

U.S. DEPARTMENT OF JUSTICE
FEDERAL BUREAU OF INVESTIGATION

FILE CONTROL ROUTINES

U.S. GOVERNMENT PRINTING OFFICE: 1964 O 3916

This manual is published by the UNIVAC Division of Sperry Rand Corporation in loose leaf format as a rapid and complete means of keeping recipients apprised of UNIVAC[®] Systems developments. The UNIVAC Division will issue updating packages, utilizing primarily a page-for-page or unit replacement technique. Such issuance will provide notification of hardware and/or software changes and refinements. The UNIVAC Division reserves the right to make such additions, corrections, and/or deletions as in the judgment of the UNIVAC Division, are required by the development of its respective Systems.

CONTENTS

CONTENTS	1 to 2
1. INTRODUCTION	1-1 to 1-1
1.1. SCOPE	1-1
1.2. GENERAL DESCRIPTION	1-1
2. TAPE INPUT/OUTPUT CONTROL ROUTINES	2-1 to 2-21
2.1. FILE1 CALL	2-1
2.2. TAPE1 CALL	2-3
2.3. MACRO INSTRUCTIONS	2-5
2.3.1. Record Handling Macro Instructions	2-6
2.3.2. Input Macro Instructions	2-9
2.3.3. Output Macro Instructions	2-12
2.3.4. Establish Rerun Point	2-15
2.4. TAPE1 (OPR) STOPS AND PROCEDURES	2-16
2.5. TAPE1 (OPR) RERUN	2-18
2.5.1. General	2-18
2.5.2. TAPE1 Rerun Utility Program RRN1	2-19
3. TAPE2 INPUT CONTROL ROUTINE	3-1 to 3-5
3.1. TAPE2 CALL	3-1
3.2. TAPE2 MACRO INSTRUCTIONS	3-2
3.2.1. Open File	3-3
3.2.2. Record Advance	3-3
3.2.3. Close File	3-4
3.3. TAPE2 (OPR) STOPS AND PROCEDURES	3-4
4. TAPE3 OUTPUT CONTROL ROUTINE	4-1 to 4-5
4.1. TAPE3 CALL	4-1
4.2. TAPE3 MACRO INSTRUCTIONS	4-3
4.2.1. Open File	4-3
4.2.2. Record Advance	4-3
4.2.3. Close File	4-4
4.2.4. Close Reel	4-5
4.3. TAPE3 (OPR) STOPS AND PROCEDURES	4-5

APPENDICES

- | | |
|--------------------------------------|------------|
| A. TAPE1 SPECIAL INPUT ERROR ROUTINE | A-1 to A-2 |
| B. FILE1 PACKET (OPR) | B-1 to B-1 |

TABLES

- | | |
|---|------|
| 1. TAPE1 (OPR) Stop Codes | 2-17 |
| 2. TAPE1 Rerun Bypass-Sentinel Block Layout | 2-19 |
| 3. TAPE2 (OPR) Stop Codes | 3-5 |

1. INTRODUCTION

1.1. SCOPE

This document provides the programmer with the information necessary to use the TAPE1, TAPE2, and TAPE3 tape input/output control routines in the environment of the OPR Executive Routine. It describes the I/O routine call statement, the file description call, and the macro instructions.

Two different types of item transfer macro instructions are described: the arithmetic form, and the transfer form. The format, entrance requirements, and exit conditions are given for each macro instruction. Stop displays and related operator procedures are given at the end of each section.

Rerun is provided with the TAPE1 Routine through the use of the rerun macro instruction. The rerun routine RRN1 is described in Section 2.5. along with the operating instructions for rerun.

1.2. GENERAL DESCRIPTION

There are three tape input/output control routines: TAPE1, TAPE2, and TAPE3. TAPE1 is a general purpose tape I/O control routine that can be used to control any number of input and output files. TAPE2 can be used to control a single input file and TAPE3 can be used to control a single output file. The TAPE2 and TAPE3 control routines require less storage and execution time and should be used for programs, or program segments, that process a single tape file (for example, a card-to-tape, tape-to-print, or the first and last passes of a sort program).

All three routines may be present in the same program if desired.

2. TAPE INPUT/OUTPUT CONTROL ROUTINES

Three program components are involved in the use of the TAPE1 routine: a FILE1 call for each file to be controlled, the TAPE1 input/output control routine call, and the macro instructions.

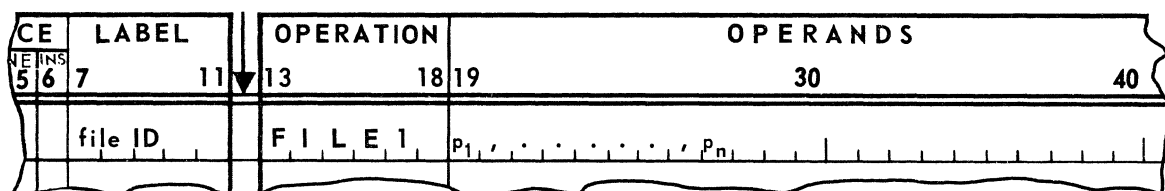
The FILE1 calls describe the files to be controlled, and direct the assembler to generate a number of constants and storage work areas for use by the TAPE1 Routine. The FILE1 call contains such information as label, input/output method to be applied, number of tape units, block size, record size, and file area(s).

The TAPE1 call directs the assembler to incorporate tape input/output coding into the program. The configuration of the incorporated coding depends upon the input/output requirements of the worker program. These requirements are specified by the values of parameters in the TAPE1 call statement. In general, the coding generated will perform the following functions: check or produce label blocks, advance records, determine when blocks should be read or written, maintain block counts, tape unit swap, submit the necessary external function (XF) instructions to the tape-handler portion of the Executive Routine, and relinquish control to the Executive Routine when necessary.

The macro instructions furnish linkages with the subroutines of the TAPE1 coding. They are used by the worker program to initiate and terminate the processing of files, to establish rerun points, and to obtain and release individual data records.

2.1. FILE1 CALL

A FILE1 call is required for each file to be controlled by the TAPE1 Routine.



The label field contains a unique 1- to 5-character label which will be used to identify the file in the macro instructions.

The operands field may contain up to 16 parameters; if necessary, it may be continued on the next line by inserting a hyphen in column 13.

