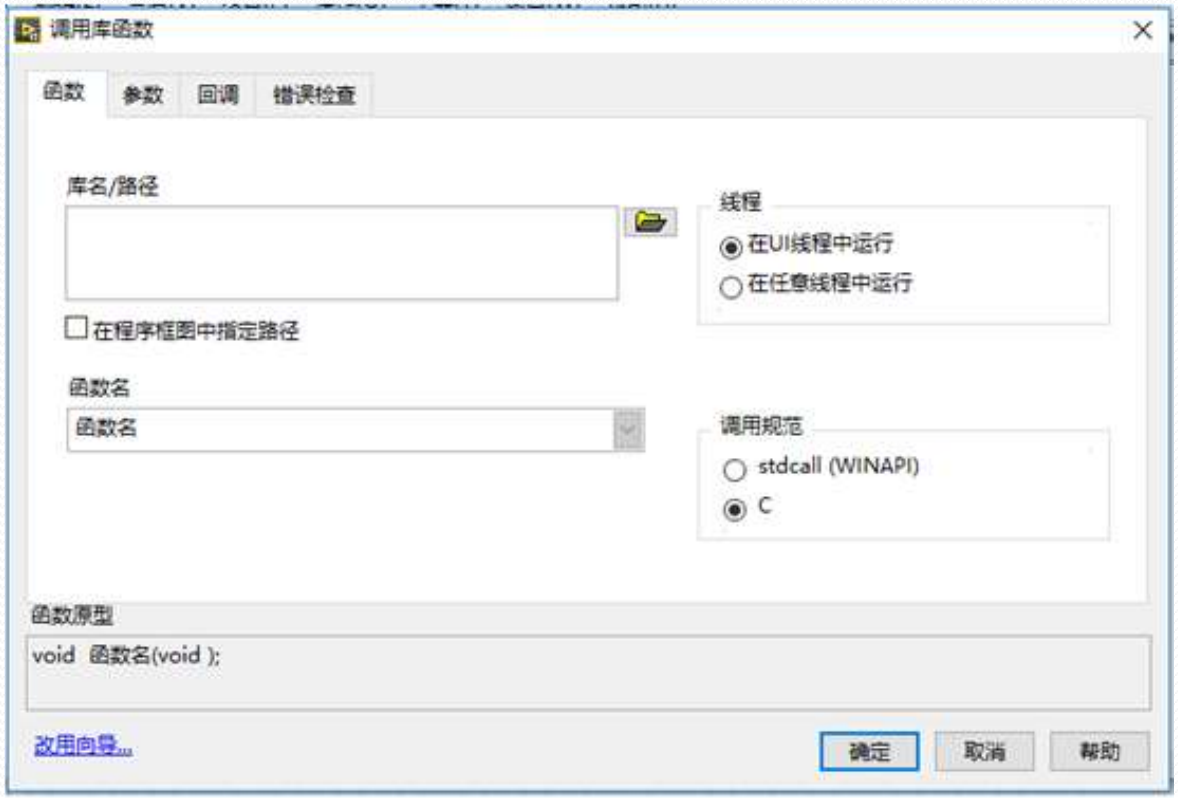
- LabVIEW作为一个强大的工具，可以调用其他很多专业的软件，实现更为复杂的LabVIEW自身不能够单独实现的功能，同时，它还可以实现在应用程序之间进行通信，把LabVIEW运算得到的数据传送给其他程序，或者用LabVIEW接收其他程序传来的数据。

- **10.1.1 DLL简介**
- **1.DLL的概念**
- 由于LabVIEW是以图形化的方式编程，它所提供的对象功能有限，因此在实际应用的过程中难免会遇到LabVIEW提供的对象不能解决的问题。所以LabVIEW提供了C语言接口及直接调用动态链接库(DLL)函数的功能。
- DLL是建立在客户端/服务器通信的概念上，包含若干个函数、类或资源的库文件，函数和数据被存储在一个DLL服务器上，并由一个或多个客户导出使用，这些客户可以是应用程序，或者是其他的DLL。DLL库不同于静态库。在静态库情况下，函数和数据被编译进—个二进制文件(通常扩展名为“.Lib”)，对应Visual C++的编译器，它在处理程序代码时将从静态库中恢复这些函数和数据，并把它们和应用程序中的其他模块组合在一起，生成可执行文件，这个过程称为“静态链接”，此时，因为应用程序所需的全部内容都是从库中复制出来的，所以静态库本身并不需要与可执行文件一起发行。
- 在动态链接的情况下，有两个文件，一个是引人库(“.Lib”)文件，一个是DLL文件。引入库文件包含从DLL导出的函数的名称和位置，DLL包含实际的函数和数据。应用程序使用“.hb”文件链接到所需要使用的DLL文件，库中的函数和数据并不复制到可执行文件中，因此，在应用程序的可执行文件中，存放的不是被调用的函数代码，而是在DLL中所要调用的函数的内存地址。这样，当一个或多个应用程序运行时，再把程序代码和被调用的函数代码链接起来，从而节省了内存资源。

- **2.调用DLL**
- 在LabVIEW中，可通过调用库函数模块调用动态链接库。在框图程序编辑窗口下，选择功能模板中的互连接口→库与可执行程序→调用库函数节点，放置在框图程序编辑窗口中，其图标如图10.1所示。



