Help and usage of PRSCRDRV utility

B. Satyanarayana
High Energy Cosmic Ray Group

April 1993

PRSCRDRV is a screen dump utility implemented under the MSDOS device driver program format. This software driver also supports reading and setting of its working configuration, an on-line help as well as reading screen dump and help function status.

1 Main features

Features such as on-line help and user-defined character support are optional, thus allowing the user to configure PRSCRDRV according to the need, memory constraints etc. PRSCRDRV Version 2.1 fully supports both 9 and 24-dot matrix printers. It supports and exhibits a uniform behaviour on MDA, CGA, MCGA, EGA, VGA and XGA screens. PRSCRDRV constructs its working environment based on the configuration file provided on the command-line, information read- out from the currently active display adapter as well as the derived information from both of these. PRSCRDRV determines the active screen type - graphics or text, dynamically at run-time. Full ASCII character set - including the control code block (0-31) as well as the box character block (128-255) is supported in case of text screens. A user-defined character set replaces the control code block, thus providing a true dump of IBM text screens.

LPT port number on the PC to which the printer is connected as well as the printer graphics density mode may be specified. It is possible to dump a portion of a text or graphics screen. The printed image can be of positive or negative logic and can be oriented as a landscape or portrait. Horizontal scale ratio of the image is programmable. Multiple screen dump is supported and if requested, the images can be stacked horizontally across the printer page or vertically one below the other. Top and left margins as well as the inter image (in case of multiple screen dump) gaps are programmable, thus providing flexibility in placing the image on the printer page. One of the character colour attributes used in a text screen may also be specified for achieving the background effect in the hardcopy for the text area displayed using the given attribute.

2 Installation

PRSCRDRV can be easily installed into any computer environment. The steps involved are:

1. Copying the driver as well as the optional configuration, help, environment handler and other document files into the system.
2. Appending the line,

   device=[drive:][path]PRSCRDRV.BIN[switches] to the CONFIG.SYS file.

   Valid switches are:

   (a) /cfg=Configuration file

   (b) /hlp=Help file

   (c) /prn=Printer type number

   If specified, configuration and help file names should also include proper paths.
3. Rebooting the system. Installation and/or error messages are displayed during booting. If loaded properly, the utility is ready for use.

If a configuration file name is specified, its contents override the defaults built into the system during booting. Either this, or any other text file may be specified as the help file. If the help file size is more than 13K, then the excess is ignored. Specifying a help file also enables the utility to display error message texts on text screens, if needed.

Following is the list of printer types recognised by PRSCRDRV. User can safely specify an equivalent recognised printer type if one wishes to use a printer not listed here. If the printer type is not specified or an unknown printer type is specified, then user-defined character feature is not supported.

Type	Make/Model
00	EPSON EX-1000
01	EPSON LX-800
02	EPSON LQ-1050
03	EPSON LQ-850
04	EPSON UP-130K
05	EPSON VP-500
06	EPSON FP-80
07	SIVA DL3400
99	Unknown printer

3 Operation

PRSCRDRV can be invoked from keyboard, from any programming environment or at MSDOS prompt.

Shift+PrtSc hot-key combination is used to invoke a screen dump. The on-line help may be requested by *PrtSc* key. It may be noted here that, help operation cannot be invoked by this method while using a 83/84 key keyboard becuase of its layout constraints.

In a programming environment, this utility is invoked through the filename *PRSCR*. Application programs can handle this file in a manner similar to that of other character/sequential files/devices. Before use, the driver is opened as:

```
open "prscr" for output as # n   (in BASIC)

fp = fopen ("prscr", "w+t");     (in C)
setbuf (fp, NULL);
```

Then a hardcopy operation is performed by a file character write statement, such as:

```
print # n, "C";    (in BASIC)

fputc ('C', fp);   (in C)
```

where, "C" is a non-null character.

