2 ASPI for DOS Specification

About This Chapter

Read this chapter to find out about

- The steps involved in programming ASPI for DOS
- How to access and call ASPI
- The SCSI request block and ASPI command codes executed by the ASPI manager

▼▼▼▼▼ 2

Using ASPI – An Overview

Two steps are required for a driver to use ASPI:

- Obtaining the ASPI entry point
- Calling ASPI

Typically, the entry point is obtained once and ASPI calls are made multiple times within a device driver. ASPI function calls are used to return data about the ASPI manager, host adapter, and SCSI devices, but they are mainly used to execute SCSI I/O requests. The ASPI layer is reentrant and can accept function calls before previous calls have completed. A call normally returns immediately with zero status indicating that the request has been successfully queued. In order to continue program flow after the function completes, the driver either polls ASPI status or enables the post bit, which turns control over to a specified routine upon completion of the ASPI call.

Note: When a program makes a call to an ASPI manager, the manager uses the caller's stack. Therefore, the program must allocate enough stack memory for itself and for the ASPI manager. There is no fixed amount of stack needed by all ASPI managers; a programmer must be aware of this constraint and must test code with individual managers for compatibility.

Accessing ASPI

Device drivers wishing to access ASPI must open the driver by performing a DOS Int 21h function call **OPEN A FILE** as follows:

On Entry:	AX	=	3D00h
	DS:DX	=	Pointer to SCSIMGR$, 0
On Return:	AX	=	File handle if carry flag is not set Error code if carry flag is set

Getting the ASPI Entry Point

Device drivers can get the entry point to ASPI by performing a DOS Int 21h function call **IOCTL READ** as follows:

On Entry:	AX	=	4402h
	DS:DX	=	Pointer to data returned (4 bytes)
	CX	=	4
	BX	=	File handle
On Return:	AX	=	Nothing

Data returned in DS:DX contains the ASPI entry point:

Byte 0-1:	ASPI Entry Point Offset
Byte 2-3:	ASPI Entry Point Segment

Closing ASPI

Device drivers can close ASPI by performing a DOS Int 21h function call **CLOSE A FILE** as follows:

On Entry:	AH	=	3Eh
	BX	=	File handle
On Return:	AX	=	Error code if carry flag is set Nothing if carry flag is not set

Calling ASPI

The following is an example of how to call the ASPI manager. As shown in this sample code, the SRB's segment is first pushed onto the stack, followed by its offset. ASPI is then called directly:

```
                    .MODEL SMALL
                    .STACK 100h                 ;100h byte stack

                    .DATA
SCSIMgrString       db      "SCSIMGR$"
                    dw      0                   ;NULL-terminate string
ASPI_Entry          db      4 dup (?)
SRB                 db      58 dup (0)          ;Initialize SRB for Host
                                                ;  Adapter Inquiry

                    .CODE
start:              mov     ax,@DATA
                    mov     ds,ax               ;Init DS
                    mov     ax,03D00h
                    lea     dx,SCSIMgrString
                    int     21h                 ;Open ASPI Manager
                    jc      NoASPIManager       ;Branch if none found
                    push    ax                  ;Save ASPI File Handle

                    mov     bx,ax               ;BX = File Handle
                    mov     ax,4402h
                    lea     dx,ASPI_Entry       ;Store entry point here
                    mov     cx,4                ;Four bytes to transfer
                    int     21h                 ;Get ASPI entry point

                    mov     ah,03Eh
                    pop     bx                  ;BX = ASPI File Handle
                    int     21h                 ;Close ASPI Manager

                    push    ds                  ;Push SRB's segment
                    lea     bx,SRB
                    push    bx                  ;Push SRB's offset
                    lea     bx,ASPI_Entry
                    call    DWORD PTR [bx]      ;Call ASPI
                    add     sp,4                ;Restore the stack
                            .
                            .
ASPI_Exit:          mov     ax,4C00h            ;Exit to DOS
                    int     21h
                    ret

NoASPIManager:              .                   ;No ASPI Manager found!!
                    jmp     ASPI_Exit           ;Handle it.
                    END
```

SCSI Request Block (SRB)

A SCSI Request Block (SRB) contains the command to be executed by the ASPI manager and is used by both drivers and application programs. An SRB consists of an 8-byte SRB header followed by additional fields dependent on the command code.