- 双击此函数模块，将弹出调用库函数对话框，如图10.2所示。在此可以对要调用的动态链接库及函数进行设定。



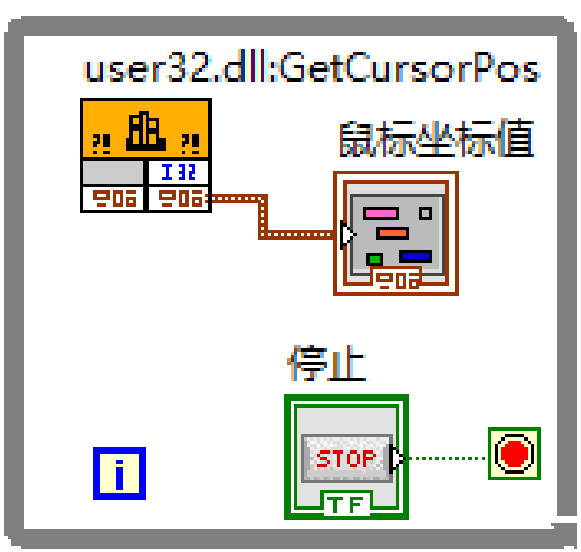
- 将节点放置在程序框图中，双击会出现它的配置对话框，共有四页。第一页用于填写被调用函数的信息Library name or path（库名/路径）需给出DLL文件名和路径，若引用操作系统路径下的DLL文件，直接输入文件名也可调用，其它的必须输入全路径。在这里已经给出名字的DLL是被静态加载到程序中的，也就是说当调用了这个DLL的VI被装入内存时，DLL同时被装入内存。LabVIEW也可动态加载DLL，只要勾选上Specify path on diagram(在程序框图中指定路径)的选项即可。选择了这个选项，在 Library name or path（库名/路径）中输入的内容就无效了，取而代之的是CLN 节点多出一对输入输出，用于指明所需要使用的DLL的路径。这样，当VI被打开时，DLL不会被装入内存，只用程序运行到需要使用这个DLL中的函数时，才把其装入内存。Function name是需要调用的函数的名称，LabVIEW会把DLL中所有的暴露出来的函数都列出，用户只要在下拉框中选取即可。Thread栏用于设定哪个线程里运行被调用的函数。用户可以通过 CLN 节点的配置面板来指定被调用函数运行所在的线程。CLN 的线程选项非常简单，只有两项： Run in UI thread和Run in any thread。LabVIEW的程序框图上直接可以看出一个 CLN节点是选用的什么线程。如果Run in UI thread，节点颜色是橙色的；Run in any thread则是浅黄色的。
- 单击确定按钮后，就可以将该调用库函数节点当做一个具有该动态链接库中被调函数功能的节点来使用。
- 对已经编译生成的DLL文件，可以按上述方法直接调用。另外，LabVIEW还提供了编写动态链接库的C源代码框架，利用此框架编写动态链接库的步骤如下所述。

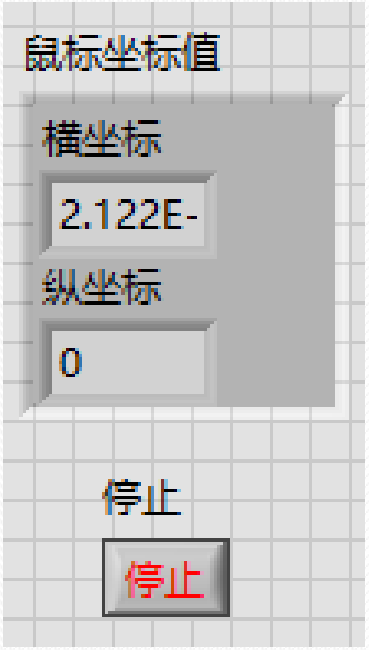
- **1.生成C源代码框架**
- 在框图程序编辑窗口创建调用库函数节点快捷菜单,选择Create.c File，这时弹出一个文件对话框，在其中键入文件名（例如labview_DLL.c），然后用文本编辑器打开保存的文件，LabVIEW在文件中已经生成了如下的框架。
- /* Call Library source file */
- #inc1ude "extcode.h"
- void funcName(void);
- void funcName(void)
- {
- /* Insert code here*/
- }

- **2.添加源代码**
- 在生成的C源代码框架中/* Insert code here*/处加人用户需要的函数体，框架中的函数名和返回类型应与用户在图10.2所示的调用库函数对话框中设置的一致。添加完源代码后保存文件即可。
- **3.编译生成DLL**
- 将保存的C源代码文件(LabVIEW DLL.c)放到VC++集成环境下进行编译，将生成DLL文件，这样就可以在LabVIEW中直接调用LabVIEW_DLL.DLL中的函数了。

- **10.1.2 API简介**
- 应用程序编程接口（Application Programming Interface，API）是一套用来控制Windows的各个部件（从桌面的外观到为一个新进程分配的内存）的外观和行为的一套预先定义的Windows函数。用户的每个动作都会引发一个或几个函数的运行以告诉Windows发生了什么，这使得它很像是Windows的天然代码，而其他的语言只是提供一种能自动而且更容易访问API的方法。如果用户用VB写了一段代码，则当这些代码在Windows环境中运行时，每行代码都会被VB转换为API函数传递给Windows，如VB中的Form1 .Print函数将会以一定的参数调用TextOut这个API函数。
- 同样，当用户单击窗体上的一个按钮时，Windows会发送一个消息给窗体（这对于用户来说是隐藏的），VB获取这个调用并经过分析后生成一个特定事件（Button_Click）。
- Windows系统下的API函数位于Windows系统目录下的动态链接库文件中(如User32.dll、GDl32.dll、She1132.dll...)，因此在LabVIEW中调用API函数和调用动态链按库的方法是一致的。具体的API函数的功能、原型以及参数等，用户可以查阅专门介绍API函数的相关书籍。

- **10.1.3 调用库函数实例**
- 图10.3给出了一个调用库函数的示例程序框图，图中“调用库函数节点”函数的主要设置为：



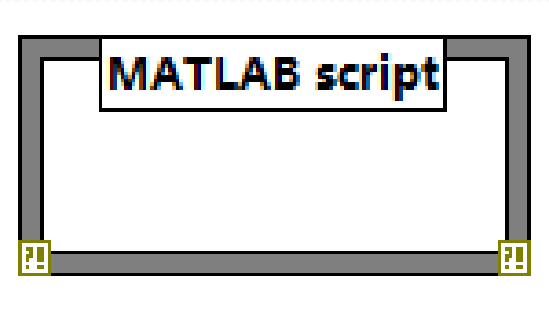


- （1）在“库名或路径”中选择的库文件为user32.dll，该文件可在“WINDOWS→system32”目录下找到，用户也可以将其复制到任何希望的路径。
- （2）在“函数名”下啦框中选择的函数为GetCursorPos函数，或直接输入函数名。
- （3）在“线程”中选择为“在UI线程中使用”。
- （4）在“调用规范”中选择为“stdcall（WINAPI）”调用。
- （5）在“参数”页中单击+添加一个参数，并命名为lpPoint，设置类型为“匹配至类型”，选择数据格式为“按值处理”。命名添加参数名为lpPoint，是因为在前面所选的GetCursorPos函数声明中已经定义了参数lpPoint。
- （6）单击“确定”按钮退出配置属性对话框后会发现“调用库函数节点”函数图标的“返回值”端口中显示为I32，说明返回值的数据类型为I32。
- 图中的“鼠标坐标值”是一个簇，包含有两个数值控件，用于分别显示鼠标在屏幕上所处位置的横、纵坐标值。运行该程序后可以发现，无论鼠标是否在VI的前面板上，当移动鼠标时，鼠标的坐标值都会随着鼠标的移动而变化，并始终显示鼠标当前的位置。

