TO: Distribution

FROM: Frank Canali
Gary Dixon
Ross Klinger
Janice Phillipps

DATE: June 10, 1974

SUBJECT: Implementation of the ANSI Standard Tape I/O Module

This MTB outlines the basic features and restrictions of the ANSI Standard Tape I/O Module.

The implementation plan listed below shows which features the I/O module will support in each of its versions. After any given version of the I/O module is completed, we may install the I/O module to make the added features of that version available to users, or we may wait until later versions are completed before we install the module. The decision to install a particular version will be based on the viability of that version of the I/O module, and the expected completion schedule for the versions which follow.

Appendix A of this MTB is a glossary of terms used in the implementation plan. Appendix B describes how the file name and file number attachment options interact in each version of the I/O module. Appendix C defines the ASCII-to-EBCDIC conversion which the I/O module performs when the file character code is EBCDIC. Appendix D defines the general error recovery strategy for the hardware errors which can arise while processing a tape.

The ANSI Standard Tape I/O Module is described in more detail in the memo, "Proposed Multics ANSI Tape DIM, Revision II", written on June 20, 1973. Copies of this memo are available to interested parties from Gary Dixon, 39-584, K3-3224.

Please direct any comments on this MTB to Gary Dixon, at the above address. Mail comments to GDixon.PDO on the MIT Multics.

VERSION ONE

Features Supported By This Version

1. I/O Switch Interfaces:
   a) ios_ interfaces: attach, read, write, detach
   b) attachment options:
      mode - read or write
      volume serial number - 6 characters
      file name - 1 to 17 characters (1)
      file number (1)
      not-yet-labelled (2)
   c) detachment disposals:
      leave, reread, rewind, or unload

2. tape labels:
   a) standard:
      American National Standard Magnetic Tape Labels for Information Interchange, ANS X3.27-1969.
   b) tape organizations:
      single-file volumes
      multi-file volumes
   c) labels read:
      VOL1 HDR1 HDR2 EOF1 EOF2 EOV1 EOV2 (3)
   d) labels written
      VOL1 HDR1 HDR2 EOF1 EOF2
   e) labels skipped on input:
      UVL1-UVL9 HDR3-HDR9 EOF3-EOF9 EOV3-EOV9
      UHLa UTLa
   f) label character code: ASCII (4)
   g) label I/O technique: synchronous I/O
   h) label error recovery strategy:
      input: backspace-block/reread 10 times
      output: backspace-block/rewrite 10 times

---

(1) The interaction between the file name and the file number is explained in detail for each version of the I/O module in Appendix B of this MTB.

(2) This attachment option is valid only when writing a file.

(3) Since this version of the I/O module does not support multi-volume files, EOV1 and EOV2 labels are treated as EOF1 and EOF2 labels to allow the file sections of a multi-volume file to be treated as separate files. Multi-volume files will be fully supported in a future version of the I/O module.

(4) 9-bit Multics bytes, each containing an ASCII character, are converted to 8-bit tape frames by ignoring the leftmost bit of each Multics byte.

3. tape files:
   a) standard:
      Magnetic Tape Labels and File Structure for Information Interchange, ANS X3L5/365T-09/27/73 (a draft proposed revision of ANS X3.27-1969).
   b) record format: U (undefined)
   c) maximum block size: 2048 characters (5)
   d) record length limits:
      U-format:
         record_length = block_size - buffer_offset (6)
   e) encoding technique: ASCII character code (4)
   f) I/O technique: synchronous I/O
   g) ANSI block prefixes:
      input: skipped
      output: not supported
   h) error recovery strategy:
      input: backspace-block/reread 10 times
      output: backspace-block/rewrite 10 times

4. recording technique: 9-track, 800-bpi density

5. access control:
   No access control facilities are provided by the I/O module, either on a per volume or on a per file basis. Access control will be provided for tape volumes by the Tape Mount Package, when it is installed and used in a future version of the ANSI Tape I/O Module.

6. tape mounting and file positioning:
   At attachment time, the requested tape volume is mounted, if not already mounted from a previous attachment in the current process. The volume label is checked when the tape is first mounted to insure that the operator has mounted the proper tape. The tape is then positioned to the requested file.