- p_1 File type: IN for input, or OUT for output.
- p_2 Tape label: 13 or fewer characters bounded by apostrophes.
- p_3 Reel number base: three decimal digits bounded by apostrophes. This value plus decimal 1 will be the reel number of the first reel.
- p_4 Tape type and recording density ¹: A, for UNISERVO IIIA tapes; B, for compatible tapes at 200 PPI; C, for compatible tapes at 556 PPI; or D, for compatible tapes at 800 PPI.
- p_5 Channel: normally 4 or 5 for an input file, or 5 for an output file.
- p_6 First tape unit number.
- p_7 Second tape unit number, if tape unit swap for alternate reels is desired; otherwise, this parameter is equal to p_6 .
- p_8 Label of an area large enough to contain one block of the file. (Same size as p_{15}). An AREA directive for this area must appear in the worker program.
- p_9 If the standby method is to be applied to the file, this parameter specifies a second area large enough to contain one block of the file; otherwise, this parameter is blank.
- p_{10} Label of a closed subroutine to be executed in addition to the standard label processing. For an input file, the subroutine is executed after the label block is read, but before it is checked. For an output file, the subroutine is executed after the label block is assembled in an output area, but before it is written onto tape. For both input and output files, the four least significant characters (LSC) of AR1 contain the absolute address of the label block when the subroutine is entered.

If there is no label subroutine to be executed for the file, this parameter is blank.

- p_{11} For an input file, this parameter specifies a label in the worker program to which control will be transferred when an end-of-file block is read (subject to the sentinel option described under p_{10} of the TAPE1 call).

For an output file, this parameter determines what the TAPE1 coding will do if an end-of-tape condition is detected while a PUT1 macro instruction is being executed.

If this parameter is blank, the TAPE1 coding will close the current reel and open the next, returning control from the PUT1 in the normal fashion. (Refer to Section 2.3.3.5, Exit Conditions in Close Output Reel, which constitutes a detailed description of the end-of-tape actions performed by the TAPE1 coding.)

If this parameter is not blank, the TAPE1 coding will transfer control to the specified label.² The worker program may then perform any desired end-of-reel processing, such as putting out summary records or hash totals or establishing a rerun point. This processing must be followed by a close reel macro instruction, after which normal processing may be resumed.

¹ Translation mode is not specified for compatible tapes because they are always read or written in the binary mode.

² This transfer of control will take place only once per reel.

If there is more than one PUT1 macro instruction that addresses the file, control may be returned to the proper point by a jump to the exit line of the appropriate PUT1 subroutine. If the arithmetic register form of the macro instruction is used, the exit line is labeled XTP60; if the transfer form is used, the exit line is labeled XTP62.

It should be noted, however, that the execution of one or more PUT1 macro instructions in the end-of-reel processing will alter the exit line. In this case, the worker program must save XTP60+1 through XTP60+3, or XTP62+1 through XTP62+3, before the macro instructions are executed.

p_{12} Form of the macro instructions: AR, if the arithmetic register form will be used; TRF, if the transfer form will be used.

p_{13} Record size. If p_{12} is TRF, this parameter cannot be greater than 1024.

p_{14} Number of fill characters (refer to p_{15}), or 0.

p_{15} Physical block size, which equals: p_{13} , times the number of records per block, plus p_{14} , plus 6.

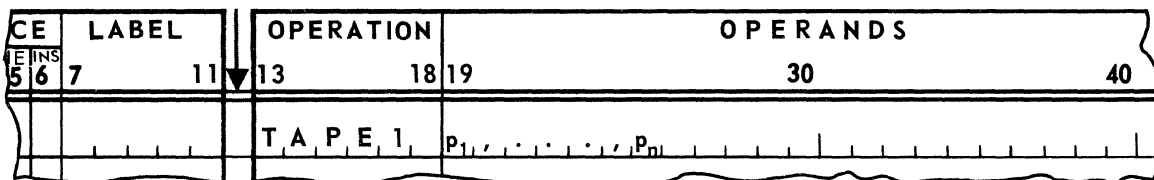
For UNISERVO IIIA tapes, this parameter must be a multiple of 4.

p_{16} (Input only) Label of a closed subroutine that is to be executed whenever a label, block count error, or block type error is detected. At the time the subroutine is executed, the address of the appropriate FILE1 packet is in the primary control index register (XTCX1), and an error flag is set in the most significant character of AR1. (Refer to Appendix A.)

If the file is an output, or if no error subroutine is to be executed for the file, this parameter is omitted.

2.2. TAPE1 CALL

The TAPE1 call statement immediately follows the FILE1 calls. Its parameters describe the tape input/output requirements of the worker program. These parameters are used to generate the input/output subroutines that perform the required file and record handling functions.



The label field must be blank.

The operands field is as follows:

p₁ Number of input files to be controlled by TAPE1. This parameter is 0 or blank if there are no input files.

p₂ Input method to be applied: STDBY, DMND, or BOTH. In the standby method, a file has 2 input areas; therefore, the next block can be read from tape while the current block, in the other area, is being processed.

In the demand method, a file has only 1 input area; therefore, a block can be read only when the previous block has been completely processed. Parameter 9 of the FILE1 call determines which method is to be applied to a particular file.

This parameter is blank if there are no input files.

p₃ Form of the input record advance macro instructions that will be used: AR, TRF, or BOTH. (Refer to Sections 2.3.1.1 and 2.3.1.2 for a description of these forms.) Parameter 12 of the FILE1 call, and the macro instructions actually used, determine which form applies to a particular file.

This parameter is blank if there are no input files.

p₄ Number of output files to be controlled by TAPE1. This parameter is 0 or blank if there are no output files.

p₅ Output method to be applied: STDBY, DMND, or BOTH. In the standby method, a file has 2 output areas; therefore, a block can be written from one area while the next block is being processed in the other area. In the demand method, a file has only 1 input area; therefore, a block cannot be processed until the write order for the previous block has been completed. Parameter 9 of the FILE1 call determines which method is to be applied to a particular file. This parameter is blank if there are no output files.

p₆ Form of the output record advance macro instructions that will be used: AR, TRF, or BOTH. (Refer to Sections 2.3.1.1 and 2.3.1.2 for a description of these forms.) Parameter 12 of the FILE1 call, and the macro instructions actually used determine which form applies to a particular file.

p₇ Primary control index register: 1 through 7. The index register designated will be used for communication between the TAPE1 coding and the macro instructions. It may be used by the worker program for other purposes, but its contents will be altered whenever a macro instruction is executed.

p₈ Secondary control index register: 1 through 7. The index register designated will be used by the TAPE1 coding. It may be used by the worker program for other purposes, but its contents will be altered whenever a macro instruction is executed.