10.2MATLAB接口

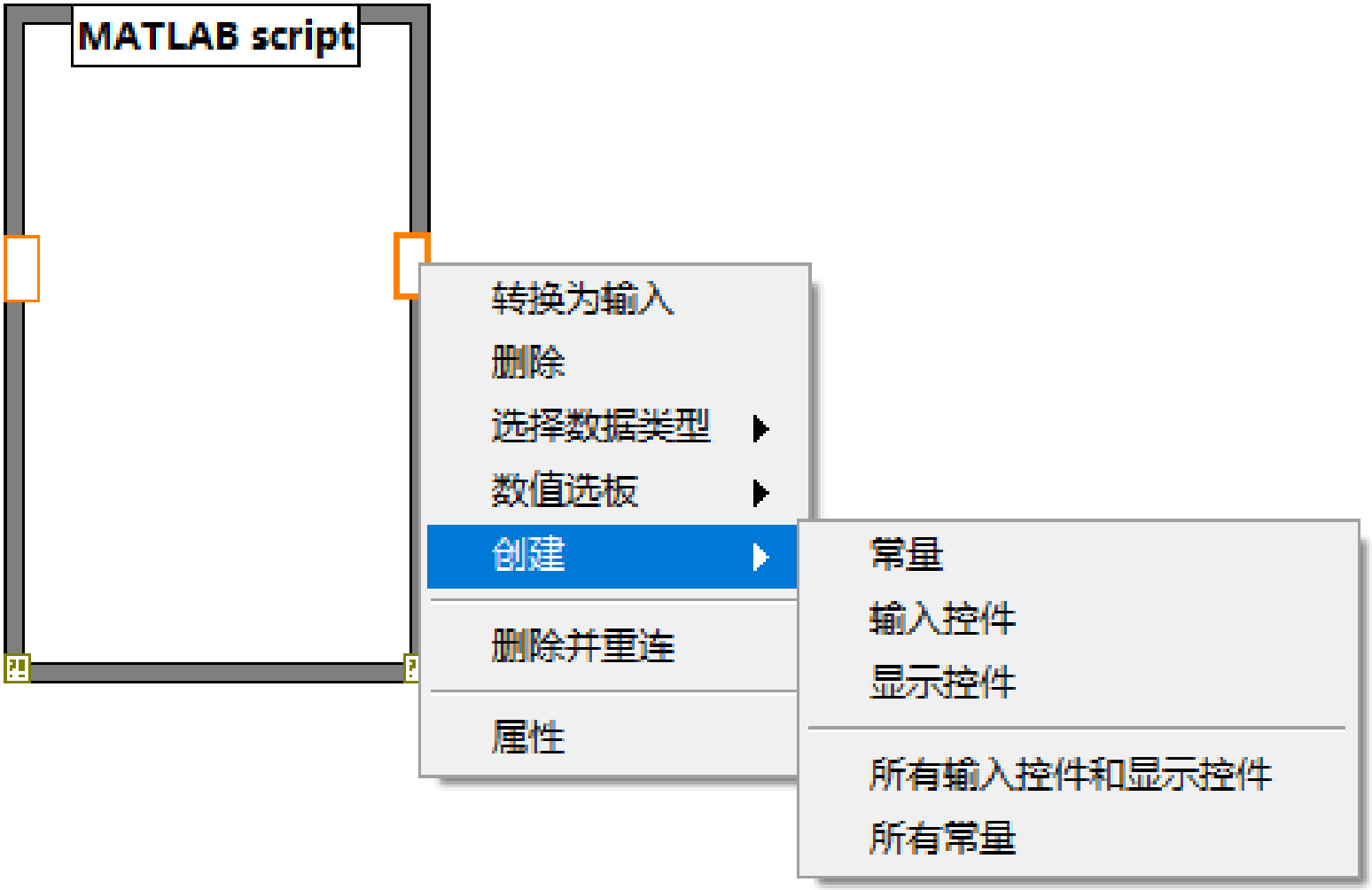
- LabVIEW中的公式节点可以实现一些基本的数学运算，如果涉及比较复杂的数学运算，则可以调用Matlab的脚本文件(“.m”文件)来实现。Matlab是一种在科学和工程可视化计算领域中广泛流行的交互式编程环境。在LabVIEW中调用Matlab的功能是通过Matlab节点来实现的。
- 注意：要使用Matlab节点，必须首先安装Mat1ab，因为Mat1ab节点要调用Mat1ab脚本服务器。

- **10.2.1 Matlab节点**
- Matlab节点位于“函数”选板下“数学→脚本与公式→脚本节点”子模板内，添加Matlab节点的方式和前面章节中添加公式节点的方式类似，如图10.4和图10.5所示。

