xSpool

User Manual

The vpxPrint Spooler

Spooling system for LINUX and Windows

Version 6.3

September 2023

4GL
vpxPrint Studio

4gl.fr

***xSpool** is dedicated to network spooling UNIX - WINDOWS.*

At first, we wanted to exchange **vpxPrint** files between two applications or two systems.

The initial goal was to provide a solution for character applications that some users still use. These applications are usually not able to call a dll or execute a Windows API.

The main principle of **xSpool** is very simple:

1. The main program creates files in a local/shared folder or on a server machine, accessible to the PC stations,
2. On the Windows PC, xSpool silently scans this directory,
3. When a file is found, xSpool starts the corresponding application and deletes the file after processing. (it can be saved in a "backup" directory)

The problems to solve:

1. The files to process can be created in a local or shared (SAMBA) directory available on the PC. If the server does not have this capability, xSpool use the FTP protocol to retrieve the files.
2. In an FTP process, each PC must process the files created for him and bypass the other files. It quickly becomes apparent that the ETHERNET address was the best choice for determining unique receivers. **– PREFIX can be used to change this option**.
3. How to determine the ETHERNET address of an emulated "character" session on the host?
4. Write a file spooler on the PC. This program must be very flexible to support a wide variety of combinations: shared device or FTP, UNIX or not, ASCII or BINARY, etc…

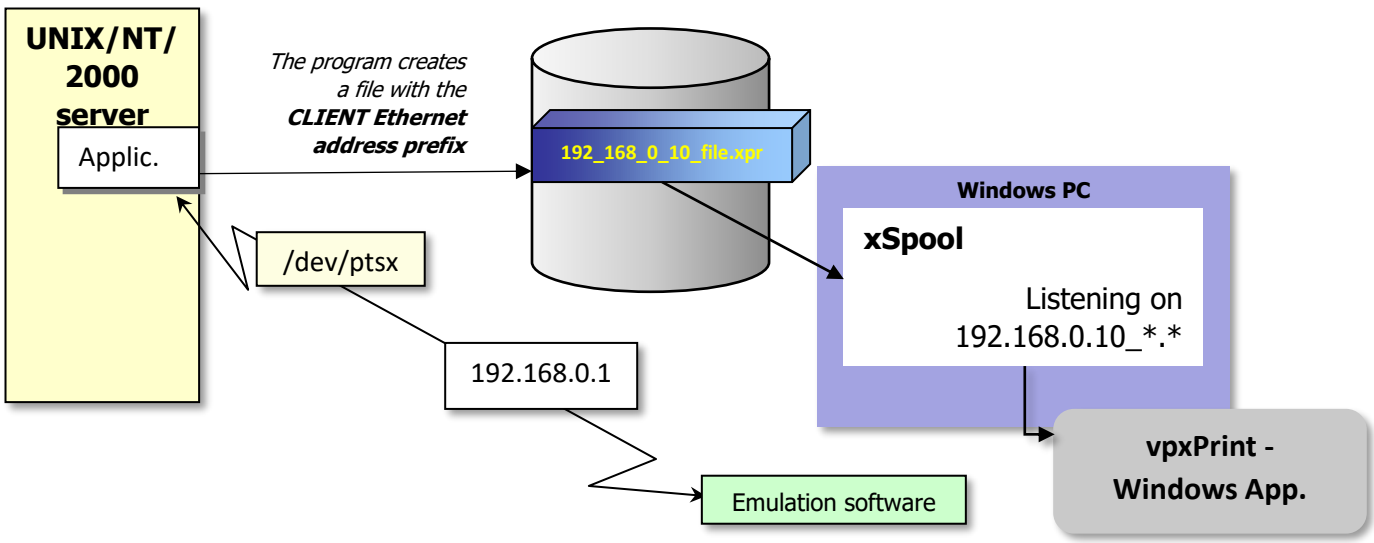


Fig 1. xSpool main principle

History of changes:

