Path Descriptors and Device Descriptors

This appendix includes the device descriptor initialization table definitions and path descriptor option tables for RBF, SCF, SBF, and PIPEMAN type devices. Refer to Appendix A for RBF, SCF, and SBF example device descriptors.

RBF Device Descriptor Modules

This section describes the definitions of the initialization table contained in device descriptor modules for RBF-type devices. The table immediately follows the standard device descriptor module header fields (see Chapter 3 for full descriptions). Figure B-1 shows a graphic representation of the table. The size of the table is defined in the M$Opt field.

Name **Description**

PD_DTP **Device type**

This field is set to one for RBF devices. (0=SCF, 1=RBF, 2=PIPE, 3=SBF, 4=NET)

PD_DRV **Drive number**

Use this field to associate a one-byte logical integer with each drive that a driver/controller handles. Number each controller's drives 0 to n-1 (n is the maximum number of drives the controller can handle and is set into V_NDRV by the driver's INIT routine). This number defines which drive table the driver and RBF access for this device. RBF uses this number to set up the drive table pointer (PD_DTB). Prior to initializing PD_DTB, RBF verifies that PD_DRV is valid for the driver by checking for a value less than V_NDRV in the driver's static storage. If not valid, RBF aborts the path open and returns an error. On simple hardware, this logical drive number is often the same as the physical drive number.

NAME DESCRIPTION

PD_STP **Step rate**

This field contains a code that sets the drive's head-stepping rate. To reduce access time, set the step rate to the fastest value of which the drive is capable. For floppy disks, the following codes are commonly used:

Step Code	5" Disks	8" Disks
0	30ms	15ms
1	20ms	10ms
2	12ms	6ms
3	6ms	3ms

For hard disks, the value in this field is usually driver dependent.

PD_TYP **Disk type**

Defines the physical type of the disk, and indicates the revision level of the descriptor.

If bit 7 = 0, floppy disk parameters are described in bits 0-6:

bit 0: 0 = 5 1/4" floppy disk (pre-Version 2.4 of OS-9)
1 = 8" floppy disk (pre-Version 2.4 of OS-9)

bits 1-3: 0 = (pre-Version 2.4 descriptor) Bit 0 describes type/rates.
1 = 8" physical size
2 = 5 1/4" physical size
3 = 3 1/2" physical size
4-7: Reserved

bit 4: Reserved

bit 5: 0 = Track 0, side 0, single density
1 = Track 0, side 0, double density

bit 6: Reserved

If bit 7 = 1, hard disk parameters are described in bits 0-6:

bits 0-5: Reserved

bit 6: 0 = Fixed hard disk
1 = Removable hard disk

Name **Description**

PD_DNS **Disk density ***

The hardware density capabilities of a floppy disk drive:

bit 0:	0 =	Single bit density (FM)
	1 =	Double bit density (MFM)
bit 1:	1 =	Double track density (96 TPI/135 TPI)
bit 2:	1 =	Quad track density (192 TPI)
bit 3:	1 =	Octal track density (384 TPI)

PD_CYL **Number of cylinders (tracks) ***

The logical number of cylinders per disk. Format uses this value, PD_SID, and PD_SCT to determine the size of the drive. PD_CYL is often the same as the physical cylinder count (PD_TotCyls), but can be smaller if using partitioned drives (PD_LSNOffs) or track offsetting (PD_TOffs).

If the drive is an autosize drive (PD_Cntl), format ignores this field.

PD_SID **Heads or sides ***

The number of heads for a hard disk (Heads) or the number of surfaces for a floppy disk (Sides). If the drive is an autosize drive (PD_Cntl), format ignores this field.

PD_VFY **Verify flag**

Indicates whether or not to verify write operations.

0 = verify disk write
1 = no verification

NOTE: Write verify operations are generally performed on floppy disks. They are not generally performed on hard disks because of the lower soft error rate of hard disks.

PD_SCT **Default sectors/track***

The number of sectors per track. If the drive is an autosize drive (PD_Cntl), format ignores this field.

PD_T0S **Default sectors/track (track 0) ***

The number of sectors per track for track 0. This may be different than PD_SCT (depending on specific disk format). If the drive is an autosize drive (PD_Cntl), format ignores this field.

*** These parameters are format specific.**

Name **Description**

PD_SAS **Segment allocation size**

The default minimum number of sectors to be allocated when a file is expanded. Typically, this is set to the number of sectors on the media track (for example, 8 for floppy disks, 32 for hard disks), but can be adjusted to suit the requirements of the system.

