TO: MSPM Distribution
FROM: J. Cecil
DATE: September 13, 1968
SUBJECT: Sections BX.99.01 and BX.99.02

These sections are being reissued to describe the changes to the calling sequences of tape_in and tape_out, and to correct the descriptions of the control files.

Published: 09/13/68
(Supersedes: BX.99.01, 05/21/68)

## Identification

Interim Facility to Write a Tape
tape_out
K. J. Martin, J. H. Cecil

## Purpose

As an interim facility, the tape_out command causes a tape to be written in CTSS disk editor (7punch) format containing segments existing in the Multics file system hierarchy. The tape is suitable 1) for input to the CTSS disk editor, and 2) for input to the Multics tape_in command (BX.99.02). The tape_out command only creates control segments and signals the tape daemon process to actually write the tape.

## Usage

tape_out reel_number seg1 seg2 seg3 ... segn

where seg1 through segn are path names (relative to the root or to the user's working directory-see BX.8.00) of segments which should be written on tape in CTSS disk editor format. The argument, reel_number, specifies the number of the tape to use. Reel_number may be scr if any scratch tape may be used.

## Assumptions

The GECOS system expects all text, link, and symbol files taken from CTSS to have headers containing the Multics segment name. In order to form these headers, the following naming assumption is made. All Multics segments with one-component names are assumed to be text segments (the string ".text" is not part of a text segment's name). Other segments will, in general, have two components; e.g., alpha.epl. Link and symbol segments have second (last) components ".link" and ".symbol". A corresponding CTSS file name is formed from the segment name as follows. The second component is truncated to six characters (if necessary) to form the second CTSS name. (A single component Multics name has CTSS second name TEXT.) A first component of six characters or less can be translated directly into the CTSS first name. Any resulting CTSS name of less than 6 characters is padded with blanks on the left. A first component of more than six characters must be treated carefully to (hopefully) avoid identical CTSS names for different segments. The algorithm used is to concatenate the first three and the last three characters of such a first component. It is thought that this is more apt to be unique than simply truncating to six characters.

## Implementation

The tape_out command calls smm$set_name_status to create a control segment in the user's working directory. The name of that segment is determined by calling unique_bits to obtain a unique 70-bit string, then calling unique_chars to convert it to a character string.

Tape_out calls find_12_dir with the working directory name (BY.99.01) to determine an appropriate CTSS directory for this user. The tape_out command then fills the control segment with appropriate control lines. For each segment in segment_list, tape_out inspects its name as described above. Tape_out forms from the CTSS file name and directory name an ascii line of the form:

```
BCD :input Prob  Prog   name1  name2
```

and places it in the control segment to indicate the CTSS name and directory where this segment should be placed. In addition, for each of the text, link and symbol segments (only, other segments are unmodified), tape_out constructs a 22-word CTSS header where the first character (9 bits) contains a character count and the rest contain the segment's Multics name (first or only component) followed by blanks.

That 22-word header is placed in the control segment as ascii data to a 7punch control line. The control line following is a 7punch control line to write out the segment.

Appropriate mount, dismount and eof control lines are included. The last control line is history with this user's mail box segment name as the argument. The tape daemon will mail a report of the tape writing to the user at its completion including any errors, operator comments and the reel number of the tape written. Reel number is crucial since it must be specified to the tape_in command.

The tape_out command creates a link in the tape daemon's working directory to the control segment. It then looks in the segment td_wakeup and signals an event over the channel whose id is found there. The channel and process ids are td_wakeup$channel and td_wakeup$process_id. The event id signalled is the same 70-bit unique bit string obtained earlier, so that the tape daemon may construct the name of the link in its working directory.

The tape_out command is finished, so it returns to command level.

Possible errors arise from smm, unique_bits, append_link and the interprocess communication facility. Any of these errors which indicate failure to perform properly (as opposed to status indications) are fatal to the command. Tape_out types a comment to the user describing the failure and returns.

Example

The command line

```
tape_out scr alpha.epl beta  beta.link
```

would generate the control segment:

```
mount     scr  w
bcd       :input Prob Prog alpha epl
7punch      alpha.epl
bcd       :        *eof*
bcd       :input Prob Prog beta text
7punch    :beta
7punch      beta
bcd       :        *eof*
bcd       :input Prob Prog beta link
7punch    :beta
7punch      beta.link
bcd       :        *eof*
bcd       :stop
bcd       :close
eof
dismount
history   personal_name.project
stop
```