

## **Adding TCP/IP Support to VPS By Kenneth Tomiak**

Recently, a client asked me to enable the TCP/IP feature in their existing VPS R6.2 subsystem. VPS (VTAM Printer Support System), a Levi, Ray and Shoup, Inc. (LRS) product, delivers printed output from your JES spool to network-connected printers. One task called for educating the customer's staff on how to enable this feature. As I configured the subsystem and found a printer to test with, I made decisions that I hoped would ease the site's turnover and long-term support requirements. Based on this experience, I have provided a step-by-step guide to enabling the TCP/IP feature and testing the first IP-connected printer as well as insight into how to select PDS member names for your IP printers.

This article will examine six areas related to using VPS. First, I will demonstrate how to configure VPS to support TCP/IP. Next, I will present a simple REXX program for converting a four-part dotted decimal IP address to a valid MVS PDS member name. Then, I will present sample definitions of Postscript and PCL (Printer Command Language) capable printers. I will then explain some additional JES2 JCL parameters that might be used to influence the look of your printed output. Finally, I will explain two more REXX programs to help verify that the printer works and show how PCL commands can be used to affect the look of the output.

### **How to Configure VPS to Support TCP/IP**

First, I had to locate the PDS with the VPS Started Task JCL. If you need to search for the JCL like I did, then start with SYS1.PROCLIB(JES2) and then search in each DD statement. If you followed the vendor documentation you will be looking for a member named VPS. You also need to know where the startup option member is located. Find the SMEM= symbolic in the VPS JCL. This is the startup member found in the //VPSLIB PDS. Once you know where the member resides you can use ISPF 3.1 to make a copy of this member in the same PDS (e.g., NEWSTART). You should edit the copy so that you have a working member on which to fall back. When it's time to start VPS you can issue the "S VPS,SMEM=NEWSTART" command to use an alternate member. Once this is done you can make any changes specific to your installation. After you know that VPS will start with the alternate member you can rename the PDS member names, making NEWSTART the normally used name, VPSSTART. Figure 1 shows the parameters I added; they are explained as follows:

- KEYTCP=xx is the key provided by LRS to unlock the TCP/IP feature.
- TCPIPID=tcPIP is used to specify the name of the TCP/IP stack. Change the name here if you are not using the default name or want to use an alternate stack .
- TCPPORT=(721-731) provides a range of TCP/IP ports to use.  
-- If you send many small reports to the same printer in a short interval you will need both a large range of ports and TCPOPTS in the printer definition member. Review the LRS documentation on what values can be coded. Also review your TCP/IP Time-To-Live value if TCPOPTS does not work for you.
- TCPDMN = 'your.domain.name' is the DOMAIN name from TCP/IP's parameters (ask the TCP/IP staff for the location of this information, as it may vary). In my case, although the LRS reference manual I used referred me to tcPIP.TCPIP.DATA, the domain names used by this installation were kept in SYS1.TCPPARMS(TCPDATA).
- TCPTYPE=IBM310 specifies which type of interface to use. This works with IBM's TCP/IP stack Version 3 Release 1 and higher.

## **Support for VPS Exits and Translation Tables**

While you are in the VPS startup options member it is a good time to check whether EXIT08 (Error Retry) is enabled. This vendor-supplied exit will recover from some error conditions and redrive a print request up to three times, five minutes apart, before placing the printer in an EDRAINED state. Using a table of error conditions, redrive interval and retry attempts, this exit helps prevent printers from being placed out of service, or EDRAINED. You can easily change intervals or retry attempts to fit your installation's needs. EXIT08 is also a good exit to use for SNA printers. If you do not already use this exit, then locate the PDS provided by LRS with the "ASM" suffix. Then assemble and link the exit into your VPS load library.

