

TO: Distribution
FROM: Ross E. Klinger
DATE: 10/15/74
SUBJECT: tape_ansi_ IO Module

Attached is the draft MPM documentation of the tape_ansi_ IO Module. This documentation includes a number of features not implemented in the initial version of the module, but planned for the second (and final) version.

I. Multi-volume Files and/or File-sets

Multi-volume files and/or file-sets are not supported in the initial version. The -volume and -device keywords will therefore not appear in the initial documentation.

II. User Label Processing

User label processing is not supported in the initial version. The -user_labels keyword, as well as the read_user_labels and write_user_labels control operations, will therefore not appear in the initial documentation.

III. Retain Options

The -retain options "device" and "volume" are not supported in the initial version. "device" will be supported when the initial version of the resource control package (rcp_) becomes available. "volume" will be supported when volume assignment is implemented by rcp_. The "device" and "volume" options will therefore not appear in the initial documentation.

IV. Opening Modes

The sequential_input_output mode is not supported in the initial version, and therefore will not appear in the initial documentation.

V. The read_length and position operations are not supported in the initial version.

Most of the above features actually are implemented in the initial code, but have not yet been tested.

I/O_MODULE

Name: tape_ansi_

This I/O Module supports reading and writing files on magnetic tape, according to Draft Proposed Revision X3L5/419T of American National Standard ANSI X3.27-1969, Magnetic Tape Labels and File Structure for Information Interchange.

Entries in the module are not called directly by users; rather, the module is accessed through the I/O system. See the MPM section, the Multics I/O System, for a general description of the I/O System, and see the MPM section, File I/O, for a discussion of files.

Attach_Description

The attach description has the following form:

tape_ansi_ reelid -opt1- -optn-

1. reelid is the tape reel identifier of the first (or only) volume.

2. opt1 may be one of the following options.

-volume v1 v2 ... vD specifies, for multi-volume file sets,
-vol v1 v2 ... vD follow-on tape reel identifiers in correct switching sequence. If additional volumes are needed, and have not been specified, tape_ansi_ will query the user. Multi-volume files written by tape_ansi_ contain volume switching information in the system-defined label fields, so explicit specification of follow-on reel identifiers is unnecessary when the file is subsequently read. Any reel identifier beginning with a hyphen, "-", must be preceded

-name fileid specifies the file identifier of the file to be
-nm fileid accessed. fileid must consist of 17 or fewer characters, and only a subset of ASCII is permitted. See Notes, below. If omitted, the -number option must specify the file to be accessed. See -number, below.

-number n specifies the sequence number of the file to be
-nb n accessed. If -name is omitted, the nth file of the file set is accessed; if -name and -number

are both specified, `fileid` and `n` must both match the actual file identifier and sequence number recorded in the file labels. `n` must be an integer in the range $1 \leq n \leq 9999$.

- `-ring`
`-rg` specifies that volumes be mounted with write rings.

- `-extend`
`-ext` specifies extension of an already existing file. Extension begins following the last data record. The version number of an extended file will increase by one. See Notes, below.

- `-modify`
`-moc` specifies modification of an already existing file. Modification begins at the first data record. The version number of a modified file will increase by one. See Notes, below.

- `-write`
`-wrt` specifies writing a new generation of an already existing file. The generation of a written file increases by one; its version is zero. This is the default output option. See Notes, below.

- `-create fileid`
`-cr fileid` specifies the creation of a new file with file identifier `fileid`. `fileid` must satisfy the character set rules for file identifiers. (See Notes, below.) If neither `-name` nor `-number` are specified, the file will be created at the end of the file set; otherwise, the accessed file will be overwritten by the created file. The generation of a created file is one; its version is zero.

- `-device n`
`-dv n` specifies the maximum number of drives which may be used. `n` must be an integer in the range $1 \leq n \leq 63$. `n` cannot exceed the process's drive limit.

- `-density n`
`-den n` specifies the density at which the file is to be recorded. `n` may be either "800" or "1600" characters per inch (dpi). Every file of a file set must be recorded at the same density, and the ANSI standard requires that files be recorded at 800 dpi. The default is therefore 800 dpi.

- `-expires date`
`-exp date` specifies the expiration date of a file being written or created. `date` must be in a form acceptable to `convert_date_to_binary_`. (See the MPM write-up of `convert_date_to_binary_` for acceptable forms.) The default is to record the expiration date as "00000"; i.e., expired.

- `-force` specifies that expiration checking is not to be performed. The default will cause every file in the file set, beginning with the accessed file, to

be checked for expiration. The user will be queried whenever an unexpired file is detected.

- format form** specifies the record format. **form** may be either
- fmt form** "f" (fixed-length records), "fu" (fixed-length records, unblocked), "d" (variable-length records), "du" (variable-length records, unblocked), "s" (spanned records), "su" (spanned records, unblocked), or "u" (undefined). The default format for file creation is "d". Note that the ANSI standard does not permit "u" format for interchange purposes.

- block n** specifies the actual or maximum block length, in
- bk n** characters. **n** must be an integer in the range $1 \leq n \leq 8192$. Note that the ANSI standard permits a maximum block length of 2048; the default for file creation is therefore 2048.

- record n** specifies the actual or maximum record length, in
- rec n** characters. **n** must be an integer in the range $1 \leq n \leq 262144$. The default for file creation is **n** = block length.

- mode c** specifies the mode used to record the file data.
- md c** **c** may be either "ascii", "ebcdic", or "binary". Note that the ANSI standard permits files to be recorded only in the ASCII character code; the default for file creation is therefore "ascii".

- user_labels** specifies that user header and trailer labels are
- ul** to be processed.

- retain opt** specifies a detach-time retention option.
- ret opt** If **opt** is "none", devices and volumes will be unassigned. If "device", devices will be retained, but volumes will be unassigned. If "volume", devices will be unassigned, but volumes will remain assigned to the process. If "all", devices and volumes will be retained. The default is the rcp_ (resource control package) default.

Opening

The opening modes supported are sequential_input, sequential_output, and sequential_input_output.

Control Operation

This I/O module supports four control operations:

1. hardware_status - this operation returns a 72 bit hardware status string.
2. status - this operation returns a structure containing a detailed interpretation of the hardware status string.
3. read_user_labels - this operation provides an entry variable for the user's label-checking procedure, called when user labels are read.
4. write_user_labels - this operation provides an entry variable for the user's label-generating procedure, called when user labels are written.

Modes Operation

This I/O module does not support the modes operation.

Note

When reading a file, -format, -block, -record, and -mode may be used to either override the equivalent information in the file's header labels, or to supply such information for files not recorded with HDR2 labels.

If a file being written in -extend, -modify, or -write mode was recorded with HDR2 labels, -format, -block, -record, and -mode are ignored.

-extend, -modify, -write, and -create are mutually exclusive. Specification of either -modify, -write, or -create precludes the use of the extend bit at open time.

ANSI permits only a subset of ASCII to be used in file and volume labels. This subset consists of the numbers, upper-case letters, blank, and some special characters. Referring to MPM Section 5.1, ASCII Character Set, this subset is defined by octal codes 040 through 136, inclusive.

WARNING: files not recorded according to ANSI standards (for example, with a block length > 2048) are not suitable for interchange purposes.