And finally the driver is closed as follows:

```
close # n       (in BASIC)

fclose (fp);    (in C)
```

Help may be requested in a programming environment by issuing a file character read statement after opening the *PRSCR* for input, such as:

```
input # n, i    (in BASIC)

fgetc (fp);     (in C)
```

An alternate way to operate PRSCRDRV is by directly calling the Print-Screen Service Interrupt; Int 5, with the function code in AH register. This is examplified by the following C code.

```
#include <dos.h>
union REGS regs;
     ...
regs.h.ah = 'P';         ... Screen dump
int86 (5, &regs, &regs);
     ...
regs.h.ah = 'H';         ... Help
int86 (5, &regs, &regs);
```

In this case, the application program doesnot need to handle the PRSCRDRV as a file or device and hence need not be opened or closed.

Lastly, PRSCRDRV can also be invoked from MSDOS prompt. Though for screen dump operation it is of little practical use, this method is useful to invoke help, especially on systems with 83/84 key keyboards. The command syntax is as follows:

- For screen dump:

```
copy con:prscr
Enter
Any  character
CTRL+Z
Enter
```

- For help:

```
copy prscr nul

     or

type prscr
```

It is convenient to keep these commands in a batch file.

A screen dump or help operation in progress may be aborted by pressing ESC key.

4 Environment and communication

Operation of the PRSCRDRV is controlled by a configuration structure, which essentially contains read/write and read only segments. The structure can be read or written by the application program via MSDOS IOCTL functions (Int 21h, Function: 44h, Subfunctions: 2 and 3) or their equivalents. Size and constituent elements of the structure to be defined in an application can be varied according to the user need. Full definition of the structure used by the PRSCRDRV is given below for reference. User can use the langauge specific syntax to define the structure as well as to communicate with the driver.

```
pscfg   struc

; Read/write segment
```

```
chkflag  dw   'Y' ; Parameter check flag ([Y]es/[N]o)
prnmode  dw   1   ; Printer mode
iorient  dw   'P' ; Image orientation ([L]andscape/[P]ortrait)
topmarg  dw   0   ; Top margin (mm)
lftmarg  dw   18  ; Left margin (mm)
graphxs  dw   0   ; Start X-pixel for graphic screen
graphys  dw   0   ; End X-pixel
graphxe  dw   639 ; Start Y-pixel
graphye  dw   349 ; End Y-pixel
nscrpag  dw   1   ; Number of screen pages
svidpag  dw   0   ; Start video page number
imaggap  dw   10  ; Gap between images (mm)
stktype  dw   'V' ; Stack type ([H]orizontal/[V]ertical)
textxs   dw   0   ; Start column for text screen
textys   dw   0   ; End column
textxe   dw   79  ; Start row
textye   dw   24  ; End row
prnbgc   dw   15  ; Character colour attribute for background effect
hexpsr   dw   2   ; Horizontal scale expansion ratio
ilogic   dw   'P' ; Image type ([P]ositive/[N]egative)
lptnum   dw   1   ; LPT port number

; Read only segment

cfgfile  db   32 dup (0)          ; Configuration file name
hlpfile  db   32 dup (0)          ; Help file name
prntype  db   2 dup (space)       ; Printer type
prnmake  db   'Unknown printer',0 ; Printer make/model
prndots  dw   9                   ; Number of printer head needles
pagewid  dw   80                  ; Number of printer columns
prnmds   db   7, 0, 1, 2, 3, 4, 5, 6, 8 dup (0) ; Printer modes
vidmode  dw   16                  ; Current video mode
scrtype  dw   'G'                 ; Current screen type([G]raphic/[T]ext)
nvpages  dw   2                   ; Number of video pages
columns  dw   80                  ; Number of text columns
rows     dw   25                  ; Rows
xpixels  dw   640                 ; Number of graphic X-pixels
ypixels  dw   350                 ; Y-pixels
cvpage   dw   0                   ; Current active page number

pscfg   ends
```

An application program, PRSCRENV distributed along with the driver, may be used to create new configuration files, read and change resident driver's configuration, edit and print the same and so on.

5 Parameter checking and error handling

When a PRSCRDRV operation is invoked with the *chkflag* turned ON ('Y'), then the parameters supplied through the configuration structure are checked against the acquired and derived system parameters. If any input parameter is found illegal, then an error code is returned and the operation is aborted. Following is the list of error codes returned and their meanings. However, the end message texts received may be different compared to those given here due to additional translation performed by the langauge environment in use.