Next, assemble and link EXIT14 (Dataset Add) to provide native PCL and Postscript writer support. These routines send control codes to the printer that affect the way the output is printed. You may want to research this topic some more, as you can enable such features as gray bar, two-up and booklet-framed to name just a few. Add a line in NEWSTART to enable each exit when you restart VPS, as shown below:

```
EXIT08=ENA,           ERROR RECOVERY  
EXIT14=ENA,           PCL & POSTSCRIPT PRINTER SUPPORT
```

When directing output from an IBM system that uses EBCDIC to a network printer that uses ASCII, data must be translated from EBCDIC to ASCII. It is important to note that ASCII code 27, the ESCape character, is used as a control code by printers and is a signal that printer commands follow. In some systems, hardware "black boxes" perform translation of the broken vertical bar symbol (EBCDIC code x'6A') to the ASCII ESCape character to allow control codes to be sent as part of the output data. The broken vertical bar does not allow passing control codes to the printer, it allows people who wrote programs to pass a character that they had on their keyboard. For VPS, EBCDIC-to-ASCII translation is controlled by member VPSXASC, found in the "ASM" dataset. To allow VPS to perform translation of the broken vertical bar to ESCape, you can modify the translate table, converting the character at location x'6A', which would normally be translated to ASCII x'7C' to ASCII x'1B' (decimal 27) instead. This allows programs that use the broken vertical bar in their output to control SNA printers to operate unchanged for TCP/IP printers. In my case, I saved the modified table under the name "BARSXASC." Control codes direct the printer to do things such as set the font, set the lines per page, and set line spacing. If your programs send control codes as part of their output, then make a copy of the member VPSSXASC, found in the ASM PDS, to some other name, e.g., BARSXASC. Make the changes shown in Figure 2, then assemble and link the translation table. The character at hex location x'6A' would normally be translated to x'7C'. Overtyping the '7C' value with '1B'. Now assemble and link your translate table. I'll show you where to use this table when you define a printer. Programs already written to use the broken-vertical bar in their output will continue to work without any changes if you let the translation table mimic the "black box" process.

## **How to Recycle the VPS Task**

Now that VPS start values and exits are in place it is time to recycle VPS. Make sure VPS is not printing anything and then bring VPS down if it is already running by using the "P VPS" command or use your VMCF dialog. With VPS down you may want to compress the VPS LINKLIB PDS after all of the linkedit work is done for the exits and refresh LLA if you need to. Now you can start VPS with "S VPS,SMEM=xxxxxxx" using the alternate options member name you edited earlier. Viewing the task's output should show your normal product license code and now the TCP/IP product license code. Next,

VPS will attach to TCP/IP. You now have a task that is dependent on VTAM and TCP/IP being active on your system. Make sure your automation package and operators know this. Once you are sure your VPS task options member is good, you can rename the member normally used, VPSSTART, for a backup and the alternate options member you just used to VPSSTART.

### **Converting a Four-Part Dotted Decimal IP Address to a Valid MVS PDS Member Name**

As you start defining IP-connected printers you will find you need to use a valid MVS PDS member name. SNA printers often get mapped to a member name using their LUNAME, whereas IP printers are not as lucky. The four-part dotted decimal IP address is not a valid MVS PDS member name. Two quick solutions are to name the printers based on the JES destid (e.g., U3204) or some other identifying name (e.g., department or person's name). However, there are other choices. I opted to convert the four-part dotted decimal IP address to hexadecimal values, which later will allow me to decipher where the printer is on the network by converting the hexadecimal value back to a four-part dotted decimal IP address. I wrote the REXX program in Figure 3 to convert each part from decimal to hex. If you convert 9.153.12.102 to hex you get 09 99 0C 66. That leading zero is not a valid beginning character for an MVS PDS member name. I would then replace the zero with an alphabetic character, such as "I", and string everything together to get I9990C66, a valid PDS member name to be used later. Why force the "I"? In days of old, VTAM devices were typically named with "D" for display and "P" for printer, a valid beginning character for MVS PDS member names. At the moment, I am able to use an "I" as the first character in the member name, since all of printers I need to define on my network have a value of "10" (hex '0A') as the first node in the IP address, so the first digit when the IP address is converted to hexadecimal is always zero.

It's easy to write a REXX program to take the eight-character name I9990C66 and convert it back to a four-part dotted decimal IP address. Contact me if you need help. The clue I'll give you is to SUBSTR two bytes at a time and convert hex to decimal (X2D). In my case, I bypass the first position. HEX2IP I9990C66 converts to 9.153.12.102. See Figure 2.