PD_ILV **Sector interleave factor ***

The sequential arrangement of sectors on a disk (for example, 1, 2, 3... or 1, 3, 5...). For example, if the interleave factor is 2, the sectors are arranged by 2's (1, 3, 5...) starting at the base sector (see PD_SOffs).

NOTE: Optimized interleaving can drastically improve I/O throughput.

NOTE: PD_ILV is typically only used when the media is formatted, as format uses this field to determine the default interleave. However, when the media format occurs (I$SetStat, SS_WTrk call), the desired interleave is passed in the parameters of the call.

PD_TFM **DMA (Direct Memory Access) transfer mode**

The mode of transfer for DMA access, if the driver is capable of handling different DMA modes. Use of this field is driver dependent.

PD_TOffs **Track base offset ***

The offset to the first accessible physical track number. Track 0 is not always used as the base track because it is often a different density.

PD_SOffs **Sector base offset ***

The offset to the first accessible physical sector number on a track. Sector 0 is not always the base sector.

*** These parameters are format specific.**

Name **Description**

PD_SSize **Sector size**

Indicates the physical sector size in bytes. The default sector size is 256. Depending upon whether the driver supports non-256 byte logical sector sizes (that is, a variable sector size driver), the field is used as follows:

- **Variable sector size driver**
  If the driver supports variable logical sector sizes, RBF inspects this value during a path open (specifically, after the driver returns “no error” on the SS_VarSect GetStat call) and uses this value as the *logical* sector size of the media. This value is then copied into PD_SctSiz of the path descriptor options section, so that application programs can know the logical sector size of the media, if required. RBF supports logical sector sizes from 256 bytes to 32,768 bytes, in integral binary multiples (256, 512, 1024, etc.).

  During the SS_VarSect call, the driver can validate or update this field (or the media itself) according to the driver’s conventions. These typically are:

  ¿ If the driver can dynamically determine the media’s sector size, and PD_SSize is passed in as 0, the driver updates this field according to the current media setting.

  ¡ If the driver can dynamically set the media’s sector size, and PD_SSize is passed in as a non-zero value, the driver sets the media to the value in PD_SSize (this is typical when re-formatting the media).

  ¬ If the driver cannot dynamically determine or set the media sector size, it usually validates PD_SSize against the supported sector sizes, and returns an error (E$SectSiz) if PD_SSize contains an invalid value.

- **Non-variable sector size driver**
  If the driver does not support variable logical sector sizes (that is, logical sector size is fixed at 256 bytes), RBF ignores PD_SSize. In this case, PD_SSize can be used to support deblocking drivers that support various physical sector sizes.

  NOTE: A non-variable sector sized driver is defined as a driver which returns the E$UnkSvc error for GetStat (SS_VarSect).

Name **Description**

PD_Cntl **Device control word**

Indicates options that reflect the capabilities of the device. You may set these options, as follows:

bit 0:	0 = Format enable 1 = Format inhibit
bit 1:	0 = Single-Sector I/O 1 = Multi-Sector I/O capable
bit 2:	0 = Device has non-stable ID 1 = Device has stable ID
bit 3:	0 = Device size determined from descriptor values 1 = Device size obtained by SS_DSize GetStat call
bit 4:	0 = Device cannot format a single track 1 = Device can format a single track
bits 5-15:	Reserved

PD_Trys **Number of tries**

Indicates whether a driver should try to access the disk again before returning an error. Depending upon the driver in use, this field may be implemented as a flag or a retry counter:

Value	Flag	Counter
0	retry ON	default number of retries
1	retry OFF	no retries
other	retry ON	specified number of retries

Drivers that work with controllers that have error correcting functions (for example, E.C.C. on hard disks) should treat this field as a flag so they can set the controller's error correction/retry functions accordingly.

When formatting media, especially hard disks, the format-enabled descriptor should set this field to one (retry OFF) to ensure that marginal media sections are marked out of the media free space.

Name **Description**

PD_LUN **Logical unit number of SCSI drive**

Used in the SCSI command block to identify the logical unit on the SCSI controller. To eliminate allocation of unused drive tables in the driver static storage, this number may be different from PD_DRV. PD_DRV indicates the logical number of the drive to the driver, that is, the drive table to use. PD_LUN is the physical drive number on the controller.

PD_WPC **First cylinder to use write precompensation**

The cylinder to begin write precompensation.