VERSION	DESCRIPTION
6.3	Linux & MacOS friendly…
5.2	Light theme, Encoding of FTP parameters (FTPEncrypt), Faster start-up and restart, Fix a problem on restart on FTP connections,
5.1	Black theme New manual
4.13	Handling and logging of vpxPrint return codes, Process the return code 18 (vpxPrint version 10.22j) to wait for the Printer Spooler to regain the control after a Windows spooler shutdown.
4.12	Trying to restart xSpool when printFile() aborts and the log file does not exist
4.9A	More information is collected in the log file in debugging mode
4.9	New keyword –ORDER sets the order to process the files: –ORDER=TIME files are processed on their modification date/time –ORDER=NAME files are processed on alphabetic names order More information is collected in the log file, Default value for –RESTART has been set to 100
4.7-4.8	If **vpxPrint 10.10a** (at least) is connected to xSpool 4.7, a complete log file is created in the –LOGFILE folder if a problem exists in vpxPrint when processing the file. xSpool processes the files in the creation time order. That was not always true.
4.6	In –SILENT mode, the keyword –SILENT is added multiple times in -RESTART mode, fixed Debug has been integrated, New keyword **–NOCHECK**
4.0-4.5	**version 4.0** • xSpool stops and restarts automatically based on the value of **–RESTART** parameter • information on vpxPrint version in the main xSpool screen **version 4.1** • xSpool 32 and **64bit** version • new keyword **–SILENT** allows to run xSpool silently in the background **version 4.2** • new keyword –BACKUP creates a backup copy of the files after they are processed • -PREFIX did not work as expected, fixed • Changes in the xSpool window ergonomics, new display “filter” • Keyword **–DLL** was not documented **version 4.3** • When a file is locked, vpxPrint is not called. This avoids the “File in use” message.

	version 4.4 • New keywords -ERRORDIR and –LOGFILE to trace the files processed **version 4.5** • Limit of command line has been set to 512 characters instead of 255
3+	new parameters: **-PASSIVE** Enhancements to the retry procedure in case of empty directories on the server. support of V7-V9 users
2+	• **xSpool** is now minimized to system tray and –**MINIMIZE** option has been added, • -PERSISTENT parameter has been added to keep the connection alive while processing the files locally. Under normal situations, do not use this option, it may cause a timeout if your server is not configured to support long connections. • Bug correction: when different stations had the same beginning IP address, then the first address may retrieve the files instead of the other. For example PC1=192.168.0.10 and PC2=192.168.0.1. PC2 scans the server for 192_168_0_1* files. **Files retrieved must have an IP prefix separated from the suffix by a '_'.** **Files must be named 192_168_0_1_myfile.xpr** • **xSpool** runs *.bat files, • Automatic **unzip** of *.zip files to extract and proceed the files. • new parameters: **-LIC and –QUIT**
1+	**8 nov 2002: version 1.1.** **-EXEC=AUTO** allows calling multiple programs in a single session of xSpool. Depending on file extensions, the correct program as defined in WINDOWS is called. When –EXEC=AUTO is specified, the file type is determined using the file extension. Txt and xpr files are 'text files'. **Other extensions are evaluated as specified in the –BINARY or ASCII flag** (default is ASCII). vpxPrint will be called on .xpr file extensions. **8 nov 2002: version 1.2.** **-LOCALIP=*IP_Address*** has been added to specify a specific local IP address. This can be used when multiple IP addresses can be selected in the local setting **-PREFIX=*Prefix*** has been added to specify a unique prefix for files to process. This prefix will be used in place of local IP address when selecting files in the remote directory. The **.. value** to specify a parent directory in the –DIR parameter is handled. **4 dec 2002: version 1.3.** **A memory leak** has been solved when calling vpxPrint DLL from xSpool. Other configurations are not concerned. When a big amount of vpxPrint files were transferred (> 800) in a single session. **16 jan 2003: version 1.4** **When unsuccessful** login occurs, then xSpool tries to reconnect 5 times with a 30 seconds delay. If unsuccessful, then abort. **17 jan 2003: version 1.5** **xSpool** now reads all the pending files, then closes the FTP connection BEFORE processing the files. There is no need to modify the time-out option on the FTP server. **16 feb 2003: version 1.7**

	On login errors, xSpool retries 5 times with a 30 sec. interval between each retry.

xSpool is a self-executable file (xSpool.exe) that runs on a WINDOWS system.