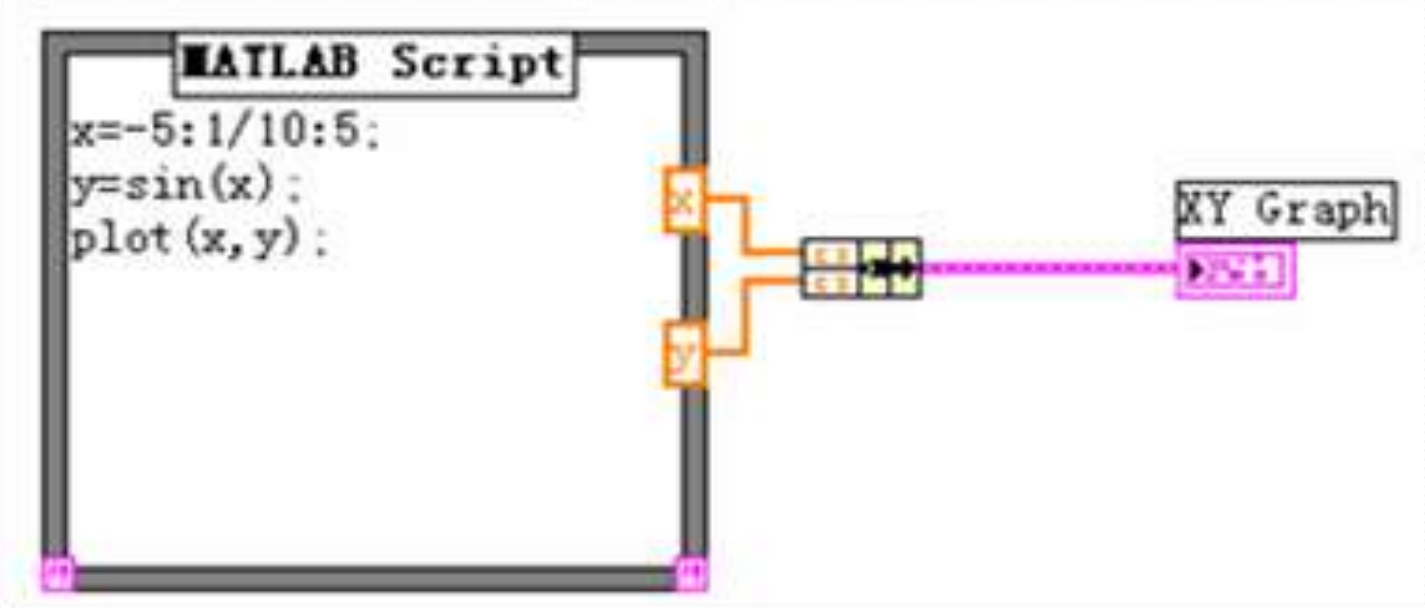


- 向Matlab节点输人Matlab文件的方法有两种：一种是直接在框内写人文件代码，另一种是在Matlab节点的边框上单击鼠标右键，在弹出的快捷菜单中选导入...，在弹出的对话框中输入Matlab脚本文件即可，如图10.6所示。输人/输出变量的添加和公式节点类似，分别用添加输入和添加输出来实现。
- 和公式节点一样，用户可以为Matlab节点的每个输人/输出变量添加指示器或者控制器，如图10.6所示。

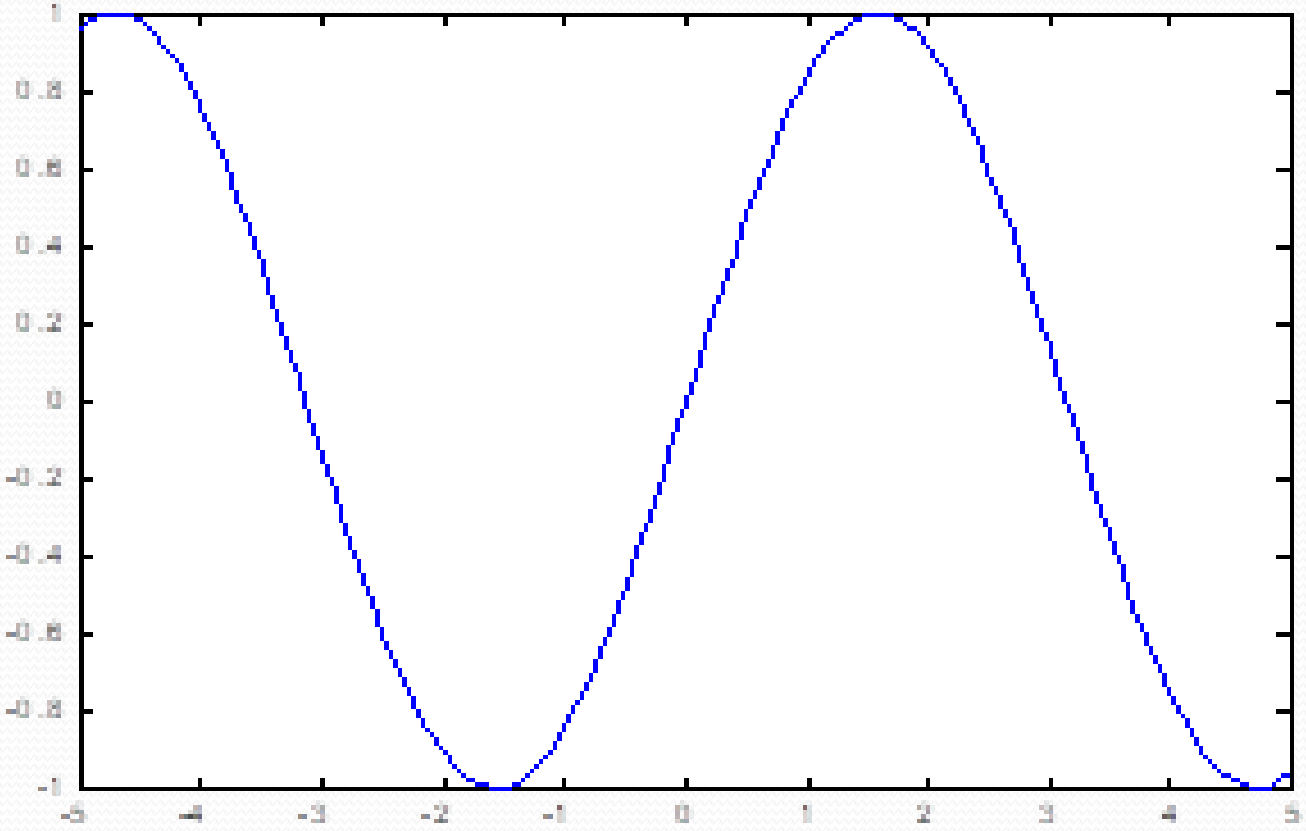

MATLAB script
显示项
帮助
范例
说明和提示...
断点
添加输入
添加输出
在服务器中编辑
导入...
导出...
清除脚本
选择脚本服务器
删除并重连
创建
属性