The allocation of this index register to the TAPE1 coding will result in a saving of storage locations and execution time, but may be omitted if the user wishes. In this case, parameter 8 is 0 or blank.

p₉ If rerun is to be allowed for, this parameter designates the output file on which rerun points are to be established; otherwise, it is blank. Rerun points are established by use of the RERN1 macro instruction.

p₁₀ Sentinel option: OPSEN if the sentinel option is desired, otherwise blank. If the sentinel option is chosen, the program will stop whenever an end-of-file or end-of-reel sentinel is read. By a trace switch setting, or typein, the operator will direct the TAPE1 coding to perform either end-of-file or end-of-reel processing for the file being read.

p₁₁ RT, if the worker program is servicing real-time devices, blank otherwise. If RT is specified, the tape XF instructions associated with the program will be flagged with a priority bit when they are delivered to the tape handler. This priority bit will give them precedence over the tape XF instructions of concurrently running programs.

2.3. MACRO INSTRUCTIONS

The worker program communicates with the TAPE1 coding by means of the following macro instructions.

The OPEN1 file macro instruction initiates processing of a file. It checks or writes a label block and sets the constants and working storages, pertaining to the file, to their initial conditions. Also, except in the case of an output file using the transfer form of the record advance, the OPEN1 macro presents the first data record, or its address, to the worker program. The open file macro instruction may be executed when

- no previous instructions have been executed for the file, or
- the file has been previously opened and was automatically closed at the end of the file, or
- the file has been previously opened and was closed by means of a close file macro instruction.

The GET1 macro instruction makes the next record available to the worker program by: transferring the record to a specified location; or giving the worker program the address of the record. (The two different record handling methods are described in the following section.) Another block of records is automatically read from this file whenever all the records in the input file area have been processed.

The PUT1 macro instruction either provides the address of the area in the output file for the next record, or transfers the record to the output file area. (Again, the two different record handling methods are described in the following section.) Records are collected in the output file area until enough have accumulated to make a full output block. A block is then automatically written onto tape.

The CLOS1 file macro instruction terminates the processing of a file. (In the case of an output file, it writes any remaining records onto tape, together with two end-of-file blocks.) It rewinds the tape, if directed to do so, with or without interlock. Control then returns to the worker program when the end-of-file processing has been successfully completed.

The CLOS1 reel macro instruction initiates, at the worker program's request, the actions that are usually performed automatically at the end of a reel. It is used when the worker program wishes to close a reel prior to the detection of an end-of-tape condition (output)¹, or an end-of-reel sentinel (input). The close reel macro instruction terminates the processing of the current reel, and initiates the processing of the next reel.

¹ Refer to parameter 11 of the FILE1 call for an exception to this.

In the case of an output file, the CLOS1 reel macro instruction writes any remaining records onto tape, together with two end-of-reel blocks. The tape is rewound, if it has been so directed, with or without interlock, and a label block is written on the next reel.

In the case of an input file, the CLOS1 reel macro instruction simply rewinds the reel and checks the label block of the next reel. In all cases, except that of an output file using the transfer form of the record advance, the macro instruction presents the first data record, or its address, to the worker program.

Beginning in Section 2.3.2, the format, entrance requirements, and exit conditions are given for each macro instruction.

In addition to the specific exit conditions given for each macro instruction, note that all macro instructions alter the contents of arithmetic registers 1 and 2, the primary control index register, and the secondary control index register, if any.

An entry in the label field of a macro instruction applies to the first instruction generated by the macro instruction.

2.3.1. Record Handling Macro Instructions

2.3.1.1. The Arithmetic Register Form

The arithmetic register form of the record handling macro instruction places the absolute address of the first character of the next record (or record area, if it is an output file) in arithmetic register one.

The worker program must then place the absolute address in an index register. All subsequent references to this record use the index register and are, therefore, record relative (that is, the first character of the record is addressed as 0, the second 1 and so on). For example, an instruction bringing the tenth through the fifteenth characters of the record to AR1 could be written BA1 14, 6, x, where x is the index register that contains the record's absolute base address.

If the programmer wants to refer to a current record and its constituent fields with labels, he may do so by defining the structure of the record with a "dummy AREA statement." The AREA statement and its associated field definitions are written in the normal fashion.¹ However, the location counter is set to zero so that the AREA statement will be zero relative.² The AREA statement's fourth parameter specifies the index register assigned to hold the record base address. After the AREA statement, the location counter is reset to its previous value. The following example shows how an AREA statement might be used to define the structure of a record.

¹ Refer to the UNIVAC 1050 Card System Assembler Manual UP-3915-1.01, Section 2.4.

² Refer to the UNIVAC 1050 Central Processor Reference Manual UP-3912, Rev. 1, Section 2.2.

SEQUENCE				LABEL		OPERATION			OPERANDS									
3	4	5	6	7	11	13	18	19	30									
1				label-a		E Q U		\$										
2						O R I G		0										
3				label-b		A R E A		record size, type,, index register										
4				F L D 1		-		6 , 1 5										

j				field-n		-		length, address										
j+1						O R I G		label-a										

Line 1 establishes a reference point to which the location counter is reset after the record has been defined. Line 2 sets the location counter to zero so the record and its field addresses will be zero relative. Lines 3 through j define the record and its constituent fields. (Note that the AREA statement cannot specify a fill character.) Line j+1 resets the location counter to the value it contained prior to the record definition.

Using the dummy AREA statement to define a record, the tenth through the fifteenth characters of a record can be brought to AR1 by the instruction BA1 FLD1. The same operation, without the dummy AREA statement, is achieved with the instruction BA1 14, 6, x, where x is the index register that will contain the record base address.

2.3.1.2. Transfer Form

The transfer form of the tape I/O macro instruction transfers an input record from the input area to a designated work storage output area; an output record is transferred from a specified work storage area or input area to the output file area.

If a record is transferred to, or from, a working storage area, it and its constituent fields can be defined by a normal AREA statement.

The area that a record is transferred from may be associated with an input file using the arithmetic register macro instructions. In that case output-transfer form macro instructions could address the input area by referencing a label assigned in a dummy AREA statement. For example, a record common to both input file A and output file B using index register 3 might be labeled RCDAB. It could then be processed as follows:

LABEL	OPERATION	OPERANDS	CO
7	11	13 18 19	30 40 45 46
	OPEN	A	Checks A label. Places record address in AR1.
	BT	AR1, X3	Places first record address in index register 3.
	OPEN1	B	Writes B label.
		Process the record. The record and component fields are referenced through labels assigned in a dummy AREA statement. The fourth parameter of the dummy AREA statement specifies index register 3.	
	PUT1	RCDAB, B	Transfers record from A area to B area.
	GET1	A	Places next record address in AR1.
	BT	AR1, X3	Places next record address in index register 3.
		Return to record processing. At the end of file A, close file B.	
	CLOS1	B, RWD	

A record can be processed in the output area instead of the input area. If that is to be the case, the tape input macro instructions would be of the transfer form, the tape output macro instructions would be of the arithmetic register form, and the records, while in the output area, would be accessed through labels assigned in a dummy AREA statement.

