Date: 01 May 1999
To: T10 Technical Committee
From: Ralph Weber, LSI Logic Alternate Member of T10
Subj: Comments on EXTENDED COPY (99-143r1)

Motivated by discussions at the March T10 Plenary, I have reviewed 99-143r0 (and the 2 April revision 99-143r1) pursuant to its proposed incorporation in SPC-2. Several changes are necessary prior to inclusion in SPC-2 and several more seem like good ideas. A mark-up 99-143r1 is attached to show the proposed changes.

Based on requirements for storage array devices, LSI requests addition of a Target descriptor format that identifies targets based on information in their Device Identification Vital Product Data page. That addition also is shown in the attached mark-up with descriptor type code E4h.

A few omissions require responses from the proposal authors. In other areas, the attached mark-up shows hopefully correct guesses for needed changes and questions regarding those changes are included at the end of this summary.

Problems to be addressed before the proposal can be approved by T10

In 7.1.3.1 and 7.1.3.3, the description lacks a definition of the WORLD WIDE NAME field. This definition must include specific identification of which of the several world wide names belonging to a device server can appear in the field and all specific references to other standards necessary to locate and understand the format/contents of such world wide names. Pending resolution of this issue, the affected field has been marked "Reserved" in the attached mark-up.

In 7.1.3.2 and 7.1.3.3, the description lacks a definition of the NPORT_ID (D_ID) field. This definition must include all specific references to other standards necessary to locate and understand the format/contents of the N_port and D_ID entities. This description also must explain how two different entities (N_port and D_ID) can share a field without any indication of which is present in any given instance. If the explanation appears in some other standard, a reference to that standard is sufficient. Pending resolution of this issue, the affected field has been marked "Reserved" in the attached mark-up.

Discussion of edits suggested or required by SPC-2 project editor

The minor edits are reflected by change bars and strike-through markings in the attached mark-up copy of the proposal text below. The substantial changes are reflected in the attached mark-up and discussed in the paragraphs below.

In the EXTENDED COPY parameter list (Table 9): Current practice is for all SCSI parameter data to be self-defining. I recognize that this practice may not have been in place when the COPY command was created, leading to an inadequate model for this proposal. However, current practice strongly recommends the addition of an INLINE DATA LENGTH field to the parameter list header. The field has been added as a 4-byte field.

In 7.1, regarding the definition of List ID:

> The List ID field is a user-defined value that is used by the copy manager to return status for that command. At any given instant, the value of the List ID field for an EXTENDED COPY command must be unique to the copy manager. If the copy manager detects a duplicate List ID, the command shall terminate with a CHECK CONDITION and the sense key shall be set to ILLEGAL REQUEST and the ASC and ASCQ shall indicate OPERATION IN PROGRESS.

There are no users in SCSI. 'User' does not appear in the SPC-2 glossary and use of the standard English meaning is not advised, since it might draw scrutiny from the drug enforcement authorities. Use of this terminology and other wording problems in this paragraph result in an imprecise definition of the interactions of application clients and copy managers with respect to the usage of this field. The paragraph has been completely rewritten in attached mark-up.

In 7.1 after the description of the LIST ID field, the description of the parameter list does not follow the order of the fields in the parameter list. The attached mark-up adjusts the location of information to match the order of the parameter list. These movements of text are not marked with change bars.

I find the Table 10 — EXTENDED COPY Descriptor Codes confusing, even though (or maybe because) it's a reasonable expansion of the equivalent table in the COPY command definition. The notes are more than notes; they are functional definitions. Excessive white space, especially in the Target descriptor area, present the reader with an information wasteland. A major restructuring of the table in presented in the attached mark-up.

Since the communications device type (09h) has been removed from SSC, I think it should not be listed as a device type for the EXTENDED COPY command in Table 10. Also, the RBC device (0Eh) should be added as a block device. The attached mark-up reflects these changes in several locations.

In 7.1, regarding the definition of the Priority field:

> The PRIORITY field of the EXTENDED COPY parameter list establishes the relative priority of this EXTENDED COPY command to other commands being executed by the same copy manager. Priority 0 is the highest priority, with increasing priority values indicating lower priorities. All other commands are assumed to have a priority of 1.

While this is basically a cut & paste from the COPY command, using "copy manager" in place of "device server" has muddied the requirements substantially. Is it intended that priority be only a relative priority among EXTENDED COPY commands, with no relationship to other work being done by the device server that contains the copy manager? The attached mark-up uses the wording from the COPY command, with the effect that priority applies to all commands being processed by the device server (not just copy commands). Modifications from the COPY wording are identified with strike-outs and underlines.

7.1 does not indicate that support for OOE=1 is optional, but 7.4.3 indicates that it is optional. The attached mark-up adds wording to 7.1 to indicate that support for OOE=1 is optional.

In 7.1, the following requirement doesn't make sense:

> If both the NRC and OOE bits are one, the command shall be terminated with CHECK CONDITION status …

What's wrong with an application client doing and out-of-order copy and not asking for results via the RECEIVE COPY RESULTS command? In fact, a copy manager that doesn't support out-of-order processing could accept a command with NRC and OOE equal to one, process the segment descriptors in order, and not have a Segment order list to return without anyone being harmed. The real problem occurs if NRC is 0 and OOE is 1 and the copy manager does not support the optional out-of-order processing feature. In that case, the application client is going to send its RECEIVE COPY RESULTS command and get an error. It probably would be best to give the error on the EXTENDED COPY command instead. The attached mark-up contains no changes in this area, but further discussion is needed.

In 7.1, the following statement is inappropriate in a command definition and has been removed:

> The Reserved fields in the EXTENDED COPY Command, Parameter List, and all descriptors must be set to 0.

In the 7.1 discussion of target descriptors:

> The target descriptors shall be indexed in ascending order beginning with 0 to a maximum of 65535.

The maximum descriptor index is not 65535. With the maximum number of bytes allowed for target descriptors being 65535 and the minimum descriptor size being 8, the maximum descriptor index is 8191. Furthermore, the usage of the 'index' is not specified. The attached mark-up includes a description of 'index' and several revisions on the topic of maximum descriptor index.

In 7.1, I fail to see the continued need for this requirement:

> If OOE is one the INLINE DATA LENGTH field shall contain zero.

Since the attached mark-up provides sufficient information for the copy manager to locate the inline data at anytime, it would seem that this requirement could be removed. However, the removal is not final in the attached mark-up.

At the same time, I see a problem if OOE is one and a stream device is used as a destination, particularly because there is no practical way to return the segment order list in this case. Would it not be good to add a restriction against such a situation? There also may be a problem if OOE is one and a stream device is used as a source.

In 7.1.2 list entry d:

> If any data has been transferred for the current segment, the residual for the current segment shall be written to the INFORMATION field, and the valid bit shall be set to 1.

The term 'current segment' lacks specificity. The attached mark-up shows it replaced with 'segment being processed at the time the error occurred'.

In 7.1.2 list entry g:

> If, during the execution of a segment descriptor, the copy manager determines that a target is not reachable, then the Sense-key specific field shall be set as described in section 7.3, with the field pointer indicating the first byte of the segment descriptor which identifies the target.

Is it truly desired that the field pointer indicate a segment descriptor, or should it indicate a target descriptor? If it is desired that the field pointer indicate a segment descriptor, how should the application client determine whether it was the source or destination target that was not reachable? Assuming the simplest case, the attached mark-up changes 'segment descriptor' to 'target descriptor'.

In 7.1.3, the Address Type field (which is a sub-field in the DESCRIPTOR TYPE CODE field), provides no additional information beyond that found parent the parent field. Inclusion of the Address Type field in the standard serves only to complicate the standard and confuse the reader. The attached mark-up removes all discussion of the Address Type field.

In 7.1.3.4 Table 16, the LOGICAL UNIT NUMBER field in the SCSI B_T_L Target Descriptor Format must be 8 bytes just like all the other logical unit number fields. There are several reasons for this requirement not the least of which is the fact that packetized SCSI on the parallel bus supports 8 byte LUN values. The attached mark-up includes an 8 byte LOGICAL UNIT NUMBER field in the SCSI B_T_L Target Descriptor Format.

In 7.1.3 and all its sub-clauses, with the size increase in the SCSI B_T_L Target Descriptor Format, all Target Descriptor formats are within 8 bytes of being the same size. Since fixed length Target Descriptors would be helpful for the indexing feature required in 7.1, the attached mark-up shows all Target Descriptors having a fixed length of 32 bytes. With this change, the maximum number of Target Descriptors in one EXTENDED COPY parameter list is 2048. Also, common fields in several Target Descriptor formats have been placed in the same bytes for all Target Descriptors in which they appear.

In the 7.1.3.4 SCSI B_T_L Target Descriptor format (Table 16), the SCSI BUS field has no meaning because the application client has no way to determine the value to place in the field. The only possible source of such information would be specific fields within a hierarchical LUN. However, the application client should use the LOGICAL UNIT NUMBER field in the Target Descriptor to pass such information, not a specialized separate field. Based on an assumption that the source of information for the SCSI BUS field is vendor unique, the attached mark-up contains a vendor-unique byte in the SCSI B_T_L Target Descriptor format definition.

In 7.1.3.5, the description of device types supported is not specific enough for a standard. The proposed mark-up identifies (by code number) all device types supported and specifically provides for the processor device type.

Regarding 7.1.4 and all its sub-clauses: Concerns are beginning to be raised about the size of the LBA fields in existing SCSI commands. Four bytes are believed to be insufficient for disks just a couple of generations in the future. Therefore, all LBA fields in the attached mark-up have been increased to eight bytes.

In 7.1.4, no provision has been made for optional segment descriptor format support, such as is already provided for target descriptor formats by the proposal. The attached mark-up provides for optional segment descriptor support.

Also in 7.1.4, embedded data should be counted in the length of a segment descriptor, to simplify the handling of out-of-order processing and to maintain consistency with other SCSI parameter list formats. Making this change requires increasing the DESCRIPTOR LENGTH field from 1 to 2 bytes. The attached mark-up reflects this change.

In 7.1.4, the use of 'should' is confusing in the following:

> If the target addressed by the Target Identifier index is not accessible to the copy manager then the command should terminate with a CHECK CONDITION and the sense key shall be set to COPY ABORTED with an ASC and ASCQ of UNREACHABLE TARGET.

SCSI standards are written from the perspective of what the author of a device serve needs to know. Making this statement 'should' is no more informative to the author of a device server than using 'may'. If there is an alternative to terminating the command, then that needs to appear in the standard. In the interim, the attached mark-up shows CHECK CONDITON as a required response to the unavailability of a copy target device.

There appears to be an omission in 7.1.4.4. While provisions are made for zero values in the BLOCK DEVICE NUMBER OF BLOCKS field, no equivalent provisions are defined for the TRANSFER COUNT field. The attached mark-up adds this provision.

In 7.1.4.5, the inline data would be much more useful if more than one series of bytes could be stored in inline data. Allowing for this ability requires an offset/count combination in the inline segment descriptor, instead of just a count. The attached mark-up adds the offset field in the inline segment descriptor. Note: this change might render the embedded data segment descriptor unnecessary.

7.1.4.5 contains a requirement that Inline data be aligned on a 4-byte boundary. The need for this requirement has been eliminated in the attached mark-up by insuring that all descriptors are a multiple of 4 bytes in length.

Regarding the following statement in 7.1.4.5:

> The inline to Stream segment descriptors must be processed in a sequential manner from first to last to ensure that the inline data at the end of the parameter list is properly assigned.

The statement creates more questions than it answers. Would this problem occur at anytime other than when the OOE bit is one? Isn't this statement suggestive of a greater concern, to whit, is OOE=1 incompatible with stream devices as sources or destinations? If OOE were one, would this statement require the copy manager take as its first action a pass through the segment descriptors processing immediately and sequentially all inline to stream descriptors? Given all the difficulties with this statement, it is marked for deletion in the attached mark-up.

The following changes have been made in 7.1.4.6 to accommodate the SCSI standards requirement that the embedded data be included in the descriptor length. The maximum length of embedded data has been reduced to 65524 bytes. Two embedded data byte count fields now appear in the segment descriptor. The number of bytes to write is as before. The length of the embedded data is required to allow the copy manager to validate the descriptor length field. Theoretically, the EMBEDDED DATA LENGTH field could be removed from the descriptor with appropriate wording changes elsewhere in the descriptor definition. However, the two bytes consumed by the field would still need to be present as reserved bytes to maintain the 4 byte alignment required for segment descriptors.

In 7.1.4.17, field definitions have been copied from SSC in that attached mark-up rather than being defined by reference.

In 7.1.4.18 and 7.1.4.18, the following wording is prohibits the copy manager from setting the NACA bit:

> All other fields of the issued [SPACE or LOCATE] command shall be set to zero.

The attached mark-up contains wording that lifts the restriction.

In 7.1.4.18, the usage of 'logical block address' leads to the possible misinterpretation that the segment descriptor can apply to block devices. Revised field naming and other wording changes in the attached mark-up attempt to clarify the fact that the segment descriptor applies only to tapes.

The image copy described in 7.1.4.20 says stream devices, but it's really good only for tapes. All of the conditions that terminate the copy apply only to tapes. I realize that 7.1.4.20 is nearly a verbatim copy of equivalent text in the COPY command, but that doesn't mean we can't write the EXTENDED COPY definition more realistically.

The following text from 7.1.4.20 has been marked for deletion in the attached mark-up because it is believed that equivalent text already has been added to this and all other operations that apply only to tape devices:

> If the copy is attempted to a device that does not support the block size, file marks, or set marks of the source then the copy manager shall terminate the command with a CHECK CONDITION. The Sense Key shall be set to COPY ABORTED and the ASC and ASCQ shall indicate INCOMPATIBLE FORMAT.