MATLAB script
转换为输入
删除
选择数据类型
数值选板
创建
删除并重连
属性
常量
输入控件
显示控件
所有输入控件和显示控件
所有常量

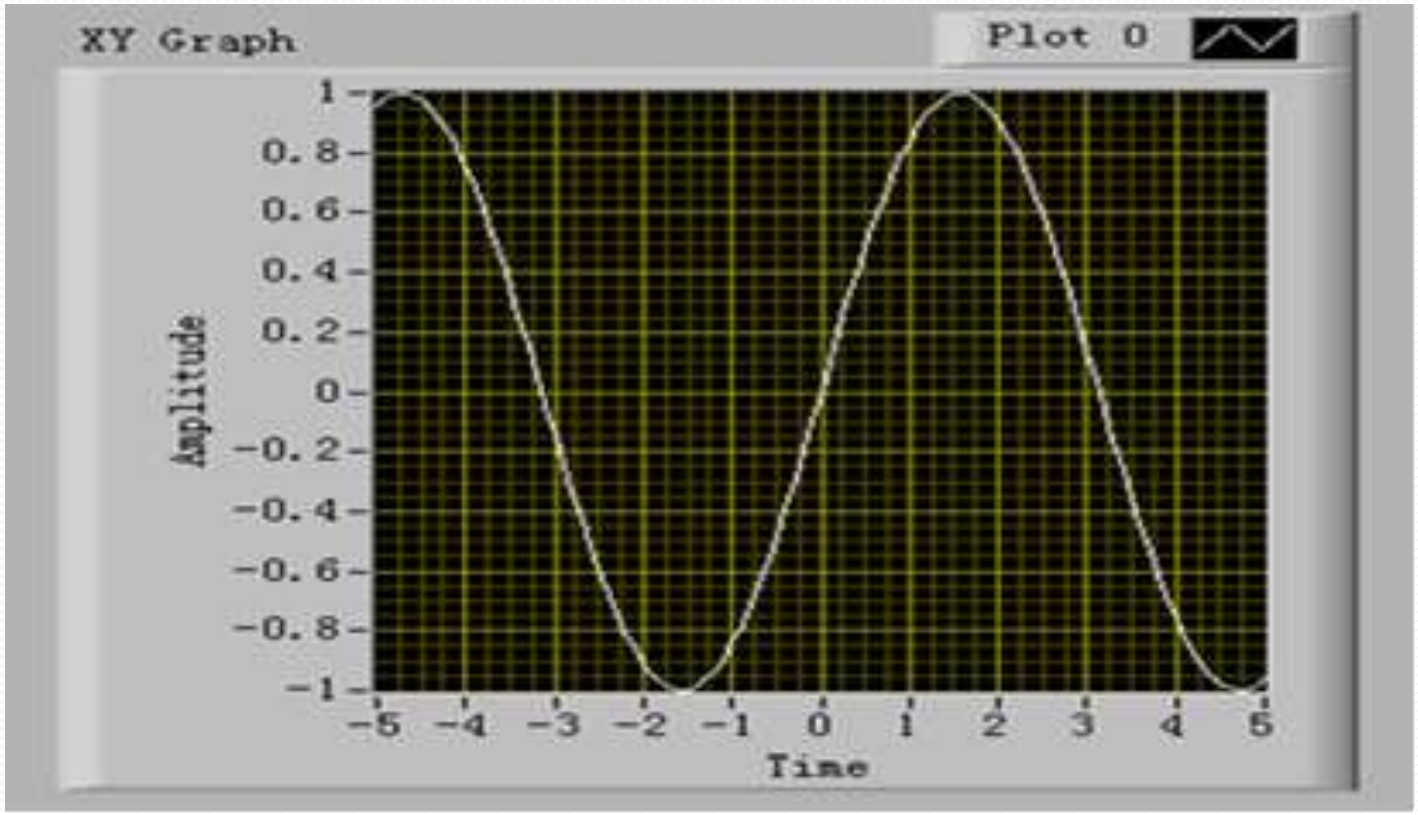
- **10.2.2 Matlab节点应用**
- 本节通过一个简单实例来说明Matlab节点是如何工作的。
- 【例10.1】 画一段正弦函数图像。
- 在Matlab节点内写人如下代码：
- X=-5:1/10:5;
- y=sin(x);
- plot(x,y);
- x是给定的一维数组向量，包含从-5~5的均匀分布的101个数据，类型是双精度浮点型；y是把x进行正弦运算后得到的结果；plot是Matlab中画二维图像的命令，它以x变量为x轴，y变量为y轴画出的结果。其框图程序如图10.8所示。


MATLAB Script
x=-5:1/10:5;
y=sin(x);
plot(x,y);
x
y
XY Graph

- 运行后，LabVIEW的Matlab节点调用Matlab服务器(相当于在Matlab环境下运行“.m”文件)，画出的图形是一个正弦函数图像，如图10.9所示。



- 为了便于比较，把x变量和y变量都定义为输出变量，类型在LabVIEW中为一维数组，用Bundle函数绑定成簇，然后用XY图形控件显示。LabVIEW中画出的正弦函数图如图10.10所示。可以看出，两者是一致的。


XY Graph
Plot 0
Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-5 -4 -3 -2 -1 0 1 2 3 4 5
Time

- 用户在实际使用中，Matlab代码可以具有非常复杂的功能，虽然LabVIEW的Matlab节点可以调用Matlab的功能，但是调试环境是不一样的，用户最后在Matlab环境下把程序调试成功后，再输入到LabVIEW的Matlab节点中。

10.3 ActiveX

- **10.3.1 ActiveX简介**
- ActiveX是Microsoft公司提出的一组使用部件对象模型(Component Object Model，CoM)使软件部件在网络环境中进行交互的技术集。它与具体的编程语言无关，即CoM是跨越语言的操作系统及标准。它定义了对象之间的存取方法，不同的应用程序可以各自开发出一系列公共对象，如控件、函数等。它们有开放的属性和方法，允许其他应用程序访问，而不同的开发平台在互相调用对象时只需要载入对象所在的EXE或DLL文件即可，对象的代码并不存在于主程序中。ActiveX采用客户机/服务器模式进行不同应用程序的链接，调用其他应用程序的对象时，这个应用程序被称作客户端；而自己创建的对象被其他应用程序调用时，这个应用程序被称作服务器。LabVIEW既可以作为ActiveX 采用客户端，又可以作为ActiveX服务器使用。