---

(5) A block of 2048 characters is the largest block which ANSI allows for tape interchange. Therefore, Version One always assumes a block size of 2048 characters.

(6) Note that ANSI allows every block to begin with a user-specified block prefix, the contents of which is independent of the data recorded in the block. In ANSI terminology, the size of this block prefix is called the buffer_offset. Therefore, the block size is related to the record length as shown above. Version One of the ANSI I/O module does not support the generation of block prefixes when writing a tape file, but it does support skipping of any block prefixes which may be present when reading a tape file.

VERSION TWO

Features Supported by This Version

1. I/O Switch Interfaces:
   a) ios_ interfaces: attach, read, write, detach
   b) attachment options:
      mode - read or write
      volume serial number - 6 characters
      file name - 0 to 17 characters (7)
      file number (7)
      record format - F, D, S, or U (8)
      blocked records (8) (9)
      record length (8)
      block size (8)
      file character code (8)
      file generation number (8)
      file version number (8)
      file expiration date (8) (10)
      not-yet-labelled (8) (11)
      density - 800 or 1600 () (12)
   c) detachment disposals: same as Version One

2. tape labels:
   a) standard: same as Version One.
   b) tape organizations: same as Version One
   c) labels read: same as Version One

---

(7) The interaction between the file name and the file number is explained in detail for each version of the I/O module in Appendix B of this MTB.

(8) This attachment option is valid only when writing a file.

(9) This option may not be used with U-format records.

(10) This version of the I/O module does not use the file expiration date to prevent rewriting of a file which has not expired. It merely allows the user to fill in the file expiration date field of the label to facilitate interchange of ANSI tapes between operating systems. A future version of the I/O module will honor the file expiration date.

(11) If this option is not specified, the tape is assumed to have an ANSI Standard volume label (VOL1), and at least one Beginning-of-File-Section header label (HDR1). These labels must be present to mount an ANSI tape, or else the not-yet-labelled attachment option must be used.

(12) If this option is not specified, a recording technique of 9-track, 800-bpi is assumed.

d) labels written: same as Version One
e) labels skipped on input: same as Version One
f) label character code: same as Version One
g) label I/O technique: synchronous I/O (13)
h) label error recovery strategy: (14)
input: backspace-block/reread 10 times
output: backspace-block/erase/rewrite 10 times

3. tape files:
a) standard: same as Version One.
b) record formats:
F (fixed-length; blocked or unblocked)
D (variable-length; blocked or unblocked)
S (spanned; blocked or unblocked)
U (undefined)
c) maximum block size: 8192 characters (15)
d) record length limits:
F-format, unblocked:
record_length = block_size - buffer_offset
F-format, blocked:
record_length =
x: mod(block_size - buffer_offset, x) = 0
D-format, unblocked:
record_length = block_size - buffer_offset
D-format, blocked:
record_length ≤ block_size - buffer_offset
S-format, blocked or unblocked:
record_length = x: 1 ≤ x ≤ 131071 (16)
U-format:
record_length = block_size - buffer_offset

---

(13) The error buffer and error order call entry points of tapeio_ are used to process tape labels so that output which is pending when a volume switch occurs can be preserved while the new volume is being mounted. The error buffer entry points of tapeio_ always perform synchronous I/O.

(14) The handling of errors which occur while processing labels is defined in more detail in Appendix D of this MTB.

(15) This limitation is imposed by the size of the wired-down, hardcore buffer from which TDCM writes, and into which it reads, tape blocks.

(16) S-format records may span several blocks and may, in general, have any length. The maximum record length of 131071 (2**17 - 1) characters is a constraint made by the maximum value of the nelem and nelemt arguments of ios_, which are declared as fixed bin(17).

e) encoding technique:
ASCII character code (4)
EBCDIC character code (17)
binary encoding (18) (19)
f) I/O technique: asynchronous I/O (20)
g) ANSI block prefixes:
input: skipped
output: not supported
h) error recovery strategy: (21)

input:	backspace-block/reread	10 times
output:	backspace-block/erase/rewrite	10 times

4. recording techniques:
9-track, 800-bpi density
9-track, 1600-bpi density

5. access control: same as Version One

6. tape mounting and file positioning: same as Version One

---

(17) The conversion from ASCII to EBCDIC is performed in two steps: (1) each 7-bit ASCII character (which is stored in the rightmost 7 bits of a 9-bit Multics byte) is converted to an 8-bit EBCDIC character (which is stored in the rightmost 8-bits of a Multics byte); (2) the Multics byte is then converted to an 8-bit tape frame by ignoring the leftmost bit of each byte. The ASCII/EBCDIC conversion performed in the first step is defined in Appendix C of this MTB.