PD_RWR **First cylinder to use reduced write current**

The cylinder to begin reduced write current.

PD_Park **Cylinder used to park head**

The cylinder at which to park the hard disk's head when the drive is shut down. Parking is usually done on hard disks when they are shipped or moved and is implemented by the SS_SQD SetStat to the driver.

PD_LSNOffs **Logical sector offset**

The offset to use when accessing a partitioned drive. The driver adds this value to the logical block address passed by RBF prior to determining the physical block address on the media. Typically, using PD_LSNOffs is mutually exclusive to using PD_TOffs.

PD_TotCyls **Total cylinders on device**

The actual number of physical cylinders on a drive. It is used by the driver to correctly initialize the controller/drive. PD_TotCyls is typically used for physical initialization of a drive that is partitioned or has PD_TOffs set to a non-zero value. In this case, PD_CYL denotes the *logical* number of cylinders of the drive. If PD_TotCyls is zero, the driver should determine the physical cylinder count by using the sum of PD_CYL and PD_TOffs.

PD_CtrlrID **SCSI controller ID**

The ID number of the SCSI controller attached to the drive. The driver uses this number to communicate with the controller.

Name **Description**

PD_ScsiOpt **SCSI driver options flags**

The SCSI device options and operation modes. It is the driver's responsibility to use or reject these values, as applicable.

bit 0: 0 = ATN not asserted (no disconnect allowed)
1 = ATN asserted (disconnect allowed)

bit 1: 0 = Device cannot operate as a target
1 = Device can operate as a target

bit 2: 0 = Asynchronous data transfer
1 = Synchronous data transfer

bit 3: 0 = Parity off
1 = Parity on

All other bits are reserved.

PD_Rate **Data transfer/rotational rate**

The data transfer rate and rotational speed of the floppy media. Note that this field is normally used only when the physical size field (PD_TYP, bits 1-3) is non-zero.

bits 0-3: Rotational speed

0 = 300 RPM
1 = 360 RPM
2 = 600 RPM

All other values are reserved.

bits 4-7: Data transfer rate

0 = 125K bits/sec
1 = 250K bits/sec
2 = 300K bits/sec
3 = 500K bits/sec
4 = 1M bits/sec
5 = 2M bits/sec
6 = 5M bits/sec

All other values are reserved.

PD_MaxCnt **Maximum transfer count**
The maximum byte count that the driver can transfer in one call. If this field is 0, RBF defaults to the value of $ffff (65,535).

NOTE: ***Offset*** refers to the location of a module field, relative to the starting address of the static storage area. Offsets are resolved in assembly code by using the names shown here and linking the module with the relocatable library, sys.l or usr.l.

Device Descriptor Offset	Path Descriptor Label	Description
$48	PD_DTP	Device Class
$49	PD_DRV	Drive Number
$4A	PD_STP	Step Rate
$4B	PD_TYP	Device Type
$4C	PD_DNS	Density
$4D		Reserved
$4E	PD_CYL	Number of Cylinders
$50	PD_SID	Number of Heads/Sides
$51	PD_VFY	Disk Write Verification
$52	PD_SCT	Default Sectors/Track
$54	PD_T0S	Default Sectors/Track 0
$56	PD_SAS	Segment Allocation Size
$58	PD_ILV	Sector Interleave Factor
$59	PD_TFM	DMA Transfer Mode
$5A	PD_TOffs	Track Base Offset
$5B	PD_SOffs	Sector Base Offset
$5C	PD_SSize	Sector Size (in bytes)
$5E	PD_Cntl	Control Word
$60	PD_Trys	Number of Tries
$61	PD_LUN	SCSI Unit Number of Drive
$62	PD_WPC	Cylinder to Begin Write Precompensation
$64	PD_RWR	Cylinder to Begin Reduced Write Current
$66	PD_Park	Cylinder to Park Disk Head
$68	PD_LSNOffs	Logical Sector Offset
$6C	PD_TotCyls	Number of Cylinders On Device

$6E	PD_CtrlrID	SCSI Controller ID
$6F	PD_Rate	Data transfer/Disk Rotation Rates
$70	PD_ScsiOpt	SCSI Driver Options Flags
$74	PD_MaxCnt	Maximum Transfer Count

Figure B-1: Initialization Table for RBF Device Descriptor Modules

RBF Definitions of the Path Descriptor

The first 26 fields of the path options section (PD_OPT) of the RBF path descriptor are copied directly from the device descriptor standard initialization table. All of the values in this table may be examined using I$GetStt by applications using the SS_Opt code. Some of the values may be changed using I$SetStt; some are protected by the file manager to prevent inappropriate changes. You can update the following fields using GetStat and SetStat system calls:

PD_STP
PD_CYL
PD_SCT
PD_TYP
PD_SID
PD_TOS
PD_DNS
PD_VFY
PD_SAS

All other fields are read-only. The RBF path descriptor option table is shown on the following page.