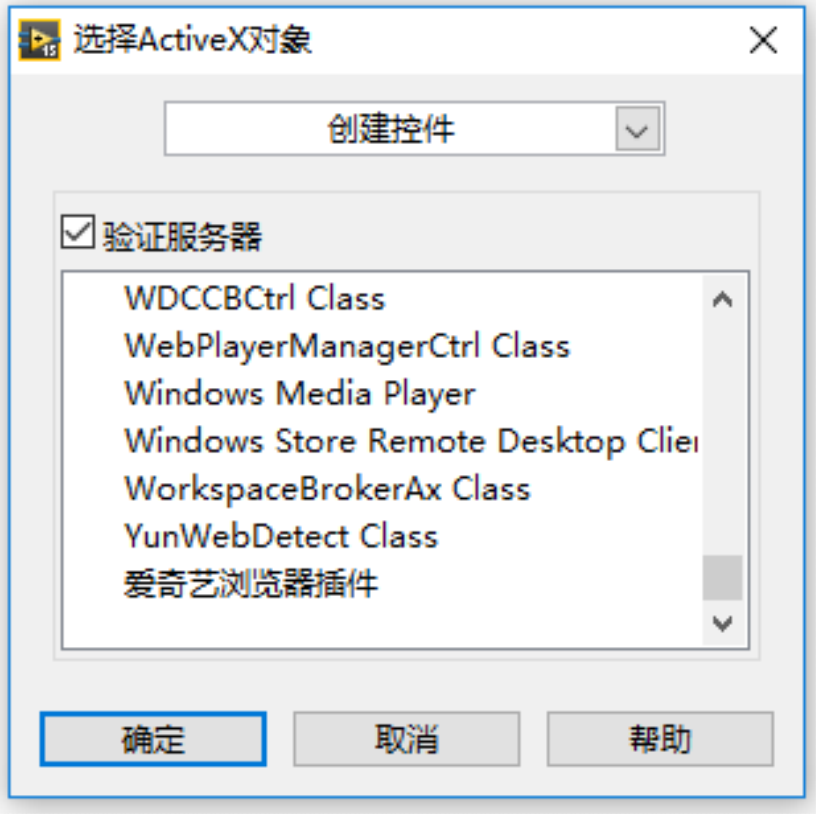
- ActiveX 既包含服务器端技术，也包含客户端技术。其主要由下列几部分构成：
- 1)ActiveX控件(ActiveX Control)：用于向Web页面、Microsoft Word等支持ActiveX的容器(Container)中插入COM对象。
- 2)ActiveX文档(ActiveX Document)：用于在Web浏览器或者其他支持ActiveX的容器中浏览复合文档(非HTML文档)，如Microsoft Word文档、Microsoft Excel文档或者用户自定义的文档。
- 3)ActiveX自动化：用于在不同的应用中传递数据和命令。
- 4)ActiveX脚本描述(ActiveX Scripting)：用于从客户端或者服务器端操纵ActiveX和Java程序传递数据，并协调它们之间的操作。

- **10.3.2 ActiveX控件**
- ActiveX控件具有独立的外观显示，因此需要要调用它的编程语言必须提供相应的容器，作为ActiveX的显示窗口。LabVIEW中的ActiveX控件模板位于“容器”模板中，如图10.11所示。利用ActiveX容器函数，用户可以调用第三方提供的各种ActiveX控件。

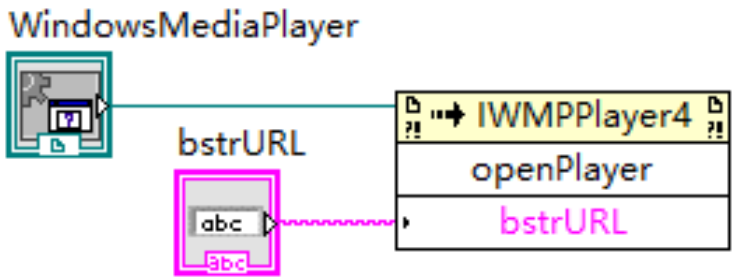

控件
搜索
自定义
新式
容器
水平分隔栏
垂直分隔栏
.NET容器
选项卡控件
子面板
ActiveX容器


ActiveX容器

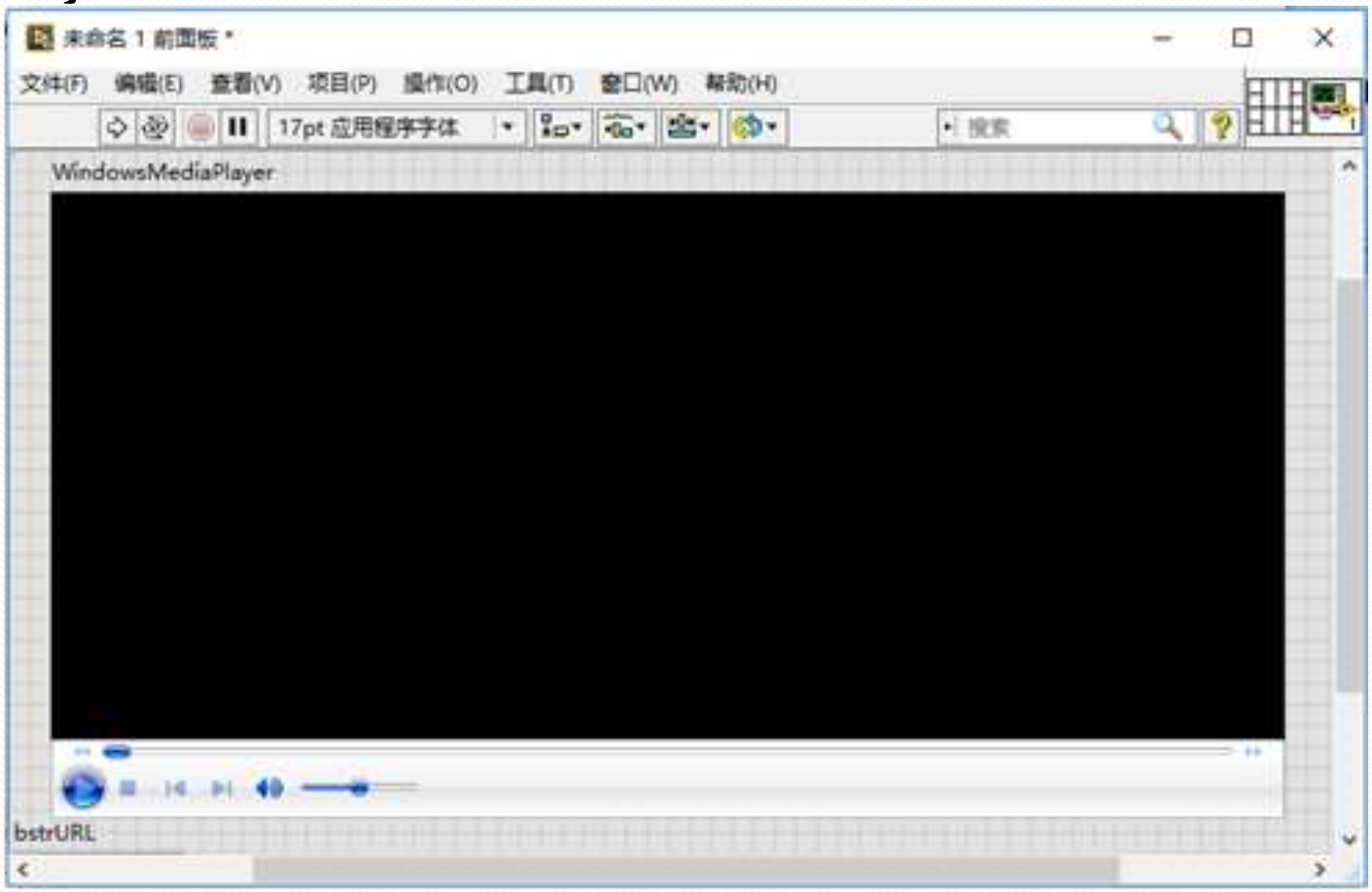
- 将ActiveX容器函数拖放到前面板中，鼠标右键单击函数图标，从弹出的快捷菜单中选择“插入ActiveX对象”项，将打开一个如图10.12所示的名为“选择ActiveX对象”的对话框。