Table 2-1. SCSI Request Block Structure

Offset	# Bytes	Description	R/W*
00h (00)	01h (01)	Command Code. This field is used to indicate which of the ASPI services is being accessed. Valid ASPI command codes are listed in Table 2-2.	W
01h (01)	01h (01)	Status Byte. This field is used to post the status of the command. ASPI status bytes are listed in Table 2-3.	R
02h (02)	01h (01)	Host Adapter Number. This field specifies which installed host adapter the request is intended for. Host adapter numbers are assigned by the SCSI manager layer, beginning with 0. An Invalid Host Adapter Number (81h) status indicates that the specified host adapter is not installed.	W
03h (03)	01h (01)	SCSI Request Flags. This field definition is command code specific.	W
04h (04)	04h (04)	Reserved = 0	—

* Indicates whether the field is sent to ASPI (W), returned from ASPI (R), or reserved (—).

DOS ASPI Commands

Table 2-2. Valid ASPI Command Codes

Command Code	Description
00h	Host Adapter Inquiry
01h	Get Device Type
02h	Execute SCSI I/O Command
03h	Abort SCSI I/O Command
04h	Reset SCSI Device
05h	Set Host Adapter Parameters
06h	Get Disk Drive Information
07h-7Fh	Reserved for Future Expansion
80h-FFh	Reserved for Vendor Unique

Table 2-3. ASPI Status Bytes

Status Byte	Description
00h	SCSI Request in Progress
01h	SCSI Request Completed Without Error
02h	SCSI Request Aborted by Host
04h	SCSI Request Completed With Error
80h	Invalid SCSI Request
81h	Invalid Host Adapter Number
82h	SCSI Device Not Installed

Host Adapter Inquiry Command

This command is used to get information on the installed host adapter hardware, including number of host adapters installed. You can issue it once specifying 0 in the Host Adapter Number field to get the number of installed host adapters. If you need more information, you can issue the command for each individual host adapter.

Table 2-4. Host Adapter Inquiry Command

Offset	# Bytes	Description	R/W*
00h (00)	01h (01)	Command Code = 0	W
01h (01)	01h (01)	Command Status. This byte always returns with a non-zero status. A SCSI Request Completed Without Error (01h) status indicates that the remaining fields are valid.	R
02h (02)	01h (01)	Host Adapter Number. This field specifies which installed host adapter the request is intended for. Host adapter numbers are assigned by the SCSI manager layer, beginning with 0. An Invalid Host Adapter Number (81h) status indicates that the specified host adapter is not installed.	W
03h (03)	01h (01)	SCSI Request Flags. This field is currently undefined for this command and should be zeroed.	—
04h (04)	02h (02)	Extended Request Signature. This field is set to AA55h (55 in byte 4) by the ASPI client. If the extended request mechanism is supported by the device driver, this signature is reversed to 55AAh when returned. This word is reserved for older drivers that do not support the extended request and will be returned unchanged.	R/W
06h (06)	02h (02)	Extended Buffer Length. The ASPI client specifies the number of extended-buffer bytes desired. Currently, 4 bytes are defined. When returned, this field contains the number of bytes that contain valid information. In no case will the length returned be greater than the length specified on input.	R/W
08h (08)	01h (01)	Number of Host Adapters. The ASPI manager sets this field with the number of host adapters installed under ASPI. For example, a return value of 2 indicates that host adapters #0 and #1 are valid.	R
09h (09)	01h (01)	Host Adapter SCSI ID. The ASPI manager sets this field with the SCSI ID of the given host adapter.	R
0Ah (10)	10h (16)	SCSI Manager ID. This field contains a 16-byte ASCII string describing the SCSI manager.	R
1Ah (26)	10h (16)	Host Adapter Identifier. This field contains a 16-byte ASCII string describing the SCSI host adapter.	R
2Ah (42)	10h (16)	Host Adapter Unique Parameters. The definition of this field is left to implementation notes specific to a particular host adapter.	
3Ah (58)	2h (2)	Supported Extensions. Bit 15-4 Reserved Bit 3 0 = Not a Wide SCSI 32-bit host adapter 1 = Wide SCSI 32-bit host adapter Bit 2 0 = Not a Wide SCSI 16-bit host adapter 1 = Wide SCSI 16-bit host adapter Bit 1 0 = Residual byte length not reported 1 = Residual byte length reported. See *Residual Byte Length* on page 2-7 Bit 0 Reserved	R

* Indicates whether the field is sent to ASPI (W), returned from ASPI (R), or reserved (—).