(18) Consecutive 8-bit strings of the buffer are treated as 8-bit tape frames. On output, the final tape frame is padded with zero bits when the number of bits to be written is not 0 mod 8. On input, these pad bits are ignored.

(19) Binary encoding is not supported for F-format records, because the actual length of the binary data written in a record has been obscured by padding the record out to a fixed length.

(20) Input/output of data is performed by the asynchronous interfaces of the tapeio_ subroutine.

(21) The handling of errors which occuring while reading or writing data blocks is defined in more detail in Appendix D of this MTB.

VERSION THREE

Features Supported by This Version

1. I/O Switch Interfaces:
   a) iox_ interfaces:
      attach, open, read_record, write_record, read_length, position, control, close, and detach
   b) attachment options:
      volume serial number - 6 characters
      file name - 0 to 17 characters (22)
      file number - a number or END (22)
      record format - F, D, S, or U (23)
      blocked records (23) (24)
      record length (23)
      block size (23)
      file character code (23)
      file generation number (23)
      file version number (23)
      file expiration date (23) (25)
      user labels option
      unregistered option (26)

---

(22) The interaction between the file name and the file number is explained in detail for each version of the I/O module in Appendix B of this MTB.

(23) This attachment option is normally used only when writing a tape file. It may be used when reading a tape file to override the file description information which is usually obtained from the file labels.

(24) This attachment option may not be used with U-format records.

(25) The file is regarded as "expired" when the current date is equal to or later than the expiration date. When a file has expired, that file and the succeeding files of the file set may be overwritten. Otherwise, overwriting is prevented. Note that, to be effective, the expiration date of a file must be earlier than or equal to the expiration date of those files in the file set which precede it. The I/O module attempts to enforce this restriction when the file expiration date is set.

(26) This option only has meaning for tapes which have not been registered with the Multics Tape Mount Package. The label characteristics and recording density of registered tapes is obtained from the tape's Volume Descriptor Segment (VDS). This information must be supplied for unregistered tapes.

not-yet-labelled option (26) (27)
density - 800 or 1600 (26) (28)
detachment disposal - leave, reread, rewind, or unload

c) opening modes:
sequential_input
sequential_output

d) control requests:
set user label processing subroutine
set block prefix processing subroutine
get detailed error information
get file description information

e) positioning:
skip to beginning of file
skip to end of file
skip forwards or backwards of a specified number of records

2. tape labels:

a) standard: same as Version One

b) tape organization:
single-file volumes
multi-file volumes
multi-volume files
multi-file multi-volume volume sets

c) labels read:

VOL1	HDR1	HDR2	EOF1	EOF2	EOV1	EOV2
	UHLa		UTLa			

d) labels written:

VOL1	HDR1	HDR2	EOF1	EOF2	EOV1	EOV2
	UHLa		UTLa			

e) labels skipped on input:

UVL1-UVL9	HDR3-HDR9	EOF3-EOF9	EOF3-EOV9

f) label character code: same as Version One

g) label I/O technique: same as Version Two

h) label error recovery strategy: same as Version Two (14)

i) overriding file description information:
input: override the file description information which is normally obtained from the HDR2 label by using the appropriate attachment options

3. tape files:

a) standard: same as Version One

b) record formats: same as Version Two

c) maximum block size: same as Version Two

---

(27) This attachment option may only be used when writing a file.

(28) If this option is not specified, then a recording technique of 9-track, 800-bpi is assumed.

d) record length limits:
same as Version Two, except:
S-format, blocked or unblocked:
record_length = x: $1 \leq x \leq 1048576$ (29)
e) encoding technique: same as Version Two
f) I/O technique: same as Version Two
g) ANSI block prefixes:
input: contents made available to caller of I/O module
output: contents generated by caller of I/O module
h) error recovery strategy: same as Version Two (21)

4. recording technique: same as Version Two

5. access control:
Access control is provided by the Tape Mount Package, which provides one ACL per tape volume. This ACL is stored as the extended ACL on the tape volume's Volume Descriptor Segment (VDS).

