Version 3.7
sep-2011

xSpool

Spooling system for UNIX and Windows…

http://www.4gl.fr/

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

At first, we wanted to exchange **vpxPrint** files between two systems. Our initial goal was to provide a solution for characters applications that a lot of PROGRESS users always use. The vpxPrint software didn't offer anything, except if a print server exists on the network.
((First, we have tried to do something to save old teletype writers…;-)))

When a connection exists through a PC and an Ethernet line, files exchange is easy.
The main principle of **xSpool** is quit simple:

1. The main program creates a file on the server machine, accessible to the PC stations
2. On the PC, a batch program scans this directory.
3. When a file is found, it sends it to the corresponding application and deletes it from the server

The problems to solve are:

1. The file can be created in a shared directory (SAMBA) available from the PC or the server does not have such capability. In this case, the client must use FTP protocol to retrieve files.
2. The PC must process files created for this PC and bypass other files. It quickly appears that the ETHERNET address is 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 session on the host ?
4. To write a file spooler on the PC. This program must be very flexible to support a large variety of combinations: shared device or FTP, UNIX or not, ASCII or BINARY, etc… and (of course) without any run-time royalty !

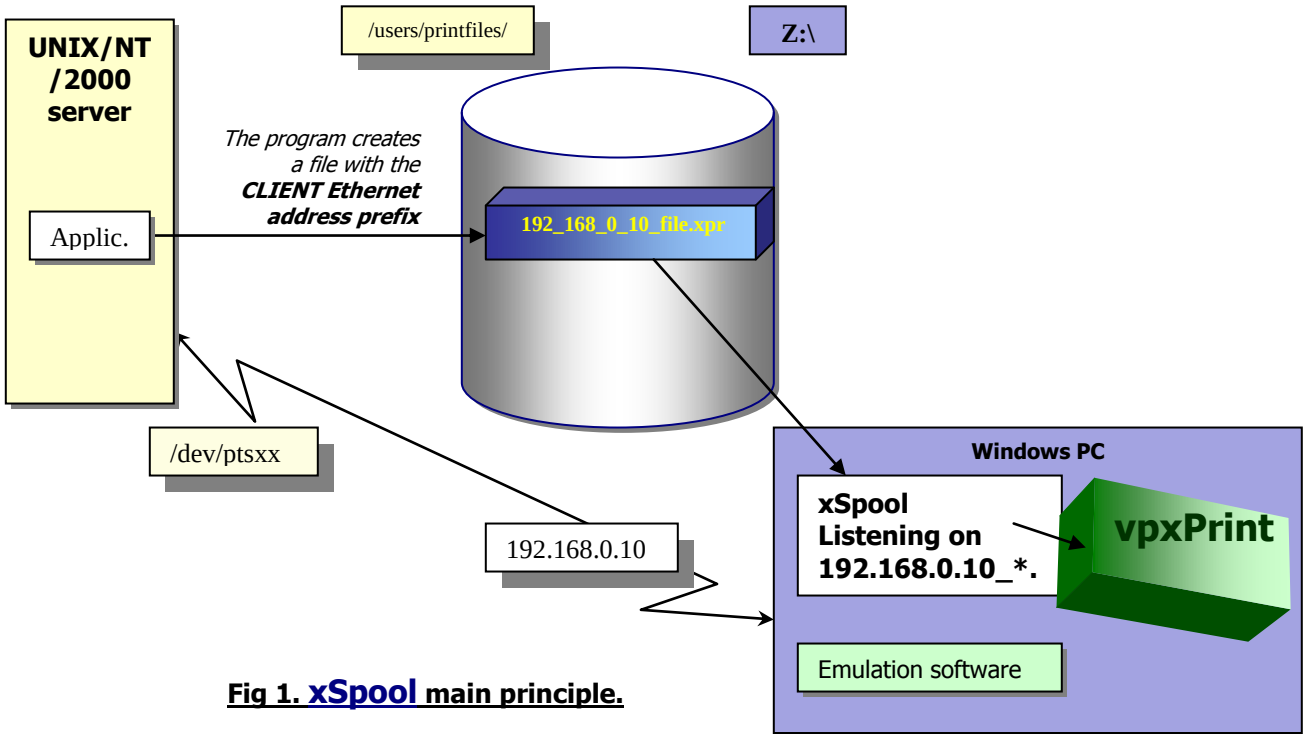

Fig 1. xSpool main principle.

History of changes:

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 fixed 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.

3-sep-2003: version 2.1

- **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 !
  Now, files retrieved must have an IP prefix separated from the suffix by a '_'.
  Files must be named 192_168_0_1_myfile.xpr

23-sep-2003: version 2.4 (cumulative 2.2,2.3)

- **xSpool** runs *.bat files,
- Automatic **unzip** of *.zip files to extract and proceed the files.

24-sep-2004: version 2.9

- new parameters: **-LIC and –QUIT**

19-sep-2005: version 3.2

- new parameters: **-PASSIVE**

7-november-2005: version 3.3

- Enhancements have been made to retry procedure in case of empty directories on the server.

September-2011: version 3.7

- support of V7-V9 users

xSpool is a self-executable file (xSpool.exe) that runs on a WINDOWS machine.
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=*host_address* or *host_name*	if the connection must establish a FTP connection
-USER=*user_name*	FTP user name
-PASS=*password*	FTP password
-DIR=*remote_or_local_directory_name*	the input directory. If case of FTP session, this directory is relative to the directory where the user is initially logged on.
-TEMP=*temporary_directory*	when FTP is used, points the local temp directory
[-INTERVAL=*time_interval*] default 2 seconds	# of seconds between two directory scans,
[-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*]	set 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 extensions, MS-Excel on .xls. Multiple programs can be called in one session of xSpool.) *program_name* is used to call a given program. **xPrint** calls the xPrint Dll on each input file.
[-PREFIX=*prefix_used_to_retrieve_files*]	specify the prefix of the files to be processed.
[-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.

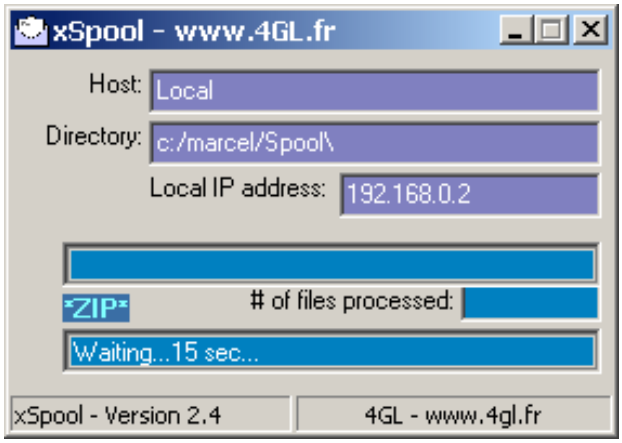

Fig 2. The xSpool screen

TECHNICAL CONSIDERATIONS:

The **xSpool** spooler program searches in the directory specified by –**dir** parameter for files matching the Internet address of the local machine. If local machine address is 192.168.0.163, then all files beginning with **192_168_0_163_** are processed ('.' Characters 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.

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 file are text files.

The default mode sends files received to the xPrint.dll or to 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.

APPLICATIONS:
A lot of applications can be considered with 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 problematic is: 'How can I know the ETHERNET address of the PC who 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 07:27 192.168.4.27
prod                         *p4   2:36  Tue Oct 29 05:30 pcdfx
root     Superuser           *1A         Tue Oct 29 08:06
```

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 the spooler retrieves 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