The input macro instruction could have been of the transfer form. In that case the record would be transferred to a working storage area and the record, after it had been processed, would then be moved to the output file area with another macro instruction of the transfer form

CE	LABEL	OPERATION	OPERANDS
5 6	7	11	13 18 19
			30 40
		PUT1	ws, B

where ws is a label assigned to the first character of the working storage area.

It should be noted that the AREA statements described in the preceding paragraphs pertain only to the records, and should be distinguished from the AREA statement or statements that allocate the input or output area(s) required by a file. (Refer to parameters 8 and 9 of the FILE1 call.)

2.3.2. Input Macro Instructions

2.3.2.1. Open Input File

CE	LABEL	OPERATION	OPERANDS
5 6	7	11	13 18 19 30 40
	label	OPEN 1	file ID
	label	OPEN 1	file ID, destination

■ Format

Destination is specified if the transfer form of the macro instructions were called for in the FILE1 line. It is either the label of a working storage area or the label of an output record.

■ Entrance Requirements

This is the first macro instruction executed for the file, or the file has been closed.

If destination is the label of a record, the output file concerned is open and the index register assigned to destination contains the address of the first character of the current record area.

■ Exit Conditions

The label block has been read from the tape unit specified by parameter 6 of the FILE1 call and has been checked.

If specified in parameter 10 of the FILE1 call, a special label subroutine has been executed.

If the AR form is used, the four least significant characters of AR1 contain the absolute address of the first record.

If the transfer form is used, the first record has been transferred to the area specified by the label destination.

2.3.2.2. Input Record Advance (Arithmetic Register Form)

CE	LABEL	OPERATION	OPERANDS
5 6	7	11	13 18 19 30 40
	label	GET 1	file ID

■ Entrance Requirements

The file is open.

■ Exit Conditions

The absolute address of the first character of the next record is in the four least significant characters of arithmetic register 1.

If an end-of-reel was detected, the current reel has been rewound with interlock and the next reel opened. The label has been checked and the worker program's special label processing, if any, has been performed. The absolute address of the first character of the next record is in the four least significant characters of arithmetic register 1.

If an end-of-file block is detected, the current reel has been rewound with interlock and control transferred to the label specified in parameter 11 of the FILE1 line. The address of the end-of-file block is not supplied, and no further macro instructions may be executed for the file until it is reopened.

2.3.2.3. Input Record Advance (Transfer Form)

CE	LABEL	OPERATION	OPERANDS
5 6	7 11	13 18 19	30 40
	label	GET 1	file ID, destination

■ Format

Destination is either the label of a working storage area or the label of an output record.

■ Entrance Requirements

The file is open.

If destination is the label of a record, the output file with which it is associated is open and its index register contains the address of the first character of the current record area.

■ Exit Conditions

The next record has been transferred to the area specified by destination.

If an end-of-reel block was detected, the current reel has been rewound with interlock and the next reel opened. The label has been checked and the worker program's special label processing, if any, has been performed. The next record is in the area specified by destination.

If an end-of-file block was detected, the current reel has been rewound with interlock and control transferred to the label specified in parameter 11 of the FILE1 line. The end-of-file block was not transferred, and no further macro instructions may be executed for the file until it is reopened.

2.3.2.4. Close Input File

CE	LABEL	OPERATION	OPERANDS
5 6	7 11	13 18 19	30 40
	label	C L O S 1	file ID, rewind option

■ Format

Rewind option is RWD, for a rewind without interlock, LOCK, for a rewind with interlock, or NORWD, if the current reel is not to be rewound.

■ Entrance Requirements

The file is open.

■ Exit Conditions

The current reel has been rewound as specified.

Control is not transferred to the worker program's end-of-file section (parameter 11 of the FILE1 line), but passes to the worker program at the point immediately following the macro instruction.

No further macro instruction may be executed for the file until it has been reopened.

2.3.2.5. Close Input Reel

CE	LABEL	OPERATION	OPERANDS
5 6	7 11	13 18 19	30 40
	label	C L O S 1	file ID, rewind option, REEL
	label	C L O S 1	file ID, destination, rewind option, REEL

■ Format

Destination is specified if the transfer form of the macro instructions were called for in the FILE1 line. It is the label of either a working storage area, or the record of an output file.

Rewind option is RWD, for a rewind without interlock, LOCK, for a rewind with interlock, or NORWD if the current reel is not to be rewind.

■ Entrance Requirements

The file is open.

If destination is the label of a record, the output file with which it is associated is open and its index register contains the address of the first character of the current record area.

■ Exit Conditions

The current reel has been rewound as specified and the next reel opened. The label has been checked and the worker program's special label subroutine, if any, has been executed.

If destination were specified, the first record of the new reel is in the area specified by destination; otherwise, the absolute address of the first record is in the four least significant characters of AR1.

2.3.3. Output Macro Instructions

2.3.3.1. Open Output File

CE	LABEL	OPERATION	OPERANDS
5 6	7 11	13 18 19	30 40
	label	O P E N 1	file ID

■ Entrance Requirements

This must be the first macro instruction executed for the file, or the first macro instruction executed since the file was closed.

■ Exit Conditions

The label block has been written on the tape unit specified by parameter 6 of the FILE1 call.

If specified in parameter 10 of the FILE1 call, a special label subroutine has been executed.

If the AR form of the PUT1 macro instruction is used, the absolute address of the first character of the first record area will be in the four least significant characters of arithmetic register 1.

2.3.3.2. Output Record Advance (Arithmetic Register Form)

CE	LABEL	OPERATION	OPERANDS
<small>E IN S</small> 5 6	7 11	13 18 19	30 40
	label	P U T 1	file ID

■ Entrance Requirements

The file is open.

■ Exit Conditions

The absolute address of the first character of the next record area will be in the four least significant characters of arithmetic register 1. The previous record is not available to the worker program.

If an end-of-tape condition was detected and parameter 11 of the FILE1 call was blank, then two end-of-reel blocks have been written on the current reel. The current reel has been rewound with interlock and a label block was written on the next reel. If specified in parameter 10 of the FILE1 call, a special label subroutine has been executed. The absolute address of the first character of the next record area is in the four least significant characters of arithmetic register 1.

If an end-of-tape condition was detected, and parameter 11 of the FILE1 call was not blank, control has been transferred to the specified label. The absolute address of the first character of the next record area is in the four least significant characters of AR1.