Residual Byte Length

Residual byte length is the number of bytes not transferred to, or received from, the target SCSI device. For example, if the ASPI buffer length for a SCSI Inquiry command is set for 100 bytes, but the target only returns 36 bytes; the residual length is 64 bytes. Another example: if the ASPI buffer length for a SCSI Write command is set for 514 bytes but the target only takes 512 bytes, the residual length is 2 bytes.

ASPI modules can determine if the loaded ASPI manager supports residual byte length by issuing an Extended Host Adapter Inquiry command, as described in the previous section.

Get Device Type Command

This command can be used by various drivers, during initialization, to identify the targets that they need to support. A CD-ROM driver, for example, can scan each Target/LUN on each installed host adapter looking for the device type corresponding to CD-ROM devices. This eliminates the need for each driver to duplicate the effort of scanning the SCSI bus for devices.

This command always returns with non-zero status.

Table 2-5. Get Device Type Command

Offset	# Bytes	Description	R/W*
00h (00)	01h (01)	Command Code = 1	W
01h (01)	01h (01)	Command Status. This byte always returns with a non-zero status. A SCSI Request Completed Without Error (01h) status indicates that the specified device is installed and the Peripheral Device Type field is valid. A SCSI Device Not Installed Error (82h) indicates that the Peripheral Device Type field is not valid.	R
02h (02)	01h (01)	Host Adapter Number. This field specifies which installed host adapter the request is intended for. Host adapter numbers are assigned by the SCSI manager layer, beginning with 0. An Invalid Host Adapter Number (81h) status indicates that the specified host adapter is not installed.	W
03h (03)	01h (01)	SCSI Request Flags. This field is currently undefined for this command and should be zeroed.	—
04h (04)	04h (04)	Reserved for Expansion = 0	—
08h (08)	01h (01)	Target ID. This field indicates the SCSI ID of the target device.	W
09h (09)	01h (01)	LUN. This field indicates the Logical Unit Number (LUN) of the device.	W
0Ah (10)	01h (01)	Peripheral Device Type of Target/LUN. The ASPI manager fills this field with the peripheral device type, as previously reported by the SCSI Inquiry command. Refer to any SCSI specification to learn more about the SCSI Inquiry command.	R

* Indicates whether the field is sent to ASPI (W), returned from ASPI (R), or reserved (—).

Execute SCSI I/O Command

This command usually returns with zero status, indicating that the request was queued successfully. Command completion can be determined by polling for nonzero status or by using the Post Routine Address field (discussed in *ASPI Command Posting* on page 2-10). If you are going to use polling, interrupts must be enabled.

Table 2-6. Execute SCSI I/O Command

Offset	# Bytes	Description	R/W*
00h (00)	01h (01)	Command Code = 2	W
01h (01)	01h (01)	Command Status.	R
02h (02)	01h (01)	Host Adapter Number. This field specifies which installed host adapter the request is intended for. Host adapter numbers are assigned by the SCSI manager layer, beginning with 0. An Invalid Host Adapter Number (81h) status indicates that the specified host adapter is not installed.	W
03h (03)	01h (01)	SCSI Request Flags. Bits 7-5 Reserved Bits 4-3 00 = Transfer direction determined by SCSI command. Length not checked 01 = Transfer direction from SCSI target to host. Length checked. 10 = Transfer direction from host to SCSI target. Length checked. 11 = No data transfer. Bit 2 0 = Residual byte length reporting disabled 1 = Residual byte length reporting enabled (if the ASPI manager supports it) Bit 1 0 = Linking disabled 1 = Linking enabled. This bit should be cleared if scatter/gather is enabled. Bit 0 0 = Posting disabled 1 = Posting enabled	W
04h (04)	04h (04)	Reserved = 0	W
08h (08)	01h (01)	Target ID. This field specifies the target ID of the peripheral device involved in the I/O operation.	W
09h (09)	01h (01)	LUN. This field specifies the LUN of the peripheral device involved in the I/O operation.	W
0Ah (10)	04h (04)	Data Allocation Length. This field indicates the number of bytes to be transferred. If the SCSI command to be executed does not transfer data (i.e., Rewind, Start Unit, etc.) the Data Allocation Length must be set to zero. If residual byte length is supported and selected, this field is returned with the residual number of bytes (usually 0). See *Residual Byte Length* on page 2-7.	W
0Eh (14)	01h (01)	Sense Allocation Length (N). This field indicates the number of bytes allocated at the end of the SRB for sense data. A request sense is automatically generated if a check condition is presented at the end of a SCSI command.	W
0Fh (15)	02h (02)	Data Buffer Pointer (Offset). This field is a pointer to the I/O data buffer. You place the logical address here. ASPI converts it to the physical address in the case of a bus master or DMA transfer.	W
11h (17)	02h (02)	Data Buffer Pointer (Segment)	
13h (19)	02h (02)	SRB Link Pointer (Offset). This field is a pointer to the next SRB in a chain. See *SCSI Command Linking* on page 2-9 for more information.	W
15h (21)	02h (02)	SRB Link Pointer (Segment)	
17h (23)	01h (01)	SCSI CDB Length (M). This field establishes the length, in bytes, of the SCSI Command Descriptor Block (CDB).	W