6. tape mounting and file positioning:
At open time, the volume will be automatically mounted if not already mounted from a previous attachment, and will be automatically positioned to the requested file.

---

(29) The maximum record length of S-format records is 1048576 (256 * 1024 * 4) characters. This constraint is based upon the size of the largest possible buffer (a 256K segment) which can hold a record.

FEATURES NOT SUPPORTED BY ANY PLANNED VERSION

1. I/O Switch Interfaces:
   a) No plans to support opening modes, other than sequential_input and sequential_output.

2. tape labels:
   a) No plans to read or write the following optional labels:
      UVL1-UVL9 HDR3-HDR9 EOF3-EOF9 EOV3-EOV9
   b) No plans to support label encoding techniques, other than ASCII character code.
   c) No plans to support mixed character codes within labels.
   d) No plans to support labels consisting only of upper-case ASCII letters.

3. tape files:
   a) No plans to support the use of mixed character codes, or mixed character and binary encoding, within a single file.
   b) No plans to support BCD character code; no plans to support encoding techniques other than ASCII character code, EBCDIC character code, and binary encoding.

4. recording techniques:
   No plans to support 7-track tapes; no plans to support 9-track tapes at densities other than 800- and 1600-bpi

5. access control:
   a) No plans to support tape volume access control through the accessibility field of the VOL1 volume label.
   b) No plans to support tape file access control through the accessibility field of the file labels.

APPENDIX A
GLOSSARY OF TAPE TERMS (1)

Definitions in the first group below fall in to one of two categories: logical structures or physical structures. Definitions in the second group are uncategorized.

GROUP ONE - logical or physical structures

record (logical)
Related data treated as a unit of information.

Note 1: The delineation of a record may be arbitrary and is determined by the designed of the information format.
Note 2: A record may be recorded in all or part of a block, or in more than one block.

block (physical)
A collection of characters written or read as a unit.

Note 1: A block may contain one or more complete records.
Note 2: A block may contain segments of one or more spanned records. A single block does not contain multiple segments of the same spanned record.
Note 3: For the purpose of this standard, blocks are separated by an inter-block gap.

block prefix (logical)
An optional, fixed-length field at the beginning of each block. The contents of block prefix is specified by the user and is independent of the record(s) in the block.

Note 1: The block prefix might contain a checksum of the record(s) in the block, might contain a block sequence number, or other error correction information. The length of this block prefix is known as the buffer offset.

file (logical)
A collection of information cosisting of records pertaining to a single subject.

Note 1: The description, content, or organization of a file may be arbitrary.
Note 2: A file may be recorded on all or part of a volume, or on more than one volume.

---

(1) These definitions are taken from: Magnatic Tape Labels and File Structure for Information Interchange, ANS X3L5/365T-09/27/73 (a draft proposed revision of ANS X3.27-1969).

volume (physical)
A dismountable physical unit of storage media (i.e., a reel of magnetic tape).

Note 1: A volume may contain part of a file, a complete file, or more than one file.
Note 2: A volume may contain sections of one or more files. A volume does not contain multiple sections of the same file.

file section (logical)
That part of a file that is recorded on any one volume.

Note 1: The sections of a file do not have sections of other files interspersed.

file set (logical)
A collection of one or more related files recorded consecutively on a volume set.

Note 1: A file set may span one or more volumes.

volume set (physical)
A collection of one or more volumes on which one and only one file set is recorded.

unspanned record (logical)
A record contained in a file in which each record, by design, ends in the same block in which it begins.

spanned record (logical)
A record contained in a file in which each record may begin in one block and end in another.

Note 1: Each record consists of one or more segments, each segment being contained in a block, the blocks being written consecutively.

record segment (logical)
That part of a spanned record that is contained in any one block.

Note 1: The segments of a record do not have segments of other records interspersed.

unblocked record (logical)
A record contained in a file in which each block, by design, contains only one record or record segment.

blocked record (logical)
A record contained in a file in which each block may contain more than one record or record segment.

fixed-length record (logical)
A record contained in a file in which all the records, by design, have the same length.

variable-length record (logical)
A record contained in a file in which the records may have different lengths.

