

To: MTB Distribution
From: Ross E. Klinger
Gary C. Dixon
Subject: A file_copy command
Date: September 8, 1975

INTRODUCTION

The `io_call` command currently provides complete facilities at command level for performing all possible operations on a single I/O switch, with the exception of `lox_$control` requiring an information pointer. As such, it provides an extremely useful tool for setting up user program input/output switches, and for debugging programs and I/O modules.

The major facility it does not provide is a copy facility. There is a need to copy from one switch to another, either until the input switch exhausts the input source, or until a certain number of read/write operations have been performed.

A switch-to-switch copy facility would support file-to-tape operations through any of the tape I/O modules, file-to-disk operations through the disk I/O module, and special-purpose file-to-printer, card reader-to-file, and file-to-card punch operations. As such, it would be a useful tool for debugging, and for quick copy operations between various media.

INTERFACE

See the attached draft MPM documentation.

CRITICISM

The above interface can copy from or to any type of file except for keyed sequential files. One can argue that these are only supported by the storage system (`vfile_`) currently, and can be copied with the `copy` command.

An alternate proposal would use a `-keyed` control argument to signify that both the input and output files were keyed sequential. The input file would be opened for

Multics Project Internal working documentation. Not to be reproduced or distributed outside the Multics Project.

keyed_sequential_input and the output file for keyed_sequential_output. Records and their keys could then be read with read_key/read_record operations, and written with seek_key/write_record operations.

Given the limited use of keyed files and the availability of the copy command, it is probably unnecessary for file_copy to provide this level of support. Note that the interface does permit a keyed sequential input file to be read and copied in a sequential manner.

While the interface can copy unstructured files, the record_stream_ I/O module is required as an intermediary. Although the transformation function could conceivably be integrated with file_copy, the record_stream_ I/O module was specifically designed to effect this type of conversion, and there is no need to duplicate its function within the file_copy command.

file_copy

file_copy

Names: file_copy, fc

This command copies records from an input file to an output file. The input and output file types must be sequential. (See Unstructured Files and Keyed Files, below, for an explanation of how these types of files may be copied.) The input file may be copied either partially, or in its entirety.

Usage

file_copy input_spec output_spec -control_args-

where:

1. **input_spec** specifies the input file from which records are read. It may be either an I/O switch name, or an attach description. (See Notes, below.)

-input_switch switchname

-lsw switchname

specifies the input file by means of an already attached I/O switch name, switchname.

-input_file XXX

-lf XXX

specifies the input file by means of an attach description, XXX.

2. **output_spec** specifies the output file to which these records are written. It may be either an I/O switch name, or an attach description. (See Notes, below.)

-output_switch switchname

-osw switchname

specifies the output file by means of an already attached I/O switch name, switchname.

-output_file XXX

-of XXX

specifies the output file by means of an attach description, XXX.

3. **control_args** may be one or more of the following optional control arguments (see Notes, below):

-from I

-fm I

specifies that records are copied beginning with the Ith record of the input file. I must be a positive integer. The default is

file_copy

file_copy

to begin copying with the "next record".
(See Notes, below.)

- to J specifies that copying is performed until the Jth record has been copied, or the input file exhausted, whichever occurs first. If this option is specified, -from must also be specified. J must be a positive integer, greater than or equal to I.
- count N
-ct N specifies that copying is performed until N records have been copied, or the input file exhausted, whichever occurs first. N must be a positive integer.
- all, -a specifies that records are copied until the input file is exhausted. This is the default.
- brief, -bf specifies that the message indicating the number of records actually copied is to be suppressed.
- long, -lg specifies that a message indicating the number of records actually copied is to be printed. This is the default.

Unstructured Files

file_copy operates by performing record I/O on structured files. If it is desired to copy from/to an unstructured file, the record_stream_ I/O module may used, e.g.:

```
file_copy -lf record_stream_ -target vfile_ pathname -osw OUT
```

The effect is to take lines from the file specified by pathname via vfile_, transform them into records via record_stream_, and then copy them to the I/O switch named OUT.

Keyed Files

file_copy processes only the sequential type of structured file. The copy command may be used to copy indexed sequential storage system files preserving keys. (See the MPM write-up of the copy command.) However, file_copy may be used to copy an indexed sequential input file in a sequential manner (disregarding keys). Note that the output file type is

file_copy

file_copy

sequential, not indexed sequential.

Notes

If either the input or output specification is an attach description, it is used to attach an uniquely-named I/O switch to the file. The switch is opened for sequential_input or sequential_output, the copy performed, and then closed and detached. If an attach description contains any of the control arguments used by file_copy, then it must be quoted. Alternately, an I/O switch may be attached using the io_call command, and the file specified by means of the I/O switch name. (See the MPM write-up of the io_call command.)

If the input file is specified by an I/O switch name and the switch is not open, file_copy opens it for sequential_input, performs the copy, and closes it. If the switch is already open when file_copy is invoked, it is not closed after the copy has been performed; the file is left positioned immediately after the last record copied.

If the -from control argument is not used to specify an absolute starting position within the input file, copying begins with the "next record". If the I/O switch is opened by file_copy, the next record is the first record of the file; otherwise, the next record is that record at which the file is positioned when file_copy is invoked.

If the output file is specified by an I/O switch name and the switch is not open, file_copy opens it for sequential_output, performs the copy, and closes it. If the switch is already open when file_copy is invoked, it is not closed after the copy has been performed; the file is left positioned immediately after the last record copied.

The -brief and -long control arguments are mutually exclusive. The copy count message appears as follows:

```
file_copy: 345 records copied. (EOI)
```

The appearance of EOI (End Of Information) indicates that the last record copied was the last record in the input file.

The -to, -count, and -all control arguments are mutually exclusive.