It scans a directory, either:

- A local or remote directory on networked drives,
- A remote directory accessible through a FTP connection.

The accessible parameters used to define an xSpool connection are:

FTP Mode

-FTP=*host_address* or *host_name*	Host name in an FTP connection
-USER=*user_name*	FTP user name
-PASS=*password*	FTP password
	-OR-
-CONNECT=*encrypted_parameters*	This encrypted string is created with **FTPEncrypt.exe** (version 5.2) It encodes the **-FTP, -USER** and/or **-PASS** parameters for confidentiality reasons.

-DIR=*remote_or_local_directory_name*	1. In local mode, it's the input directory. 2. **If case of FTP session, this directory selects the relative directory where the user is initially logged on.**
-TEMP=*temporary_directory*	In FTP mode, points the local temp directory where the files are stored before processing. If not specified, the local Windows temp folder is used.
[-INTERVAL=*time_interval*]	# of seconds between two directory scans, default 2 seconds
[-BINARY or –ASCII]	BINARY or ASCII transfer in FTP mode ***.txt and *.xpr** files are always processed as 'text' files others as specified by this option. Default is ASCII
[-LOCALIP=*local_IP_Address*]	sets the local IP address if case of multiple local addr.
[-EXEC=***AUTO*** | *program_name* | *XPRINT*]	- when 'AUTO' is specified, xSpool tries to run the associated program as defined in the WINDOWS settings: MSWORD, for example, will be automatically called on .doc and .docX extensions, MS-Excel on .xls or .xlsX. Multiple file types can be processed in a single session. *program name* is used to call a given program.

xPrint calls the xPrint Dll on each input file.

[-PREFIX=*prefix_used_to_retrieve_files*] specifies the prefix **or the mask** of the files to process.

[-MINIMIZE] initializes xSpool in the system tray.

[-LIC=*vpxPrint_License_File*] Path and file name of the vpxPrint license file. When xSpool initializes, it copies the license file into the PC system directory.

[-QUIT] On close, vpxPrint shuts down silently without any dialog box.

[-PASSIVE] Set the FTP connection as passive.

[-DLL=*vpxPrintDLL*] Sets the name and the folder of xPrint.dll to use

[-RESTART=*nnn*] Sets the value of number of files processed before vpxPrint shuts down and restarts. **Default value 100**.
Set it to zero to disable this feature

[-SILENT] specifies that xSpool will run silently in the background

[-BACKUP=*folder*] specifies the folder where xSpool will save the files after processing (version 4.2)

[-LOGFILE=*log_file_name*] specifies the name and the path of the log file.

In this file, a trace of all xSpool files, timestamp.
It's your responsibility to manage/clear this file,

[-ERRORDIR=*error_directory*] specifies the folder where xSpool will store bad vpxPrint files if a crash occurs during processing.

[-NOCHECK] xSpool will not check if the file is accessible.

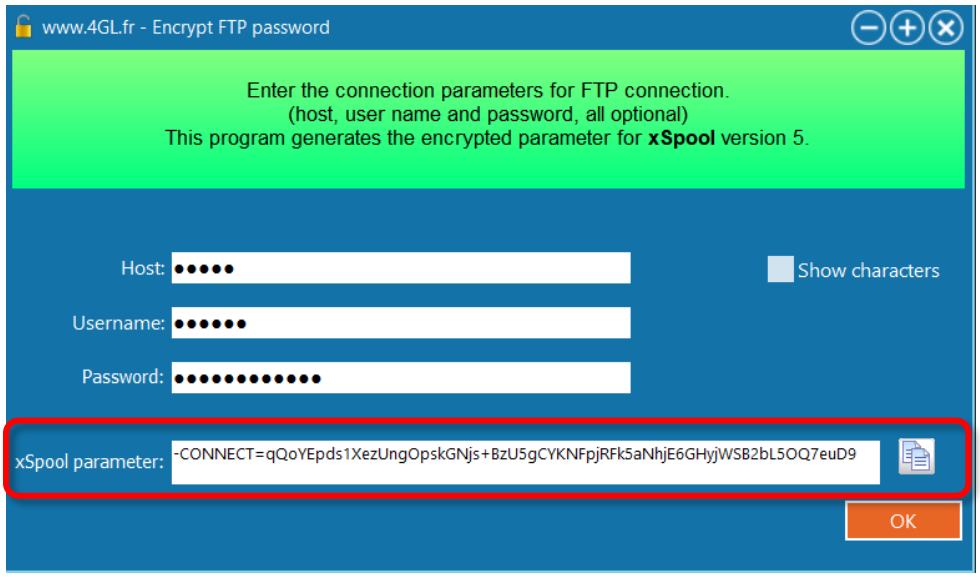
[-SORT=*xxxxx*] sets the order to process the files (**not active in FTP mode**):