GROUP TWO - uncategorized definitions

label
A record at the beginning of a volume, or at the beginning or end of a file section, or at the end of a file, that identifies, characterizes, and/or delimits that volume of file section. A label is not considered to be part of a file.

label set
A collection of one or more contiguous labels with the same three initial characters (label identifier).

label group
A collection of one or more contiguous label sets that delimit one end of a volume, of a file section, or of a file.

tape mark
A delimiter used to indicate the boundary between file data and label groups and also between certain label groups.

double tape mark
A delimiter, consisting of two consecutive tape marks, that is used to indicate the end of a volume or of a file set.

Note: Two consecutive tape marks also occur when an empty file section or an empty file exists on a volume, in which case they are not interpreted as a double tape mark but rather as two single tape marks framing an empty file section. In this context, "empty" means that no blocks are present between the tape mark following the Beginning-of-File-Section label group and the tape mark preceding the End-of-File-Section label group of that file section or file.

APPENDIX B
INTERACTION BETWEEN THE FILE NAME AND FILE NUMBER ATTACHMENT OPTIONS

In the descriptions below, the term file name refers to the file name attachment option while the term file identifier refers to the file name stored in the HDR1 label of a file; the term file number refers to the file number attachment option while the term file sequence number refers to the file number recorded in the HDR1 label of a file, which equals the position of the file within the file set.

VERSION ONE

Both the file name and the file number must be specified for each attachment. The file name must be a non-null character string from 1 to 17 characters in length. The file number must be a non-negative integer.

The file name is used to select the file to be accessed. If the file number is not 0, then it is compared with the file sequence number of the selected file as a further check on the correctness of the attachment.

If the file number is not 0, then the file is selected as follows.

1) On input, the first file in the file set whose file identifier matches the file name is selected for reading. If the file number matches the file sequence number of this file, then the file is attached for reading. If no file is selected, or if the selected file's sequence number does not match the file number, then an error occurs and the attachment fails.

2) On output, if one or more files in the file set have a file identifier which matches the file name, then the first such file is selected for rewriting. If the file sequence number of this file matches the file number, then the file is attached for writing. Otherwise, an error occurs and the attachment fails.

   On output, if no files in the file set have a file identifier which matches the file name, then a file is appended to the file set. The file number must be $n+1$, where n is the file sequence number of the final file in the file set; otherwise, an error occurs and the attachment fails.

If the file number is 0, then the file is selected as follows.

1) On input, the first file in the file set which matches the file name is attached for reading. If no matching file is found, then an error occurs and the attachment fails.

2) On output, if one or more files in the file set have a file identifier which matches the file name, then the first such file is selected for rewriting, and is attached for writing.

   On output, if no files in the file set have a file identifier which matches the file name, then a file is appended to the file set and this new file (whose file sequence number is n+1) is attached for writing.

   In either output case, the position of the file within the file set is recorded as the file sequence number in the HDR1 label of the file.

VERSION TWO

As in Version One, both the file name and the file number must be specified for each attachment. The file name may be any character string from 0 to 9 characters in length. A file name composed of 0 characters is called a null file name. The file number must be a non-negative integer. A null file name and a zero file number cannot be used in the same attachment.

If the file name is non-null, then attachment proceeds as outlined for Version One, above.

If a null file name is given, then the file number is used to identify the tape file to be accessed.

1) On input, if the file identified by the file number exists within the file set, then it is attached for reading; otherwise, an error occurs and the attachment fails. No check is made on the file identifier of that file.

2) On output, if the file whose file sequence number equals the file number exists within the file set, then it is rewritten.

   If the requested file number does not identify an existing file within the file set, then if the file number is n+1 (where n is the file sequence number of the final file of the file set), then this new file is attached for writing. If the file number is not n+1, then an error occurs and the attachment fails.

In either output case, the position of the file within the file set is recorded as the file sequence number in its file labels.

VERSION THREE

As in Versions One and Two, both the file name and the file number must be specified for each attachment. The file name may be any character string from 0 to 9 characters. The file number must be a non-negative integer, or else it must be END. A null file name and a zero file number cannot be used in the same attachment. A file number of END may be used with a null file name, however.