### **Examples of TCP/IP Printer Definitions**

Now that a valid MVS PDS member name has been created, let's configure some printers. I will show a printer configured to support PCL as printer S9990C66 and the same printer configured to support Postscript as printer I9990C66. However, I do not recommend this for every printer. Pick one style and stick with it. There are many more default Postscript styles to choose from than PCL but PCL may be easier to handle if you want to code some yourself. I prefer a Postscript configuration for JCL support. Let the programmers pass PCL directly as part of the program's output. You could replace a leading 0 through 9 character with a unique letter from the alphabet or select one letter, such as "I", for IP device. Since you should know what network your printers are on, perhaps converting your IP addresses to hex will never yield a leading numeric and then you can just use the converted name. If it does, you could replace the leading 0 through 9 character with another letter, like G though P.

Assuming you have an SNA-defined printer already, copy one of your SNA printer definition members into the member for the TCP/IP printer being defined, I9990C66. Some parameters are common to SNA and IP printers. Continue to use the ones that make sense. GRPNAME allows you to define a printer as part of a group. DEST defines the internal JES name for which to select output. In OS/390 V2R5 the internal JES2 name is in the range from U1 to U32767. I'll provide some sample definitions first and then in the JCL show you how to get the benefits of PCL or Postscript writers. I sent some output to each format and decided I like Postscript the best. PCL codes may still be sent embedded within a report.

The first printer definition, shown in Figure 3, will allow Postscript support using VPS EXIT 14. SEPINFO=PS will allow EXIT 14 to send pre-defined Postscript commands to the printer. DEVTYPE=V.TCPIP is the first parameter to define an IP printer. Another is the TCPHOST parameter in place of the LUNAME parameter found with an SNA printer. PRTXLATE defines the EBCDIC-to-ASCII translation table used. Since I wanted to support the broken-vertical bar translation I coded PRTXLATE=(Y,BARSXASC) using the EBCDIC-to-ASCII translate table described earlier. TCPOPTS defines some special options. You should refer to the LRS reference manual for all of the options. I coded X'02' to set a delay of 10 seconds between reports. This is necessary when sending many small reports to the same printer. Reports are sent to a HOST:PORT combination, which must be unique. Our start-up member lists the range of ports that can be used. Exceeding the HOST:PORT limit will result in an error. To avoid this, EXIT 08 is used to redrive the print request after waiting five minutes. Also, ask your TCP/IP systems programmer if it is possible to lower the time-to-live value.

The next printer definition, as shown in Figure 4, supports PCL code. By comparing these two definitions you will see that only the SEPINFO parameter has changed. The manual will explain additional values you can use. I found SEPINFO=HPL was sufficient to get the writers to work with EXIT 14.

### **The Additional JCL Parameters That Might Be Used**

The SEPINFO parameter allows you to have pre-defined styles that can be invoked via JCL. The JCL statement below is destined for a PCL printer. The SSTD writer listed tells VPS which member will be read from the VPS LINKLIB and sent to the printer ahead of the report. There are more Postscript choices than PCL. When it comes time to create your own, you may find PCL easier to code. I prefer the look of the vendor-provided Postscript writers, such as two-up graybar.

```
//REPORT DD SYSOUT=(A,SSTD),DEST=U5044
```

Most printers accept control codes to help format output. Hewlett Packard printers support PCL and often Postscript code as well. I first learned to pass control codes by way of BASIC programs. Having a PC-oriented ASCII Character Code table is invaluable for this sort of work. Your printer should have come with a manual that shows the common codes it will accept. Printers vary, so you may want to gather a manual of each type for future reference.

### **Two More REXX Programs**

In this section I will examine two more REXX programs to help verify that the printer works and show how PCL commands can be used to affect the look of the output. The REXX program in Figure 5 builds a small report to change the printer to LANDSCAPE, print some text, change the font, and print a few more lines. This is nothing fancy, but take note that I do not have PCL codes in separate records. Although printers will vary, my experience has been that PCL codes should be part of existing records or you may get unpredictable results. One example is an extra form feed when changing the orientation.

The REXX program in Figure 6, TESTDEST, reads a printer definition member and sends a test page requesting a confirmation call back. This is a convenient way to tailor output for a specific printer and get the user to verify that it works. For best results, you or the user need to fax a page to the other person to get final visual confirmation. The sample JCL in Figure 7 shows how to run the TSO program in batch that sets the condition code within the job to the code set in TESTDEST upon exiting.

