## VS2010,VS2005串口控件MSComm加入方法

首先，我用的是VC2010，但在VC2010下没有这个控件，所以要借用一下VC6。控件名是MSCOMM32.OCX，如果安装了VC6或VB6的话，就能在/WINDOWS/System32下找到MSCOMM32.OCX，MSCOMM32.SRG，MSCOMM32.DEP三个文件。如果在开发的机器上没有安装VC6或VB6，同时也没有安装过使用的这个控件，则首先要对这个控件进行注册。注册方法是使用命令regsvr32 MSCOMM32.OCX，比如我就直接把找来的这三个文件放到我的System32下，同时新建一个文本文件，把命令regsvr32 MSCOMM32.OCX写到文本文件中，改扩展名为bat，直接双击这个bat就完成空间注册。另外，为了在VC2005中开发的时候能正常使用这个控件，还需要修改一下注册表，否则会出错。修改方法是将以下内容写到一个文本文件中，扩展名改为reg，然后双击这个文件导入就可以了。（此处注册表修改的内容也可以在MSCOMM32.SRG中找到）

REGEDIT

HKEY\_CLASSES\_ROOT\Licenses = Licensing: Copying the keys may be a violation of established copyrights.

// Comm Control 6.0 license key

HKEY\_CLASSES\_ROOT\Licenses\4250E830-6AC2-11cf-8ADB-00AA00C00905 = kjljvjjjoquqmjjjvpqqkqmqykypoqjquoun

完成以上的控件注册和注册表的修改后就可以开始程序的开发了。首先新建一个MFC对话框程序。然后在IDE的右侧打开“工具箱”，鼠标右击，在弹出菜单中点击“选择项”，在COM组件下找到Microsoft Communications Control，在其前面的选择框中打勾，一个黄色电话图标就会出现在工具箱里，用鼠标把这个图标拖进对话框资源编辑器里，控件就被添加到项目的程序中。

为了便于使用，我们可以给这个控件绑定一个控件变量，比如叫做m\_mscomm，程序会自动添加一个CMscomm类封装这个控件，而m\_mscomm就是这个类的一个对象。接下来就可以以此控件变量来使用控件了。

在正式开始收发数据前，要给控件设置适当的几个参数来初始化。

首先是指定端口号，使用方法m\_mscomm.put\_CommPort(1);，参数就是端口号，比如此处程序要使用COM1端口。

然后是波特率、奇偶校验、数据位和停止位，使用方法put\_Settings(LPCTSTR newValue)，参数是一个形如”\*,\*,\*,\*”字符串，比如我需要波特率4800，无奇偶校验，8位数据位和1作为停止位就可以如此操作，m\_mscomm.put\_Settings(\_T(“4800, n, 8, 1”))；。

m\_mscomm.put\_RThreshold(1)和m\_mscomm.put\_SThreshold(0)分别设定接收和发送数据的时候，引发接收数据的OnComm事件时缓冲区中的字符数量，0表示不触发OnComm事件，比如put\_RThreshold(1)表示每接收到一个字符就处罚OnComm事件。

m\_mscomm.put\_InputLen(100)设定当前接收区数据长度为0,表示全部读取。

m\_mscomm.put\_InputMode(0)用来设定数据接收模式，1表示二进制，0表示文本。

m\_mscomm.put\_OutBufferSize(1024)和m\_mscomm.put\_InBufferSize(1024)分别设定输出和接收的缓冲区大小，单位是字节。

比如，现在要接收数据的话，可以如下初始化程序。

view plaincopy to clipboardprint?

m\_mscomm.put\_CommPort(1);    
           m\_mscomm.put\_Settings(\_T(“4800, n, 8, 1”));    
           m\_mscomm.put\_RThreshold(1);    
           m\_mscomm.put\_SThreshold(0);    
           m\_mscomm.put\_InputLen(100);    
           m\_mscomm.put\_InputMode(0);    
           m\_mscomm.put\_InBufferSize(1024);

 m\_mscomm.put\_CommPort(1);  
            m\_mscomm.put\_Settings(\_T(“4800, n, 8, 1”));  
            m\_mscomm.put\_RThreshold(1);  
            m\_mscomm.put\_SThreshold(0);  
            m\_mscomm.put\_InputLen(100);  
            m\_mscomm.put\_InputMode(0);  
            m\_mscomm.put\_InBufferSize(1024); 初始化完成后就可以用m\_mscomm.put\_PortOpen(TRUE)来打开端口开始接收数据了。

根据我们在RThreshold中设定的参数，程序会在接收到参数中指定的字符后触发一个OnComm事件，我们就需要处理这个事件来提取接收到的数据。我们需要通过查询m\_mscomm.get\_CommEvent()来确定具体的事件类型，比如可能是发送也可能接收到了数据，比如在此我们要提取接收数据，事件对应的返回值应当是2。然后获得的数据就可以用m\_mscomm.get\_Input()来获得，这个函数返回一个VARIANT结构变量，如果我们接收的是字符串数据，那就是保存在bstrVal中，比如下面的代码就是在接收到数据时将接收到的字符串保存在str变量中。

view plaincopy to clipboardprint?  
void CMScomDlg::OnCommMscomm()    
            {    
            // TODO: 在此处添加消息处理程序代码    
            CString str;    
            switch(m\_mscomm.get\_CommEvent())    
            {    
            case 2:    
            str=CString(m\_mscomm.get\_Input().bstrVal);    
            break;    
            default:    
            break;    
            }    
            }

void CMScomDlg::OnCommMscomm()  
            {  
            // TODO: 在此处添加消息处理程序代码  
            CString str;  
            switch(m\_mscomm.get\_CommEvent())  
            {  
            case 2:  
            str=CString(m\_mscomm.get\_Input().bstrVal);  
            break;  
            default:  
            break;  
            }  
            } 如果要发送数据的话，只需调用m\_mscomm.put\_Output(VARIANT newValue)将保存在newValue中的数据发送出去。在此我们可以使用一个COleVariant类对象来代替直接使用VARIANT类型变量，COleVariant直接提供了将字符串变量转化为兼容VARIANT类型的构造函数，比如我们将数据保存在CString类型的str变量中，然后用COleVariant 包一下送给put\_Output ，如m\_mscomm.put\_Output(COleVariant(str))就可以将str中的字符通过串口发送出去。