2.3.3.3. Output Record Advance (Transfer Form)

CE	LABEL	OPERATION	OPERANDS
<small>E IN S</small> 5 6	7 11	13 18 19	30 40
	label	P U T 1	origin, file ID

■ Format

Origin is either the label of a working storage area, or the label of an input record.

■ Entrance Requirements

The file is open.

If origin is an input record, the input file with which it is associated is open and its index register contains the address of the first character of the current record area.

■ Exit Conditions

The record has been transferred to the output area.

If an end-of-tape condition was detected and parameter 11 of the FILE1 call was blank, then two end-of-reel blocks have been written onto the current reel, which was then rewound with interlock. A label block was written on the next reel. If specified in parameter 10 of the FILE1 call, a special label sub-routine has been executed.

If an end-of-tape condition was detected, and parameter 11 of the FILE1 call was not blank, control has been transferred to the specified label.

2.3.3.4. Close Output File

CE	LABEL	OPERATION	OPERANDS
5 6	7 11	13 18 19	30 40
	label	C L O S E	file ID, rewind option

■ Format

Rewind option is RWD, for a rewind without interlock, LOCK for a rewind with interlock, or NORWD if the current reel is not to be rewound.

■ Entrance Requirements

The file is open.

■ Exit Conditions

All records committed to output have been written onto tape. Two end-of-file blocks have also been written.

The current reel has been rewound as specified.

No further macro instructions may be executed for the file until it has been reopened.

2.3.3.5. Close Output Reel

CE	LABEL	OPERATION	OPERANDS
REINS 5 6	7 11	13 18 19	30 40
	label	C L O S 1	file ID, rewind option, REEL

- Format

Rewind option is RWD for a rewind without interlock, LOCK for a rewind with interlock, or NORWD if the current reel is not to be rewound.

- Entrance Requirement

The file is open.

- Exit Conditions

All records committed to output have been written onto tape. Two end-of-reel blocks have also been written.

The current reel has been rewound as specified, and the label block has been written on the next reel.

If specified in parameter 10 of the FILE1 call, a special label subroutine has been executed.

If the AR form of the PUT1 macro instruction is used, the absolute address of the first character of the first record area will be in the four least significant characters of AR1.

2.3.4. Establish Rerun Point

CE	LABEL	OPERATION	OPERANDS
REINS 5 6	7 11	13 18 19	30 40
	label	R E R N 1	return, P ₁

- Format

Return is the label to which control will be returned when the program is rerun from this point.

P₁ is FR if the Fastrand File Control Routine is part of the worker program. This parameter is blank if Fastrand is not used.

■ Entrance Requirements

The file on which the rerun point is to be established is open.

The minimum block size of the file on which the rerun point is to be established must be 450 characters.

All typewriter messages have been completed.

No unsolicited typeins to the worker program are allowed from the time the rerun memory dump is started (stop display 15uull, where uu is the tape unit, to the time the rerun memory dump is completed (stop display 15uu12).

■ Exit Conditions

■ When the rerun point has been established:

All records committed to the rerun file have been written onto tape.

A rerun memory dump, bracketed by bypass-sentinels, has been written.

If the AR form of the PUT1 macro instruction is used, the absolute address of the first character of the next record area is in the four least significant characters of arithmetic register 1.

Control is returned to the worker program at the instruction following the macro instruction.

The contents of arithmetic register 1, arithmetic register 2, and the indicators (HI, LO, EQ) are destroyed.

■ When the program is being rerun:

All TAPE1 files have been repositioned.

The storage allocated to the worker program has been restored.

If the AR form of the PUT1 macro instruction is used, the absolute address of the first character of the next record area is in the four least significant characters of arithmetic register 1.

Control is returned to the worker program at the label specified in the operands field of the macro instruction.

All index registers are restored.

The sense indicators are restored.

2.4. TAPE1 (OPR) STOPS AND PROCEDURES

All typeouts (stop displays) are in the format 1cuuxx, where: c = channel, uu = unit, and xx = a code indicating the reason for the typeout.

For a description of the correct typein procedure for entering responses refer to the OPR Executive Routine document.

STOP CODE	MEANING	OPERATOR ACTION
01	Label Error	<ol style="list-style-type: none"> 1. To try another tape: <ol style="list-style-type: none"> a. If tape unit is rewind: <ol style="list-style-type: none"> (1) Mount new reel on same tape unit. (2) Type in a value other than 00 or 01. (3) Program will check label block of new reel. b. If tape unit is not rewind: <ol style="list-style-type: none"> (1) Type in 00. (2) Tape unit will rewind, and typeout will be repeated. (3) Continue as in 1-a above. 2. To ignore error and continue: <ol style="list-style-type: none"> a. Type in 01. b. Program will continue. 3. To perform label check on next block: <ol style="list-style-type: none"> a. Type in a value other than 00 or 01. b. Program will read next block, and perform a label check on it.
03	Block count Error	<ol style="list-style-type: none"> 1. To skip the error block: <ol style="list-style-type: none"> a. Type in 00. b. Program will forward the tape one block. c. Program will accept new tape block count and continue. 2. To continue processing the error block: <ol style="list-style-type: none"> a. Type in 01. b. Program will accept block count from tape and will continue processing (with block type check). 3. Typing in a value other than 00 or 01 will cause the program to loop through OPR, to wait to be jettisoned by the operator.
06* 07*	End-of-Reel End-of-File	<ol style="list-style-type: none"> 1. To treat sentinel as end-of-reel: <ol style="list-style-type: none"> a. Type in 06. b. Program will close the current reel and continue. 2. To treat sentinel as end-of-file: <ol style="list-style-type: none"> a. Type in a value other than 06. b. Program will close the file and continue.
10	Block-type error	Refer to block count error above. Note that typing in 01 will cause program to repeat error typeout cycle.
45	write error; zero character count due to programmer error	The recovery procedure is described in the OPR Executive Routine description
55	unrecoverable memory parity error	
66	tape parity error	
76	non-ready first block condition	
77	unit offline or non-ready	

* This typeout will occur only if the sentinel option was chosen in the TAPE 1 call.

Table 1. TAPE1 (OPR) Stop Codes

The following four stops may appear only if rerun points are being established through the RERN1 macro instruction.

STOP DISPLAY	MEANING	OPERATOR ACTION
30 15 uu 11 60	A rerun point is being established on tape unit uu.	Press the PROGRAM START button to display the rerun point number.
30 15 rrrr 60	Rerun point rrrr is being established.	Record rrrr for future reference and press the PROGRAM START button to continue.
30 15 uu 12 60	A rerun point has been established on tape unit uu.	Press the PROGRAM START button to display the rerun point number.
30 15 rrrr 60	Rerun point rrrr has been established.	Press the PROGRAM START button to continue the worker program.

