DOS printing and Windows printers.

Enabling DOS applications to print to Windows printers isn't that straightforward:

- In the DOS days, printers were connected to a parallel (LPT1-3) or serial (COM1-4) port. Printers now connect to an USB port or (wireless) network.
- DOS applications can only print to a parallel or serial port; they don't know about future USB ports. Modern PC's don't have parallel or serial ports anymore.
- DOS printers expect to receive text. An application adds printer control codes to end a line (line feed), eject a page (form feed), eventually print in bold (DOS printer model specific codes)… When a line ended, it was instantly printed (dot-matrix printers). Modern Windows GDI printers mostly don't support text input and are not line orientated: They expect formatted, complete pages.
- DOS text is ASCII based: 128 (minus 32 control codes) characters, basically those found on an US International keyboard. The ASCII character set was extended to support an additional 128 characters. Due to ASCII's limitations (256-32 characters), a language specific DOS code page is needed to define what those extra characters translate to in Windows text (Unicode with thousands of characters).

So some work is needed to facilitate a DOS application to print to a Windows printer, mainly:

- The stream of text and printer control codes sent to a printer port has to be collected, at some moment interpreted and formatted to Windows GDI pages.
- Those pages then have to be redirected to a Windows printer (mostly not parallel/serial).

When a DOS application prints to a LPTx/COMx printer port, vDos will:

- Save that text to a #LPTx/COMx.asc file. Not of any use to vDos itself, but so you can optionally use an alternative external Windows program to convert and print that ASCII text.
- Convert the ASCII text to Windows Unicode text (based on the current DOS code page), and save that to a #LPTx/COMx.txt file. You can open that file in a Windows editor, or read it into a Windows application.
- By default convert the collected text and printer commands to a format accepted by a Windows printer.

Printing in vDos.

If you don't tell vDos (in config.txt) what to do with text sent to a printer, it lets you select the Windows printer to print to. If the printer output is plain text, with line/form feeds, eventually extra printer codes inserted by your DOS application as if it was printing to an Epson, IBM Proprinter or PCL compatible printer, this should mostly work just fine.

You may however want to adjust the default settings of how this text is handled by adding LPTx = or COMx = option lines to config.txt:

LPTx/COMx = options (x = 1-9, though many DOS programs will at most support LPT1-3 and COM1-4):

Left to the equal sign defines what the application expects to print to. The options to the right what vDos is considered to do with that:

SPOOL
If used, it has to be the very first option. It postpones printing until Win+Ctrl+S is pressed, the title bar of the vDos window will show spooling is in progress. To solve persistent timeouts issues (see TIO = below), or to accumulate DOS print jobs.

TIO: seconds
Printer output is collected as print jobs (Windows printers need this concept), considered to be finished when the application doesn't send further text for some time. The default setting is: If the last print command was a form feed (eject page), wait for 1, else 5 seconds. If you get broken pages, or want the printer to respond faster; there're two options. Specify the amount of time vDos will wait (1 to 99 seconds). Or disable the timeout functionality by TIO: 0. Applications can address a printer in various ways, this will only work for the "nice" way: Open, send text, and finally close a DOS LPT/COM device. Mind, applications aren't always that nice; nothing could get printed with TIO: 0!

SEL: "Windows printer name"
Explicitly sets the printer to be used. SEL: "" will select the default Windows printer.
* Default: display a printer selection dialog.

RAW
To use this option, the printer must support DOS text input (PCL/PostScript printers do). The (ASCII) text is sent directly to the printer. For instance to print to an actual PCL compatible printer if such a (mostly HP named) printer is set in your DOS application. Mind, the remaining options don't apply with RAW.

NOLINES
Sequences of three or more "-" or "=" characters are by default replaced by line drawing characters. For instance "- - - -" will be printed as "——". With NOLINES the original characters will be printed.

FONT: "Windows installed font"
Print the text in this font, this should be a monospaced one: Each character has the same width.
* Default: The built-in vDos font also used for displaying text.

HORZ: left[,right,chars]
left: The left paper margin in mm's.
right: The right margin in mm's.
chars: The number of characters that has to fit between these margins.
* Default: 15,10,80.

VERT: top[,bottom,lines]
top: The top paper margin in mm's.
bottom: The bottom margin in mm's.
lines: The number of lines that has to fit between these margins.
* Default: 10,15,lines calculated based on the paper size.

Example:
LPT1 = spool font:"Courier New" horz:20,10,132
Postpones printing, displays a dialog to select the Windows printer (none set), prints in the Courier New font, sized so 132 characters will fit between the set margins.

Other print options.

DUMMY
Don't print; you just want to ignore the printers output. Or use an alternative DOS-to-Windows printer program, monitoring the vDos directory for new .asc files to print.
Example: LPT3 = DUMMY

CLIP
Don't print, copy the translated Unicode text to the Windows clipboard to paste into another program.
Example: COM4 = CLIP

"Windows command/program" [WAIT][HIDE][program options] #LPTx/COMx.asc/txt
Open the .asc(ASCII)/.txt(Unicode) file by the specified Windows command/program.
Example: LPT2 = "%windir%\system32\notepad.exe" HIDE /p #lpt2.txt
(Let Notepad print (/p) the #lpt2.txt file, while hiding (HIDE) its window).

"Windows device":
Essentially not meant for printing; communicate with the specified device interactively (expect no text to be collected as a print job). The colon at the end is required.
Basic support for serial devices, the port has to be setup correctly before vDos is started. In the device manager, or by MODE COM… at the (Windows!) command prompt.
Example: COM1 = "COM1":

DOS application prints to PCL printer.

If your DOS application can be set to print to a PCL printer: Most laser printers still support that text input. If you don't have an actual PCL printer and the vDos print processor is too basic: vDos will detect a PCL printer is addressed (the text will contain some PCL specific code sequences). If provided in the vDos directory, the PCL6 program is then started to convert the saved .asc file to a PDF document. That is then opened in the default Windows PDF viewer. PCL6 (GhostPCL) can be downloaded and is documented at: www.columbia.edu/~em36/ghostpcl.html.

Use a third-party program to print.

The built-in support of Epson, IBM Proprinter and PCL printers especially lacks graphics (images). vDos is also distinct at handling margins, ignoring those eventually set by embedded printer codes. So scaling and positioning by HORZ: and VERT: actually works. You could be unhappy with the end result on paper, possibly with an application doing extensive print formatting. Since vDos saves the (ASCII) text to an .asc file, you can use any external DOS-to-Windows print program to do the printing instead. Three of those programs, among others available online:

- **DOSPRN**
  Shareware, $14.95, price drops at more licenses.
  DOSPRN's author added some extensions to make it work better with vDos. DOSPRN also supports IBM Proprinter and limited PCL printing, you can even define alternative sets of printer codes.
  Example: LPT1 = "%ProgramFiles(x86)%\DOSPRN\DOSprn.exe" DOSPRN options #lpt1.asc
  Note: Omit (x86) with Windows 32-bit.
- **DOSPrinter** (Epson-only)
  Shareware, $40.00 for a standard (single user) license.
  Example: LPT2 = "DOSPrinter.exe" WAIT DOSPrinter options #lpt2.asc
  (Provided the DOSPrinter program is located in the vDos directory).
- **WinPrint** (Epson-only)
  Freeware.
  WinPrint installs to the Windows system tray, monitoring a directory for new files to be printed.
  Example: COM1 = dummy
  (vDos should only create the #com1.asc/.txt files, WinPrint will be set to monitor the vDos directory for new (#com1).asc files).