* Indicates whether the field is sent to ASPI (W), returned from ASPI (R), or reserved (—).

Table 2-6. Execute SCSI I/O Command *(Continued)*

Offset	# Bytes	Description	R/W*
18h (24)	01h (01)	Host Adapter Status. 00h - Host adapter did not detect any error 11h - Selection time-out 12h - Data overrun/underrun 13h - Unexpected Bus Free 14h - Target Bus phase sequence failure	R
19h (25)	01h (01)	Target Status. 00h - No target status 02h - Check status (sense data is in sense allocation area) 08h - Specified target/LUN is busy 18h - Reservation conflict	R
1Ah (26)	02h (02)	Post Routine Address (Offset). This field, if specified, is called when the I/O is completed. See *ASPI Command Posting* on page 2-10 for more information.	W
1Ch (28)	02h (02)	Post Routine Address (Segment)	W
1Eh (30)	22h (34)	Reserved for ASPI Workspace	—
40h (64)	M	SCSI Command Descriptor Block (CDB). This field contains the CDB as defined by the target's SCSI command set. The length of the SCSI CDB is specified in the SCSI Command Length field.	W
40h+M	N	Sense Allocation Area. This field is filled with sense data on a check condition. The maximum length of this field is specified in the Sense Allocation Length field. The target can return fewer than the number of sense bytes requested.	R

* Indicates whether the field is sent to ASPI (W), returned from ASPI (R), or reserved (—).

SCSI Command Linking

ASPI provides the ability to use SCSI linking to guarantee the sequential execution of several commands. This feature requires that the involved target(s) support SCSI linking.

To use SCSI linking, a chain of SRBs is built with the SRB link pointer used to link the elements together. The Link bit should be set in the SCSI Request Flags byte of all SRBs except the last in the chain. When a SCSI target returns indicating that the linked command is complete, the next SRB is immediately processed, and the appropriate CDB is dispatched. When using SCSI linking, be sure the linking flags in the SCSI CDB agree with the Link bit in the SCSI request flags. Inconsistencies can cause unpredictable results. For example, if you set the CDB up for linking but fail to set the Link bit, a random address may be used for the next SRB pointer.

Any error returned from the target on a linked command breaks the chain. If linking without tags is used, as defined in SCSI, posting may not occur on any elements in the chain until the chain is complete. If you have the post bit set in each SRB's SCSI request flags byte, then each SRB's post routine will be called.

Note: We *strongly* recommend that you *do not* use SCSI linking. Many SCSI targets and SCSI host adapters do not handle SCSI linking and will not work with your ASPI module.

ASPI Command Posting

To use posting, the Post bit must be set in the SCSI request flags. Posting refers to the SCSI manager making a FAR call to a post routine as specified in the SRB. The post routine is called to indicate that the SRB is complete. The specific SRB completed is indicated by a 4-byte SRB pointer on the stack. It is assumed that all registers are preserved by the post routine.

The ASPI manager first pushes the completed SRB's 2-byte segment onto the stack followed by its 2-byte offset. The following is a sample ASPI post handler:

```
ASPI_Post       proc    far
                push    bp
                mov     bp,sp

                pusha                           ;Save all registers
                push    ds
                push    es

                mov     bx,[bp+6]               ;BX = SRBs offset
                mov     es,[bp+8]               ;ES = SRBs segment
                        .                       ;ES:BX points to SRB
                        .
                        .
                pop     es
                pop     ds
                popa
                pop     bp                      ;Restore all registers
                retf                            ;  and return to ASPI
ASPI_Post       endp
```

When your post routine is first entered, the stack looks like this:

Top of Stack	[SP+0] —>	Return Address (Offset)
	[SP+2] —>	Return Address (Segment)
	[SP+4] —>	SRB Pointer (Offset)
	[SP+6] —>	SRB Pointer (Segment)
		...
		...
		...

