

Re: Redirecting lpt1 output to an installed printer

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-08/msg01714.html>

- *From:* "Ralph" <nt_consulting64@xxxxxxxxxx>
 - *Date:* Wed, 9 Aug 2006 14:31:26 -0500
-

"Jim Carlock" <anonymous@localhost> wrote in message
news:%23pwdDI9uGHA.324@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

"Jerry" <xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

The application that I am referring to is an old DOS app that prints to the lpt1 port. Printing to the default printer isn't working. What I am doing now is to redirect the output using the "net use" command. I would like to fully automate the operation.

If there's a recognized printer on that port, it shows up in the Windows Printer applet, right? Or is the printer not listed in the list of printers when you start the the printers explorer listing.

I found a pretty neat vbScript in the system32 folder of an XP system dated 8/23/2001, named prnport.vbs. Definitely an interesting piece of software. The top of the file reads...

```
'-----  
'  
' Copyright (c) Microsoft Corporation. All rights reserved.  
'  
' Abstract:  
' prnport.vbs – Port script for WMI on Whistler  
' used to add, delete and list ports  
' also for getting and setting the port configuration  
'  
' Usage:  
' prnport [-adlgt?] [-r port] [-s server] [-u user name] [-w password]  
' [-o raw|lpr] [-h host address] [-q queue] [-n number]  
' [-me | -md ] [-i SNMP index] [-y community] [-2e  
| -2d]"  
'  
' Examples
```

Re: Redirecting lpt1 output to an installed printer

```
' prnport -a -s server -r IP_1.2.3.4 -e 1.2.3.4 -o raw -n 9100  
' prnport -d -s server -r c:\temp\foo.prn  
' prnport -l -s server  
' prnport -g -s server -r IP_1.2.3.4  
' prnport -t -s server -r IP_1.2.3.4 -me -y public -i 1 -n 9100  
'  
'  
-----
```

And then all sorts of information about it and the other .vbs scripts exists on the net. Just searching for its name turns up quite a bit of detail.

Microsoft provides some information about it here:

<http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/prnport.mspx>

<http://technet2.microsoft.com/windowsserver/en/library/24390518-1C91-40FF-8046-E75B10C27E121033.mspx>

Then, PrnAdmin.dll exists on Windows NT4 and Windows 2000 when the appropriate resource kit gets installed. Microsoft provides some help with that here:

<http://support.microsoft.com/?kbid=321025>

Converting the vbScript to actual VB code takes little time, and using ShellExecute to run the items instead of using wscript.run works for the most part (for running any sort of external executable).

But then going back to what I posted, you can list the names of all the "VB recognized" printers connected to the system (meaning if the printer actually shows up in the explorer applet [named "Printers and Faxes" on Windows XP]). You failed to identify if the printer gets recognized by the system, so no one here knows exactly how the situation over there exists.

A rereading of your initial question brings up some things...

I have an application that outputs to lpt1. I have a vb6 application that lists all my printers and allows me to select one. I then pass the selected printer to a batch file that calls "net use lpt1" to

redirect

the output. This doesn't work to well. The program does not know when the batch job is complete, or if it completed successfully. Is there a way to redirect the lpt1 output to a windows printer from within a vb6 application?

Re: Redirecting lpt1 output to an installed printer

I read that as...

2 applications work together. The first application I don't own the source code for, the second application is my VB6 app. The first app sends its data to LPT1:. So I need to intercept the LPT1: port, which I do via a "net use LPT1" command inside a batch file.

So I guess that in your batch file you issue the "Net Use" command, then start your VB program, then start the program that sends data sends data to LPT1, whereby the VB App intercepts the data that comes in over LPT1.

And your looking for a way to get VB to hook up to intercept all stuff coming out of the LPT1 port and negotiate with the other app as its occurring?

That seems a little more than I initially garnered. I seem to recall a mode.com command getting used for some redirection things, but I could be way off target there. Perhaps not though...

MODE LPT1:=CON:

should stream information to the console, and then a redirection symbol >> could be used to put it in a file. Just not sure how to get this done with a VB app right at the moment, but it seems like it might be worth investigating. In other words run the MODE statement in the batch file, then run your app like:

```
C:\PathTo\MyApp.Exe >> file.ext
```

I await your answer. Ralph might be on the right track. There a way to redirect the stuff from LPT1 to a file, then read the data in the file with VB? A timer could check for the presence of the file and work things from there.

—

Jim Carlock
Post replies to the group.

I like everyone else's answers better than mine. <g>

One thing I might comment on. There is a problem when one attempts to redirect output from a "Console" or "DOS" program, as there are actually two kinds of "DOS Prompt" APIs and many Window programmers don't know the difference.

1) Low-level and buffered-stream System I/O thru the command-interpretter. Highlighted by puts(), gets(), iostream(), etc, that reads and writes using the stdin/stdout of the command-interpretter (cmd/command).

Re: Redirecting lpt1 output to an installed printer

You can redirect output/input by simply using the redirection operators of the command-line interpreter.

This includes your true MS-DOS programs as well as 32-bit "command line programs".

2) The Console and Character-Mode support.

This is your AllocConsole, ReadConsole, WriteConsole, etc

This API creates a separate screen buffer, and by default reads and writes from and to its own buffer and NOT stdin/stdout. Outputs for example will show in the DOSPrompt window, but is actually in its own 'window'. It is possible to internally (by code) redirect or re-pipe the commands to stdin and stdout, but it often isn't done.

You often can't tell the difference between the programs by looking at them, so the usual test is to run them and try to redirect output.

net commands are Console apps which only write to their own screen buffer, which you can test with ... "c:> net /? > test.txt". "test.txt" will be empty.

[You can also use dumpbin and see what libraries it uses.]

That's the trouble when somebody comes on here and says "I have a *DOS* program that I want it to ...". What is it really? Is it a 16-bit MS-DOS program, a 32-bit System I/O program, or a true Windows Console program? And if the latter does it internally redirect to stdin/stdout/stderr or not?

It can get worse. You can find Console apps that write results to their screen buffer only, but write errors to stderr, for example. Or Window apps that create consoles (the DOSPrompt app for example).

[Shear guess, but since the OP's program is hard-wired to print to lpt1, my guess it was written using System I/O.]

hth

-ralph

.