Refer to the previous section on RBF device descriptors for descriptions of the first 26 fields. The last five fields contain information provided by RBF:

Name **Description**

PD_ATT **File attributes** (D S PE PW PR E W R)
The file’s attributes are defined as follows:

bit 0: Set if owner read.

bit 1: Set if owner write.

bit 2: Set if owner execute.

bit 3: Set if public read.

bit 4: Set if public write.

bit 5: Set if public execute.

bit 6: Set if only one user at a time can open the file.

bit 7: Set if directory file.

PD_FD **File descriptor**
The LSN (Logical Sector Number) of the file’s file descriptor is written here.

PD_DFD **Directory file descriptor**
The LSN of the file’s directory file descriptor is written here.

PD_DCP **File’s directory entry pointer**
The current position of the file’s entry in its directory.

PD_DVT **Device table pointer (copy)**
The address of the device table entry associated with the path.

Name	**Description**
PD_SctSiz	**Logical sector size** The logical sector size of the device associated with the path. If this is 0, assume a size of 256 bytes.
PD_NAME	**File name**

NOTE: In the following chart, ***offset*** refers to the location of a path descriptor field relative to the starting address of the path descriptor. Path descriptor offsets are resolved in assembly code by using the names shown here and linking with the relocatable library: sys.l or usr.l.

Figure B-2: Option Table for RBF Path Descriptor

Offset	Name	Description
$80	PD_DTP	Device Class
$81	PD_DRV	Drive Number
$82	PD_STP	Step Rate
$83	PD_TYP	Device Type
$84	PD_DNS	Density
$85		Reserved
$86	PD_CYL	Number of Cylinders
$88	PD_SID	Number of Heads/Sides
$89	PD_VFY	Disk Write Verification
$8A	PD_SCT	Default Sectors/Track
$8C	PD_TOS	Default Sectors/Track 0
$8E	PD_SAS	Segment Allocation Size
$90	PD_ILV	Sector Interleave Factor
$91	PD_TFM	DMA Transfer Mode
$92	PD_TOffs	Track Base Offset
$93	PD_SOffs	Sector Base Offset
$94	PD_SSize	Sector Size (in bytes)
$96	PD_Cntl	Control Word
$98	PD_Trys	Number of Tries
$99	PD_LUN	SCSI Unit Number of Drive

$9A	PD_WPC	Cylinder to Begin Write Precompensation
$9C	PD_RWR	Cylinder to Begin Reduced Write Current
$9E	PD_Park	Cylinder to Park Disk Head
$A0	PD_LSNOffs	Logical Sector Offset

Offset	Name	Description
$A4	PD_TotCyls	Number of Cylinders On Device
$A6	PD_CtrlrID	SCSI Controller ID
$A7	PD_Rate	Data Transfer/Rotational Rates
$A8	PD_ScsiOpt	SCSI Driver Option Flag
$AC	PD_MaxCnt	Maximum Transfer Count
$B0		Reserved
$B5	PD_ATT	File Attributes
$B6	PD_FD	File Descriptor
$BA	PD_DFD	Directory File Descriptor
$BE	PD_DCP	File’s Directory Entry Pointer
$C2	PD_DVT	Device Table Pointer (copy)
$C6		Reserved
$C8	PD_SctSiz	Logical Sector Size
$CC		Reserved
$E0	PD_NAME	File Name

SCF Device Descriptor Modules

Device descriptor modules for SCF-type devices contain the device address and an initialization table which defines initial values for the I/O editing features, as listed below. The initialization table immediately follows the standard device descriptor module header fields (see Chapter 3 for full descriptions). The size of the table is defined in the M$Opt field. The initialization table is graphically shown in Figure B-3 and the following table.

NOTE: You can change or disable most of these special editing functions by changing the corresponding control character in the path descriptor. You can do this with the I$SetStt service request or the tmode utility. A permanent solution may be to change the corresponding control character value in the device descriptor module. You can easily change the device descriptors with the xmode utility.