You may issue any ASPI command from within your post routine except for an Abort command. Your post routine should get in and out as quickly as possible. You should not poll for completion of an ASPI request from your post routine.

Posting can be used by device drivers and by Terminate and Stay Resident (TSR) programs which need to operate in an interrupt driven fashion.

Abort SCSI I/O Command

This command is used to request that an SRB be aborted. It should be issued on any I/O request that has not completed if the driver wishes to timeout on that request. Success of the Abort command is never assured.

This command always returns with SCSI Request Completed Without Error, but the actual failure or success of the abort operation is indicated by the status eventually returned in the SRB specified.

Table 2-7. Abort SCSI I/O Command

Offset	# Bytes	Description	R/W*
00h (00)	01h (01)	Command Code = 3	W
01h (01)	01h (01)	Command Status. This byte always returns with a non-zero status. A SCSI Request Completed Without Error (01h) status indicates that the remaining fields are valid, and the abort will be attempted.	R
02h (02)	01h (01)	Host Adapter Number. This field specifies which installed host adapter the request is intended for. Host adapter numbers are assigned by the SCSI manager layer, beginning with 0. An Invalid Host Adapter Number (81h) status indicates that the specified host adapter is not installed.	W
03h (03)	01h (01)	SCSI Request Flags. This field is currently undefined for this command and should be zeroed.	—
04h (04)	04h (04)	Reserved for Expansion = 0	—
08h (08)	02h (02)	SRB Pointer to Abort (Offset). This field contains a pointer to the SRB which is to be aborted.	W
0Ah (10)	02h (12)	SRB Pointer to Abort (Segment)	W

* Indicates whether the field is sent to ASPI (W), returned from ASPI (R), or reserved (—).

Note: An Abort command should not be issued during a post routine.

Reset SCSI Device Command

This command is used to reset a specific SCSI target. The structure passed is nearly identical to the execute SCSI I/O SRB except that some of the fields are not used. This command usually returns with zero status, indicating that the request was queued successfully. Command completion can be determined by polling for nonzero status or through the use of posting.

Table 2-8. Reset SCSI Device Command

Offset	# Bytes	Description	R/W*
00h (00)	01h (01)	Command Code = 4	W
01h (01)	01h (01)	Status	R
02h (02)	01h (01)	Host Adapter Number. This field specifies which installed host adapter the request is intended for. Host adapter numbers are assigned by the SCSI manager layer, beginning with 0. An Invalid Host Adapter Number (81h) status indicates that the specified host adapter is not installed.	W
03h (03)	01h (01)	SCSI Request Flags Bit 7-1 Reserved Bit 0 1 = Posting enabled 0 = Posting disabled	W
04h (04)	04h (04)	Reserved = 0	—
08h (08)	01h (01)	Target ID	W
09h (09)	01h (01)	LUN	W
0Ah (10)	0Eh (14)	Reserved = 0	—
18h (24)	01h (01)	Host Adapter Status	R
19h (25)	01h (01)	Target Status	R
1Ah (26)	02h (02)	Post Routine Address (Offset)	W
1Ch (28)	02h (02)	Post Routine Address (Segment)	W
1Eh (30)	02h (02)	Reserved for ASPI Workspace	—

* Indicates whether the field is sent to ASPI (W), returned from ASPI (R), or reserved (—).

Set Host Adapter Parameters Command

Table 2-9. Set Host Adapter Parameters Command

Offset	# Bytes	Description	R/W*
00h (00)	01h (01)	Command Code = 5	W
01h (01)	01h (01)	Command Status.	R
02h (02)	01h (01)	Host Adapter Number. This field specifies which installed host adapter the request is intended for. Host adapter numbers are assigned by the SCSI manager layer, beginning with 0. An Invalid Host Adapter Number (81h) status indicates that the specified host adapter is not installed.	W
03h (03)	01h (01)	SCSI Request Flags	W
04h (04)	04h (04)	Reserved = 0	—
08h (08)	10h (16)	Host Adapter Unique Parameters. The definition of the this field is left to implementation notes specific to a particular host adapter.	W

* Indicates whether the field is sent to ASPI (W), returned from ASPI (R), or reserved (—).