The proposed mode page (7.2) contradicts the principle that SCSI mode pages have a fixed length. (The page length field is present primarily to assist in processing a series of mode pages when multiple pages are transferred by either MODE SELECT or MODE SENSE.) Fortunately, the proposed mode page intends communication of information in only one direction (from copy manager to application client). Since the information flow is the same as for the RECEIVE COPY RESULTS command, the best transfer method would be adding a 'mode' to RECEIVE COPY RESULTS that causes the data in the mode page to be sent to the application client. This change has the side benefit of eliminating the use of a shared mode page, which was preventing implementation of a copy manager in a CD-ROM or DVD device server. The attached mark-up includes this change, with minor changes in the data format to maintain 4-byte alignment and to reflect the fact that it is no longer a mode page.

In 7.2, the definition of the MAXIMUM SEGMENT LENGTH field is unclear. Is it the maximum length of a segment descriptor? This seems unlikely. It's probably the maximum length of data transferred by a segment, but even this would be unclear. The attached mark-up defines it as the maximum length of data written by a segment. In addition, the attached mark-up specifies a value of zero as indicating that there is no limit to the amount of data written by a single segment.

Several definitions in 7.2 fail to consider the affects of PAD=1. Wording has been added in the attached mark-up to indicate that the affects of PAD=1 are ignored for the purposes of the definitions. Similarly, many of the 7.2 definitions use the word 'can' which is not acceptable in a standard. The attached mark-up rewords the definitions to remove the forbidden word.

The 7.2 definition for the MAXIMUM INBOUND INLINE DATA LENGTH (non-Buffered Implementation) field is inconsistent with the RECEIVE COPY RESULTS description of the inbound inline data feature in 7.4 and 7.4.2. The attached mark-up proposes removal of the MAXIMUM INBOUND INLINE DATA LENGTH (non-Buffered Implementation) field.

In 7.2, the following sentence in the definition of the MAXIMUM INBOUND INLINE DATA LENGTH (Buffered Implementation) field makes no sense in the context of data stored by the copy manager for delivery to the application client via a RECEIVE COPY DATA command:

> The amount of data transferred by a single segment descriptor must be a multiple of the granularity.

There is no data transfer taking place, instead the copy manager is placing data in a buffer. The attached mark-up deletes the above sentence and adds a more applicable statement.

In 7.3, the use of the new sense key specific format needs to be qualified by the sense key value, in this case COPY ABORTED. Once the sense key specific format is properly qualified by sense key value, there is no need to show the C/D bit in the format definition (the bit can be marked reserved). These changes are shown in the attached mark-up.

In 7.4, the CDB MODE field is can be compared to the SERVICE ACTION field in the persistent reservations commands. Since the use of a SERVICE ACTION field as a qualifier for operation code is accepted practice in SCSI, the MODE field has been changed to a SERVICE ACTION field and wording throughout 7.4 and it's sub-clauses has been modified accordingly.

In 7.4 table 38, the following wording changes should be made. For code 1, the word 'executing' should be deleted since the status can be returned for commands that have finished execution. For code 2, the word 'Inline' should be deleted because the data returned in not the Inline data described in the EXTENDED COPY command definition. These changes are reflected in the attached mark-up.

In 7.4, the following text is unnecessary because undefined code values in the 'mode' field are identified as reserved and thus are covered by the reserved code value handling specified in SPC-2 clause 3.3.8.

> If an invalid Mode Value is specified, then the copy manager shall terminate the command with a CHECK CONDITION. The Sense Key shall be set to ILLEGAL REQUEST and the ASC and ASCQ shall indicate INVALID FIELD IN CDB.

The following information from 7.4 has been included in the table that defines the service action [mode] codes:

> If the EXTENDED COPY command is active and the Mode field has a value of 1 (Inline Data) or 2 (Segment Order), the copy manager shall wait for the completion of that active EXTENDED COPY command. Otherwise, if the EXTENDED COPY command has completed, or it the Mode field has a value of 0 (Copy Status), the copy manager shall return data and status immediately.

In several places, 7.4 uses the phrase 'for a reasonable period of time'. This is not precise enough for a standard. In the absence of any specific time interval definition, the attached mark-up replaces 'for a reasonable period of time' with 'for a vendor specific period of time'. This may be the most acceptable solution since it makes the time interval a purchase specification requirement, not a standards requirement. If a standards requirement is desired, the recommend mechanism would be placing the amount of the time interval in the copy manager's operating parameters, perhaps specified as a byte field in number of seconds.

The definition for the ALLOCATION LENGTH field given in 7.4 is not consistent with other SPC-2 definitions for this field. The definition from PERSISTENT RESERVE IN has been substituted in the attached mark-up. This change requires that an AVAILABLE DATA field be added to the Copy Status parameter data and the attached mark-up reflects that change too.

In 7.4.1, the definition of the SEGMENT NUMBER field is unclear because the segments are not numbered. In addition, a segment number may not be a valid indicator of progress if OOE=1. The attached mark-up contains changes to correct this.

Regarding 7.4.1 table 41, there are usually at least two definitions of terms like Gigabyte and Terabyte. Enhancements to the table in the attached mark-up attempt to define the usage in this context.

In the definition of the TRANSFER COUNT field in 7.4.1, the phrase 'data transferred' is not specific. It could mean the 'data read' or 'data written' or the sum of the two. The attached mark-up changes it to the more specific 'data written'.

Several of the sub-clauses of 7.4 are missing common SCSI requirements, such as requirements regarding the value in the AVAILABLE DATA field. These requirements are added in the attached mark-up.

7.4.2 does not discuss the copy manager's actions in the event that it is asked to hold more bytes for the application client than it can accommodate. The attached mark-up adds a discussion of this situation.

In 7.4.2, the format of a Segment order descriptor provides no opportunity for future enhancements. The attached mark-up addresses this problem by: 1) adding a 1-byte DESCRIPTOR TYPE CODE field to return the descriptor type followed by a reserved byte; 2) reducing the offset field to 2 bytes; and 3) repositioning fields so that the block field is still 4 byte aligned; 4) and adding a reserved bye. The descriptor type should be useful additional information to the application client and the reserved byte can be used as necessary to define future enhancements to the Segment order descriptor.

The following text in 7.4.2 produces more questions than it answers:

> The copy manager will return the block number relative to the first block written by any segment of this EXTENDED COPY command. Thus, the first block written following the start of an EXTENDED COPY command is always written to block 0.

Does this mean that the copy manager must scan all segment descriptors to find the lowest numbered logical block written and compute relative block numbers against that LBA? Or, would LBA values smaller than the very first LBA ever written result in negative relative block number values? Are the relative block numbers relative to a given segment descriptor, or as suggested by the 'any segment' phrase, are the relative block numbers relative to all segments? How does the copy manager handle segments that reference different destination disks? Do they count in the 'any segment' relative block numbers, or is there a different base block number for the relative calculations on each destination disk?

If the above text were deleted, the copy manager would be expected to return absolute LBA values in each Segment order descriptor. This seems like a pragmatic solution to all the questions listed above. Therefore, the attached mark-up deletes the text.

Questions

Would it be useful to define an IMMED bit for the EXTENDED COPY command?

How should the EXTENDED COPY command specify the interactions of the NRC and OOE bits?

Should the presence of inline data be an error if OOE=1?

Should the copy manager be required to generate an error if a stream destination device is encountered and the OOE bit is 1?

What, if anything, needs to be said about OOE=1 and encountering a stream source device?

In 7.1.2 list entry g, is it desired that the field pointer indicate a segment descriptor, or should it indicate a target descriptor?

Is the embedded data segment descriptor useful with the embedded data length restricted to 65524 or fewer bytes? Does the definition of offset/count addressing for inline data allow removal of the embedded data segment descriptor?

Should the Verify segment descriptor have both source and destination devices? If both were present, the copy manager could verify that the device is appropriate as a source or destination. For example, a read-only device would not be an appropriate destination device. If both were added, control bits might be needed to tell the copy manager which to test.

Is it acceptable to specify the Copy Status TRANSFER COUNT field as being the data written?

Are absolute LBA values an acceptable substitute for the confusing relative block numbers in Segment order descriptors?

This page intentionally left blank

7.1 EXTENDED COPY command

The EXTENDED COPY command (see table 8) provides a means to copy data from one set of logical units to another or the same set of logical units. The entity within a device server that receives and performs the EXTENDED COPY command is called the copy manager. The copy manager is responsible for copying data from the ~~logical unit(s) (~~source device(s)~~)~~ to the ~~logical unit(s) (~~destination device(s)~~)~~. The~~se~~ source and destination copy devices are logical units that may reside on different SCSI devices or the same SCSI device (in fact all the devices and the copy manager may be the same logical unit).

Table 8 — EXTENTED COPY command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (83h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	(MSB)			PARAMETER LIST LENGTH				
11								
12								
13								(LSB)
14	Reserved							
15	CONTROL							

Before the copy manager is instructed to move data, the application controlling the data movement shall independently execute any activities necessary to prepare the source and destination devices for the EXTENDED COPY command. These activities could include media changer commands, loading of tapes, MODE SELECT commands, positioning of tape, and etc. After all preparatory actions have been accomplished, the EXTENDED COPY command should ~~can~~ be issued to the copy manager to start the data transfer.

The PARAMETER LIST LENGTH field specifies the length in bytes of the parameter~~s and~~ data that shall be contained in the Data-Out Buffer. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. If the copy manager detects a parameter list length that exceeds the maximum length specified by the MAXIMUM DESCRIPTOR LIST LENGTH field in its operating parameters (see 7.77.4)~~the Extended Copy mode page (0Eh)~~, the command shall terminate with a CHECK CONDITION, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ~~PARAMETER LIST LENGTH TOO LONG~~ PARAMETER LIST LENGTH ERROR.

The EXTENDED COPY parameter list (see table 9) begins with a twelve byte header that contains the LIST IDENTIFIER field, the STR, NRC, and OOE bits, the command's priority, the length of the target descriptor list, the length

of the segment descriptor list, and the length of the optional inline data. Immediately following the header is one or more target descriptors, followed by one or more segment descriptors, followed by any optional inline data.

Table 9 — EXTENDED COPY parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	LIST IDENTIFIER							
1	Reserved		STR	NRC	OOE	PRIORITY		
2	(MSB)							
3				TARGET DESCRIPTOR LIST LENGTH (n-12)				(LSB)
4	(MSB)							
5								
6				SEGMENT DESCRIPTOR LIST LENGTH (m-k)				
7								(LSB)
8	(MSB)							
9								
10				INLINE DATA LENGTH				
11								(LSB)
	Target descriptor(s)							
12								
43				Target descriptor 0				
				. .				
n-31								
n				Target descriptor x				
	Segment descriptor(s)							
k								
k+l				Segment descriptor 0 (See specific table for length.)				
				. .				
m				Segment descriptor y (See specific table for length.)				
	Inline data							

The LIST IDENTIFIER field is a unique value selected by the application client to identify the extended copy operation to the copy manager. The list identifier also may ~~can~~ be used in the RECEIVE COPY RESULTS command to request status for a specific EXTENDED COPY command. The LIST IDENTIFIER value shall be unique for each concurrent EXTENDED COPY command sent by an initiator. If the copy manager detects a duplicate LIST IDENTIFIER value the command shall be terminated with a CHECK CONDITION, the sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to OPERATION IN PROGRESS.

The PRIORITY field ~~of the COPY parameter list~~ establishes the relative priority of this EXTENDED COPY command to other commands being executed by the same device server. All commands other than copy commands have a priority of 1. Priority 0 is the highest priority, with increasing PRIORITY values indicating lower priorities.

A Sequential Stripped bit (STR) value of one indicates to the copy manager that most of the disk references in the parameter list represent sequential access of several stripped disks. This may be used by the copy manager to implement read-ahead. A STR value of zero indicates to the copy manager that disk references are not necessarily sequential.

If the No Receive Copy (NRC) bit is one, the application client shall ~~will~~ not send a RECEIVE COPY RESULTS command to collect the results of this parameter list. If NRC is zero, the application client may send RECEIVE COPY RESULTS command to receive the results of this parameter list.

If the Out of Order Enable (OOE) bit is 0, the copy manager shall process the segment descriptors in the order that they appear ~~are found~~ in the segment descriptor list. If OOE is 1, the copy manager may process the segment descriptors in any order. Copy manager support for OOE being set to 1 is optional. When OOE is 1, the order in which the segment descriptors were processed is returned to the application client in response to a RECEIVE COPY RESULTS command with SEGMENT ORDER LIST service action ~~for the parameter list specified by the List ID field~~. If both the NRC and OOE bits are one, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to sense key shall be set to INVALID FIELD IN CDB.

~~The Reserved fields in the EXTENDED COPY Command, Parameter List, and all descriptors must be set to 0.~~

The TARGET DESCRIPTOR LIST LENGTH contains the length in bytes of the target descriptor list that immediately follows the ~~eight-byte~~ parameter list header. The number of target descriptors equals the length in bytes of the target descriptor list divided by 32.

An EXTENDED COPY command may reference one or more target devices (which are the source and/or the destination logical units). Each target device is described by a target descriptor. All target descriptors ~~shall~~ have their formats specified by an EXTENDED COPY descriptor code. A copy manager need not support all target descriptor formats ~~for its device type~~. ~~The target descriptors shall be indexed in ascending order beginning with 0 to a maximum of 65535.~~ See 7.1.4 for a detailed description of the target descriptors.

Segment descriptors reference target descriptors by their position, or index, in the target descriptor list. The index for a target descriptor is the starting byte number for the target descriptor minus 12 divided by 32. The maximum number of target descriptors permitted within a parameter list is defined by the ~~copy manager's~~ MAXIMUM TARGET COUNT field in the copy manager's operating parameters (see 7.77.4) ~~Extended Copy mode page (0Eh)~~. If the number of target descriptors exceeds the allowed number, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to TOO MANY TARGET DESCRIPTORS (26h/06h).