Name **Description**

PD_DTP **Device type**
Set to zero for SCF devices. (0=SCF, 1=RBF, 2=PIPE, 3=SBF, 4=NET)

PD_UPC **Letter case**
If PD_UPC is not equal to zero, input or output characters in the range “a..z” are made “A..Z”.

PD_BSO **Destructive backspace**
If PD_BSO is zero when a backspace character is input, SCF echoes PD_BSE (backspace echo character). If PD_BSO is non-zero, SCF echoes PD_BSE, space, PD_BSE.

PD_DLO **Delete**
If PD_DLO is zero, SCF deletes by backspace-erasing over the line. If PD_DLO is not zero, SCF deletes by echoing a carriage return/line-feed.

PD_EKO **Echo**
If PD_EKO is not zero, all input bytes are echoed, except undefined control characters, which are printed as periods. If PD_EKO is zero, input characters are not echoed.

PD_ALF **Automatic line feed**
If PD_ALF is not zero, line-feeds automatically follow carriage returns.

PD_NUL **End of line null count**
Indicates the number of NULL padding bytes to send after a carriage return/line-feed character.

PD_PAU **End of page pause**

If PD_PAU is not zero, an auto page pause occurs upon reaching a full screen of output. See PD_PAG for setting page length.

Name **Description**

PD_PAG **Page length**

Contains the number of lines per screen (or page).

PD_BSP **Backspace “input” character**

Indicates the input character recognized as backspace. See PD_BSE and PD_BSO.

PD_DEL **Delete line character**

Indicates the input character recognized as the delete line function. See PD_DLO.

PD_EOR **End of record character**

Defines the last character on each line entered (I$Read, I$ReadLn). An output line is terminated (I$Writln) when this character is sent. Normally PD_EOR should be set to $0D. **WARNING:** If PD_EOR is set to zero, SCF’s I$ReadLn will *never* terminate, unless an EOF or error occurs.

PD_EOF **End of file character**

This field defines the end-of-file character. SCF returns an end-of-file error on I$Read or I$ReadLn if this is the first (and only) character input.

PD_RPR **Reprint line character**

If this character is input, SCF (I$ReadLn) reprints the current input line. A carriage return is also inserted in the input buffer for PD_DUP (see below) to make correcting typing errors more convenient.

PD_DUP **Duplicate last line character**

If this character is input, SCF (I$ReadLn) duplicates whatever is in the input buffer through the first PD_EOR character. Normally, this is the previous line typed.

PD_PSC **Pause character**

If this character is typed during output, output is suspended before the next end-of-line. This also deletes any “type ahead” input for I$ReadLn.

PD_INT **Keyboard interrupt character**

If this character is input, SCF sends a keyboard interrupt signal to the last user of this path. It terminates the current I/O request (if any) with an error identical to the keyboard interrupt signal code. PD_INT is normally set to a control-C character.

PD_QUT **Keyboard abort character**

If this character is input, SCF sends a keyboard abort signal to the last user of this path. It terminates the current I/O request (if any) with an error code identical to the keyboard abort signal code. PD_QUT is normally set to a control-E character.

Name **Description**

PD_BSE **Backspace "output" character (echo character)**

This field indicates the backspace character to echo when PD_BSP is input. See PD_BSP and PD_BSO.

PD_OVF **Line overflow character**

If I$ReadLn has satisfied its input byte count, SCF ignores any further input characters until an end-of-record character (PD_EOR) is received. It echoes the PD_OVF character for each byte ignored. PD_OVF is usually set to the terminal's bell character.

PD_PAR **Parity code, number of stop bits and bits/character**

Bits zero and one indicate the parity as follows:

0 = no parity
1 = odd parity
3 = even parity

Bits two and three indicate the number of bits per character as follows:

0 = 8 bits/character
1 = 7 bits/character
2 = 6 bits/character
3 = 5 bits/character

Bits four and five indicate the number of stop bits as follows:

0 = 1 stop bit
1 = 1 1/2 stop bits
2 = 2 stop bits

Bits six and seven are reserved.