Get Disk Drive Information Command

This command is intended for use by SCSI disk drivers that need to determine which disk drives are already being controlled by some BIOS/DOS, and which disk drives are available for use by the disk driver. It also provides a means to determine which drives are not under control of the BIOS/DOS, yet are still accessible via Int 13h. This is useful because many disk caching utilities cache Int 13h requests but do not cache disk driver requests. Also, some disk utility programs allow the user to access physical sectors on a disk via Int 13h.

Table 2-10. Get Disk Drive Information Command

Offset	# Bytes	Description	R/W*
00h (00)	01h (01)	Command Code = 6	W
01h (01)	01h (01)	Command Status. ASPI managers that support this command code always return with a status of SCSI Request Completed Without Error (01h). ASPI managers that do not support this command code always return with a status of Invalid SCSI Request (80h).	R
02h (02)	01h (01)	Host Adapter Number. This field specifies which installed host adapter the request is intended for. Host adapter numbers are assigned by the SCSI manager layer, beginning with 0. An Invalid Host Adapter Number (81h) status indicates that the specified host adapter is not installed.	W
03h (03)	01h (01)	SCSI Request Flags. This field is currently undefined for this command and should be zero	—
04h (04)	04h (04)	Reserved = 0	—
08h (08)	01h (01)	Target ID. This field indicates the SCSI ID of the target device.	W
09h (09)	01h (01)	LUN. This field indicates the Logical Unit Number (LUN) of the device.	W
0Ah (10)	01h (01)	Drive Flags Bits 7-2 Reserved Bits 1-0 These bits return information pertaining to the Int 13 drive field: 00 The given drive (HA #/Target/LUN) is not accessible via Int 13h. If you wish to read/write to this drive, you must send ASPI read/write requests to the drive. The Int 13h Drive field is invalid. 01 The given drive (HA #/Target/LUN) is accessible via Int 13h. The Int 13h Drive field contains the drive's Int 13h drive number. This drive is under the control of DOS. 10 The given drive (HA #/Target/LUN) is accessible via Int 13h. The Int 13h Drive field contains the drive's Int 13h drive number. This drive is not under control of DOS and can be used, for example, by a SCSI Disk Driver. 11 Invalid.	R
0Bh (11)	01h (01)	Int 13h Drive. This field returns the Int 13 drive number for the given host adapter number, target ID, and LUN. Valid Int 13 drive numbers range for 00-FFh.	R
0Ch (12)	01h (01)	Preferred Head Translation. This field indicates the given host adapter's/disk drive's preferred head translation method. A typical value is 64 heads.	R
0Dh (13)	01h (01)	Preferred Sector Translation. This field indicates the given host adapter's/disk drive's preferred sector translation method. A typical value is 32 sectors per track.	R
0Eh (14)	0Ah (10)	Reserved = 0	—

* Indicates whether the field is sent to ASPI (W), returned from ASPI (R), or reserved (—).

ASPI Under Windows 3.x

Windows is a graphical user interface which runs under DOS as an application; however, writing a device driver or application capable of making ASPI calls in a Windows environment is not as simple as in the strictly DOS case. The problem arises because ASPI for DOS uses a Real Mode interface and Windows uses the DOS Protected Mode Interface (DPMI). Whereas ASPI expects a Real Mode segment and offset for the SRB and the entry point of ASPI, Windows uses a *Selector* and offset to address data and code. In order to program correctly in this environment, a consortium of companies (Microsoft and Intel among them) have written a specification called the *DOS Protected Mode Interface Specification*. The details of the specification are too complex to describe here, but we recommend that you obtain a copy from the DPMI committee for programming purposes. Here is a brief overview of the two steps you need to follow to access ASPI for DOS from a Windows application:

1. Allocate all SRBs and buffers down in Real Mode memory. You can do this with the Windows GlobalDosAlloc routine or with DPMI interrupt 31h, function 100h. This makes it possible for the ASPI module and manager to locate the SRB and data buffers using segments and offsets.
2. Call the Real Mode procedure with Far Return Frame Function (interrupt 31h, function 301h). This makes it possible to call the ASPI manager which is a Real Mode procedure.

For more information on programming ASPI in the Windows environment, refer to Chapter 3, *ASPI for Win16 Specification* and Chapter 4, *ASPI for Win32 Specification*.

❒