The SEGMENT DESCRIPTOR LIST LENGTH contains the length in bytes of the segment descriptor list that follows the target descriptor parameters. See 7.1.5 for a detailed description of the segment descriptors. The maximum number of segment descriptors permitted within a parameter list is defined by the ~~copy manager's~~ MAXIMUM SEGMENT COUNT field in the copy manager's operating parameters (see 7.77.4) ~~Extended Copy mode page (0Eh)~~. If the number of segment descriptors exceeds ~~the number established through the mode sense value~~ the allowed number, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to TOO MANY SEGMENT DESCRIPTORS (26h/08h).

The maximum length of the descriptors permitted within a parameter list is defined by the ~~copy manager's~~ MAXIMUM DESCRIPTOR LIST LENGTH field in the copy manager's operating parameters (see 7.77.4) ~~Extended Copy mode page (0Eh)~~. If the combined length of the target and segment descriptors exceeds ~~the length established through the~~

~~mode sense value~~ the allowed value, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ~~INCORRECT PARAMETER LIST LENGTH~~ PARAMETER LIST LENGTH ERROR.

~~Optional Inline Data may follow the segment descriptors. Inline data may only be valid at the end of the parameter list for parameter lists with the OOE bit cleared to zero. Out of Order processing does not support inline data at the end of the parameter list. Out of order processing requires the data to be embedded in the segment descriptors.~~

The INLINE DATA LENGTH field contains the number of bytes of Inline data, after the last segment descriptor. A value of zero indicates that no inline data is present. If OOE is one the INLINE DATA LENGTH field shall contain zero. Out of order processing requires the data to be embedded in the segment descriptors. If OOE is one and the INLINE DATA LENGTH field contains a non-zero value, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to sense key shall be set to INVALID FIELD IN PARAMETER LIST. ~~If the inline data length field contains a non-zero value, the Inline data must begin on a 4-byte boundary and should be padded up to a 4-byte boundary.~~

The copy manager shall move data from the source devices to the destination devices in the manner proscribed by the segment descriptors. ~~In general,~~ The specific commands issued by the copy manager to the source and destination ~~logical~~ devices ~~to execute~~ while processing the segment descriptors is ~~implementation dependent~~ vendor specific. ~~Data must be moved from the source devices to the destination devices in a manner that is consistent with the segment descriptors; the movement must always results in the destination device being at a deterministic state when any intermediate or final status is returned.~~ Upon completion of an EXTENDED COPY command that returns GOOD status, the source and destination devices (particularly stream devices) shall be positioned at deterministic locations such that the device could be repositioned to the same location by the applicaiton client with appropriate commands.

7.1.1 Errors detected before starting ~~execution~~ processing of the segment descriptors

Errors may occur during ~~execution~~ processing of an EXTENDED COPY command before the first segment descriptor is processed. These conditions include parity errors while transferring the EXTENDED COPY command, invalid parameters in the CDB or parameter data, invalid segment descriptors, and inability of the copy manager to continue operating. In the event of such an exception condition, the copy manager shall:

a) terminate the EXTENDED COPY command with CHECK CONDITION status; and
b) set the VALID bit in the sense data to zero. The sense key shall contain the sense key code describing the exception condition (i.e.: not COPY ABORTED).

7.1.2 Errors detected during ~~execution~~ processing of segment descriptors

Errors may occur after the copy manager has begun ~~executing~~ processing segment descriptors. These include invalid parameters in segment descriptors, invalid segment descriptors, unavailable targets referenced by target descriptors, inability of the copy manager to continue operating, and errors reported by source or destination target devices. If the copy manager receives a CHECK CONDITION status from one of the target devices, it shall recover the sense data associated with the exception condition and clear the ACA condition (if any) associated with the CHECK CONDITION status.

Following an exception condition detected during segment descriptor processing, the copy manager shall:

a) terminate the EXTENDED COPY command with CHECK CONDITION status;
b) set the sense key code to COPY ABORTED;
c) indicate the segment that was being processed at the time of the exception by writing the segment number to third and forth bytes of the COMMAND-SPECIFIC INFORMATION field. The segment number is based on the relative position of the segment descriptor in the EXTENDED COPY parameter list. The first segment descriptor in the parameter list is assigned descriptor number zero, the second is assigned one, etc.;

d) If any data has been transferred for the ~~current~~ segment being processed at the time the error occurred, the residual for the ~~current~~ segment shall be placed in the INFORMATION field, and the VALID bit shall be set to 1. If the segment descriptor specifies a transfer count in blocks, then the residual count is the number of blocks remaining for transfer, otherwise, the residual count is the number of bytes remaining for transfer. If no data has been transferred for the segment being processed at the time the error occurred, then the VALID bit shall be set to 0 and the content of the INFORMATION field is not defined. Segment descriptors that do not specify a transfer count shall not have a valid residual count returned;

e) If the exception condition is reported by the source logical unit, then the first byte of the COMMAND-SPECIFIC INFORMATION field shall specify the starting byte number, relative to the first byte of sense data, of an area that contains (unchanged) the source logical unit's status byte and sense data. A zero value indicates that no status byte ~~or~~ and sense data is being returned for the source logical unit;

f) If the exception condition is reported by the destination logical unit, then the second byte of the COMMAND-SPECIFIC INFORMATION field shall specify the starting byte number, relative to the first byte of sense data, of an area that contains (unchanged) the destination logical unit's status byte and sense data. A zero value indicates that no status byte ~~or~~ and sense data is being returned for the destination logical unit;

g) If, during the ~~execution~~ processing of a segment descriptor, the copy manager determines that a target is not reachable, then the SENSE-KEY SPECIFIC field shall be set as described in ~~section~~ 7.20.1, with the FIELD POINTER field indicating the first byte of the ~~segment~~ target descriptor which identifies the target; and

h) If, during the ~~execution~~ processing of a segment descriptor, the copy manager detects an error in the segment descriptor, then the SENSE-KEY SPECIFIC field shall be set as described in ~~section~~ 7.20.1, with the FIELD POINTER field indicating byte in error. The FIELD POINTER field may ~~can~~ be used to indicate an offset into either the parameter data or the segment descriptor. The SD bit is used to differentiate between these two cases. The SD bit shall be set to zero to indicate the FIELD POINTER field contains an offset from the start of the parameter data. The SD bit shall be set to one to indicate the FIELD POINTER field contains an offset from the start of the segment descriptor.

7.1.3 Descriptor type codes

Target descriptors and segment descriptors share a single set of code values that identify the type of descriptor (see table 10). Segment descriptors use codes in the range 00h to BFh. The definitions of codes between C0h and DFh are vendor specific. Target descriptors use codes in the range E0h to FFh.

Table 10 — EXTENDED COPY descriptor type codes

Descriptor type code	Reference	Description [1]	Shorthand [1]
00h	7.1.5.1	Copy from block device to stream device	block→stream
01h	7.1.5.2	Copy from stream device to block device	stream→block
02h	7.1.5.3	Copy from block device to block device	block→block
03h	7.1.5.4	Copy from stream device to stream device	stream→stream
04h	7.1.5.5	Copy inline data to stream device	inline→stream
05h	7.1.5.6	Copy embedded data to stream device	embedded→stream

NOTES

[1] Block devices are those with peripheral device type codes 0h, 4h, 5h, 7h, and Eh. Stream devices are those devices with peripheral device type codes 1h and 3h. Tape devices are those with peripheral device type code 01h. See 7.4.1 for peripheral device type code definitions.

[2] The application client ~~should~~ shall use the RECEIVE COPY RESULTS with a RECEIVE DATA service action to retrieve data held for it by the copy manager (see 7.77.2).

Table 10 — EXTENDED COPY descriptor type codes

Descriptor type code	Reference	Description [1]	Shorthand [1]
06h	7.1.5.7	Read from stream device and discard	stream→discard
07h	7.1.5.8	Verify block or stream device operation	
08h	7.1.5.9	Copy block device with offset to stream device	block<o>→stream
09h	7.1.5.10	Copy stream device to block device with offset	stream→block<o>
0Ah	7.1.5.11	Copy block device with offset to block device with offset	block<o>→block<o>
0Bh	7.1.5.1	Copy from block device to stream device and hold a copy of read data for application client **[2]**	block→stream +application client
0Ch	7.1.5.2	Copy from stream device to block device and hold a copy of read data for application client **[2]**	stream→block +application client
0Dh	7.1.5.3	Copy from block device to block device and hold a copy of read data for application client **[2]**	block→block +application client
0Eh	7.1.5.4	Copy from stream device to stream device and hold a copy of read data for application client **[2]**	stream→stream +application client
0Fh	7.1.5.7	Read from stream device and hold a copy of read data for application client **[2]**	stream→discard +application client
10h	7.1.5.12	Write filemarks to sequential-access device	filemark→tape
11h	7.1.5.13	Space records or filemarks on sequential-access device	space→tape
12h	7.1.5.14	Locate on sequential-access device	locate→tape
13h	7.1.5.15	Image copy from tape device to tape device	<i>tape→<i>tape
14h - BFh		Reserved for segment descriptors	
C0h - DFh		Vendor unique descriptors	
E0h	7.1.4.1	World Wide Name target descriptor	
E1h	7.1.4.2	N_port D_ID target descriptor	
E2h	7.1.4.3	World Wide Name & N_port D_ID target descriptor	
E3h	7.1.4.4	SCSI B_T_L target descriptor	
E4h	7.1.4.5	Identification descriptor target descriptor	
E5h - FFh		Reserved for target descriptors	

NOTES

[1] Block devices are those with peripheral device type codes 0h, 4h, 5h, 7h, and Eh. Stream devices are those devices with peripheral device type codes 1h and 3h. Tape devices are those with peripheral device type code 01h. See 7.4.1 for peripheral device type code definitions.

[2] The application client ~~should~~ shall use the RECEIVE COPY RESULTS with a RECEIVE DATA service action to retrieve data held for it by the copy manager (see 7.77.2).

7.1.4 Target descriptors

All target descriptors are 32 bytes in length and begin with a four-byte header that contains the DESCRIPTOR TYPE CODE field, that identifies the format of the descriptor (see table 11). The assigned values for target descriptors type codes are shown in table 10. Support for each target descriptor format is optional. If copy manager receives an unsupported descriptor type code in a target descriptor, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to UNSUPPORTED TARGET DESCRIPTOR TYPE CODE (26h/07h).

Table 11 — Target descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (E0 - FFh)							
1	Reserved			PERIPHERAL DEVICE TYPE				
2	Reserved							
3	~~target descriptor length (n-4)~~ Reserved							
4	Target descriptor parameters							
27								
28	Device type specific parameters							
31								

The DESCRIPTOR TYPE CODE field is described in 7.1.3. The PERIPHERAL DEVICE TYPE field is described in 7.4.1. The value in the DESCRIPTOR TYPE CODE field determines the format of the target descriptor parameters that follow the four-byte header and precede the device type specific parameters. The values in the DESCRIPTOR TYPE CODE field are listed in table 10.

~~The Address Type specifies the length and format of the target descriptor. Table 12 lists the target descriptor formats enumerated by Address Type. The Device Type field defines the peripheral device type as defined in the standard inquiry data (i.e. direct access, etc.).~~

~~The target descriptor length field contains the number of bytes in the Target descriptor parameters.~~

The value in the PERIPHERAL DEVICE TYPE field determines the format of the device type specific parameters that follow the target descriptor parameters. The device type specific parameters convey information specific to the type of device identified by the target descriptor. Table 12 lists the peripheral device type code values having formats defined for the device type specific parameters in a target descriptor. Peripheral device types with code values not listed in table 12 are reserved in the EXTENDED COPY parameter list.

Table 12 — Device type specific parameters in target descriptors

Peripheral Device Type	Reference	Description	Shorthand
00h, 04h, 05h, 07h, and 0Eh	7.1.4.6	Block devices	Block
01h	7.1.4.7	Sequential-access devices	Stream or Tape
03h	7.1.4.8	Processor devices	Stream

The copy manager may, prior to ~~executing~~ processing a segment descriptor, verify the information in a ~~the~~ target descriptor's device specific fields. However, the copy manager shall not issue any commands that change the state of the target device to verify the information.

~~Device specific information is currently defined for device types 00h and 01h. Device specific information for device types other than 00h and 01h are not yet defined. Until device specific information is defined for these other types, all block devices shall use device specific information of type 00h and all stream devices shall use the device specific information of type 01h.~~

7.1.4.1 World Wide Name target descriptor format

The target descriptor format shown in table 13 is used to identify a target using its World Wide Name.

Table 13 — World Wide Name target descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (E0h)							
1	Reserved			PERIPHERAL DEVICE TYPE				
2	Reserved							
3	~~target descriptor length (n-4)~~ Reserved							
4	(MSB)			LOGICAL UNIT NUMBER				
11								(LSB)
12	(MSB)			Reserved {WORLD WIDE NAME}				
19								(LSB)
20				Reserved				
27								
28				Device type specific parameters				
31								

The DESCRIPTOR TYPE CODE and PERIPHERAL DEVICE TYPE fields and the device type specific parameters are described in 7.1.4.

The LOGICAL UNIT NUMBER field specifies the logical unit within the SCSI device addressed by the data in the WORLD WIDE NAME field that shall be the target (source or destination) for EXTENDED COPY operations.

The WORLD WIDE NAME field shall contain TBD.

7.1.4.2 N_port D_ID target descriptor format

The target descriptor format shown in table 14 is used to identify a target using its N_port D_ID.