PD_BAU **Software adjustable baud rate**

This one-byte field indicates the baud rate as follows:

0 =	50 baud	6 =	600 baud	C =	4800 baud
1 =	75 baud	7 =	1200 baud	D =	7200 baud
2 =	110 baud	8 =	1800 baud	E =	9600 baud
3 =	134.5 baud	9 =	2000 baud	F =	19200 baud
4 =	150 baud	A =	2400 baud	10 =	38400 baud
5 =	300 baud	B =	3600 baud	FF =	External

Name **Description**

PD_D2P **Offset to output device descriptor name string**

SCF sends output to the device named in this string. Input comes from the device named by the M$PDev field. This permits two separate devices (a keyboard and video display) to be one logical device. Usually PD_D2P refers to the name of the same device descriptor in which it appears.

PD_XON **X-ON character**

See PD_XOFF below.

PD_XOFF **X_OFF character**

The X-ON and X-OFF characters are used to support software handshaking. Output from a SCF device is halted immediately when PD_XOFF is received and does not resume until PD_XON is received. This allows the distant end to control its incoming data stream. Input to a SCF device is controlled by the driver. If the input FIFO is nearly full, the driver sends PD_XOFF to the distant end to halt input. When the FIFO has been emptied sufficiently, the driver resumes input by sending the PD_XON character. This allows the driver to control its incoming data stream.

NOTE: When software handshaking is enabled, the driver consumes the PD_XON and PD_XOFF characters itself.

PD_Tab **Tab character**

In I$WritLn calls, SCF expands this character into spaces to make tab stops at the column intervals specified by PD_Tabs. **NOTE:** SCF does not know the effect of tab characters on particular terminals. Tab characters may expand incorrectly if they are sent directly to the terminal.

PD_Tabs **Tab field size**

See PD_Tab.

NOTE: ***Offset*** refers to the location of a module field, relative to the starting address of the module. Module offsets are resolved in assembly code by using the names shown here and linking the module with the relocatable library: sys.l or usr.l.

Device Descriptor Offset	Path Descriptor Label	Description
$48	PD_DTP	Device Type
$49	PD_UPC	Upper Case Lock
$4A	PD_BSO	Backspace Option
$4B	PD_DLO	Delete Line Character
$4C	PD_EKO	Echo
$4D	PD_ALF	Automatic Line Feed
$4E	PD_NUL	End Of Line Null Count
$4F	PD_PAU	End Of Page Pause
$50	PD_PAG	Page Length
$51	PD_BSP	Backspace Input Character
$52	PD_DEL	Delete Line Character
$53	PD_EOR	End Of Record Character
$54	PD_EOF	End Of File Character
$55	PD_RPR	Reprint Line Character
$56	PD_DUP	Duplicate Line Character
$57	PD_PSC	Pause Character
$58	PD_INT	Keyboard Interrupt Character
$59	PD_QUT	Keyboard Abort Character
$5A	PD_BSE	Backspace Output
$5B	PD_OVF	Line Overflow Character (bell)
$5C	PD_PAR	Parity Code, # of Stop Bits, and # of Bits/Character
$5D	PD_BAU	Adjustable Baud Rate
$5E	PD_D2P	Offset To Output Device Name
$60	PD_XON	X-ON Character
$61	PD_XOFF	X-OFF Character
$62	PD_TAB	Tab Character
$63	PD_TABS	Tab Column Width

Figure B-3: Device Descriptor Initialization Table

SCF Definitions of the Path Descriptor

The first 27 fields of the path options section (PD_OPT) of the SCF path descriptor are copied directly from the SCF device descriptor initialization table. The table is shown on the following page.

You can examine or change the fields with the I$GetStt and I$SetStt service requests or the tmode and xmode utilities.

You may disable the SCF editing functions by setting the corresponding control character value to zero. For example, if you set PD_INT to zero, there is no "keyboard interrupt" character.

NOTE: Full definitions for the fields copied from the device descriptor are available in the previous section. The additional path descriptor fields are defined below:

Name	Description
PD_TBL	**Device table entry** A user-visible copy of the device table entry for the device.
PD_COL	**Current column** The current column position of the cursor.
PD_ERR	**Most recent error status** The most recent I/O error status.

NOTE: ***Offset*** refers to the location of a module field, relative to the starting address of the module. Module offsets are resolved in assembly code by using the names shown here and linking the module with the relocatable library: sys.l or usr.l.