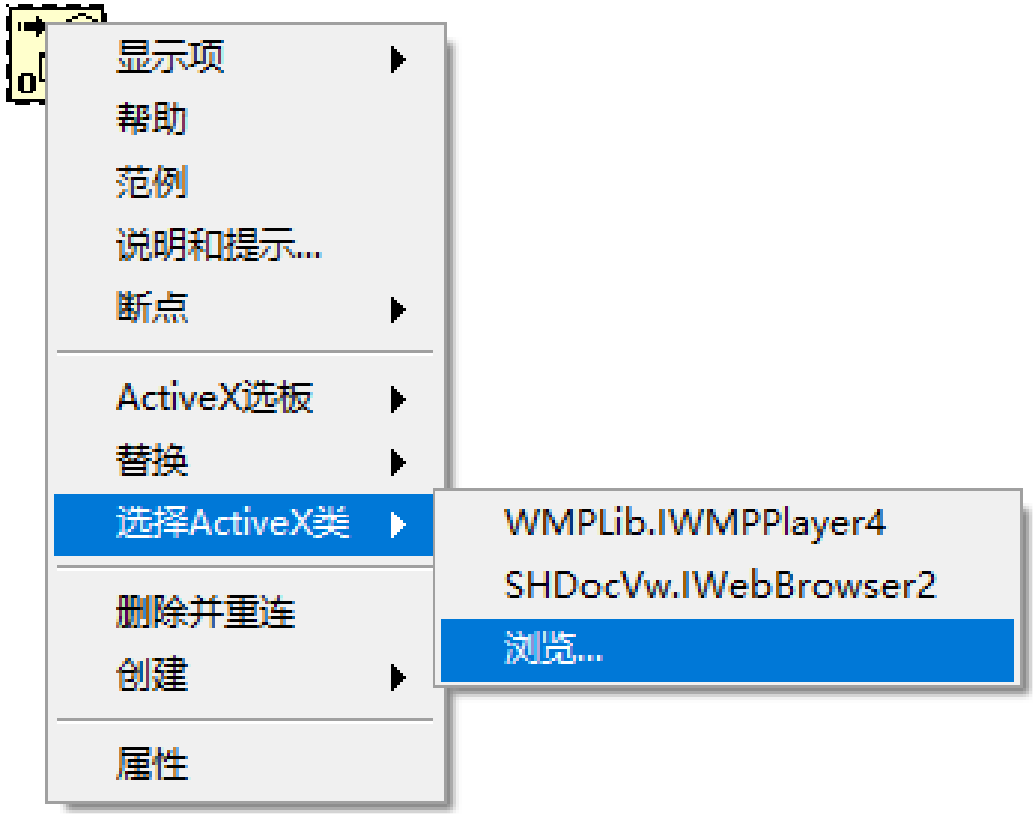
- 若想在LabVIEW中调用Windows Media Player，可在如图10. 12所示的对话框中选择Windows Media Player，此时在前面板中可以看到一个类似于Media Player播放器的界面。此时在程序框图中，可以用鼠标右键单击控件图标，从弹出的快捷菜单中选择“ActiveX选板”下的“调用节点”函数，将Windows Media Player控件的数据输出端口有函数的数据输入端口相连，并选择方法为openPlayer，同时在下面的数据输入端口bstrURL创建一个路径输入控件，程序框图如图10.13所示。



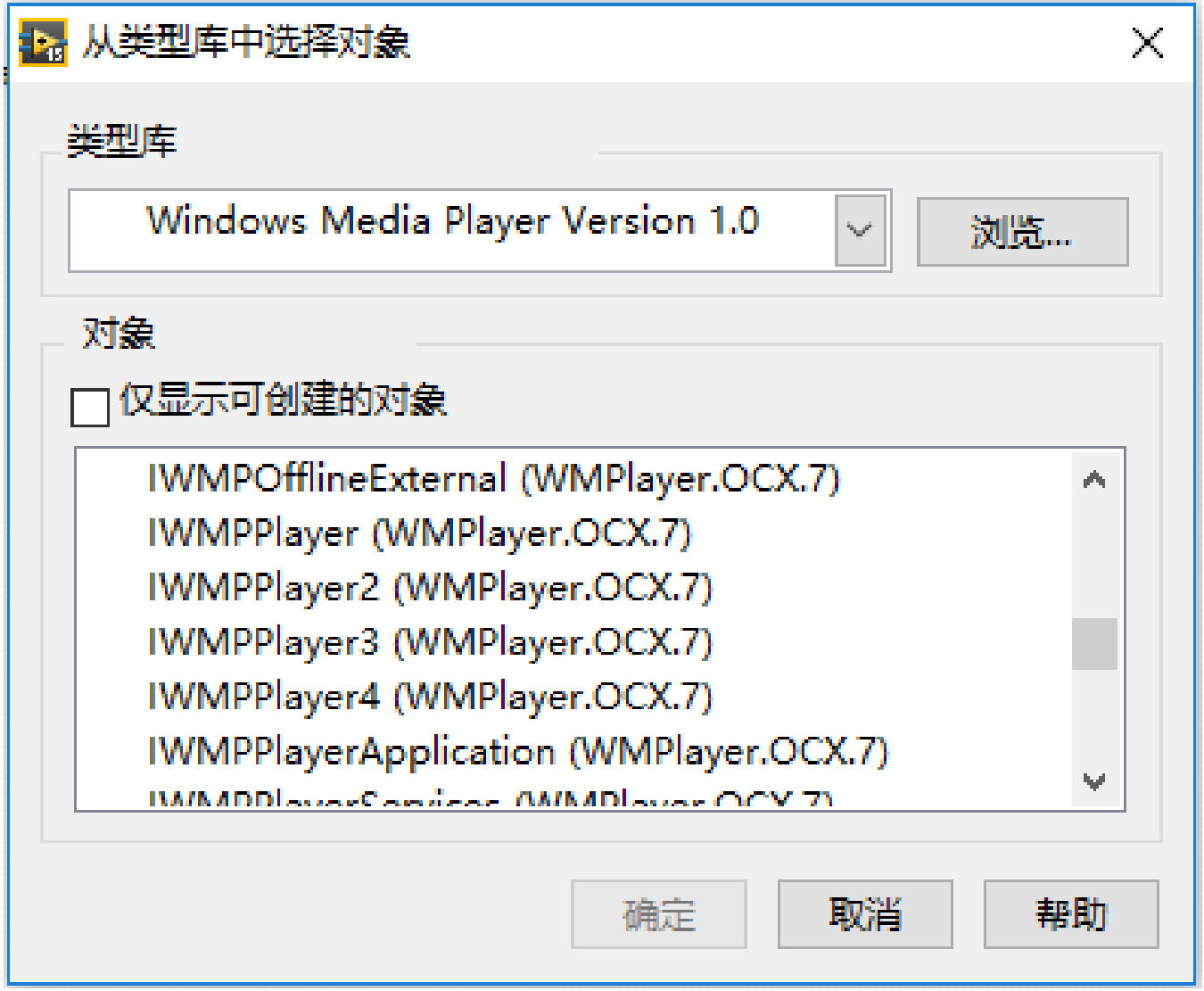
- 当用户在如图10.14所示前面板的路径输入控件中输入文件路径后，运行程序，将会打开Windows Media Player程序并播放相应文件。