Table 14 — N_port D_ID target descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (E1h)							
1	Reserved			PERIPHERAL DEVICE TYPE				
2	Reserved							
3	~~target descriptor length (n-4)~~ Reserved							
4	(MSB)			LOGICAL UNIT NUMBER				
11								(LSB)
12				Reserved				
20								
21	(MSB)							
22				Reserved {N_PORT (D_ID)}				
23								(LSB)
24				Reserved				
27								
28				Device type specific parameters				
31								

The DESCRIPTOR TYPE CODE and PERIPHERAL DEVICE TYPE fields and the device type specific parameters are described in 7.1.4.

The LOGICAL UNIT NUMBER field specifies the logical unit within the SCSI device addressed by the data in the N_PORT (D_ID) field that shall be the target (source or destination) for EXTENDED COPY operations.

The N_PORT (D_ID) field shall contain TBD.

7.1.4.3 World Wide Name and N_port D_ID target descriptor format

Targets addressed using World Wide Name along with their N_port D_ID are identified using the target descriptor format shown in table 15.

Table 15 — World Wide Name and N_port D_ID target descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (E2h)							
1	Reserved			PERIPHERAL DEVICE TYPE				
2	Reserved							
3	~~target descriptor length (n-4)~~ Reserved							
4	(MSB)			LOGICAL UNIT NUMBER				
11								(LSB)
12	(MSB)			Reserved {WORLD WIDE NAME}				
19								(LSB)
20	Reserved							
21	(MSB)			Reserved {N_PORT (D_ID)}				
22								
23								(LSB)
24	Reserved							
27								
28	Device type specific parameters							
31								

The DESCRIPTOR TYPE CODE and PERIPHERAL DEVICE TYPE fields and the device type specific parameters are described in 7.1.4.

The LOGICAL UNIT NUMBER field specifies the logical unit with in the SCSI device addressed by the data in the WORLD WIDE NAME and N_PORT (D_ID) fields that shall be the target (source or destination) for EXTENDED COPY operations.

The WORLD WIDE NAME field shall contain TBD.

The N_PORT (D_ID) field shall contain TBD.

7.1.4.4 Parallel Interface T_L target descriptor format

Targets addressed using their parallel SCSI bus Target ID, and logical unit number are identified using the target descriptor format shown in table 16.

Table 16 — Parallel Interface T_L target descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (E3h)							
1	Reserved			PERIPHERAL DEVICE TYPE				
2	Reserved							
3	~~target descriptor length (n-4)~~ Reserved							
4	(MSB)			LOGICAL UNIT NUMBER				
11								(LSB)
12	Vendor unique							
13	TARGET IDENTIFIER							
14	Reserved							
27								
28	Device type specific parameters							
31								

The DESCRIPTOR TYPE CODE and PERIPHERAL DEVICE TYPE fields and the device type specific parameters are described in 7.1.4.

The LOGICAL UNIT NUMBER field specifies the logical unit with in the SCSI device addressed by the data in the TARGET IDENTIFIER field that shall be the target (source or destination) for EXTENDED COPY operations.

The TARGET IDENTIFIER field specifies the SCSI target identifier to be used when this target descriptor identifies the source or destination of an EXTENDED COPY operation.

7.1.4.5 Identification descriptor target descriptor format

The target descriptor format shown in table 17 instructs the copy manager to locate a target and logical unit that returns a device identification VPD page (see 8.4.3) containing an Identification descriptor having the specified CODE SET, ASSOCIATION, IDENTIFIER TYPE, IDENTIFIER LENGTH, and IDENTIFIER field values. The copy manager may use any N_port, target identifier and logical unit number values the result in matching VPD field values to address the copy device. If multiple N_port, target identifiers and logical unit number combinations access matching VPD field values, the copy manager may use any combination to address the copy device and shall try another combination in the event that one combination becomes non-operational during the processing of an EXTENDED COPY command.

Table 17 — Identification descriptor target descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (E4h)							
1	Reserved			PERIPHERAL DEVICE TYPE				
2	Reserved							
3	~~target descriptor length (n-4)~~ Reserved							
4	Reserved				CODE SET			
5	Reserved		ASSOCIATION		IDENTIFIER TYPE			
6	Reserved							
7	IDENTIFIER LENGTH (n-7)							
8	(MSB) IDENTIFIER							
n								(LSB)
n+1	Reserved							
27								
28	Device type specific parameters							
31								

The DESCRIPTOR TYPE CODE and PERIPHERAL DEVICE TYPE fields and the device type specific parameters are described in 7.1.4.

The contents of the CODE SET, ASSOCIATION, IDENTIFIER TYPE, IDENTIFIER LENGTH, and IDENTIFIER fields is specified in 8.4.3.

The identifier length shall be 20 or less. If the identifier length is 20 there shall be no reserved bytes between the target descriptor parameters and the device type specific parameters.

Some combinations of code set, association, identifier type, identifier length and identifier do not uniquely identify a logical unit to serve as a copy target device. The application client shall not send such combinations to the copy manager.

7.1.4.6 Device type specific target descriptor parameters for block device types

The format for the device type specific target descriptor parameters for block device types (device type code values 00h, 04h, 05h, 07h, and 0Eh) is shown in table 18.

Table 18 — Device type specific target descriptor parameters for block device types

Bit / Byte	7	6	5	4	3	2	1	0
28	Reserved					PAD	Reserved	
29	(MSB)							
30	DISK BLOCK LENGTH							
31								(LSB)

~~The pad bit is used in combination with the cat bit to control determine the actions on Inexact Segments. The interaction for various combinations of Pad and Cat bits is shown in Table 21.~~The PAD bit is used in conjunction with the CAT bit (see 7.1.5) in the segment descriptor to determine what action should be taken when a segment of the copy does not fit exactly into an integer number of destination blocks.

~~The only information required for the block device is the Disk Block Length for the device and the Pad bit.~~ The DISK BLOCK LENGTH is the number of bytes in a disk block for the logical device being addressed.

7.1.4.7 Device type specific target descriptor parameters for stream device types

The format for the device type specific target descriptor parameters for the sequential-access ~~stream~~ device type (device type code value 01h) is shown in table 19.

Table 19 — Device type specific target descriptor parameters for processor device types

Bit / Byte	7	6	5	4	3	2	1	0
28	Reserved					PAD	SILI	FIXED
29	(MSB)							
30	STREAM BLOCK LENGTH							
31								(LSB)

The contents of the FIXED bit and STREAM BLOCK LENGTH field ~~in the Device Specific information~~ are combined with the STREAM DEVICE TRANSFER LENGTH field in the segment descriptor to determine the length of the stream read or write operation see table 20.

Table 20 — Stream device transfer lengths

FIXED bit	STREAM BLOCK LENGTH field	Description
0	0	Use variable length reads or writes. The number bytes for each read or write is specified by the STREAM DEVICE TRANSFER LENGTH field in the segment descriptor.
0	not 0	The command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

Table 20 — Stream device transfer lengths

FIXED bit	STREAM BLOCK LENGTH field	Description
1	0	The command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST
1	not 0	Use fixed record length reads or writes. The number of bytes for each read or write shall be the product of the STREAM BLOCK LENGTH field and the STREAM DEVICE TRANSFER LENGTH field in the segment descriptor.

~~The pad bit is used in combination with the cat bit to control determine the actions on Inexact Segments. The interaction for various combinations of Pad and Cat bits is shown in Table 21.~~The PAD bit is used in conjunction with the CAT bit (see 7.1.5) in the segment descriptor to determine what action should be taken when a segment of the copy does not fit exactly into an integer number of destination blocks.

The SILI bit indicates the value used in the SILI bit of any read commands issued to the target.

7.1.4.8 Device type specific target descriptor parameters for processor device types

The format for the device type specific target descriptor parameters for the processor device type (device type code value 03h) is shown in table 21.

Table 21 — Device type specific target descriptor parameters for stream device types

Bit Byte	7	6	5	4	3	2	1	0
28	Reserved					PAD	Reserved	
29	Reserved							
31								

The PAD bit is used in conjunction with the CAT bit (see 7.1.5) in the segment descriptor to determine what action should be taken when a segment of the copy does not fit exactly into an integer number of destination blocks.

When the processor device is a source, the number of bytes to be transferred by a SEND command shall be specified by STREAM DEVICE TRANSFER LENGTH field in the segment descriptor. When the processor device is a destination, the number of bytes to be transferred by a RECEIVE command shall be specified by STREAM DEVICE TRANSFER LENGTH field in the segment descriptor.

7.1.5 Segment Descriptors

All segment descriptors begin with the eight byte header shown in table 22. ~~The first byte of the segment descriptor is the descriptor type code field which is used to describe the operation for the segment. A list of defined segment descriptor type codes can be found in Table 10.~~

Table 22 — Segment descriptor header

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (00h-~~13h~~3Fh)							
1	Reserved						DC	CAT
2	(MSB)			DESCRIPTOR LENGTH				
3								(LSB)
4	(MSB)			SOURCE TARGET DESCRIPTOR INDEX				
5								(LSB)
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3. Support for each segment descriptor format is optional. If copy manager receives an unsupported descriptor type code in a segment descriptor, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to UNSUPPORTED SEGMENT DESCRIPTOR TYPE CODE (26h/09h).

The destination count (DC) bit is used in ~~with~~ those segment descriptors where both the source and destination devices have the same storage structure (block or stream). The DC bit is only applicable to segment descriptors with DESCRIPTOR TYPE CODE values of 02h, 03h, 0Dh, and 0Eh. The DC bit is reserved ~~ignored~~ for all other segment descriptors. Details of usage for the DC bit appear in the clauses defining the segment descriptors that use the bit.

The CAT bit is used in conjunction with the PAD bit in the ~~segment~~ target descriptors to define what action should be taken when a segment of the copy does not fit exactly into an integral number of destination blocks. Table 23 defines the operation of the PAD and CAT bits. ~~See Table 21 for a description on how the Pad and Cat bits interact.~~

Table 23 — PAD and CAT bit definitions (part 1 of 2)

PAD bit in		CAT bit	Copy manager action
Source target descriptor	Destination target descriptor		
1	0	0	On inexact segments, the copy manager shall strip input characters from the final source block(s), always stopping at the end of a complete block.
0 or 1	1	0	On inexact segments, the copy manager shall add pad characters (00h) to the destination block to completely fill the block.

Table 23 — PAD and CAT bit definitions (part 2 of 2)

PAD bit in		CAT bit	Copy manager action
Source target descriptor	Destination target descriptor		
0 or 1	0 or 1	1	The copy manager shall always write or read complete blocks. On inexact segments, the remainder of the destination block data shall be taken from the next segment. If the CAT bit is one on the last segment of an EXTENDED COPY command the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to an COPY ABORTED and the additional sense code shall be set to UNEXPECTED INEXACT SEGMENT (26h/0Ah).
0	0	0	The occurrence of an inexact segment when both PAD bits and the CAT bit are all zero shall be an error. The the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to an COPY ABORTED and the additional sense code shall be set to UNEXPECTED INEXACT SEGMENT (26h/0Ah).

The DESCRIPTOR LENGTH field contains the length in bytes of the fields that follow the DESCRIPTOR LENGTH field in the ~~current~~ segment descriptor. ~~The length is defined as the actual length of the descriptor - 4. The Descriptor Length does not include any embedded data where applicable.~~ The copy manager shall verify value in the DESCRIPTOR LENGTH field based on the descriptor type and the length of the embedded data, if any. If the length is not correct, the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INCORRECT DESCRIPTOR LENGTH (26h/0Bh).

~~The Source and Destination Target Identifiers specify the indexes into the Target Descriptors (See Section 7.1.3). For some segment descriptors, either the Source or Destination Target Identifier is not applicable; in those cases, the field should be coded as zero.~~

The SOURCE TARGET DESCRIPTOR INDEX field contains an index into the target descriptor list (see 7.1) identifying the source target device. The DESTINATION TARGET DESCRIPTOR INDEX field contains an index into the target descriptor list (see 7.1) identifying the destination target device. Some segment descriptor formats do not require a SOURCE TARGET DESCRIPTOR INDEX field or a DESTINATION TARGET DESCRIPTOR INDEX field, in which case the field is reserved.

If the target identified ~~addressed~~ by a SOURCE TARGET DESCRIPTOR INDEX field or a DESTINATION TARGET DESCRIPTOR INDEX field is not accessible to the copy manager, then the command ~~should~~ shall be terminated with a CHECK CONDITION status, the sense key shall be set to COPY ABORTED and the additional sense code shall be set to UNREACHABLE COPY TARGET (08h/04h).

7.1.5.1 Block device to stream device operations

The segment descriptor format shown in table 24 is used for all copy operations that move data from a block device to a stream device.

Table 24 — Block device to or from stream device segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (00h, 01h, 0Bh, or 0Ch)							
1	Reserved							CAT
2	(MSB)			DESCRIPTOR LENGTH (20)				
3								(LSB)
4	(MSB)			SOURCE TARGET DESCRIPTOR INDEX				
5								(LSB)
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	Reserved							
9	(MSB)			STREAM DEVICE TRANSFER LENGTH				
10								
11								(LSB)
12	Reserved							
13	Reserved							
14	(MSB)			BLOCK DEVICE NUMBER OF BLOCKS				
15								(LSB)
16	(MSB)			BLOCK DEVICE LOGICAL BLOCK ADDRESS				
23								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Two DESCRIPTOR TYPE CODE values use the segment descriptor format described in this clause.

For descriptor type code 00h (block→stream), the copy manager shall copy the data from the source block device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination stream device identified by the DESTINATION TARGET DESCRIPTOR INDEX field using the logical blocks starting at the location identified by the BLOCK DEVICE LOGICAL BLOCK ADDRESS field and continuing for the number of blocks specified in the BLOCK DEVICE NUMBER OF BLOCKS field. The data shall be written to the stream device starting at the current position of the media.