## Conclusion

You should find adding support is not difficult. Make sure you implement VPS exits to keep the printers going, watch out for special translation requirements, review TCPOPTS and TCPPOPT values, and read the vendor manuals for release dependent information. And document what JCL values can be used at your installation to affect the look of output.

NaSPA member Kenneth Tomiak has more than 20 years in the Information Systems field. He is currently employed by Schunk and Associates Inc. Ken started as an operator trainee, moved into applications programming, and finally systems programming. Ken has worked with platform vendors including Honeywell, IBM, and DEC and installed software from ADSM through Xpediter, including AIX, CICS, DB2, IMS, MVS, NCP, OS/390, TCP/IP, VTAM, Windows, and many more supporting products on a range of platforms in many types of industries. Visit [WWW.Schunk-Associates.com](http://WWW.Schunk-Associates.com) to download the REXX programs.

Figure 1: Parameters Added to VPSSTART

KEYTCP=xx,	as provided by the vendor
TCPIPID=tcPIP,	name of your TCP/IP started task
TCPPOPT=(721-731),	up to outputs in two minutes
TCPDMN=your.domain.name,	from TCP/IP DOMAINORIGIN
TCPTYPE=IBM310,	version/release of TCP/IP

Figure 2: Convert Four-Part Dotted Decimal IP Address to Hex

```
+-----+
Name:      IP2HEX
Type:      REXX EXEC
Purpose:   Convert four part dotted decimal ip address to hex.
Release:   1.0.0
Programmer: Kenneth E. Tomiak
Date:      01999-01-14
Expectations: regular ip address is passed in.
Syntax:    TSO %IP2HEX ipaddr
           Replace ipaddr with your four part dotted decimal ip
           Address. Sample: 9.153.12.102
           Example: TSO %IP2HEX 9.153.12.102
           Output:   IP address 9.153.12.102 converts to 09 99 0C 66.
+-----+ */
PARSE UPPER ARG IPAddr
IF (IPAddr = "") then,
  Do
    Say "IPAddr is required."
    Exit 16
  End
IPAddr = Strip(IPAddr,"B"," ")
ip_as_hex = ""
PARSE VAR ipaddr part.1 '.' part.2 '.' part.3 '.' part.4
```

```

Do ix = 1 to 4
  ip_part = Strip(part.ix,"B"," ")
  If (ip_part = "") then,
    Do
      Say "IPaddr is a four part dotted decimal number like",
        "9.153.12.102"
    Exit 16
  End
  If (DataType(ip_part) <> "NUM") then,
    Do
      Say "IPaddr is a four part dotted decimal number like",
        "9.153.12.102"
      Say "fix" ip_part
    Exit 16
  End
  part = Right("00"d2x(part.ix),2)
  ip_as_hex = ip_as_hex part
End
Say "IP address" IPaddr "converts to" ip_as_hex"."
Exit 0

```

Figure 3: Sample Printer Definition for Post Script Support

```

*=====*
*
* I9990C66 - PRINTER AT CLIENT ACROSS FROM SHIPPING MANAGER
*
* IPADDR=9.153.12.102
*
*-----*
AUTOEJCT=(N,N,N,N),          ELIMINATE BLANK PAGE
AUTONL=N,
CRCHAR=0D,                   DEFAULT
COMPRESS=N,
CPI=0,                       DEFAULT
CKPTLINE=60,
CKPTPAGE=5,
CLASS=AZ,
DEST=U5044,
DEVTYPE=V.TCPIP,
GRPNAME=CLIENT,
FCB=Y,                       ALLOW FCB= IN JCL
*FORM=SSTD,                  LANDSCAPE 132 BY 60
  FORMFEED=Y,                DEFAULT
*FFSEQ=0D0C0D,              DEFAULT
  MAXLPG=0,                  DEFAULT
  MPP=0,                     DEFAULT
  NCCSPACE=1,                NO CARRIAGE CONTROL SINGLE SPACED
  NLSEQ=0D0A,
  PRTXLATE=(Y,BARSXASC),
  RELREQ=I,

```

Kenneth E. Tomiak  
 Schunk and Associates, Inc. (www.schunk-associates.com)  
 Kenneth.Tomiak@schunk-associates.com

```
SEPAR=(S,VPSSSEP5,DATASET),      BLOCK LETTER OUTPUT JCL INFO
SEPINFO=PS,                       SUPPORTS POSTSCRIPT VIA EXIT 14
SVFACTN=(N,N,N),
*WRITER=(,SSTD),
TCPMRD=15,                         MINUTES FOR TIMEOUT
TCPOPTS=02000080,
TCPHOST=9.153.12.102
```