In case of operating PRSCRDRV through direct interrupt calls (INT 5), the error code is returned in AL register. When invoked by *PrtSc* key, an alarm is issued on error. Error message text is also displayed if help option has been installed and if a text screen is currently active. In either of these types of operations, *PrtSc* return status location, 0000:0050 is updated. Codes expected at this location are 0 for no error, 1 for operation in progress and 15 for error.

Error	PRSCRDRV meaning	Original MSDOS meaning
0	No error	Write-protect violation
1	Wrong LPT port number	Unknown unit
2	Printer power is off	Drive not ready
3	Unknown command	Unknown command
4	Invalid parameter(s)	Data error (CRC)
6	Unknown video mode number	Seek error
7	No help on non-text screens	Unknown medium
9	Printer out of paper	Printer out of paper
10	Printer is off-line	Write fault
11	Help utility not installed	Read fault
12	Function aborted by user	General failure

6 Performance

PRSCRDRV offers a host of new features compared to similar packages already available. Minimum resident memory required by this utility is about 2.8K, which is same on any type of screen. MSDOS's GRAPHICS command for example needs 5.2K, 11.5K and 10.9K of memory on CGA, EGA and VGA screens respectively. PRSCRDRV screen dumps are faster by about 20%. PRSCRDRV offers a complete control on the screen dump operation from the calling process in terms of selecting the required configuration, the operation as well as receiving the return status. It also supports a variety of calling mechanisms which are suitable in any programming environment.

Resident memory requirements of PRSCRDRV in different configurations are given below:

Memory	Remarks
2.9K	No help, no user-defined character support
21.3K	Help, no user-defined character support
21.6K	Help, 9-pin user-defined character support
25.1K	Help, 24-pin user-defined character support

(Assuming size of the help file to be the maximum allowed by PRSCRDRV)

7 Calling syntax in PRSCRDRV

PRint SCReen device DRiVer Version 2.1 supports the following functions:

- Screen dump
- On-line help
- Reading current configuration
- Changing current configuration
- Reading screen dump operation status
- Reading on-line help operation status

Following is the description of calling syntax as well as return status for these functions in various programming environments. There are essentially two methods by which the screen dump and on-line help operations may be invoked. Since the operating system recognises PRSCRDRV as a device named *prscr*, the above mentioned operations can be invoked by openning the device and by issuing a character/byte write or read calls respectivey. This is called the file I/O method. Alternately, it is also possible to invoke these functions directly by calling the Print Screen Service Interrupt (Int 5), because PRSCRDRV also traps this interrupt and executes the functions when this interrupt is generated.

Rest of the functions are invoked either by using the IOCTL functions of the langauge, if supported or by calling the MSDOS IOCTL function (44H) directly. Some more varieties in calling syntax are possible depending on the fact whether the langauge supports direct calls to software interrupts or not. It is advisable to include an error handler routine to the main program using PRSCRDRV so as to trap the error codes, if any returned by PRSCRDRV. It may be noted here that aborting a screen dump operation in progress by pressing ESC key generates a critical error which needs special treatment in terms of error recovery.

7.1 Hot-keys

Screen dump and on-line help operations are invoked by *Shift+PrtSc* and *PrtSc* hot-key combinations respectively. In both the cases return status code of the operation is available at address 0000:0050. If the byte at this address has a value of FFH, then a previous operation was not completed successfully. A value of 00H indicates that no error occurred, and that the operation is ready to go. A value of 01H indicates that the operation is currently in progress; any request for a second one will be ignored.

On error, a beep is given and the error message will also be displayed on the screen if the on-line help option is installed and the current video mode is of text type. Rest of the functions are not suitable to be invoked by hot-key combinations.

7.2 MSDOS prompt

Screen dump operation is invoked by typing the following at the MSDOS command prompt:

```
copy con:prscr
Enter
Any character
CTRL+Z
Enter
```

On-line help operation is invoked by issuing one of the following MSDOS internal commands:

```
copy prscr nul

     or

type prscr
```

It is convenient to keep these commands in batch files. MSDOS displays an appropriate error message, if any. The final message text displayed may be different from the corresponding PRSCRDRV message due to the transalation performed by MSDOS.