For descriptor type code 0Bh (block→stream+application client), the copy manager shall copy the data from the source block device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination stream device identified by the DESTINATION TARGET DESCRIPTOR INDEX field using the logical blocks starting at the location identified by the BLOCK DEVICE LOGICAL BLOCK ADDRESS field and continuing for the number of blocks specified in the BLOCK DEVICE NUMBER OF BLOCKS field. The data shall be written to the stream device starting at the current position of the media. The copy manager also shall hold a copy of the read data for delivery to the application client upon completion of the EXTENDED COPY command in response to a RECEIVE COPY RESULTS command with RECEIVE DATA service action as described in 7.77.2. If the copy manager supports the 0Bh

descriptor type code it also shall support the RECEIVE COPY RESULTS command with RECEIVE DATA service action.

The CAT bit is described in 7.1.5.

The DESCRIPTOR LENGTH field shall contain 20 (0014h) and the copy manager shall verify this value as described in 7.1.5. The SOURCE TARGET DESCRIPTOR INDEX and DESTINATION TARGET DESCRIPTOR INDEX fields are described in 7.1.5.

The STREAM DEVICE TRANSFER LENGTH field specifies the amount of data to be written on each write operation to the ~~tape~~ stream device. See 7.1.4.7 ~~Table 20~~ for ~~information on~~ a description of how data in the STREAM DEVICE TRANSFER LENGTH field in the segment descriptor interacts with data in the STREAM BLOCK LENGTH field in the device type specific target descriptor parameters for the stream device type.

The BLOCK DEVICE NUMBER OF BLOCKS field specifies the number logical blocks to be read in the ~~current~~ segment. A value of zero indicates that no blocks shall be transferred in this segment. This shall not be considered as an error.

The BLOCK DEVICE LOGICAL BLOCK ADDRESS field specifies the starting logical block address on the ~~logical unit~~ block device for this segment.

7.1.5.2 Stream device to block device operations

The segment descriptor format shown in table 24 (see 7.1.5.1) also is used for all copy operations that move data from a stream device to a block device. Two DESCRIPTOR TYPE CODE values use the segment descriptor format described in this clause.

For descriptor type code 01h (stream→block), the copy manager shall copy the data from the source stream device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination block device identified by the DESTINATION TARGET DESCRIPTOR INDEX field using the stream data starting at the current position of the stream device. The data shall be written to logical blocks starting at the location identified by the BLOCK DEVICE LOGICAL BLOCK ADDRESS field and continuing for the number of blocks specified in the BLOCK DEVICE NUMBER OF BLOCKS field.

For descriptor type code 0Ch (stream→block+application client), the copy manager shall copy the data from the source stream device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination block device identified by the DESTINATION TARGET DESCRIPTOR INDEX field using the stream data starting at the current position of the stream device. The data shall be written to logical blocks starting at the location identified by the BLOCK DEVICE LOGICAL BLOCK ADDRESS field and continuing for the number of blocks specified in the BLOCK DEVICE NUMBER OF BLOCKS field. The copy manager also shall hold a copy of the read data for delivery to the application client upon completion of the EXTENDED COPY command in response to a RECEIVE COPY RESULTS command with RECEIVE DATA service action as described in 7.77.2. If the copy manager supports the 0Ch descriptor type code it also shall support the RECEIVE COPY RESULTS command with RECEIVE DATA service action.

The CAT bit is described in 7.1.5.

The DESCRIPTOR LENGTH field shall contain 20 (0014h) and the copy manager shall verify this value as described in 7.1.5. The SOURCE TARGET DESCRIPTOR INDEX and DESTINATION TARGET DESCRIPTOR INDEX fields are described in 7.1.5.

The STREAM DEVICE TRANSFER LENGTH field specifies the amount of data to be read from the source stream device on each read operation. See 7.1.4.7 ~~Table 20~~ for ~~information on~~ a description of how data in the STREAM DEVICE TRANSFER LENGTH field in the segment descriptor interacts with data in the STREAM BLOCK LENGTH field in the device type specific target descriptor parameters for the stream device type.

The BLOCK DEVICE NUMBER OF BLOCKS field specifies the number blocks to be written by the ~~current~~ segment. A value of zero indicates that no blocks shall be transferred in this segment. This shall not be considered as an error.

The BLOCK DEVICE LOGICAL BLOCK ADDRESS field specifies the starting logical block address on the block device for this segment ~~descriptor~~.

7.1.5.3 Block device to block device operations

The segment descriptor format shown in table 25 is used for all copy operations that move data from a block device to a block device.

Table 25 — Block device to block device segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (02h or 0Dh)							
1	Reserved						DC	CAT
2	(MSB)			DESCRIPTOR LENGTH (24)				
3								(LSB)
4	(MSB)			SOURCE TARGET DESCRIPTOR INDEX				
5								(LSB)
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	Reserved							
9	Reserved							
10	(MSB)			BLOCK DEVICE NUMBER OF BLOCKS				
11								(LSB)
12	(MSB)			SOURCE BLOCK DEVICE LOGICAL BLOCK ADDRESS				
19								(LSB)
20	(MSB)			DESTINATION BLOCK DEVICE LOGICAL BLOCK ADDRESS				
27								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Two DESCRIPTOR TYPE CODE values use the segment descriptor format described in this clause.

For descriptor type code 02h (block→block), the copy manager shall copy the data from the source block device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination block device identified by the DESTINATION TARGET DESCRIPTOR INDEX field using the logical blocks starting at the location identified by the source BLOCK DEVICE LOGICAL BLOCK ADDRESS field and continuing for the number of blocks specified in the BLOCK DEVICE NUMBER OF BLOCKS field. The data shall be written to logical blocks starting at the location identified by the DESTINATION BLOCK DEVICE LOGICAL BLOCK ADDRESS field.

For descriptor type code 0Dh (block→block+application client), the copy manager shall copy the data from the source block device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination block device identified by the DESTINATION TARGET DESCRIPTOR INDEX field using the logical blocks starting at the location identified by the source BLOCK DEVICE LOGICAL BLOCK ADDRESS field and continuing for the number of blocks specified in the BLOCK DEVICE NUMBER OF BLOCKS field. The data shall be written to logical blocks starting at the

location identified by the DESTINATION BLOCK DEVICE LOGICAL BLOCK ADDRESS field. The copy manager also shall hold a copy of the read data for delivery to the application client upon completion of the EXTENDED COPY command in response to a RECEIVE COPY RESULTS command with RECEIVE DATA service action as described in 7.77.2. If the copy manager supports the 0Dh descriptor type code it also shall support the RECEIVE COPY RESULTS command with RECEIVE DATA service action.

The CAT bit is described in 7.1.5.

The destination count (DC) bit indicates whether the BLOCK DEVICE NUMBER OF BLOCKS field refers to the source or destination device. A DC bit of zero indicates that the BLOCK DEVICE NUMBER OF BLOCKS field refers to the source device. A DC bit of one indicates that the BLOCK DEVICE NUMBER OF BLOCKS field refers to the destination device.

The DESCRIPTOR LENGTH field shall contain 24 (0018h) and the copy manager shall verify this value as described in 7.1.5. The SOURCE TARGET DESCRIPTOR INDEX and DESTINATION TARGET DESCRIPTOR INDEX fields are described in 7.1.5.

The BLOCK DEVICE NUMBER OF BLOCKS field specifies the number of blocks to be transferred from the source block device to the destination block device. A value of zero indicates that no blocks are to be transferred. This shall not be considered as an error.

The SOURCE BLOCK DEVICE LOGICAL BLOCK ADDRESS field specifies the logical block address from which the read of data will start.

The DESTINATION BLOCK DEVICE LOGICAL BLOCK ADDRESS field specifies the block address at which the write operation will begin.

7.1.5.4 Stream device to stream device operations

The segment descriptor format shown in table 26 is used for all copy operations that move data from a stream device to a stream device.

Table 26 — Stream device to stream device segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (03h or 0Eh)							
1	Reserved						DC	CAT
2	(MSB)			DESCRIPTOR LENGTH (16)				
3								(LSB)
4	(MSB)			SOURCE TARGET DESCRIPTOR INDEX				
5								(LSB)
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	Reserved							
9	(MSB)			SOURCE STREAM DEVICE TRANSFER LENGTH				
10								
11								(LSB)
12	Reserved							
13	(MSB)			DESTINATION STREAM DEVICE TRANSFER LENGTH				
14								
15								(LSB)
16	(MSB)			TRANSFER COUNT				
19								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Two DESCRIPTOR TYPE CODE values use the segment descriptor format described in this clause.

For descriptor type code 03h (stream→stream), the copy manager shall copy the data from the source stream device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination stream device identified by the DESTINATION TARGET DESCRIPTOR INDEX field. Data shall be read from the source stream device starting at the current position of the source stream device. Data shall be written to the destination stream device starting at the current position of the destination stream device. The TRANSFER COUNT and DC fields define the number of read or write operations to be performed by the copy manager.

For descriptor type code 0Eh (stream→stream+application client), the copy manager shall copy the data from the source stream device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination stream device identified by the DESTINATION TARGET DESCRIPTOR INDEX field. Data shall be read from the source stream device starting at the current position of the source stream device. Data shall be written to the destination stream device starting at the current position of the destination stream device. The TRANSFER COUNT and DC fields define the number of read or write operations to be performed by the copy manager. The copy manager also shall hold a copy of the read data for delivery to the application client upon completion of the EXTENDED COPY command in response to a RECEIVE COPY RESULTS command with RECEIVE DATA service action as described in 7.77.2. If

the copy manager supports the 0Eh descriptor type code it also shall support the RECEIVE COPY RESULTS command with RECEIVE DATA service action.

The CAT bit is described in 7.1.5.

The destination count (DC) bit indicates whether the TRANSFER COUNT field refers to the source or destination device. A DC bit of zero indicates that the TRANSFER COUNT field indicates the number of read operations to be performed on the source device. A DC bit of one indicates that the TRANSFER COUNT field indicates the number of write operations to be performed on the destination device.

The DESCRIPTOR LENGTH field shall contain 16 (0010h) and the copy manager shall verify this value as described in 7.1.5. The SOURCE TARGET DESCRIPTOR INDEX and DESTINATION TARGET DESCRIPTOR INDEX fields are described in 7.1.5.

The source STREAM DEVICE TRANSFER LENGTH field specifies the amount of data to be read from the source stream device on each read operation. See 7.1.4.7 ~~Table 20~~ for ~~information on~~ a description of how data in the SOURCE STREAM DEVICE TRANSFER LENGTH field in the segment descriptor interacts with data in the STREAM BLOCK LENGTH field in the device type specific target descriptor parameters for the source stream device type.

The DESTINATION STREAM DEVICE TRANSFER LENGTH field specifies the amount of data to be written to the destination stream device on each write operation. See 7.1.4.7 ~~Table 20~~ for ~~information on~~ a description of how data in the DESTINATION STREAM DEVICE TRANSFER LENGTH field in the segment descriptor interacts with data in the STREAM BLOCK LENGTH field in the device type specific target descriptor parameters for the destination stream device type.

The TRANSFER COUNT field specifies the number of read/write operations that ~~must~~ shall be executed for this segment descriptor. The DC bit indicates whether the number of reads or writes is specified. A value of zero indicates that no reads or writes are to be performed. This shall not be considered as an error.

7.1.5.5 Inline data to stream device operation

The segment descriptor format shown in table 27 instructs the copy manager to write inline data from the EXTENDED COPY parameter list to a stream device.

Table 27 — Inline data to stream device segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (04h)							
1	Reserved							CAT
2	(MSB)			DESCRIPTOR LENGTH (16)				
3								(LSB)
4	Reserved							
5	Reserved							
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	Reserved							
9	(MSB)			STREAM DEVICE TRANSFER LENGTH				
10								
11								(LSB)
12	(MSB)			INLINE DATA OFFSET				
15								(LSB)
16	(MSB)			INLINE DATA NUMBER OF BYTES				
19								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 04h (inline→stream) instructs the copy manager to write inline data from the EXTENDED COPY parameter list to a stream device. The inline data shall be read from the optional inline data at the end of the EXTENDED COPY parameter list. The data shall be written to the destination stream device identified by the DESTINATION TARGET DESCRIPTOR INDEX field starting at the current position of the stream device.

~~The inline to stream segment descriptors must be processed in a sequential manner from first to last to ensure that the inline data at the end of the parameter list is properly assigned.~~

The CAT bit is described in 7.1.5.

The DESCRIPTOR LENGTH field shall contain 16 (0010h) and the copy manager shall verify this value as described in 7.1.5. The DESTINATION TARGET DESCRIPTOR INDEX field is described in 7.1.5.

The STREAM DEVICE TRANSFER LENGTH field specifies the amount of data to be written to the stream device on each write operation. See 7.1.4.7 ~~Table 20~~ for ~~information on~~ a description of how data in the STREAM DEVICE TRANSFER LENGTH field in the segment descriptor interacts with data in the STREAM BLOCK LENGTH field in the device type specific target descriptor parameters for the destination stream device type.

The value in the INLINE DATA OFFSET field is added to the location of the first byte of inline data in the EXTENDED COPY parameter list (see table 9) to locate the first byte of inline data to be written to the stream device. The INLINE DATA OFFSET value shall be a multiple of 4.

The INLINE DATA NUMBER OF BYTES field specifies the number of bytes of inline data that are to be transferred to the stream device.

