

Re: Receiving single bytes with MSComm

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2005-02/1501.html>

From: Joseph M. Newcomer (*newcomer_at_flounder.com*)

Date: 02/16/05

Date: Wed, 16 Feb 2005 14:27:21 -0500

I guess I've never before seen such convoluted code whose sole purpose is to receive a single byte from a serial port. I tend to favor code that looks like

```
ReadFile(h, &buffer, 1, &bytesRead, NULL);
```

as being the most complicated code I want, and in addition, if I'm doing asynchronous I/O, I'll create an event to indicate the completion, e.g.,

```
OVERLAPPED ovl;  
BYTE buffer;  
ovl.hEvent = event; // created a long time ago  
if(!ReadFile(h, &buffer, 1, &bytesRead, NULL))  
{ /* error */  
    DWORD err = ::GetLastError();  
    if(err != ERROR_IO_PENDING)  
        { /* fatal error */  
            ... deal with it  
        } /* fatal error */  
    HANDLE waiters[2];  
    waiters[0] = shutdownEvent;  
    waiters[1] = event;  
    DWORD wait = WaitForSingleObject(2, waiters, FALSE, INFINITE);  
    switch(wait)  
        { /* wait */  
            case WAIT_OBJECT_0:  
                // shut down  
                break;  
            case WAIT_OBJECT_0+1:  
                break;  
            default:  
                ASSERT(FALSE);  
                ... deal with error  
        } /* wait */  
    } /* error */  
wnd->PostMessage(UWM_BYTE_RECEIVED, buffer, NULL);
```

I've used code like this in on the order of a dozen major applications, all of which are out there running right now. Had I used MSCOMM, I probably wouldn't have a single running

app.

I note that nowhere in your code is there any error detection. Nor is there a way to recover from a device that is hung waiting for a byte. Overall, there is an amazing amount of convoluted work to interface to what is probably an inappropriate interface in the first place. If you simply take as a premise that MSCOMM is the wrong approach, the problem becomes trivial by comparison. And it has issues such as reliability and robustness that appear to be absent in the MSCOMM solution.

I suppose I might be able to offer more advice if MSCOMM were documented, but I am unable to locate any documentation in the MSDN. Probably one of the reasons I don't use it; I don't use undocumented interfaces. If it were an important control, there would be documentation on it. That is, I could type "MSCOMM" to the index, or possibly to the search window of the MSDN, and get an actual, real, live, document that describes how to use the control. (The best I found was a pointer to a nonexistent file, comm98.chm, which does not exist on my machine, and I have the full Visual Studio 6 and Visual Studio 7 installations, the latest platform SDK, and the second-latest MSDN library installed, so the lack of documentation clearly means this is an inappropriate control). In the absence the the ability to find documentation, the presumption is that this control is distributed as a joke control, whose purpose is not to solve problems but to create them. Why elect to use something that is undocumented, hard to use, and probably completely inappropriate, and then try to force a solution out of it?

joe

On Wed, 16 Feb 2005 04:01:39 GMT, "Me" <me@right.her> wrote:

```
>I got the transmit code working sending binary data for MSComm.
>
>I have the following code for receiving a string of bytes.
>
>bool CPSMDlg::GetChar()
>{
> COleVariant myVar;
> int hr;
> long lLen = 0;
> BYTE *pAccess;
> char buffer[255];
>
> myVar.Attach (m_Comm.GetInput());
>
> hr = SafeArrayGetUBound (myVar.parray, 1, &lLen); // Get the length
> if (hr == S_OK)
> {
> lLen++; // upper bound is
> zero based index
> hr = SafeArrayAccess (myVar.parray,(void**)&pAccess); // lock array so
> you can access it
> if (hr == S_OK)
> {
> for (int i 0; i < lLen; i++) // Make a copy of
> the data
```

```
> buffer[i] = pAccess[i];
> SafeArrayUnaccessData (myVar.parray); // unlock the
> data
> }
> }
> // COleVariant cleans itself up
> }
>
> How do I receive single bytes one at a time since the device may only send 1
> byte in response to commands???
>
> Thanks
> eddie@eddie1.net
>
```

Joseph M. Newcomer [MVP]
email: newcomer@flounder.com
Web: <http://www.flounder.com>
MVP Tips: http://www.flounder.com/mvp_tips.htm