Execute PRSCRENV from the command prompt for handling the PRSCRDRV configuration functions. It handles these and other related functions using user-friendly menus as well as data, help, input and status windows. Please note that PRSCRENV handles only read/write segment - doesn't include read only segment of configuration structure of PRSCRDRV. Rest of the functions are not supported by PRSCRENV and hence cannot be invoked from MSDOS prompt.

7.3 Assembly langauge

There is little reason to anticipate the possibility of calling a PRSCRDRV function directly from an assembly langauge application program. However, the following program segment should serve as an example for calling some of the PRSCRDRV functions from high-level languages such as Fortran or GW-BASIC which donot support MSDOS interrupt calls.

In fact, the following program is an independent module which when interfaced to Fortran or BASIC does support all the PRSCRDRV functions. Refer sections 7.5 and 7.6 for more details on interfacing and calling conventions.

```
        title   PRSRCDRV interface to high-level languages
        public  prscr

pscode  segment 'code'
```

```
        assume  cs: pscode
prscr   proc    far
        mov     bp, sp
        mov     bx, 8[bp]
        mov     cl, ds:[bx]       ; Function code
        mov     si, 6[bp]         ; Data buffer addresss
        mov     di, 4[bp]         ; Return status address
        mov     cs:[prvss], ss    ; Save previous SS and SP
        mov     cs:[prvsp], sp
        mov     ax, psstak
        mov     ss, ax            ; Use local stack area
        mov     sp, offset stktop
        push    ds
        mov     ax, psdata        ; Get handle for 'prscr'
        mov     ds, ax
        mov     dx, offset psfile
        mov     ax, 3d02h
        int     21h
        mov     bx, ax
        cmp     cl, 1             ; Help or screen dump functions?
        jg      prscrct
        mov     dx, offset datbuf ; Yes; execute
        mov     ah, cl
        add     ah, 3fh
        mov     cx, 1
        int     21h
        pop     ds
        jmp     prscrex
prscrct:pop     ds
        mov     ah, 44h
        cmp     cl, 3             ; Read or change configuration functions?
        jg      prscrst
        mov     al, cl            ; Yes; execute
        mov     cx, 42
        mov     dx, si
        int     21h
        jmp     prscrex
prscrst:cmp     cl, 5             ; Help or screen dump status functions?
        jg      prscrex           ; No; illegal function code
        mov     al, cl            ; Yes; execute
        add     al, 2
        int     21h
        mov     ds:[di], al
prscrex:mov     ah, 3eh
```

```
        int     21h
        mov     ss, cs:[prvss]    ; Load previous SS and SP
        mov     sp, cs:[prvsp]
        ret     6
prscr   endp
pscode  ends

psdata  segment 'data'
        assume  ds:psdata
psfile  db      'prscr', 0
prvss   dw      ?
prvsp   dw      ?
datbuf  db      'C'
psdata  ends

psstak  segment 'stack'
        assume  ss:psstak
        dw      32 dup (0)
stktop  label   word
psstak  ends

        end
```

7.4 C-langauge

Screen dump and on-line help operations are invoked by using file I/O syntax as given in the following example.

```
#include <stdio.h>
File *fp;
char c = 'C';                /* 'C' is any non-null character */
   ...
fp = fopen ("prscr", "w+t"); /* For screen dump operation */
fp = fopen ("prscr:, "r+t"); /* For on-line help operation */
setbuf (fp, NULL);
   ...
fputc (c, fp);               /* For screen dump operation */
c = getc (fp);               /* For on-line help operation */
   ...
```

```
fclose (fp);
```

The language environment receives the error code, if any from PRSCRDRV and may perform necessary translation before returning the same to the program.

Screen dump and on-line help operations are invoked by calling Print Screen Service Interrupt as given in the following example.

```
#include <dos.h>
union REGS regs;
   ...
regs.h.ah = 'P'; /* For screen dump operation */
regs.h.ah = 'H'; /* For on-line help operation */
   ...
int86 (5, &regs, &regs);
```

Error codes, if any are returned to the calling process in register AL.