Figure 4: Sample Printer Definition for PCL Support

```
*=====*
```

```
*
* S9990C66 - PRINTER AT CLIENT ACROSS FROM SHIPPING MANAGER
*
* IPADDR=9.153.12.102
*
*-----*
```

```
AUTOEJCT=(N,N,N,N),              ELIMINATE BLANK PAGE
AUTONL=N,
CRCHAR=0D,                       DEFAULT
COMPRESS=N,
CPI=0,                            DEFAULT
CKPTLINE=60,
CKPTPAGE=5,
CLASS=AZ,
DEST=U5045,
DEVTYPE=V.TCPIP,
GRPNAME=CLIENT,
FCB=Y,                            ALLOW FCB= IN JCL
*FORM=SSTD,                       LANDSCAPE 132 BY 60
  FORMFEED=Y,                      DEFAULT
*FFSEQ=0D0C0D,                   DEFAULT
  MAXLPG=0,                        DEFAULT
  MPP=0,                           DEFAULT
  NCCSPACE=1,                      NO CARRIAGE CONTROL SINGLE SPACED
  NLSEQ=0D0A,
  PRTXLATE=(Y,VPSSXASC),
  RELREQ=I,
  SEPAR=(S,VPSSSEP5,DATASET),      BLOCK LETTER OUTPUT JCL INFO
  SEPINFO=HPL,                     SUPPORTS PCL VIA EXIT 14
  SVFACTN=(N,N,N),
*WRITER=(,SSTD),
TCPMRD=15,                         MINUTES FOR TIMEOUT
TCPOPTS=02000080,
TCPHOST=9.153.12.102
```

Figure 5: REXX Program to Produce Test Output

```
/*          REXX PROGRAM          */
/*
  SMFID = MVSVAR("SYSNAME")
  SYSUID = USERID(); SAY "HELLO "SYSUID
```

```
QUEUE A2D("1B45") /*RESET */
QUEUE A2D('1B266C3153') /*SELECT DUPLEX PRINT- VERT BIND*/
QUEUE A2D('1B266C2B32333555') /*LEFT OFFSET=235 DECIPOINTS */
QUEUE A2D('1B266C3548') /*FEED FROM PAPER DECK */
QUEUE A2D('1B266C314F') /*LANDSCAPE */
QUEUE A2D('1B283855') /*ROMAN-8 SYMBOL SET */
QUEUE A2D('1B28733050') /*PRI SPACING IS FIXED */
QUEUE A2D('1B2873313548') /*PRI PITCH IS 15 */
QUEUE A2D('1B2873382E3556') /*PRI POINT SIZE IS 8.5 */
QUEUE A2D('1B28733053') /*PRI STYLE IS UPRIGHT */
QUEUE A2D('1B28733042') /*PRI STROKE WEIGHT IS NORMAL */
QUEUE A2D('1B28733054') /*PRI TYPEFACE IS LINE PRINTER */
QUEUE A2D('1B266C352E3237323743') /*VMI = 5.2727 */
QUEUE A2D('1B266B3848') /*HMI= 8 15 PITCH */
QUEUE A2D('1B266C3145') /*TOP MARGIN IS 1 */
QUEUE A2D('1B266130344C') /*LEFT MARGIN IS 04 */
QUEUE A2D('1B266C314C') /*ENABLE PERF SKIP */
QUEUE A2D('1B26733043') /*ENABLE END-OF-LINE WRAP */
QUEUE A2D('1B26613052') /*VERT CUR POS = ROW 0 */
QUEUE "ATTENTION VPS PRINTER USER"
QUEUE " "
QUEUE "PLEASE FAX THIS BACK TO ME"
QUEUE " "
QUEUE " "
QUEUE "THIS IS LANDSCAPE SMALL"
DO IX = 1 TO 5
  QUEUE COPIES("-----+-----",24)
END
QUEUE A2D('1B2836301B28733170372E307631733362323437303354')
QUEUE "THIS IS INTERNAL FONT #60:"
QUEUE "NEW CENTURY SCHOOLBOOK BOLD ITALIC - ON MY PRINTER"
QUEUE "POINT SIZE 7.0"
DO IX = 1 TO 5
  QUEUE COPIES("-----+-----",30)
END
DO IX = 1 TO 5
  QUEUE COPIES(" SOME TEXT TO TEST IF THIS IS A PROPORTIONAL FONT.",5)
END
DO IX = 1 TO 5
  QUEUE COPIES("-----+-----",30)
END
QUEUE A2D("1B45") /*RESET */
SIGNAL SPIT_IT_OUT

QUEUE A2D('1B266C334F2052657665727365204C616E647363617065')
QUEUE A2D('1B266C324F205265766572736520506F72747261697420')
QUEUE A2D('1B266C314F20526567756C6172204C616E647363617065')
QUEUE A2D('1B266C304F20526567756C617220506F72747261697420')
QUEUE "THIS IS LANDSCAPE SMALL"
QUEUE A2D('1B266C304F') /*PORTRAIT */
QUEUE "THIS IS PORTRAIT SMALL"
SPIT_IT_OUT:
QUEUE " "
```