–SORT=TIME	modification date/time order
–SORT=NAME	alphabetic names order

If –SORT is not specified, the files order is the Windows sort order.

 if the files are created in the input directory by multiple processes, the sort may provide incorrect order.

FTPEncrypt program



FTPEncrypt creates an encrypted string to store the host name and connection parameters in FTP mode.

It creates the **-CONNECT** parameter to copy and paste into the xSpool command line.

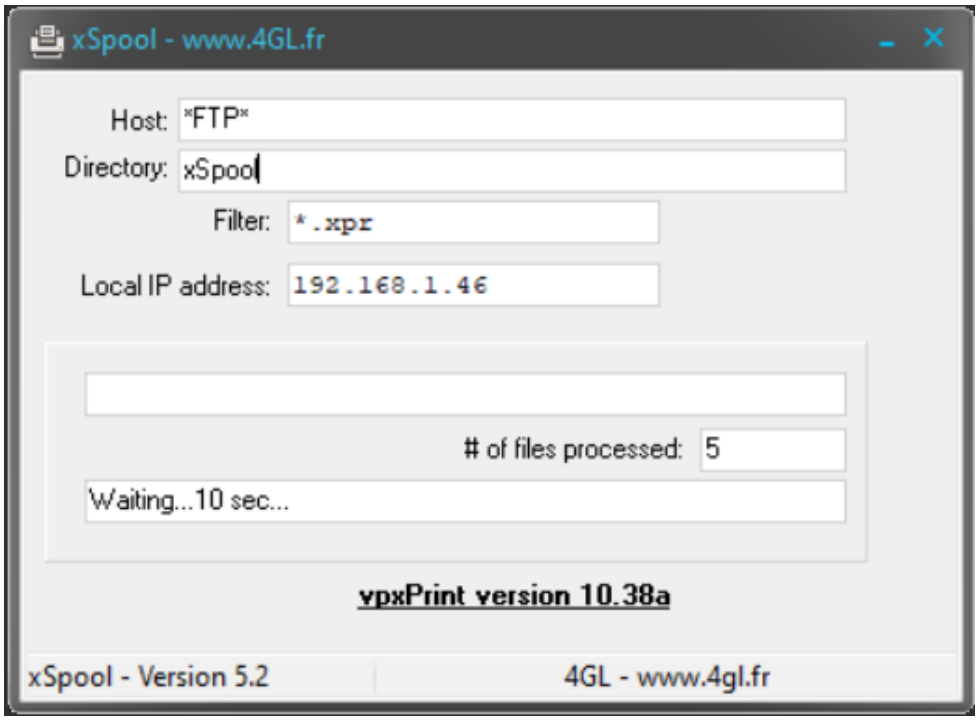


Fig 2. The xSpool screen

TECHNICAL CONSIDERATIONS:

xSpool searches in the directory specified by –**dir** parameter for files matching the Internet address of the local machine. If local machine address is 192.168.1.46, then all files beginning with **192_168_1_46_** are processed ('.' characters of the local IP address are replaced by '_') **unless if the –PREFIX keyword is used**.

NO MORE APPLICABLE with the version 1.5:

As files are processed as soon as received, the FTP connection (in case of FTP connection) remains active during the file processing.
Thus the FTP server should not limit the duration of FTP connection (time-out).You may override this option with the –PERSISTENT parameter

When files are processed, they are deleted from the server/local source directory. If you use a FTP connection, then the user must be allowed to delete these files from the server.