Offset	Name	Description
$80	PD_DTP	Device Type
$81	PD_UPC	Upper Case Lock
$82	PD_BSO	Backspace Option
$83	PD_DLO	Delete Line Character
$84	PD_EKO	Echo
$85	PD_ALF	Automatic Line Feed
$86	PD_NUL	End Of Line Null Count
$87	PD_PAU	End Of Page Pause
$88	PD_PAG	Page Length
$89	PD_BSP	Backspace Input Character
$8A	PD_DEL	Delete Line Character
$8B	PD_EOR	End Of Record Character
$8C	PD_EOF	End Of File Character
$8D	PD_RPR	Reprint Line Character
$8E	PD_DUP	Duplicate Line Character
$8F	PD_PSC	Pause Character
$90	PD_INT	Keyboard Interrupt Character
$91	PD_QUT	Keyboard Abort Character
$92	PD_BSE	Backspace Output
$93	PD_OVF	Line Overflow Character (bell)
$94	PD_PAR	Parity Code, # of Stop Bits, and # of Bits/Character
$95	PD_BAU	Adjustable Baud Rate
$96	PD_D2P	Offset To Output Device Name
$98	PD_XON	X-ON Character
$99	PD_XOFF	X-OFF Character
$9A	PD_TAB	Tab Character
$9B	PD_TABS	Tab Column Width
$9C	PD_TBL	Device Table Entry
$A0	PD_Col	Current Column
$A2	PD_Err	Most Recent Error Status

$A3	Reserved

Figure B-4: Path Descriptor Module Option Table for I/O Editing

SBF Device Descriptor Modules

This section describes the definitions of the initialization table contained in device descriptor modules for SBF devices. The initialization table immediately follows the standard device descriptor module header fields (see Chapter 3 for full descriptions). A graphic representation of the table is shown in Figure B-5. The size of the table is defined in the M$Opt field.

Device Descriptor Offset	Path Descriptor Label	Description
$48	PD_DTP	Device Type
$49	PD_TDrv	Tape Drive Number
$4A	PD_SBF	Reserved
$4B	PD_NumBlk	Maximum Number of Blocks to Allocate
$4C	PD_BlkSiz	Logical Block Size
$50	PD_Prior	Driver Process Priority
$52	PD_SBFFlags	SBF Path Flags
$53	PD_DrivFlag	Driver Flags
$54	PD_DMAMode	Direct Memory Access Mode
$56	PD_ScsiID	SCSI Controller ID
$57	PD_ScsiLUN	LUN on SCSI Controller
$58	PD_ScsiOpts	SCSI Options Flags

Figure B-5: Initialization Table for SBF Device Descriptor Module

NOTE: In this table the offset values are the device descriptor offsets, while the labels are the path descriptor offsets. To correctly access these offsets in a device descriptor using the path descriptor labels, make the following adjustment: (M$DTyp - PD_OPT).

For example, to access the tape drive number in a device descriptor, use the following value: PD_TDrv + (M$DTyp - PD_OPT). To access the tape drive number in the path descriptor, use PD_TDrv. Module offsets are resolved in assembly code by using the names shown here and linking with the relocatable library: sys.l or usr.l.

Name **Description**

PD_DTP **Device class**

This field is set to three for SBF devices. (0=SCF, 1=RBF, 2=PIPE, 3=SBF, 4=NET)

PD_TDrv **Tape drive number**

This is used to associate a one-byte integer with each drive that a controller will handle. If using dedicated (for example, non-SCSI bus) controllers, this field usually defines both the *logical* and *physical* drive number of the tape drive. If using tape drives connected to SCSI controllers, this number defines the *logical* number of the tape drive to the device driver. The *physical* controller ID and LUN are specified by the PD_ScsiID and PD_ScsiLUN fields. Each controller's drives should be numbered 0 to n-1 (n is the maximum number of drives the controller can handle). This number also defines how many drive tables are required by the driver and SBF. SBF verifies this number against SBF_NDRV prior to calling the driver.

PD_NumBlk **Number of buffers/blocks used for buffering**

Specifies the maximum number of buffers to be allocated by SBF for use by the auxiliary process in buffered I/O. If this field is set to 0, unbuffered I/O is specified.

PD_BlkSiz **Logical block size used for I/O**

Specifies the size of the buffer to be allocated by SBF. This buffer size is used when allocating multiple buffers used in buffered I/O. Unless the driver manages partial physical blocks, this size should be an integer multiple of the physical tape block size.

PD_Prior **Driver process priority**

The priority at which SBF's auxiliary process will run. This value is used during initialization. Changing this value after initialization has no effect.