IOCTL function may be used to read and change the PRSCRDRV's current configuration as well as to read the status of its screen dump and on-line help operations. The program syntax is shown below.

```
#include <io.h>
#include <fcntl.h>
int handle, status, curcfg [21];
  ...
handle = open ("prscr", O_RDWR);
status = ioctl (handle, 2, curcfg, 42); /* Read current configuration */
   ...
status = ioctl (handle, 3, curcfg, 42); /* Change current configuration */
 ...
status = ioctl (handle, 6); /* Get on-line help operation status */
   ...
status = ioctl (handle, 7); /* Get screen dump operation status */
   ...
close (handle);
```

For the first two functions, *status* gives the number of bytes of data actually transferred. For the last functions, it indicates the status of the concerned operation. Status = 00H means the operation is in progress; status = FFH means it is idle.

7.5 Fortran

Screen dump and on-line help operations are requested as follows:

```
 character c
 open (n, access = 'sequential', file = 'prscr', form = 'binary',
+mode = 'readwrite', status = 'old')
    ...
 write (n) 'C'
    ...
 read (n) c
    ...
 close (n)
```

The language environment receives the error code, if any from PRSCRDRV and may perform necessary translation before returning the same to the program.

Rest of the operations are not suitable to be called directly from Fortran as it doesn't allow software interrupt calls. However, any of the operations may be called from Fortran by adding a machine langauge interface to it and implementing the calling sequences in the same. Refer section 7.3 for an example of such interface. Refer Fortran manuals for interfacing details.

7.6 GW-BASIC

Screen dump and on-line help operations are requested as follows:

```
open "prscr" for output as # n    ' For screen dump operation
open "prscr" for input as # n     ' For on-line help operation
   ...
```

```
print # n, "C";     ' For screen dump operation ['C' is non-null character]
i = input # n       ' For on-line help operation
   ...
close # n
```

The language environment receives the error code, if any from PRSCRDRV and may perform necessary translation before returning the same to the program.

Rest of the operations are not suitable to be called directly from GW-BASIC as it doesn't allow software interrupt calls. Though it supports IOCTL functions, the syntax is not suitable to be used with PRSCRDRV. However, any of the operations may be called from GW-BASIC by adding a machine langauge interface to it and implementing the calling sequences in the same. Refer section 7.3 for an example of such interface.

The program given in section 3 should be compiled and linked. The EXE file thus created should be loaded into high memory using DEBUG utility and BSAVEd to a BIN file by executing GW-BASIC. The following program segment may be then be used to invoke any of the PRSCRDRV functions from GW-BASIC. Refer to MSDOS and GW-BASIC manuals more details.

```
10 DEF SEG = &H7024 ' Define segment for loading machine code
20 DEFINT I-N
30 DIM ICTRL (21) ' Buffer for holding PRSCRDRV configuration data
40 BLOAD "PRSCR.BIN", 0 ' Loading machine code
50 PRSCR = &H100 ' Procedure offset address
60 IFUNCTION = N ' Codes for various PRSCRDRV functions are given below
70 ISTATUS = 0
80 CALL PRSCR (IFUNCTION, ICTRL (0), ISTATUS) ' Calling convention
90 END
```

Value of N in line number 60 is defined as follows:

Value	Function
0	Invoke on-line help
1	Invoke screen dump
2	Read PRSCRDRV configuration data to ICTRL buffer
3	Replace PRSCRDRV configuration with that specifed in ICTRL buffer
4	Read on-line help operation status into ISTATUS
5	Read screen dump operation status into ISTATUS

7.7 PASCAL

Screen dump and on-line help operations are invoked by using file I/O syntax as given in the following example.

```
Program PSPASCAL;
Var
   PSfile: File of Char;
   PSdata: Char;
Begin
     PSdata := 'C';
     Assign (PSfile, 'prscr');
     ...
(* For screen dump operation *)
     Rewrite (PSfile);
     Write (PSfile, PSdata);
     ...
(* For on-line help operation *)
     Reset (PSfile);
     Read (PSfile, PSdata);
     ...
     Close (PSfile);
End.
```

The language environment receives the error code, if any from PRSCRDRV and may perform necessary translation before returning the same to the program.

Screen dump and on-line help operations are invoked by calling Print Screen Service Interrupt as given in the following example.

```
Program PSPASCAL;
Uses DOS;
Var
   Regs: Registers;

Begin
     Regs.ah := Ord ('P'); (* For screen dump operation *)
     Regs.ah := Ord ('H'); (* For on-line help operation *)
     Intr (5, Regs);
End.
```

Error code, if any is returned to the calling process in Regs.al.