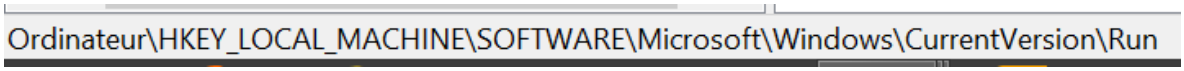
The **–BACKUP** keyword allows saving a copy of the file in a backup folder (version 4.2).

In FTP mode, server can be UNIX or WINDOWS. It just has to run an FTP server. For UNIX servers and text files, you can specify –**ASCII** mode if all files are text files.

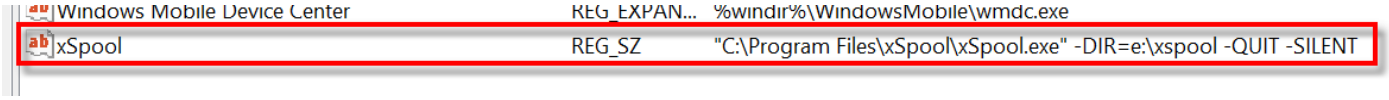
The default mode sends files received to the xPrint.dll or to the corresponding programs. If you want to send all the files to a specific program, then specify its name with the –exec parameter. –exec=notepad.exe sends all files to the notepad program.

xSpool does not run as a service. But it's possible to run it silently in the background with the –SILENT parameter. To start xSpool at Windows start-up, set the xSpool initialization in the Windows start-up parameters or in the register base:

With regedit, go to the folder:



and create a key like:



APPLICATIONS:

A lot of applications can be connected to **xSpool**.

- A print and preview server/spooler when combined with vpxPrint.
- A print and preview system for character applications from emulated sessions.
- A communication system: on a network, different systems can exchange their documentation files like Word, Excel spreadsheets. **xSpool** and associated programs open files as soon as received.
- …

EMULATOR CONNECTION:

One of the most common uses of **xSpool** is to retrieve files created from an emulated session from the PC (character applications on an UNIX system for example).

In this case, the problem is to know the ETHERNET address of the PC that started the current session. This is a prerequisite to create a spooled file.

How to get the ETHERNET address of the PC who started the current session:

UNIX systems are a lot different and provide specific results on shell commands.

However, the most useful command to get this information is the **finger** command.

```
# finger
Login    Name                    Tty   Idle   Login Time        Where
expe                            *p5      10   Tue Oct 29 07:27
```

The Ethernet address of the 'expe' user can be seen under the 'Where' column (SCO example).

```
# tty
/dev/ttyp5
```

Then, we just need to isolate in the finger result the line associated with /dev/ttyp5.