2.5. TAPE1 (OPR) RERUN

2.5.1. General

This section presents operating instructions and considerations for TAPE1 rerun in the environment of the OPR Executive.

The rerun point macro instruction, RERN1, establishes a rerun point on the output file specified by parameter 9 of the TAPE1 call. When written on tape, a rerun point comprises a bypass – sentinel block (see Table 1), a number of tape file positioning blocks, a number of tape memory dump blocks, and a second bypass – sentinel block.

The worker program can be loaded and rerun by using the utility program RRN1. It locates a specified rerun point, repositions all tape files, and reloads storage from the rerun memory dump. Control is then returned to the worker program at the label specified in the RERN1 macro instruction that established the rerun point.

The OPR rerun dump routine dumps only the memory associated with the worker program from which the rerun dump is being made. Rerun points may be established at any time during the running of the worker program, including during concurrent program running. If the worker program is absolute, the rerun dump will start with the highest Executive Routine location, plus one, and end with the highest location in store.

It is not necessary to close the reader, punch or printer routines before executing the RERN1 macro instruction. All pertinent I/O will be stopped by TAPE1 rerun coding before writing out the memory dump.

Including fill characters, 350 characters of the rerun file area are used by TAPE1 coding during the establishment of the rerun dump. The estimated storage requirement for the RERUN coding is 1000 characters. The storage requirement for each RERN1 macro instruction is 10 characters. If the Fastrand File Control Routine is used, 15 characters are required for each RERN1 macro instruction.

2.5.2. TAPE1 Rerun Utility Program RRN1

RRN1 is used to rerun a program from a rerun point established by the RERN1 macro instruction of the TAPE1 I/O routine. RRN1 will reposition all files in use, and reload the worker program into storage, from the memory dump, overlaying itself in the process. Control is then returned to the program being run at the address that was specified by the RERN1 macro instruction when the rerun point was established.

BLOCK CHARACTER NUMBER	ITEM CHARACTER NUMBER	NUMBER OF CHARACTERS	CONTENTS
0	-	-	05
1,2,3	-	3	bbb - block number in binary.
4,5	-	2	Not used.
6,7	0-1	2	Rerun number in binary.
8-35	2-29	28	Contents of IR1 through IR7.
36-59	30-53	24	Contents of tetrads 16 through 21.
60-62	54-56	3	Worker program Class 2 entry address.
63-65	57-59	3	Worker program starting address.
66-68	60-62	3	Worker program ending address.
69-71	63-65	3	Rerun coding return address to TAPE1.
72	66	1	Number of I/O files.
73	67	1	Sense Indicator settings and FR indicator bit.
74-81	68-75	8	Memory dump coding start and end address.
82-99	76-93	18	Not used.
100-227	94-221	128	Contents of tetrads for channel 0 through 7.
228-291	222-285	64	Contents of interrupt entry for channels 0 through 7.
292-328	286-322	37	Not used.
329-343	323-337	15	Switch return addresses - channels 0 through 3, and 7.
344-375	338-369	32	Not used
376-390	370-384	15	Peripheral I/O issue addresses channels 0 through 3, and 7.
391-449	385-443	59	Not used. (Minimum block size 450 characters.)

Table 2. TAPE1 Rerun Dump Bypass-Sentinel Block Layout.

2.5.2.1. RRN1 Operating Instructions

- (1) Mount tapes and set peripheral units as specified by the worker program rerun operating procedures.
- (2) Load RRN1 from the master instruction tape using the normal call procedure. The seven following stop displays should appear in sequence.
- (3) 020050 Initial load stop.

Specify recording density of the rerun file in the least significant character of the trace address as follows, then depress the PROGRAM START button.

DENSITY	SPECIFICATION
UNIVAC III	00
Compatible tape 200 PPI	21
556 PPI	01
800 PPI	31

- (4) 020051 Specify the tape unit number on which the rerun point is located in the least significant character of the trace address. Depress the PROGRAM START button.
- (5) 02uu52 Specify the most significant character of the rerun point number in the address portion of the trace address and depress the PROGRAM START button.
- (6) 02uu53 Specify the least significant character of the rerun point number in the address portion of the trace address and depress the PROGRAM START button.
- (7) 02rrrr If the rerun point number displayed (rrrr) is correct, set the trace address to 00 and depress the PROGRAM START button. (RRN1 will search for the rerun point selected, reposition the tapes, reload the worker program memory, and stop with the following display.) If the rerun point number (rrrr) is incorrect, set the trace address to a number other than 00 and depress the PROGRAM START button. This will bring the program back to the initial load stop.
- (8) 15uu13 This stop occurs after the worker program has been reloaded and is ready to be rerun. Peripheral units must be set at this time.
- (9) 15rrrr This stop shows the number of the rerun point just selected (rrrr). Depress PROGRAM START and the worker program will be executed through control being transferred to the return line of the rerun point macro instruction, RERN1.

2.5.2.2. RRN1 Error Stops

- 02uu60 Block count error on repositioning tape file uu. To ignore block count error set the least significant character of the trace address to 01 and depress the PROGRAM START Button.
- 02uu61 Block count error on reading file repositioning information, or memory dump block on tape unit uu. Restart rerun procedure.
- 02uu62 Program being rerun is not within the limits of storage. Restart the rerun procedure when more storage is available.
- 02uu63 End of tape or tape mark detected. Restart rerun procedure.
- 02uu64 End of file block encountered on tape unit uu. Rerun point specified not found. Restart.
- 02uu66 Label error in rerun file. Restart rerun procedure.
- 02uu67 Label error on file other than the rerun file.

To accept the label and reposition the tape file, set the least significant character of the address in the trace switches to 01 and depress the PROGRAM START button.

To bypass repositioning of tape file, set the least significant character of the address in the trace switches to 02 and depress the PROGRAM START button.

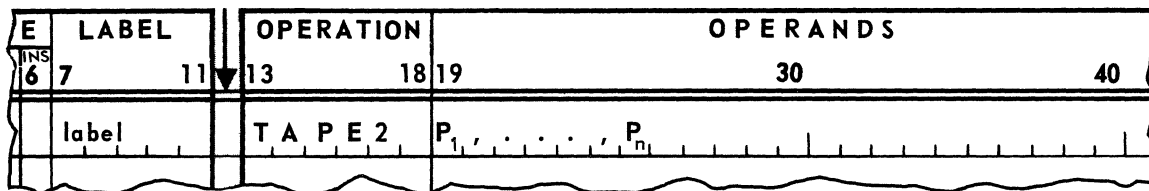
To jettison the rerun program, set the least significant character of the trace address to other than 01 or 02 and depress the PROGRAM START button.

