

Re: lineGetID() on PocketPC / Compact Framework

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.tapi/2005-02/0532.html>

From: Matthias Moetje (*moetje_at_terasens_nospam_de*)

Date: 02/24/05

Date: Thu, 24 Feb 2005 18:20:10 +0100

Peter,

from the PSDK:

comm/datamodem/portname

The comm/datamodem/portname device class consists of the device names to which modems are attached. When this device name is specified in a call to the lineGetID function, the function fills the VARSTRING structure with a null-terminated ANSI (not Unicode) string specifying the name of the port to which the specified modem is attached, such as "COM1\0". This is intended primarily for identification purposes in the user interface, but could be used under some circumstances to open the device directly, bypassing the service provider (if the service provider does not already have the device open itself). If there is no port associated with the device, a null string ("\0") is returned in the VARSTRING structure (with a string length of 1).

So, since the service provider already has the device open, you cannot use the com port for doing communications. You really need to use the file API and probably you would also need to do overlapped I/O. I believe that this is not possible from managed code, especially not on CE platforms, so you should create a component in eVC++ for the I/O and use that from your managed code (there are ways to do this...).

Best regards,

Matthias Moetje

TERASENS GmbH
Ackermannstraße 3
80797 München

Fon: +49 89 143370-0
Fax: +49 89 143370-22
e-mail: moetje at terasens dot de
www: www.terasens.de

microsoft.public.win32.programmer.tapi: Re: lineGetID() on PocketPC / Compact Framework

"Peter Beedell" <p-beedell@wellcom.de> wrote in message
news:%23noWNqoGFHA.3352@TK2MSFTNGP10.phx.gbl...

Hi,

The TAPI function lineGetID() is documented as returning the com port I can
use to transfer data once a GSM DataCall has been established:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tapi/tapi2/comm_datamodem.asp

Here is my C# implementation:

```
int iResult = lineGetID(myLine.hLine, myCall.AddressID, myCall.hCall,  
LINECALLSELECT.LINE, baVarString, "comm/datamodem/portname");
```

For me, running on an MDA II / XDA II, this fails. Reducing the TAPI Device
Class to "comm/datamodem" works but this returns a IntPtr handle to the port
which forces me to use the File APIs because C# can not open a stream using
a device handle... am I missing the point here?

How is a C# programmer supposed to exchange data using the DataCall he has
established using TAPI?

Best regards,

Peter Beedell p-beedell@wellcom.de