If the file number is not END, then attachment proceeds as outlined for Version Two, above.

If the file number is END, then the attachment proceeds as follows.

1) On input, the final file of the file set is selected for reading. If the file name is not a null string, and the file identifier matches the file name, then the file is attached for reading; otherwise, an error occurs and the attachment fails. If the file name is a null string, then the file is attached for reading without checking the file identifier.

2) On output, if the file name is null, then a new file is appended to the end of the file set. A file identifier of "(no name)" is recorded in the file labels of this file. The position of the file within the file set is recorded as the file sequence number in the file labels.

   On output, if the file name is a non-null character string, then the file name is compared with the file identifier of the final file of the file set. If there is a match, then the final file of the file set is selected for rewriting. Otherwise, a tape file is appended to the file set. In either case, the file is attached for writing, and the position of the file within the file set is recorded as the file sequence number in the file labels.

APPENDIX C
ISOMORPHIC ASCII/EBCDIC CONVERSION TABLE

The table below defines the mapping which the ANSI Tape I/O Module performs between ASCII characters and EBCDIC characters when the EBCDIC character code is used for tape file data.

ASCII		EBCDIC	
GRAPHIC	OCTAL	HEXADECIMAL	GRAPHIC
NUL	000	00	NUL
SOH	001	01	SOH
STX	002	02	STX
ETX	003	03	ETX
EOT	004	37	EOT
ENQ	005	2D	ENQ
ACK	006	2E	ACK
BEL	007	2F	BEL
BS	010	16	BS
HT	011	05	HT
LF	012	25	NL
VT	013	0B	VT
FF	014	0C	NP
CR	015	0D	CR
SO	016	0E	SO
SI	017	0F	SI
DLE	020	10	DLE
DC1	021	11	DC1
DC2	022	12	DC2
DC3	023	13	TM
DC4	024	3C	DC4
NAK	025	3D	NAK
SYN	026	32	SYN
ETB	027	26	ETB
CAN	030	18	CAN
EM	031	19	EM
SUB	032	3F	SUB
ESC	033	27	ESC
FS	034	1C	IFS
GS	035	1D	IGS
RS	036	1E	IRS
US	037	1F	IUS

ASCII		EBCDIC	
GRAPHIC	OCTAL	HEXADECIMAL	GRAPHIC
space	040	40	space
!	041	5A	!
"	042	7F	"
#	043	7B	#
$	044	5B	$
%	045	6C	%
&	046	50	&
'	047	7D	'
(	050	4D	(
)	051	5D	)
*	052	5C	*
+	053	4E	+
,	054	6B	,
-	055	60	-
.	056	4B	.
/	057	61	/
0	060	F0	0
1	061	F1	1
2	062	F2	2
3	063	F3	3
4	064	F4	4
5	065	F5	5
6	066	F6	6
7	067	F7	7
8	070	F8	8
9	071	F9	9
:	072	7A	:
;	073	5E	;
<	074	4C	<
=	075	7E	=
>	076	6E	>
?	077	6F	?

ASCII		EBCDIC	
GRAPHIC	OCTAL	HEXADECIMAL	GRAPHIC
@	100	7C	@
A	101	C1	A
B	102	C2	B
C	103	C3	C
D	104	C4	D
E	105	C5	E
F	106	C6	F
G	107	C7	G
H	110	C8	H
I	111	C9	I
¢	112	D1	¢
K	113	D2	K
L	114	D3	L
M	115	D4	M
N	116	D5	N
O	117	D6	O
P	120	D7	P
Q	121	D8	Q
R	122	D9	R
S	123	E2	S
T	124	E3	T
U	125	E4	U
V	126	E5	V
W	127	E6	W
X	130	E7	X
Y	131	E8	Y
Z	132	E9	Z
[	133	8D	[(1)
\	134	E0	\
]	135	9D	] (1)
^	136	5F	logical NOT
_	137	6D	_

(1) These graphics do not appear in (or map into any graphics which appear in) the standard EBCDIC character set. They have been assigned to otherwise "illegal" EBCDIC code values, in conformance with the mapping defined in MPM section 5.2, Punched Card Codes.