3. TAPE2 INPUT CONTROL ROUTINE

The TAPE2 routine controls a single input file, performing all necessary label checking, block handling, record handling, and related functions. It is essentially a subset of TAPE1 providing certain economies in storage requirements and execution time for a program or program segment that has only one tape input file. Two program components are involved in its use: the TAPE2 call, and a set of macro instructions.

3.1. TAPE2 CALL

The TAPE2 CALL SPECIFIES 16 parameters that describe the input file and how it is to be handled. These parameters are used to generate the input subroutines, constants, and working storages necessary to perform the required input functions.



The label field contains a unique 1- to 5-character label that will identify the file in the macro instructions.

The operands field contains up to 16 parameters:

- p₁ Tape label: 13 or less characters bounded by apostrophes.
- p₂ Reel base number: three decimal digits bounded by apostrophes. This value plus decimal one will be checked against the reel number of the first reel.
- p₃ Tape type and recording density:¹ A for UNISERVO IIIA tapes; B for compatible tapes at 200 PPI; C for compatible tapes at 556 PPI; or D for compatible tapes at 800 PPI.
- p₄ Channel: 4 or 5.
- p₅ First tape unit number.
- p₆ Second tape unit number, if tape unit swap for alternate reels is desired; otherwise this parameter is blank.
- p₇ Label of an area large enough to contain one block of the file. (Refer to p₁₄.) This area is defined by the worker program.

¹ Translation mode is not specified for compatible tapes because they are always read or written in the binary mode.

- p_8 If the standby method is to be applied to the file, this parameter specifies a second area, defined by the worker program, large enough to contain one block of the file; otherwise it is blank.
- p_9 Form of macro instructions: AR if the arithmetic register form is to be used, or TRF if the transfer form is to be used.
- p_{10} Label of a closed subroutine which is to be performed in addition to the standard label processing, or blank. The subroutine is executed after the label block is read, but before it is checked. The four least significant characters of AR1 contain the absolute address of the label block when the subroutine is entered.
- p_{11} Label in the worker program to which control will be transferred when an end-of-file block is read. (Subject to p_{15} .)
- p_{12} Record size. If p_9 is TRF, this parameter cannot be greater than 1024.
- p_{13} Number of fill characters, or 0. (Refer to p_{14} .)
- p_{14} Physical block size, which equals: p_{12} , times number of records per block, plus p_{13} , plus 6.
- For UNISERVO 111A tapes, this parameter must be a multiple of 4.
- p_{15} Sentinel option: OPSEN, if sentinel option is desired, otherwise it is blank. If the sentinel option is chosen, the program will stop whenever an end-of-file or end-of-reel sentinel is read. By means of a trace switch setting, or typein, the operator will direct the TAPE2 coding to perform either end-of-file or end-of-reel processing for the file.
- p_{16} RT, if the worker program is servicing real-time devices; blank, otherwise. If RT is specified, the tape XF instructions associated with the worker program will be flagged with a priority bit when they are delivered to the tape handler. This priority bit will give them precedence over the tape XF instructions of concurrently running programs.

3.2. TAPE2 MACRO INSTRUCTIONS

The worker program communicates with the TAPE2 coding by means of the following macro instructions. Format, entrance requirements, and exit conditions are given for each macro instruction.

In addition to the specified exit conditions given for each macro instruction, it should be noted that all macro instructions alter the contents of arithmetic registers 1 and 2.

An entry in the label field of a macro instruction applies to the first instruction generated.

3.2.1. Open File

E	INS	LABEL	OPERATION		OPERANDS					
			6	7	11	13	18	19	30	40
		label		O P E N 2		file ID				
		label		O P E N 2		file ID, destination				

■ Format

Destination is specified if the transfer form of the macro instruction was called for in TAPE2. It is the label of an area large enough to contain one record of the file.

■ Entrance Requirement

This must be the first macro instruction executed for the file.

■ Exit Conditions

The label block has been read from the tape unit specified by parameter 5 of the TAPE2 call and has been checked.

If specified in parameter 10 of the TAPE2 call, a special label subroutine has been performed.

If the arithmetic register form is used, the four least significant characters of AR1 contain the absolute address of the first data record.

If the transfer form is used, the first data record has been transferred to destination.

3.2.2. Record Advance

E	INS	LABEL	OPERATION		OPERANDS					
			6	7	11	13	18	19	30	40
		label		G E T 2		file ID				
		label		G E T 2		file ID, destination				

■ Format

Destination is specified if the transfer form was called for in TAPE2. It is the label of an area large enough to contain one record of the file.

■ Entrance Requirement

The file is open.

■ Exit Conditions

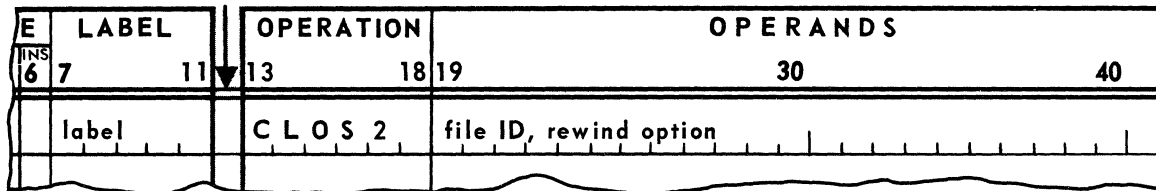
If the arithmetic register form is used, the absolute address of the first character of the next record is in the four least significant characters of AR1.

If the transfer form is used, the next record has been transferred to destination.

If an end-of-reel block was detected, the current reel has been rewound with interlock and the next reel has been opened. The label block has been checked and the worker program's special label processing, if any, has been performed.

If an end-of-file block was detected, the current reel has been rewound with interlock and control transferred to the label specified in parameter 11 of the TAPE2 call. The end-of-file block is not available to the worker program, and no further macro instructions may be executed for the file.

3.2.3. Close File



■ Format

Rewind option is LOCK for a rewind with interlock, or RWD for a rewind without interlock.

■ Entrance Requirement

The file is open.

■ Exit Conditions

The current reel has been rewound as specified.

Control is not transferred to the worker program's end-of-file section (parameter 11 of the TAPE2 call), but passes to the worker program at a point immediately following the macro instruction.

No further macro instructions may be executed for the file.

3.3. TAPE2 (OPR) STOPS AND PROCEDURES

All typeouts (stop displays) are of the form 1cuuxx, where c = channel, uu = unit, and xx = a code indicating the reason for the typeout.