Rest of the functions can be implemented in PASCAL through IOCTL function (44H) of MSDOS interrupt (21H) directly as shown in the following example.

```
Program PSPASCAL;
Uses DOS;
Var
   Regs:   Registers;
   PSfile: String;
   Buffer: Array [1..21] of Integer;
   Handle,
   Service,
   i:      Integer;
   ...
Begin
     PSfile := 'PRSCR' + #0;
     With Regs Do
     Begin
          Al := 2;
          Ah := $3d;
          Dx := Ofs (PSfile) + 1;
          Ds := Seg (PSfile);
     End;
     Msdos (Regs);
     Handle := Regs.ax;
     ...
     Service := n (* n = 2 for reading current configuration *)
                  (* n = 3 for changing current configuration *)
```

```
                  (* n = 6 for reading on-line help operation status *)
                  (* n = 7 for reading screen dump operation status *)
     ...
     With Regs Do
     Begin
          Al := Service;
          Ah := $44;
          Bx := Handle;
          Cx := 42;
          Dx := Ofs (Buffer [1]); (* Necessary for n = 2 or 3 *)
          Ds := Seg (Buffer);     (* Necessary for n = 2 or 3 *)
     End;
     Msdos (Regs);
     (* For n = 6 or 7, Regs.al gives the status *)
     (* When PRSCRDRV is free, then AL = FFH else AL = 0 *)
     ...
     Regs.ah := $3e;
     Regs.bx := Handle;
     Msdos (Regs);
End.
```

7.8 Quick BASIC

Screen dump and on-line help operations are invoked by using file I/O syntax as given in the following example.

```
open "prscr" for output as # n   ' For screen dump operation
open "prscr" for input as # n    ' For on-line help operation
   ...
print # n, "C";     ' For screen dump operation ['C' is non-null character]
i = input # n       ' For on-line help operation
   ...
close # n
```

The language environment receives the error code, if any from PRSCRDRV and may perform necessary translation before returning the same to the program.

Screen dump and on-line help operations are invoked by calling Print Screen Service Interrupt as given in the following example.

```
' $INCLUDE: 'qb.bi'
DIM inregs AS RegType, outregs AS RegType
   ...
inregs.ax = ASC("P") * 256 ' For screen dump operation
inregs.ax = ASC("H") * 256 ' For on-line help operation
CALL INTERRUPT (5, inregs, outregs)
```

(outregs.ax mod 256) is the error code returned to the calling process.

Though Quick BASIC offers IOCTL and IOCTL$ functions to send and receive control data from the device drivers, the syntax is not suitable to be used with PRSCRDRV. Instead, it is advisable to implement these functions using the MSDOS function 44H directly as shown below.

```
' $INCLUDE: 'qb.bi'
DEFINT I-N
DIM inregs AS RegTypeX, outregs AS RegTypeX
DIM ictrl(21) 'PRSCRDRV configuration buffer
psdev$ = "prscr" + CHR$(0)
   ...
inregs.ds = VARSEG(psdev$)
inregs.dx = SADD(psdev$)
inregs.ax = &H3D02
CALL interruptx(&H21, inregs, outregs)
ihandle = outregs.ax
   ...
ifunction = n 'where n = 2 for reading current configuration
             '      n = 3 for changing current configuration
             '      n = 6 for reading on-line help operation status
             '      n = 7 for reading screen dump operation status
   ...
inregs.bx = ihandle
inregs.cx = 42
inregs.ax = ifunction + 256 * &H44
inregs.dx = VARPTR(ictrl(0)) ' necessary for n = 2 or 3
inregs.ds = VARSEG(ictrl(0)) ' necessary for n = 2 or 3
```

```
CALL interruptx(&H21, inregs, outregs)
' For n = 6 or 7, (outregs.ax MOD 256) gives the status
, When PRSCRDRV is free, then AL = FFH else AL = 0
   ...
inregs.bx = ihandle
inregs.ax = &H3E00
CALL interruptx(&H21, inregs, outregs)
```