We provide **getIPaddr.p**, a sample program that returns this IP address from the shell commands:

```
/*  ====================================

    getIPaddr.p         (October 2002)
    -----------
    Get the IP Address of the session
    CopyRight 4GL & Marcel FONDACCI
    www.4GL.fr

    The subject of this program is to get the IP address of a pseudo-terminal.
    This situation occurs when PC are connected to a UNIX host.

    LINUX displays the IP address on 'who am I' statement but this is not
    supported by SCO UNIX and others.
    The logic of getIPaddr.p is the use of the 'finger' command.
    The 'finger' command provides a list of all active processes with pseudo-term
    identification and IP address. It seems to be understood by all systems.
    Different formats exist however in the display.
    We have build getIPaddr.p to take advantage of the title line of 'finger'
    to determine the position of tty and IP address. However, some differences
    may exist with different OS and you could have to adapt this program to
    meet your requirements.
```

```
    DISCLAIMER:
    This software is free and is distributed as-is. The author makes no warranties
    about the suitability of this software, either express or implied.
    The author and/or owner shall not be liable for any damages suffered as a
    result of using, modifying or distributing this software and application that
    uses it.
    ============================================================================*/
DEF VAR termid      AS CHAR format "x(20)" NO-UNDO.
DEF VAR fTitle      AS CHAR                NO-UNDO.
DEF VAR fLine       AS CHAR                NO-UNDO.
DEF VAR iTTY        AS INT                 NO-UNDO.
DEF VAR IIp         AS INT                 NO-UNDO.
DEF VAR fTTY        AS CHAR                NO-UNDO.
DEF VAR IPAddress   AS CHAR format "x(60)" NO-UNDO.

IF OPSYS <> "UNIX" THEN             /* Trap non-UNIX versions */
                RETURN '127.0.0.1'.

/*_________________Get the tty value _________________*/
INPUT  THROUGH     tty.
IMPORT UNFORMATTED termid.
INPUT  FROM TERMINAL.

/*_________________Get the current sessions__________*/
INPUT THROUGH "finger".                 /* the finger statement displays all sessions */
IMPORT UNFORMATTED fTitle.              /* get the title line                         */
/*                  Title line format   (subject to change with OS)
                    =================
    Login     Name          Tty    Idle     Login       Where
                            ^                          ^
                           *1A              master      192.168.0.12
                            ^-- Itty                    ^--- IIp
_____________________________________________________________________________________*/
iTTY    = INDEX(fTitle, "Tty").         /* The tty name is under tty literal          */
IIp     = INDEX(fTitle, "Where").       /* The IP address is under 'Where' literal    */
IIp     = IIp - iTTY + 1.               /* set IIp relative to iTTY               */

REPEAT :
    IMPORT UNFORMATTED fLine.
    fLine = SUBSTRING(fLine, iTTY).     /* start at ITTY                              */

    fTTY = SUBSTRING(FLine, 1, INDEX(fLine, " ") - 1).   /* the meaning part of tty   */

    IF termid MATCHES '*' + fTTY THEN DO:               /* prefix may change (OS)    */
          ipAddress = SUBSTRING(FLine, IIp).            /* if =, get the IP adress   */
          LEAVE.                                        /* and leave !               */
          END.
    END.

/*____________________________________________________________________*/

INPUT FROM TERMINAL.

RETURN ipAddress.
```

The **getIPaddr.p** program must be evaluated regarding your own environment and some adaptations should be done if the finger command does not return the same list than the one specified. It can be modified on **LINUX** systems because the '**who am I**' command directly returns the IP address.

How to do now?

The file creation process:

```
DEF VAR myIPaddr   AS CHAR NO-UNDO.
RUN getIPaddr.p.            /*       the return value contains the IP address */
MyIPaddr = replace(RETURN-VALUE, '.', '_').    /* replace . by _ */
OUTPUT STREAM O TO VALUE( "/users/spool/" + myIPaddr + "_myFile.xpr")
                              PAGED PAGE-SIZE 66.
PUT STREAM O CONTROL "<PREVIEW>".
PUT STREAM O
                    "<R20><C30><FARIAL><P20>Hello World !".

OUTPUT STREAM O CLOSE.
```

The file is created with a correct prefixed name under the local UNIX directory.

If the creation process is time-consuming, it may be that xSpoolretrieves this file **before the end of the building program**.

It's safe to create a temporary file with another name then rename it at the end of process.

```
DEF VAR myIPaddr   AS CHAR NO-UNDO.
RUN getIPaddr.p.            /*       the return value contains the IP address */
MyIPaddr = replace(RETURN-VALUE, '.', '_').    /* replace . by _ */
OUTPUT STREAM O TO VALUE( "/users/spool/myFile.xpr") PAGED PAGE-SIZE 66.
PUT STREAM O CONTROL "<PREVIEW>".
PUT STREAM O
                    "<R20><C30><FARIAL><P20>Hello World !".
OUTPUT STREAM O CLOSE.
OS-RENAME VALUE("/users/spool/myFile.xpr")
          VALUE( "/users/spool/" + myIPaddr + "_myFile.xpr").
```

With **xSpool**, PROGRESS characters users can see full-featured and polished printing directly from their own WINDOWS station.

Of course, this goal is reached when xSpool is used together with **vpxPrint** !

xSpool can be used in a lot of other situations like creating a centralized print server. You just have to replace the IP prefix by the IP prefix of the print server.

Enjoy with **xSpool**,

Marcel FONDACCI