Kenneth E. Tomiak  
Schunk and Associates, Inc. (www.schunk-associates.com)  
Kenneth.Tomiak@schunk-associates.com

```
"ALLOC FILE(NEWALLOC) SYSOUT(A) DEST(U5044) WRITER(DLET)",
"RECFM(V B) BLKSIZE(27998) LRECL(304) NOHOLD"
LAST_RC=RC
IF LAST_RC > 0 THEN,
  DO
    SAY "SYSOUT ERROR RC="LAST_RC
    SIGNAL END_IT
  END
"EXECIO * DISKW NEWALLOC (FINIS"
LAST_RC=RC
IF LAST_RC > 0 THEN,
  DO
    SAY "WRITE ERROR RC="LAST_RC
    SIGNAL END_IT
  END
END_IT:
"FREE FILE(NEWALLOC)"
EXIT LAST_RC
A2D:
PARSE ARG ASCII_TEXT
GO_AROUND = LENGTH(ASCII_TEXT)
IF (GO_AROUND // 2) <> 0 THEN RETURN ""
EBCDIC_TEXT = ""
DO IX = 1 TO GO_AROUND
  PAIR = SUBSTR(ASCII_TEXT, IX, 2)
  SELECT
    WHEN PAIR = "0A" THEN TXLTE = "0A"
    WHEN PAIR = "0B" THEN TXLTE = "0B"
    WHEN PAIR = "0C" THEN TXLTE = "0C"
    WHEN PAIR = "0D" THEN TXLTE = "0D"
    WHEN PAIR = "0E" THEN TXLTE = "0E"
    WHEN PAIR = "0F" THEN TXLTE = "0F"
    .....
    .....
    .....
    WHEN PAIR = "73" THEN TXLTE = "A2"
    WHEN PAIR = "74" THEN TXLTE = "A3"
    WHEN PAIR = "75" THEN TXLTE = "A4"
    WHEN PAIR = "76" THEN TXLTE = "A5"
    WHEN PAIR = "77" THEN TXLTE = "A6"
    WHEN PAIR = "78" THEN TXLTE = "A7"
    WHEN PAIR = "79" THEN TXLTE = "A8"
    WHEN PAIR = "BF" THEN TXLTE = "4A"
    WHEN PAIR = "F8" THEN TXLTE = "38"
    WHEN PAIR = "FE" THEN TXLTE = "AE"
    OTHERWISE NOP
  END
  IX=IX+1
  TXLTE=X2C(TXLTE)
  EBCDIC_TEXT=EBCDIC_TEXT||TXLTE
END
RETURN EBCDIC_TEXT
```

Kenneth E. Tomiak  
Schunk and Associates, Inc. (www.schunk-associates.com)  
Kenneth.Tomiak@schunk-associates.com

```
P2D:
  PARSE ARG BYTE_1 BYTE_2
  CHRS=""
  DO_ABLE = LENGTH(BYTE_1)
  DO IU = 1 TO DO_ABLE
    CHRS = CHRS||SUBSTR(BYTE_1,IU,1)||SUBSTR(BYTE_2,IU,1)
  END
  GETS_ME = A2D(CHRS)
  RETURN GETS_ME
```