If the sum of the INLINE DATA OFFSET and the INLINE DATA NUMBER OF BYTES values exceeds the value in the INLINE DATA LENGTH field (SEE TABLE 9), the copy manager shall terminate the command with a CHECK CONDITION status. The sense key shall be set to COPY ABORTED and the additional sense code shall be set to ~~DATA UNDERRUN~~ INLINE DATA LENGTH EXCEEDED (26h/0Ch).

7.1.5.6 Embedded data to stream device operation

The segment descriptor format shown in table 28 instructs the copy manager to write embedded data from the segment descriptor to a stream device.

Table 28 — Embedded data to stream device segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (05h)							
1	Reserved							CAT
2	(MSB)			DESCRIPTOR LENGTH (n-4)				
3								(LSB)
4	Reserved							
5	Reserved							
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	Reserved							
9	(MSB)			STREAM DEVICE TRANSFER LENGTH				
10								
11								(LSB)
12	(MSB)			EMBEDDED DATA NUMBER OF BYTES				
13								(LSB)
14	(MSB)			EMBEDDED DATA LENGTH (n-16)				
15								(LSB)
16				EMBEDDED DATA				
n								

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 05h (embedded→stream) instructs the copy manager to write embedded data from the segment descriptor to a stream device. The embedded data shall be read from the segment descriptor. The data shall be written to the destination stream device identified by the DESTINATION TARGET DESCRIPTOR INDEX field starting at the current position of the stream device.

The CAT bit is described in 7.1.5.

The DESCRIPTOR LENGTH field shall contain the length in bytes of the fields that follow the the DESCRIPTOR LENGTH field (including embedded data) and the copy manager shall verify this value. If the value in the DESCRIPTOR LENGTH field does not equal the value in the EMBEDDED DATA LENGTH field plus 12, then a INCORRECT DESCRIPTOR LENGTH error shall be reported as described in 7.1.5.

The DESTINATION TARGET DESCRIPTOR INDEX field is described in 7.1.5.

The STREAM DEVICE TRANSFER LENGTH field specifies the amount of data to be written to the stream device on each write operation. See 7.1.4.7 ~~Table 20~~ for ~~information on~~ a description of how data in the STREAM DEVICE TRANSFER LENGTH field in the segment descriptor interacts with data in the STREAM BLOCK LENGTH field in the device type specific target descriptor parameters for the destination stream device type.

The EMBEDDED DATA NUMBER OF BYTES field specifies the number of bytes of embedded data that are to be transferred to the stream device. The EMBEDDED DATA NUMBER OF BYTES value shall be less than or equal to the EMBEDDED DATA LENGTH value.

The EMBEDDED DATA LENGTH field specifies the number of bytes of embedded data in the segment descriptor. The value shall be a multiple of 4.

7.1.5.7 Stream device to discard operation

The segment descriptor format shown in table 29 instructs the copy manager to read data from a stream device and not copy it to any destination device.

Table 29 — Stream device to discard segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (06h or 0Fh)							
1	Reserved							CAT
2	(MSB)			DESCRIPTOR LENGTH (12)				
3								(LSB)
4	(MSB)			SOURCE TARGET DESCRIPTOR INDEX				
5								(LSB)
6	Reserved							
7	Reserved							
8	Reserved							
9	(MSB)			STREAM DEVICE TRANSFER LENGTH				
10								
11								(LSB)
12	(MSB)			NUMBER OF BYTES				
15								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Two DESCRIPTOR TYPE CODE values use the segment descriptor format described in this clause.

For descriptor type code 06h (stream→discard), the copy manager shall read data from the source stream device identified by the SOURCE TARGET DESCRIPTOR INDEX field starting at the current position of the source stream device. The data read shall not be transferred to any destination device.

For descriptor type code 0Fh (stream→discard+application client), the copy manager shall read data from the source stream device identified by the SOURCE TARGET DESCRIPTOR INDEX field starting at the current position of the source stream device and hold a copy of the read data for delivery to the application client upon completion of the EXTENDED COPY command in response to a RECEIVE COPY RESULTS command with RECEIVE DATA service action as described in 7.77.2. If the copy manager supports the 0Fh descriptor type code it also shall support the RECEIVE COPY RESULTS command with RECEIVE DATA service action.

The CAT bit is described in 7.1.5.

The DESCRIPTOR LENGTH field shall contain 12 (000Ch) and the copy manager shall verify this value as described in 7.1.5. The DESTINATION TARGET DESCRIPTOR INDEX field is described in 7.1.5.

The source STREAM DEVICE TRANSFER LENGTH field specifies the amount of data to be read from the source stream device on each read operation. See 7.1.4.7 ~~Table 20~~ for ~~information on~~ a description of how data in the SOURCE STREAM DEVICE TRANSFER LENGTH field in the segment descriptor interacts with data in the STREAM BLOCK LENGTH field in the device type specific target descriptor parameters for the source stream device type.

The NUMBER OF BYTES field specifies the number of bytes to be read from ~~discarded by~~ the stream device.

7.1.5.8 Verify device operation

The segment descriptor format shown in table 30 instructs the copy manager to verify the accessibility of a device.

Table 30 — Verify device operation segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (07h)							
1	Reserved							
2	(MSB)			DESCRIPTOR LENGTH (4)				
3								(LSB)
4	(MSB)			SOURCE TARGET DESCRIPTOR INDEX				
5								(LSB)
6	Reserved							
7	Reserved							

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 07h instructs the copy manager to verify the accessibility of the device identified by the SOURCE TARGET DESCRIPTOR INDEX field. The accessibility ~~target existence~~ should be verified without disturbing established unit attention or ACA conditions ~~contingent allegiance states~~, for example, using the INQUIRY command (see 7.4).

The DESCRIPTOR LENGTH field shall contain 4 (0004h) and the copy manager shall verify this value as described in 7.1.5. The SOURCE TARGET DESCRIPTOR INDEX field is described in 7.1.5.

7.1.5.9 Block device with offset to stream device operation

The segment descriptor format shown in table 31 is used to instruct the copy manager to move data from a block device with a byte offset to a stream device.

Table 31 — Block device with offset to or from stream device segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (08h or 09h)							
1	Reserved							CAT
2	(MSB)			DESCRIPTOR LENGTH (24)				
3								(LSB)
4	(MSB)			SOURCE TARGET DESCRIPTOR INDEX				
5								(LSB)
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	Reserved							
9	(MSB)			STREAM DEVICE TRANSFER LENGTH				
10								
11								(LSB)
12	(MSB)			NUMBER OF BYTES				
15								(LSB)
16	(MSB)			BLOCK DEVICE LOGICAL BLOCK ADDRESS				
23								(LSB)
24	Reserved							
25	Reserved							
26	(MSB)			BLOCK DEVICE BYTE OFFSET				
27								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 08h (block→stream) instructs the copy manager to copy the data from the source block device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination stream device identified by the DESTINATION TARGET DESCRIPTOR INDEX field using data starting at the location identified by BLOCK DEVICE BYTE OFFSET field in the logical block identified by the BLOCK DEVICE LOGICAL BLOCK ADDRESS field and continuing for the number of bytes specified in the NUMBER OF BYTES field. The data shall be written to the stream device starting at the current position of the media.

The CAT bit is described in 7.1.5.

The DESCRIPTOR LENGTH field shall contain 24 (0018h) and the copy manager shall verify this value as described in 7.1.5. The SOURCE TARGET DESCRIPTOR INDEX and DESTINATION TARGET DESCRIPTOR INDEX fields are described in 7.1.5.

The STREAM DEVICE TRANSFER LENGTH field specifies the amount of data to be written on each write operation to the ~~tape~~ stream device. See 7.1.4.7 ~~Table 20~~ for ~~information on~~ a description of how data in the STREAM DEVICE

TRANSFER LENGTH field in the segment descriptor interacts with data in the STREAM BLOCK LENGTH field in the device type specific target descriptor parameters for the stream device type.

The NUMBER OF BYTES field specifies the number bytes to be read ~~in the current segment to be copied~~. A value of zero indicates that no bytes shall be transferred in this segment. This shall not be considered as an error.

The BLOCK DEVICE LOGICAL BLOCK ADDRESS field specifies the starting logical block address on the ~~logical unit~~ source block device for this segment.

The BLOCK DEVICE BYTE OFFSET field ~~is~~ specifies the offset into the first source block at which to begin ~~transferring~~ reading bytes ~~to the destination device~~.

7.1.5.10 Stream device to block device with offset operation

The segment descriptor format shown in table 31 (see 7.1.5.9) also is used to instruct the copy manager to move data from a stream device to a block device with a byte offset.

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 09h (stream→block<o>) instructs the copy manager to copy the data from the source stream device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination block device identified by the DESTINATION TARGET DESCRIPTOR INDEX field using the stream data starting at the current position of the stream device. The data shall be written starting at the location identified by BLOCK DEVICE BYTE OFFSET field in the logical block identified by the BLOCK DEVICE LOGICAL BLOCK ADDRESS field and continuing for the number of bytes specified in the NUMBER OF BYTES field.

The content of the starting logical block on the destination device before the starting offset ~~is~~ shall be preserved. The content on the ending logical block beyond the end of the transfer ~~is~~ shall be preserved. The copy manager may implement this operation by reading ~~in~~ the starting and ending logical blocks, modifying ~~the~~ a portion of the blocks as required, and writing the full blocks to the destination device.

The CAT bit is described in 7.1.5.

The DESCRIPTOR LENGTH field shall contain 24 (0018h) and the copy manager shall verify this value as described in 7.1.5. The SOURCE TARGET DESCRIPTOR INDEX and DESTINATION TARGET DESCRIPTOR INDEX fields are described in 7.1.5.

The STREAM DEVICE TRANSFER LENGTH field specifies the amount of data to be written on each write operation to the ~~tape~~ stream device. See 7.1.4.7 ~~Table 20~~ for ~~information on~~ a description of how data in the STREAM DEVICE TRANSFER LENGTH field in the segment descriptor interacts with data in the STREAM BLOCK LENGTH field in the device type specific target descriptor parameters for the stream device type.

The NUMBER OF BYTES field specifies the number bytes to be read ~~in the current segment to be copied~~. A value of zero indicates that no bytes shall be transferred in this segment. This shall not be considered as an error.

The BLOCK DEVICE LOGICAL BLOCK ADDRESS field specifies the starting logical block address on the ~~logical unit~~ destination block device for this segment.

The BLOCK DEVICE BYTE OFFSET field is the offset into the first source block at which to begin writing data to the destination block device.

7.1.5.11 Block device with offset to block device with offset operation

The segment descriptor format shown in table 32 instructs the copy manager to move data from a block device with a byte offset to a block device with a byte offset.

Table 32 — Block device with offset to block device with offset segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (0Ah)							
1	Reserved							CAT
2	(MSB)			DESCRIPTOR LENGTH (28)				
3								(LSB)
4	(MSB)			SOURCE TARGET DESCRIPTOR INDEX				
5								(LSB)
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	(MSB)			NUMBER OF BYTES				
11								(LSB)
12	(MSB)			SOURCE BLOCK DEVICE LOGICAL BLOCK ADDRESS				
19								(LSB)
20	(MSB)			DESTINATION BLOCK DEVICE LOGICAL BLOCK ADDRESS				
27								(LSB)
28	(MSB)			SOURCE BLOCK DEVICE BYTE OFFSET				
29								(LSB)
30	(MSB)			DESTINATION BLOCK DEVICE BYTE OFFSET				
31								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 0Ah (block<o>→block<o>) instructs the copy manager to copy the data from the source block device identified by the SOURCE TARGET DESCRIPTOR INDEX field to the destination block device identified by the DESTINATION TARGET DESCRIPTOR INDEX field using data starting at the location identified by SOURCE BLOCK DEVICE BYTE OFFSET field in the logical block identified by the SOURCE BLOCK DEVICE LOGICAL BLOCK ADDRESS field and continuing for the number of bytes specified in the NUMBER OF BYTES field. The data shall be written starting at the location identified by DESTINATION BLOCK DEVICE BYTE OFFSET field in the logical block identified by the DESTINATION BLOCK DEVICE LOGICAL BLOCK ADDRESS field.

The content of the starting logical block on the destination device before the starting offset ~~is~~ shall be preserved. The content on the ending logical block beyond the end of the transfer ~~is~~ shall be preserved. The copy manager may implement this operation by reading ~~in~~ the starting and ending logical blocks, modifying ~~the~~ a portion of the blocks as required, and writing the full blocks to the destination device.

The CAT bit is described in 7.1.5.

The DESCRIPTOR LENGTH field shall contain 28 (001Ch) and the copy manager shall verify this value as described in 7.1.5. The SOURCE TARGET DESCRIPTOR INDEX and DESTINATION TARGET DESCRIPTOR INDEX fields are described in 7.1.5.

The NUMBER OF BYTES field specifies the number bytes to be read ~~in the current segment to be copied~~. A value of zero indicates that no bytes shall be transferred in this segment. This shall not be considered as an error.

The SOURCE BLOCK DEVICE LOGICAL BLOCK ADDRESS field specifies the starting address on the ~~logical unit~~ source block device for this segment.

The DESTINATION BLOCK DEVICE LOGICAL BLOCK ADDRESS field specifies the starting logical block address on the ~~logical unit~~ destination block device for this segment.

The SOURCE BLOCK DEVICE BYTE OFFSET field ~~is~~ specifies the offset into the first source block at which to begin ~~transferring~~ reading bytes ~~to the destination device~~.

The DESTINATION BLOCK DEVICE BYTE OFFSET field is the offset into the first source block at which to begin writing data to the destination block device.

7.1.5.12 Write filemarks operation

The segment descriptor format shown in table 33 instructs the copy manager to write filemarks or setmarks on the destination tape device.