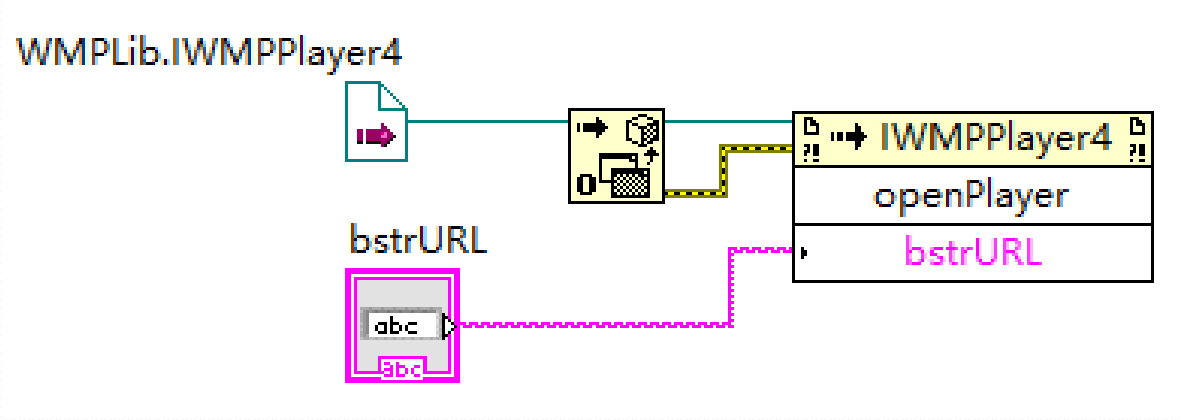
- **10.3.3 ActiveX自动化**
- ActiveX广义上是指Microsoft公司的整个COM架构。LabVIEW可以通过ActiveX自动化来调用那些基于COM架构提供出来的、而又没有对应的ActiveX控件的服务。
- 由于没有对应的控件，LabVIEW就不能直接从控件得到这个ActiveX对象的引用。因此，需要使用“互连接口”→“ActiveX”→“打开自动化”函数来创建一个ActiveX对象并得到它的引用。使用这个函数时，需要为创建的ActiveX对象指定类型：首先把“打开自动化”拖到流程框图程序上，然后为其“自动化引用句柄”输入创建一个常量，再用右键单击创建出来的常量，为其选择ActiveX类，如图10.15所示。


显示项
帮助
范例
说明和提示...
断点
ActiveX选板
替换
选择ActiveX类
删除并重连
创建
属性
WMPLib.IWMPPlayer4
SHDocVw.IWebBrowser2
浏览...

- 在图10.13和图10.14所示的程序框图中，系统是直接通过ActiveX容器来选择ActiveX对象类型为Windows Media Player的。若使用ActiveX自动化，则用户不需调用ActiveX容器，因此也就需要另外调用Windows Media Player控件。下面介绍通过ActiveX自动化来实现对Windows Media Player控件的操作。
- 首先用户需要在程序框图中放置一个“打开自动化”函数，鼠标右键选择函数图标左上角的“自动化引用句柄”端子创建一个输入控件，然后鼠标右键单击该输入控件，从弹出的快捷菜单中选择“选择ActiveX类→浏览”选项，打开一个名为“从类型库中选择对象”的对话框，如图10.16所示。在对话框中选择类型库为“Windows Media Player Version 1.0”，选择对象为“IWMPPlayer4”，单击“确定”按钮退出对话框，这样就完成了自动化引用句柄与Windows Media Player控件的连接。


从类型库中选择对象
类型库
Windows Media Player Version 1.0
浏览...
对象
仅显示可创建的对象
IWMPOfflineExternal (WMPlayer.OCX.7)
IWMPPlayer (WMPlayer.OCX.7)
IWMPPlayer2 (WMPlayer.OCX.7)
IWMPPlayer3 (WMPlayer.OCX.7)
IWMPPlayer4 (WMPlayer.OCX.7)
IWMPPlayerApplication (WMPlayer.OCX.7)
确定
取消
帮助

- 用户完成了上面的连接之后，只需将“打开自动化”函数右上角的“自动化引用句柄”端子与“调用节点”函数连接即可完成Windows Media Player控件的调用，从而实现希望达到的功能。需要注意的是，“打开自动化” 函数在左上角与右上角都有一个“自动化引用句柄”端子，用户要注意这两个端子之间的区别，不要在连线的时候由于混淆以致于程序出现错误。图10.17给出了使用“打开自动化” 函数来实现Windows Media Player控件调用的程序框图以及前面板界面，单击运行按钮即可打开Windows Media Player程序并播放指定路径的文件。可以看到，此时面板中没有出现类似Windows Media Player控件的界面


WMPLib.IWMPPlayer4
IWMPPlayer4
openPlayer
bstrURL
bstrURL
abc
abc


WMPLib.IWMPPlayer4
bstrURL
C:\Music\小城故事.mp3