Figure 6: REXX Program to Generate a Test Page

```
/* REXX -
+-----+
Name:      TESTDEST
Type:      REXX EXEC
Purpose:   Read SYSUT1 input file, extract list DESTID, output
           destination specific instructions for verifying
           the print works.
Release:   VPS R6.2+
Programmer: Kenneth E. Tomiak
Date:      01999-01-21
+-----+ */

  PARSE UPPER ARG cntl_dsn vps_contact trace
  IF (TRACE = "TRACE") then,
    DO
      TRACE I
    END
  Call Set_up
  Call Main_Line
  Call All_Done
  Exit 0

Set_up:
  x = MSG(OFF)
  bare_dsn = Strip(cntl_dsn,"B","")
  parse var bare_dsn just_dsn "(" just_mbr ")"
  quoted_dsn="'"bare_dsn'"
  mbr_ok = Sysdsn(quoted_dsn)
  if mbr_ok <> "OK" then,
    Do
      Say quoted_dsn "=" mbr_ok
      exit 16
    End
  "ALLOC FILE(VPSRTR) DATASET("quoted_dsn") SHR REUSE"
  last_rc=RC
  if last_rc > 0 then,
    Do
```

Kenneth E. Tomiak  
Schunk and Associates, Inc. (www.schunk-associates.com)  
Kenneth.Tomiak@schunk-associates.com

```
        Say quoted_dsn "alloc rc="last_rc
        exit 16
    End
last_trap=Outtrap("nomsg.")
"EXECIO * DISKR VPSRTR (FINIS Stem mbrdef."
last_rc=RC
last_trap=Outtrap(last_trap)
if last_rc > 0 then,
    Do
        Say quoted_dsn "diskr rc="last_rc
        exit 16
    End
if mbrdef.0 = 0 then,
    do
        Say quoted_dsn "is empty"
        exit 16
    End
Return
```

```
Main_line:
do ix = 1 to mbrdef.0
    If Left(mbrdef.ix,1) = "*" then Iterate
    parse var mbrdef.ix keyword "=" varval " " comment
    Select
        When keyword = "TCPHOST" then,
            Do
                TCPHOST = Strip(varval,"T",",")
            End
        When keyword = "DEST" then,
            Do
                UDEST = Strip(varval,"T",",")
            End
        When keyword = "CLASS" then,
            Do
                SCLASS = Strip(varval,"T",",")
            End
        When keyword = "SEPINFO" then,
            Do
                prtr_language = Left(varval,2) /* either PS or HP */
            End
        Otherwise,
            Do
                Nop
            End
    End
End
one_writer = "SSTD"
one_class = Left(sclass,1)
if prtr_language = "PS" then one_writer = "FPS13266"
Queue " Line 1" Copies(" ",29) "VPS IP PRINTER VERIFICATION"
Queue " Line 2"
Queue " Line 3 Please call" vps_contact
Queue " Line 4 to confirm this output has printed."
```

```
Queue " Line 5"
Queue " Line 6 This printer supports vendor provided print"
Queue " Line 7 format support using the WRITER parameter in JCL."
Queue " Line 8"
Queue " Line 9 Example:"
Queue " Line 10"
Queue " Line 11 //REPORT DD SYSOUT=("one_class","one_writer||,
    "),DEST="udest
Queue " Line 12"
Queue " Line 13",
"PCL writer formats sorted by name"
Queue " Line 14",
" DBOK Portrait 85x60 indented for 3-hole punch"
Queue " Line 15",
" DLET Portrait 72x54 indented for 3-hole punch"
Queue " Line 16",
" DSTD Landscape 132x66 indented for 3-hole punch"
Queue " Line 17",
" D132X100 Portrait 132x100"
Queue " Line 18",
" SBOK Portrait 85x60"
Queue " Line 19",
" SLET Portrait 72x54"
Queue " Line 20",
" SSTD Landscape 132x66 indented for 3-hole punch"
Queue " Line 21",
" S132X100 Portrait 132x100"
Queue " Line 22",
" ZZZRESET special writer to reset to defaults"
Queue " Line 23"
Queue " Line 24",
"Postscript writer formats sorted by name"
Queue " Line 25",
" FPS1GRYB Landscape 132x66 1up boxed graybar"
Queue " Line 26",
" FPS13266 Landscape 132x66 1up"
Queue " Line 27",
" FPS18560 Portrait 85x60 1up"
Queue " Line 28",
" FPS18566 Portrait 85x66 1up"
Queue " Line 29",
" FPS2GRYB Portrait 132x66 2up boxed graybar"
Queue " Line 30",
" FPS2UP60 Portrait 132x60 2up boxed"
Queue " Line 31",
" FPS2UP66 Portrait 132x66 2up boxed"
Queue " Line 32",
" FPS23260 Landscape 132x60 2up boxed/header pagecount"
Queue " Line 33",
" FPS23266 Landscape 132x66 2up boxed/header pagecount"
Queue " Line 34",
" FPS28560 Landscape 85x60 2up boxed/VPS pagecount"
Queue " Line 35",
```