ASCII		EBCDIC	
GRAPHIC	OCTAL	HEXADECIMAL	GRAPHIC
~	140	79	~
a	141	81	a
b	142	82	b
c	143	83	c
d	144	84	d
e	145	85	e
f	146	86	f
g	147	87	g
h	150	88	h
i	151	89	i
j	152	91	j
k	153	92	k
l	154	93	l
m	155	94	m
n	156	95	n
o	157	96	o
p	160	97	p
q	161	98	q
r	162	99	r
s	163	A2	s
t	164	A3	t
u	165	A4	u
v	166	A5	v
w	167	A6	w
x	170	A7	x
y	171	A8	y
z	172	A9	z
{	173	C0	{
|	174	6A	|
}	175	D0	}
~	176	A1	~
DEL	177	07	DEL

APPENDIX D
ERROR RECOVERY STRATEGIES WITHIN THE I/O MODULE

When the tape hardware returns an error status to the tapeio_ subroutine, tapeio_ attempts to recover from some types of errors by rereading or rewriting the block in question, or by re-executing the order request in question. If the retry fails 10 times (or if retry in inappropriate for the error), tapeio_ signals the tape_error_ condition. The ANSI I/O module's tape_error_ on unit invokes irec_error_handler_ to process the error. This handler recognizes 7 classes of errors, and handles each class as described below.

Note that several references are made in the descriptions below to trm_$error_log, an error logging entry point in the Tape Mount Package. Until the Tape Mount Package is installed, no error logging facility will be provided. Furthermore, those errors which require the error logging facility to request some error recovery operation from the operator will be treated as non-recoverable errors until trm_$error_log is available.

1) errors with major status: Command Reject

2) errors with major status: MPC Command Reject

3) errors with major status: Device Busy

   The occurrence of an error in any of the above 3 classes implies a logic error in tapeio_. The error is unrecoverable.

4) errors with major status: Device Attention

   a) minor status: Handler in Standby

      A message is sent to the operator via trm_$error_log requesting the drive to be readied. If the operator can ready the drive and reply to the message, then I/O operations are requeued; otherwise, the error is unrecoverable.

   b) any other minor status: Write Protected, No Such Handler, Handler Check, Blank Tape on Write

      These imply a hardware malfunction or configuration error. The error is unrecoverable. It is logged via trm_$error_log.

5) errors with major status: MPC Device Attention

   a) minor status: Multiple BOT

      The error implies that the user's tape is defective. The error is unrecoverable.

   b) any other minor status: Configuration Switch Error, Multiple Devices, Illegal Device I.D. Number, Incompatible Mode, TCA Malfunction, MTH Malfunction

      These imply a hardware malfunction. The error is unrecoverable. It is logged via trm_$error_log.

6) errors with major status: MPC Device Data Alert

   The error implies that the data being read is invalid, or that the tape being written is defective, assuming no hardware malfunction. tapeio_ will have already attempted error recovery; such an error is therefore unrecoverable. Currently, the I/O module will not honor further I/O requests. An option could be provided to attempt further reads, in the hope that subsequent records were readable. As tapeio_ would have already performed multiple backspace-block/erase/rewrite sequences, further write attempts would be superfluous. In addition, such errors could be recorded in a VDS error count.

7) errors with major status: Device Data Alert

   a) while reading, minor status: Blank Tape on Read

      The error implies that the file structure of the tape is invalid. The I/O module is informed that an end-of-volume while reading has occurred. The error is handled as equivalent to reading an EOF record (i.e., fully recoverable, but with the additional knowledge that trailer labels are missing).

   b) while writing, minor status: End of Tape Mark

      The I/O module is called to write an EOV trailer label set, dismount the volume, mount a new volume, and write VOL and HDR label sets. If unable to do so, the error is unrecoverable; if able, suspended operations are requeued and I/O recommenced.

c) any other minor status: Transfer Timing Alert, Bit Detected During Erase Operation, Transmission Parity Alert, Lateral Tape Parity Alert, Longitudinal Tape Parity Alert

These imply that the data being read is invalid, or that the tape being written is defective, assuming no hardware malfunction. The error is unrecoverable. For possible strategy extensions. (See (6) above.)

Note that errors are interpreted in the specific context of processing data records within a structured file. A tape error can, in general, have different meanings under different circumstances. Furthermore, a hardware malfunction can always cause an error to be erroneously indicated.