Table 33 — Write filemarks operation segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (10h)							
1	Reserved							
2	(MSB)			DESCRIPTOR LENGTH (8)				
3								(LSB)
4	Reserved							
5	Reserved							
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	Reserved						WSMK	Reserved
9	(MSB)			TRANSFER LENGTH				
10								
11								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 10h (filemark→tape) instructs the copy manager to write filemarks or setmarks to the destination tape device identified by the DESTINATION TARGET DESCRIPTOR INDEX field starting at the current position of the tape device. If the PERIPHERAL DEVICE TYPE field in the target descriptor identified by the DESTINATION TARGET DESCRIPTOR INDEX field does not contain 01h, the copy manager shall terminate the command with a CHECK CONDITION status. The sense key shall be set to COPY ABORTED and the additional sense code shall be set to INVALID OPERATION FOR COPY SOURCE OR DESTINATION (26h/0Dh).

The DESCRIPTOR LENGTH field shall contain 8 (0008h) and the copy manager shall verify this value as described in 7.1.5. The DESTINATION TARGET DESCRIPTOR INDEX field is described in 7.1.5.

If the write setmark (WSMK) bit is one, the TRANSFER LENGTH field specifies the number of setmarks to be written. If the WSMK bit is zero, the TRANSFER LENGTH field specifies the number of filemarks to be written.

7.1.5.13 Space operation

The segment descriptor format shown in table 34 instructs the copy manager to send a SPACE command (see SSC) to the destination tape device.

Table 34 — Space operation segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (11h)							
1	Reserved							
2	(MSB)							
3	DESCRIPTOR LENGTH (8)							(LSB)
4	Reserved							
5	Reserved							
6	(MSB)							
7	DESTINATION TARGET DESCRIPTOR INDEX							(LSB)
8	Reserved					CODE		
9	(MSB)							
10	COUNT							
11								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 11h (space→tape) instructs the copy manager to send a SPACE command to the destination tape device identified by the DESTINATION TARGET DESCRIPTOR INDEX field. If the PERIPHERAL DEVICE TYPE field in the target descriptor identified by the DESTINATION TARGET DESCRIPTOR INDEX field does not contain 01h, the copy manager shall terminate the command with a CHECK CONDITION status. The sense key shall be set to COPY ABORTED and the additional sense code shall be set to INVALID OPERATION FOR COPY SOURCE OR DESTINATION (26h/0Dh).

The DESCRIPTOR LENGTH field shall contain 8 (0008h) and the copy manager shall verify this value as described in 7.1.5. The DESTINATION TARGET DESCRIPTOR INDEX field is described in 7.1.5.

The CODE and COUNT field contents in the SPACE command sent to the destination tape device shall be copied from the CODE and COUNT fields in the segment descriptor. All other fields in the SPACE command sent to the destination tape device that affect the positioning of the tape shall be set to zero.

7.1.5.14 Locate operation

The segment descriptor format shown in table 35 instructs the copy manager to send a LOCATE command (see SSC) to the destination tape device.

Table 35 — Locate operation segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (12h)							
1	Reserved							
2	(MSB)			DESCRIPTOR LENGTH (8)				
3								(LSB)
4	Reserved							
5	Reserved							
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	(MSB)			BLOCK ADDRESS				
11								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 12h (locate→tape) instructs the copy manager to send a LOCATE command to the destination tape device identified by the DESTINATION TARGET DESCRIPTOR INDEX field. If the PERIPHERAL DEVICE TYPE field in the target descriptor identified by the DESTINATION TARGET DESCRIPTOR INDEX field does not contain 01h, the copy manager shall terminate the command with a CHECK CONDITION status. The sense key shall be set to COPY ABORTED and the additional sense code shall be set to INVALID OPERATION FOR COPY SOURCE OR DESTINATION (26h/0Dh).

The DESCRIPTOR LENGTH field shall contain 8 (0008h) and the copy manager shall verify this value as described in 7.1.5. The DESTINATION TARGET DESCRIPTOR INDEX field is described in 7.1.5.

The BLOCK ADDRESS field contents in the LOCATE command sent to the destination tape device shall be copied from the BLOCK ADDRESS field in the segment descriptor. All other fields in the LOCATE command sent to the destination tape device that affect the positioning of the tape shall be set to zero.

NOTE 4 The restrictions described above for the LOCATE command limit the operation to locating SCSI logical block addresses in the current tape partition.

7.1.5.15 Tape device image copy operation

The segment descriptor format shown in table 36 instructs the copy manager to perform an image copy from the source tape device to the destination tape device.

Table 36 — Tape device image copy segment descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE (13h)							
1	Reserved							
2	(MSB)			DESCRIPTOR LENGTH (8)				
3								(LSB)
4	(MSB)			SOURCE TARGET DESCRIPTOR INDEX				
5								(LSB)
6	(MSB)			DESTINATION TARGET DESCRIPTOR INDEX				
7								(LSB)
8	(MSB)			COUNT				
11								(LSB)

The DESCRIPTOR TYPE CODE field is described in 7.1.3 and 7.1.5. Descriptor type code 13h (<i>tape→<i>tape) instructs the copy manager to create a compatible image of the source device medium identified by the SOURCE TARGET DESCRIPTOR INDEX field on the destination device medium identified by the DESTINATION TARGET DESCRIPTOR INDEX field beginning at their current positions. If the PERIPHERAL DEVICE TYPE field in the target descriptor identified by the SOURCE TARGET DESCRIPTOR INDEX field or the DESTINATION TARGET DESCRIPTOR INDEX field does not contain 01h, the copy manager shall terminate the command with a CHECK CONDITION status. The sense key shall be set to COPY ABORTED and the additional sense code shall be set to INVALID OPERATION FOR COPY SOURCE OR DESTINATION (26h/0Dh).

The DESCRIPTOR LENGTH field shall contain 8 (0008h) and the copy manager shall verify this value as described in 7.1.5. The SOURCE TARGET DESCRIPTOR INDEX and DESTINATION TARGET DESCRIPTOR INDEX fields are described in 7.1.5.

The tape image copy operation ~~function~~ terminates when:

a) the source device encounters an end-of-partition as defined by the source device;
b) the source device encounters an end-of-data as defined by the source device (i.e., BLANK CHECK sense key);
c) the copy manager has copied the number of consecutive filemarks specified in the COUNT field from the source device to the destination device; or
d) the copy manager has copied the number of consecutive filemarks and/or setmarks specified in the COUNT field from the source device to the destination device, if the RSMK bit in the device configuration page (see SSC) of the source device is one.

The COUNT field of zero indicates that the EXTENDED COPY command shall not terminate due to any number of consecutive filemarks or setmarks. Other error or exception conditions (e.g., early-warning, end-of-partition on destination device) may cause the EXTENDED COPY command to terminate prior to completion. In such cases, it is not possible to calculate a residue, so the INFORMATION field in the sense data shall be set to zero.

~~If the copy is attempted to a device that does not support the block size, file marks, or set marks of the source then the copy manager shall terminate the command with a CHECK CONDITION. The Sense Key shall be set to COPY ABORTED and the ASC and ASCQ shall indicate INCOMPATIBLE FORMAT.~~

7.77 RECEIVE COPY RESULTS command

The RECEIVE COPY RESULTS command (see table 37) provides a means for the application client to receive information about the copy manager or the results of a previous (or current) EXTENDED COPY command (see 7.1). The results that may ~~can~~ be returned from the previous (or current) EXTENDED COPY command are ~~either~~ copy manager status information, held ~~inline~~ data from read operations, or information about the order in which the segments were processed during the EXTENDED COPY command.

Table 37 — RECEIVE COPY RESULTS command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (84h)							
1	Reserved			SERVICE ACTION				
2	LIST IDENTIFIER							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	(MSB)							
11				ALLOCATION LENGTH				
12								
13								(LSB)
14	Reserved							
15	CONTROL							

The service actions defined for the RECEIVE COPY RESULTS command are shown in table 38.

Table 38 — RECEIVE COPY RESULTS service action codes

Code	Name	Description	Returns Data
00h	COPY STATUS	Return the current copy status of the ~~executing~~ EXTENDED COPY command identified by the LIST IDENTIFIER field.	Immediately
01h	RECEIVE DATA	Return the ~~Inline~~ held data read by EXTENDED COPY command identified by the LIST IDENTIFIER field.	When identified command has completed
02h	SEGMENT ORDER LIST	Return the segment order list that shows the order in which the segments where written to the destination device during the EXTENDED COPY command identified by the LIST IDENTIFIER field.	
03h	OPERATING PARAMETERS	Return copy manager operating parameters.	Immediately
04h-1Fh	Reserved.		

~~If an invalid Mode [service action] Value is specified, then the copy manager shall terminate the command with a CHECK CONDITION. The Sense Key shall be set to ILLEGAL REQUEST and the ASC and ASCQ shall indicate INVALID FIELD IN CDB.~~

The LIST IDENTIFIER field identifies the EXTENDED COPY command about which information is desired. The RECEIVE COPY RESULTS command ~~will~~ shall return information from the EXTENDED COPY command originated from the same initiator with a list identifier that matches the list identifier specified in the RECEIVE COPY RESULTS CDB. If no EXTENDED COPY command known to the copy manager has a matching list identifier, then the command ~~will~~ shall be terminated with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

The actual length of the RECEIVE COPY RESULTS parameter data is available in the AVAILABLE DATA parameter data field. The ALLOCATION LENGTH field in the CDB indicates how much space has been reserved for the returned parameter list. If the length is not sufficient to contain the entire parameter list, the first portion of the list shall be returned. This shall not be considered an error. If the remainder of the list is required, the application client should send a new RECEIVE COPY RESULTS command with a ALLOCATION LENGTH field large enough to contain the entire parameter list.

7.77.1 COPY STATUS service action

In response to the COPY STATUS service action, the copy manager shall return the current status of the EXTENDED COPY command (see 7.1) identified by the LIST IDENTIFIER field in the CDB. Table 40 shows the format of the information returned by the copy manager in response to the COPY STATUS service action. If a device server supports the EXTENDED COPY command, then it shall also support the RECEIVE COPY RESULTS command with COPY STATUS service action.

Table 39 — Parameter data for the COPY STATUS service action

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			AVAILABLE DATA (00000008h)				
3								(LSB)
4	STATUS							
5	(MSB)			SEGMENTS PROCESSED				
6								(LSB)
7	TRANSFER COUNT UNITS							
8	(MSB)			TRANSFER COUNT				
11								(LSB)

After completion of an EXTENDED COPY command, the copy manager shall preserve all data returned by a COPY STATUS service action ~~will be preserved by the copy manager~~ for a vendor specific ~~reasonable~~ period of time. The copy manager shall ~~will~~ discard the COPY STATUS data ~~completion status~~ when:

a) a RECIEVE COPY RESULTS command with COPY STATUS service action is received from ~~issued by~~ the same initiator ~~host~~ with a matching list identifier;
b) when another EXTENDED COPY command is received from ~~issued by~~ the same initiator ~~host~~ using the same list identifier;
c) when the copy manager detects a hard reset condition; or
d) when the copy manager requires the resources used to preserve the data.

The AVAILABLE DATA field shall contain the number of bytes present in the parameter data that follows, eight.

The STATUS field ~~is~~ contains the current status of the ~~copy manager~~ EXTENDED COPY command identified by the LIST IDENTIFIER field in the CDB~~. Valid status values that the manager can report in the Status field are as shown~~ as defined in table 40.

Table 40 — COPY STATUS STATUS values

STATUS value	Meaning
00h	Operating in progress
01h	Operation completed without errors
02h	Operation completed with errors
04h - FFh	Reserved

The SEGMENTS PROCESSED field ~~is~~ contains the number of segments the copy manager has processed for the EXTENDED COPY command identified by the LIST IDENTIFIER field in the CDB including the segment currently being processed~~ the segment descriptor (zero based) that is currently executing on the copy manager~~. This field shall be zero if the copy manager has not yet begun ~~executing~~ processing segment descriptors.

The TRANSFER COUNT UNITS field specifies the units for the TRANSFER COUNT field~~. The Transfer Count Format is coded as shown~~ as defined in table 41.

Table 41 — COPY STATUS TRANSFER COUNT UNITS values

Value	Meaning	Multiplier to convert TRANSFER COUNT field to bytes
00h	Bytes	1
01h	Kilo-bytes	2^{10} or 1024
02h	Mega-bytes	2^{20}
03h	Giag-bytes	2^{30}
04h	Tera-bytes	2^{40}
05h	Peta-bytes	2^{50}
06h - FFh	Reserved	

The TRANSFER COUNT field specifies the amount of data written to a destination device ~~transferred~~ for the EXTENDED COPY command identified by the LIST IDENTIFIER field in the CDB ~~by the EXTENDED COPY command~~ prior to receiving the RECEIVE COPY RESULTS command with COPY STATUS service action.

7.77.2 RECEIVE DATA service action

If the copy manager supports those segment descriptors that ~~transfer copied data to the host~~ hold read data for transfer to the application client, then the RECEIVE DATA service action causes the copy manager to return any held data using the format shown in table 42. If a copy manager supports any of the segment descriptor type codes that require read data to be held for the application client (see 7.1.3), then it shall also support the RECEIVE COPY RESULTS command with RECEIVE DATA service action.