Kenneth E. Tomiak  
Schunk and Associates, Inc. (www.schunk-associates.com)  
Kenneth.Tomiak@schunk-associates.com

```
" FPS28566 Landscape 85x66 2up boxed/VPS pagecount "  
Queue " Line 36",  
" FPS43260 Landscape 132x60 4up boxed/header pagecount "  
Queue " Line 37",  
" FPS43266 Landscape 132x66 4up boxed/header pagecount "  
Queue " Line 38",  
" FPS48560 Landscape 85x60 4up boxed/header pagecount "  
Queue " Line 39",  
" FPS48566 Landscape 85x66 4up boxed/header pagecount "  
Queue " Line 40",  
" L1BOK2C Portrait 85x60 2up boxed/header pagecount "  
Queue " Line 41",  
" L1BOK4C Portrait 85x60 4up boxed/header pagecount "  
Queue " Line 42",  
" L1GRY1C Landscape 132x66 1up graybar "  
Queue " Line 43",  
" L1SSTD1C Landscape 132x66 1up "  
Queue " Line 44",  
" L132002C Landscape 132x100 2up boxed/header pagecount "  
Queue " Line 45",  
" PGRYBR2C Portrait 132x66 2up graybar "  
Queue " Line 46",  
" PSBOK1C Portrait 85x60 1up "  
Queue " Line 47",  
" PSSTD2C Portrait 132x66 2up "  
Queue " Line 48",  
" P3200G1C Portrait 132x100 1up graybar "  
Queue " Line 49",  
" P32001C Portrait 132x100 1up "  
Queue " Line 50",  
" P32351C Portrait 132x135 1up "  
Queue " Line 51"  
Queue " Line 52"  
Queue " Line 53 Printer:" just_mbr "located at" TCPHOST  
if prtr_language = "PS" then,  
Queue " Line 54 This printer is configured for Postscript"  
Else,  
Queue " Line 54 This printer is configured for PCL"  
Queue " Line 55"  
Queue " Line 56 You can still send your own codes if you know how"  
Queue " Line 57"  
Queue " Line 58 Routed:" Date() Time()  
Queue " Line 59"  
Queue " Line 60 This is te last line that should appear on Page 1"  
Return
```

```
All_Done:  
"ALLOC FILE(OUTDD) SYSOUT("one_class"),  
"DEST("udest") WRITER("one_writer")"  
LAST_RC=RC  
IF LAST_RC > 0 THEN,  
DO  
SAY "SYSOUT ERROR RC="LAST_RC
```

Kenneth E. Tomiak  
Schunk and Associates, Inc. (www.schunk-associates.com)  
Kenneth.Tomiak@schunk-associates.com

```
SIGNAL END_IT  
END  
Queue ""  
"EXECIO * DISKW OUTDD"  
Return
```

Figure 7: JCL to Run TESTDEST Program

```
//jobname JOB (acct-info), 'programmer-name',  
//          CLASS=jobclass,MSGCLASS=sysout-class  
//*  
//STEP001 EXEC PGM=IKJEFT1B  
//SYSTSPRT DD  SYSOUT=*  
//SYSPROC  DD  DISP=SHR,DSN=your.clist.library  
//SYSTSIN  DD  *  
%TESTDEST VPS.R62.CNTL(I9990C66) `Ken Tomiak @ 973-727-4664`  
%TESTDEST VPS.R62.CNTL(H9990C66) `Ken Tomiak @ 973-727-4664`  
/*  
//
```