PD_SBFFlags **SBF path flags**

Specifies the actions that SBF takes when the path is closed. You can update this field using GetStat/SetStat (SS_Opt). SBF supports the following flag definitions:

bit 0: (f_rest_b) 0 = No rewind on close.
1 = Rewind on close.

bit 1: (f_offl_b) 0 = Do not put drive off-line on close.
1 = Put drive off-line on close.

bit 2: (f_eras_b) 0 = Do not erase to end-of-tape on close.
1 = Erase to end-of-tape on close.

Name **Description**

PD_DrivFlag **Driver flags**

This field is available for use by the device driver.

NOTE: References to these flags are often made using the PD_Flags offset (defined in sys.l and usr.l). This reference is equivalent to PD_SBFFlags. References to PD_DrivFlag should use a value of PD_Flags + 1.

PD_DMAMode **Direct memory access mode**

This field is hardware specific. If available, you can use this word to specify the DMA Mode of the driver.

PD_ScsiID **SCSI controller ID**

This is the ID number of the SCSI controller attached to the device. The driver uses this number when communicating with the controller.

PD_ScsiLUN **Logical unit number of SCSI device**

This number is the value to use in the SCSI command block to identify the logical unit on the SCSI controller. This number may be different from PD_TDrv to eliminate allocation of unused drive table storage. PD_TDrv indicates the logical number of the drive to the driver and SBF (drive table to use). PD_ScsiLUN is the physical drive number on the controller.

PD_ScsiOpts **SCSI driver options flags**

This field allows SCSI device options and operation modes to be specified. It is the driver’s responsibility to use or reject these if applicable:

bit 0: 0 = ATN not asserted (no disconnects allowed).
1 = ATN asserted (disconnects allowed).

bit 1: 0 = Device cannot operate as a target.
1 = Device can operate as a target.

bit 2: 0 = asynchronous data transfers.
1 = synchronous data transfers.

bit 3: 0 = parity off.
1 = parity on.

All other bits are reserved.

SBF Definitions of the Path Descriptor

The reserved section (PD_OPT) of the path descriptor used by SBF is copied directly from the initialization table of the device descriptor. The following table is provided to show the offsets used in the path descriptor. For a full explanation of the path descriptor fields, refer to the previous pages.

Offset	Name	Description
$80	PD_DTP	Device Type
$81	PD_TDrv	Tape Drive Number
$82	PD_SBF	Reserved
$83	PD_NumBlk	Maximum Number of Blocks to Allocate
$84	PD_BlkSiz	Logical Block Size
$88	PD_Prior	Driver Process Priority
$8A	PD_SBFFlags*	SBF Path Flags
$8B	PD_DrivFlag*	Driver Flags
$8C	PD_DMAMode	Direct Memory Access Mode
$8E	PD_ScsiID	SCSI Controller ID
$8F	PD_ScsiLUN	LUN on SCSI controller
$90	PD_ScsiOpts	SCSI Options Flags

***** References to these flags are often made using the PD_Flags offset (defined in sys.l and usr.l). This reference is equivalent to PD_SBFFlags. References to PD_DrivFlag should use a value of PD_Flags + 1.

NOTE: ***Offset*** refers to the location of a path descriptor field relative to the starting address of the path descriptor. Path descriptor offsets are resolved in assembly code by using the names shown here and linking the module with the relocatable library: sys.l or usr.l.

Pipeman Definitions of the Path Descriptor

The table shown below describes the option section (PD_OPT) of the path descriptor used by pipeman.

NOTE: ***Offset*** refers to the location of a module field, relative to the starting address of the module. Module offsets are resolved in assembly code by using the names shown here and linking the module with the relocatable library: sys.l or usr.l.

Offset		**Description**
$80	DV_DTP	Device type
$81		Reserved
$82	PD_BufSz	Default pipe buffer size
$86	PD_IOBuf	Reserved I/O buffer
$E0	PD_Name	Pipe file name

Figure B-7: Path Descriptor PD_OPT for PIPEMAN

Name **Description**

DV_DTP **Device type**

This field is set to two for PIPE devices. (0 = SCF, 1 = RBF, 2 = PIPE, 3 = SBF, 4 = NET)

PD_BufSz **Default pipe buffer size**

Contains the default size of the FIFO buffer used by the pipe. If no default size is specified and no size is specified when creating the pipe, PD_IOBuf is used.

PD_IOBuf **Reserved I/O buffer**

This contains the small I/O buffer to be used by the pipe if no other buffer is specified.

PD_Name **Pipe file name (if any)**

End of Appendix B