Table 42 — Parameter data for the RECEIVE DATA service action

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			AVAILABLE DATA (n-4)				
3								(LSB)
4				Held data				
n								

~~Inline data which is held by the copy manager~~ Following completion of an EXTENDED COPY command, the copy manager shall preserve all data returned by a RECIEVE DATA service action ~~will be preserved~~ for a ~~reasonable~~ vendor specific period of time ~~by the copy manager~~. The application should issue a RECEIVE COPY RESULTS command with RECEIVE DATA service action immediately following completion of the EXTENDED COPY command to insure that the data is not discarded by the copy manager. The copy manager shall ~~will~~ discard the buffered inline data:

a) after all data held for a specific EXTENDED COPY command ~~it~~ has been successfully transferred to the application client ~~host~~;
b) when a RECIEVE COPY RESULTS command with RECEIVE DATA service action has been received from ~~is issued by~~ the same initiator ~~host~~ with a matching list identifier ~~for the same List ID~~, with the ALLOCATION LENGTH field set to 0;
c) when another EXTENDED COPY command is received from ~~issued by~~ the same initiator ~~host~~ using the same list identifier;
d) when the copy manager detects a hard reset condition; or
e) when the copy manager requires the resources used to preserve the data.

The AVAILABLE DATA field shall contain the number of bytes of held data available for delivery to the application client. If the amount of held data sent to the application client is reduced due to insufficient allocation length, the AVAILABLE DATA field shall not be altered.

The held data is the data held by the copy manager for delivery to the application client as proscribed by several segment descriptor type codes. The oldest byte read and held is returned in byte 4 and the byte most recently read and held is returned in byte n.

The maximum number of bytes that the copy manager will hold for the application client is returned in the HELD DATA LIMIT field in the copy manager's operating parameters (see 7.77.4). If the processing of segment descriptors requires more data to be held, the copy manager shall discard the oldest held data bytes to accommodate the new read data. When making room for new read data, the copy manager may discard more old data bytes than are needed immediately, but at any one time the copy manager shall never discard more than the smaller of 64 times the HELD DATA GRANULARITY value (see 7.77.4) or one quarter of the HELD DATA LIMIT value. Discarding of held data bytes shall not be considered an error. The only way the application client may detect the possible discarding of held data bytes is to compare the AVAILABLE DATA field to the HELD DATA LIMIT field.

7.77.3 SEGMENT ORDER LIST service action

If the copy manager supports out-of-order segment processing ~~writes~~ and the EXTENDED COPY command enabled out-of-order segment processing ~~writes~~ by setting OOE to 1 (see 7.1), then executing a RECEIVE COPY RESULTS command with SEGMENT ORDER LIST service action shall cause ~~result in~~ the copy manager to return ~~returning~~ a segment list that describes the actual order in which segments were processed resulting in data transfers to the destination device. The format of the data returned by the copy manager is shown in table 43. If a copy manager supports setting the OOE bit to 1, then it shall also support the RECEIVE COPY RESULTS command with RECEIVE DATA service action.

Table 43 — Parameter data for the SEGMENT ORDER LIST service action

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
3		AVAILABLE DATA (n-4)						(LSB)
4								
n		Segment order descriptors list						

If an EXTENDED COPY command is processed ~~issued~~ with ~~Out of Order Enable~~ OOE set to 1 (see 7.1), then the copy manager shall maintain ~~will hold a copy of~~ the Segment order descriptors list described in this clause. ~~The Segment Order List which is held by the copy manager~~ Following completion of the EXTENDED COPY command, the copy manager shall preserve the Segment order descriptors list ~~will be preserved~~ for a ~~reasonable~~ vendor specific period of time ~~by the copy manager~~. The application should issue a RECEIVE COPY RESULTS command with SEGMENT ORDER LIST service action immediately following completion of the EXTENDED COPY command to insure that the data is not discarded by the copy manager. The copy manager shall ~~will~~ discard the Segment order descriptors list:

a) after the entire Segment order descriptors list ~~it~~ has been successfully transferred to the application client ~~host~~;
b) when a RECIEVE COPY RESULTS command with SEGMENT ORDER LIST service action is received from ~~issued by~~ the same initiator ~~host~~ with a matching list identifier ~~for the same List ID~~, with the ALLOCATION LENGTH field set to 0;
c) when another EXTENDED COPY command is received from ~~issued by~~ the same initiator ~~host~~ using the same list identifier;
d) when the copy manager detects a hard reset condition; or
e) when the copy manager requires the resources used to preserve the data.

The AVAILABLE DATA field shall contain the length of the Segment order descriptors list that immediately follows the AVAILABLE DATA field. If the EXTENDED COPY command identified by the LIST IDENTIFIER field in the CDB did not have OOE set to 1, the AVAILABLE DATA field shall contain 0. If length of the Segment order descriptors list is greater than four plus allocation length, the AVAILABLE DATA field shall not be altered.

The Segment order descriptors list ~~field~~ is a ~~n ordered~~ list of descriptors that is ordered to match ~~describes~~ the order in which the segment descriptors from the EXTENDED COPY command parameter list produced data transfers

~~were transferred~~ to the destination device~~ for the EXTENDED COPY command~~. The format of a ~~the~~ Segment order descriptor is shown in table 44.

Table 44 — Segment order descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE CODE							
1	Reserved							
2	(MSB)			SEGMENT NUMBER				
3								(LSB)
4	(MSB)			PROCESSING ORDER				
5								(LSB)
6	(MSB)			DESTINATION BYTE OFFSET				
7								(LSB)
8	(MSB)			DESTINATION LOGICAL BLOCK				
15								(LSB)

The DESCRIPTOR TYPE CODE field contains the descriptor type from the segment descriptor in the EXTENDED COPY parameter list (see 7.1.5).

The SEGMENT NUMBER field contains the number of the segment descriptor to which this segment order descriptor applies. The segment~~ descriptor~~ number is based on ~~its~~ the relative position of the segment descriptor in the EXTENDED COPY parameter list. The first segment descriptor in the parameter list is ~~specified as~~ assigned descriptor number zero, the second is assigned one, etc. A given segment number may appear in more than one segment order descriptor.

The PROCESSING ORDER field is equal to the number of segment descriptors that were ~~executed~~ processed prior to the segment descriptor to which this segment order descriptor applies.

The DESTINATION BYTE OFFSET FIELD contains the offset, in bytes, into the destination logical block~~ that block beginning~~ at which the data described by this segment order descriptor~~ the segment~~ was written~~ transferred~~.

The DESTINATION LOGICAL BLOCK field contains the number of the first logical block to which~~ destination block beginning in which the~~ data described by this segment order descriptor~~ the segment~~ was written~~ transferred~~. ~~The copy manager will return the block number relative to the first block written by any segment of this EXTENDED COPY command. Thus, the first block written following the start of an EXTENDED COPY command is always written to block 0.~~

7.77.4 OPERATING PARAMETERS service action

In response to the OPERATING PARAMETERS service action, the copy manager shall return its operating parameter information in the format shown in table 45. If a device server supports the EXTENDED COPY command, then it shall also support the RECEIVE COPY RESULTS command with OPERATING PARAMETERS service action.

Table 45 — Parameter data for the OPERATING PARAMETERS service action

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			AVAILABLE DATA (n-4)				
3								(LSB)
4	(MSB)			MAXIMUM TARGET DESCRIPTOR COUNT				
5								(LSB)
6	(MSB)			MAXIMUM SEGMENT DESCRIPTOR COUNT				
7								(LSB)
8	(MSB)			MAXIMUM DESCRIPTOR LIST LENGTH				
11								(LSB)
12	(MSB)			MAXIMUM SEGMENT LENGTH				
15								(LSB)
16	(MSB)			MAXIMUM INLINE DATA LENGTH				
19								(LSB)
20	(MSB)			HELD DATA LIMIT				
23								(LSB)
24	(MSB)			~~maximum inbound inline data length (non-buffered)~~				
27								(LSB)
28	MAXIMUM CONCURRENT COPIES							
29	DATA SEGMENT GRANULARITY (log 2)							
30	INLINE DATA GRANULARITY (log 2)							
31	HELD DATA GRANULARITY (log 2)							
32	Reserved							
33	Reserved							
34	Reserved							
35	IMPLEMENTED DESCRIPTOR LIST LENGTH (n-36)							
36	List of implemented descriptor type codes (ordered)							
n								

The AVAILABLE DATA field shall contain the length of the total length of the parameter data minus 4.

The MAXIMUM TARGET COUNT field contains ~~is~~ the maximum number of target descriptors that the copy manager allows in a single EXTENDED COPY target descriptor list.

The MAXIMUM SEGMENT COUNT field contains ~~is~~ the maximum number of segment descriptors that the copy manager allows in a single EXTENDED COPY segment descriptor list.

The MAXIMUM DESCRIPTOR LIST LENGTH field contains ~~is~~ the maximum length, in bytes, of the target descriptor list and segment descriptor list. This length includes the embedded data but excludes inline data that follows the parameter list.

The MAXIMUM SEGMENT LENGTH field indicates the length, in bytes, of the largest amount of data that the copy manager supports writing via ~~can be specified in~~ a single segment. Bytes introduced as a result of the PAD bit being set to 1 (see 7.1.5) are not counted towards this limit. A value of zero indicates that the copy manager places no limits on the amount of data written by a single segment.

The MAXIMUM INLINE DATA LENGTH field indicates the length, in bytes, of the largest amount of inline data that copy manager supports in ~~can be included after~~ the EXTENDED COPY parameter list. This does not include data included as embedded data within the segment descriptors. The MAXIMUM INLINE DATA LENGTH field applies only to ~~the 04h~~ segment descriptors containing the 04h descriptor type code (see 7.1.5.5). The field shall be set to zero when the 04h ~~segment~~ descriptor type code is not supported by the copy manager.

The HELD DATA LIMIT field indicates the length, in bytes, of the largest amount of data the copy manager is capable of holding for return to the application client via the RECEIVE COPY RESULTS command with RECEIVE DATA service action (see 7.77.2). ~~to be buffered by the copy manager and returned to the host.~~

~~The Maximum Inbound Inline Data Length (non-Buffered Implementation) indicates the length, in bytes, of the largest amount of inline data to be returned to a host but not buffered by the copy manager; this implies the ability to have an outstanding RECEIVE COPY RESULTS concurrent with the EXTENDED COPY command.~~

The MAXIMUM CONCURRENT COPIES field contains ~~is~~ the maximum number of EXTENDED COPY commands supported for concurrent processing by the copy manager ~~CDBs that the copy manager can be executing at a single instant of time~~.

The DATA SEGMENT GRANULARITY field indicates the length of the smallest data block that copy manager permits in a non-inline segment descriptor (segment descriptors with type codes other than 04h) ~~can describe~~. The amount of data transferred by a single segment descriptor ~~must~~ shall be a multiple of the granularity. The DATA SEGMENT GRANULARITY value is expressed as a power of 2. Bytes introduced as a result of the PAD bit being set to 1 (see 7.1.5) are not counted towards the data length granularity.

The INLINE DATA GRANULARITY field indicates the length of the of the smallest block of inline data that the copy manager permits being written by a segment descriptor containing the 04h descriptor type code (see 7.1.5.5) ~~can be transferred from the host to the copy manager~~. The amount of data transferred by a single segment descriptor ~~must~~ shall be a multiple of the granularity. The INLINE DATA GRANULARITY value is expressed as a power of 2. Bytes introduced as a result of the PAD bit being set to 1 (see 7.1.5) are not counted towards the length granularity.

The HELD DATA GRANULARITY field indicates the length of the smallest block of held data that the copy manager shall transfer to the application client in response to a RECEIVE COPY RESULTS command with RECEIVE DATA service action (see 7.77.2) ~~can be transferred to the host from the copy manager~~. The amount of data held by the copy manager in response to an one segment descriptor shall be a multiple of this granularity. ~~The amount of data transferred by a single segment descriptor must be a multiple of the granularity.~~ The HELD DATA GRANULARITY value is expressed as a power of 2.

The IMPLEMENTED DESCRIPTOR LIST LENGTH field contains ~~is~~ the length, in bytes, of the List of implemented descriptor type codes ~~Descriptor Length field~~.

The List of implemented descriptor type codes contains one byte for each segment or target DESCRIPTOR TYPE CODE value (see 7.1.3) supported by the copy manager, with a unique supported DESCRIPTOR TYPE CODE value in each byte. The DESCRIPTOR TYPE CODE values shall appear in the list in ascending numerical order.

~~The Descriptor List is a list of one byte descriptor types that are supported by the copy manager. The descriptor list shall be ordered in ascending order.~~

7.20.1 Sense-key specific *changes*

The following text should be added at the end of 7.20.1.

If the sense key is COPY ABORTED and the SKSV bit is one, the SENSE-KEY SPECIFIC field shall be as shown in table 46.

Table 46 — Segment pointer bytes

Bit / Byte	7	6	5	4	3	2	1	0
15	SKSV	Reserved	SD	Reserved	BPV	BIT POINTER		
16	(MSB)	FIELD POINTER						
17								(LSB)

The segment descriptor (SD) bit indicates whether ~~is used to indicate if~~ the field pointer is with reference to the start of the parameter list or to the start of a segment descriptor. An SD value of zero indicates that the field pointer is relative to the start of the parameter list. An SD value of one indicates that the field pointer is relative to the start of the segment descriptor indicated by the third and fourth bytes of the COMMAND SPECIFIC INFORMATION field (see 7.1.2).

A bit pointer valid (BPV) bit of zero indicates that the value in the BIT POINTER field is not valid. A BPV bit of one indicates that the BIT POINTER field specifies which bit of the byte designated by the FIELD POINTER field is in error. When a multiple-bit field is in error, the BIT POINTER field shall point to the most-significant (left-most) bit of the field.

The FIELD POINTER field indicates which byte of ~~is used to indicate where in~~ the parameter list or segment descriptor was in error ~~that an error occurred. When the sd bit is zero, then the field pointer field points to the first byte in error of the parameter list~~

NOTE 5 If the parameter list is in excess of 65528 bytes in length and SD is 0, the FIELD POINTER value may not fit in two bytes provided by the sense key specific format definition. ~~that the Field Pointer can not point to the byte in error due to the eight bytes required for the parameter list header.~~