For a description of the correct typein procedure for entering responses refer to the OPR Executive Routine document.

STOP CODE	MEANING	OPERATOR ACTION
01	Label Error	<ol style="list-style-type: none"> 1. To try another tape: <ol style="list-style-type: none"> a. If servo is rewound: <ol style="list-style-type: none"> (1) Mount new reel on same tape unit. (2) Type in a value other than 00 or 01. (3) Program will check label block of new reel. b. If tape unit is not rewound: <ol style="list-style-type: none"> (1) Type in 00. (2) Tape Unit will rewind, and typeout will be repeated. (3) Continue as in 1-a above. 2. To ignore error and continue: <ol style="list-style-type: none"> a. Type in 01. b. Program will continue. 3. To perform label check on next block: <ol style="list-style-type: none"> a. Type in a value other than 00 or 01. b. Program will read next block, and perform a label check on it.
03	Block Count Error	<ol style="list-style-type: none"> 1. To skip the error block: <ol style="list-style-type: none"> a. Type in 00. b. Program will forward the tape one block. c. Program will accept new tape block count and continue. 2. To continue processing the error block: <ol style="list-style-type: none"> a. Type in 01. b. Program will accept block count from tape and will continue processing (with block type check). 3. Typing in a value other than 00 or 01 will cause the program to loop through OPR, to wait to be jettisoned by the operator.
06* 07*	End-of-File End-of-Reel	<ol style="list-style-type: none"> 1. To treat sentinel as end-of-reel: <ol style="list-style-type: none"> a. Type in 06. b. Program will close the current reel and continue. 2. To treat sentinel as end-of-file: <ol style="list-style-type: none"> a. Type in a value other than 06. b. Program will close the file and continue.
10	Block Type Error	Refer to block count error above. Note that typing in 01 will cause program to repeat error typeout cycle.
55	Unrecoverable, memory parity error	The recovery procedure is described in the OPR Executive Routine description.
66	Tape parity error	
77	Unit offline or non-ready	

*This typeout will occur only if the sentinel option was chosen in the TAPE2 call.

Table 3. TAPE2 (OPR) Stop Codes.

4. TAPE3 OUTPUT CONTROL ROUTINE

The TAPE3 routine controls a single output file, performing all necessary block handling, record handling, and related functions. It is essentially a subset of TAPE1 providing certain economies in storage requirements and execution time for a program or program segment that has only one output file. Two program components are involved in its use: a TAPE3 call, and a set of macro instructions.

4.1. TAPE3 CALL

The TAPE3 call specifies 15 parameters which describe the output file and how it is to be handled. These parameters are used to generate the output subroutines, constants, and working storages necessary to perform the required output functions. The TAPE3 calling statement has the following format:

E	LABEL		OPERATION			OPERANDS					
6	7	11	13	18	19	30				40	
	label		T A P E 3	p ₁ p _n							

The label field contains a unique 1- to 5-character label that will identify the file in the macro instructions.

The operands field contains the following:

- p₁ Tape label: 13 or less characters bounded by apostrophes.
- p₂ Reel number base: three decimal digits bounded by apostrophes. This value plus decimal one will be the reel number of the first reel.
- p₃ Tape type and recording density:¹ A for UNISERVO IIIA tapes; B for compatible tapes at 200 PPI; C for compatible tapes at 556 PPI; or D for compatible tapes at 800 PPI.
- p₄ Channel: 5.
- p₅ First tape unit number.
- p₆ Second tape unit number, if tape unit swap for alternate reels is desired; otherwise this parameter is blank.

¹ Translation mode is not specified for compatible tapes because they are always read or written in the binary mode.

- p_7 Label of an area large enough to contain one block of the file. This area is defined by the worker program.
- p_8 If the standby method is to be applied to the file, this parameter specifies a second area, defined by the worker program, large enough to contain one block of the file; otherwise it is blank.
- p_9 Form of the macro instructions: AR if the arithmetic register form is to be used, or TRF if the transfer form is to be used.
- p_{10} Label of a closed subroutine which is to be executed in addition to the standard label processing, or blank. The subroutine is executed after the label block is assembled in an output area but before it is written onto tape. The four least significant characters of AR1 contain the absolute address of the label block when the subroutine is being entered.
- p_{11} This parameter determines what the TAPE3 coding will do when an end-of-tape condition is detected while a PUT3 macro instruction is being executed.

If this parameter is blank, the TAPE3 coding will close the current reel and open the next, returning control from the PUT3 in the normal fashion. (Refer to Exit Conditions in Close Reel, which constitutes a detailed description of the end-of-tape actions performed by the TAPE3 coding.)

If this parameter is not blank, the TAPE3 coding will transfer control to the specified label.¹ The worker program may then perform any desired end-of-reel processing, such as putting out summary records or hash totals. This processing must be followed by a close reel macro instruction after which normal processing may be resumed.

If there is more than one PUT3 macro instruction, control may be returned to the proper point by a jump to the exit line of the PUT3 subroutine. If the arithmetic register form of the macro instructions is used, the exit line is labeled XTO60; if the transfer form is used, the exit line is labeled XTO62. It should be noted, however, that the execution of one or more PUT3 macro instructions in the end-of-reel processing will alter the exit line. In this case, the worker program must save XTO60+1 through XTO60+2, or XTO62+1 through XTO60+3, before the macro instructions are executed.

- p_{12} Record size. If p_9 is TRF, this parameter cannot be greater than 1024.
- p_{13} Number of fill characters, or 0. (Refer to p_{14} .)
- p_{14} Physical block size, which equals: p_{12} , times number of records per block, plus p_{13} plus 6.
- For UNISERVO IIIA tapes, this parameter must be a multiple of four.
- p_{15} RT, if the worker program is servicing real-time devices; blank, otherwise. If RT is specified, the tape XF instructions associated with the program will be flagged with a priority bit when they are delivered to the tape handler. This priority bit will give them precedence over the tape XF instructions of concurrently running programs.

¹This transfer of control will take place only once per reel.

4.2. TAPE3 MACRO INSTRUCTIONS

The worker program communicates with the TAPE3 coding by means of the macro instructions described below. Format, entrance requirements, and exit conditions are given for each macro instruction.

In addition to the specified exit conditions given for each macro instruction, it should be noted that all macro instructions alter the contents of arithmetic registers 1 and 2.

An entry in the label field of a macro instruction refers to the first instruction generated.

4.2.1. Open File

E	LABEL	OPERATION	OPERANDS
INS 6	7 11	13 18 19	30 40
	label	O P E N 3	file ID

- Entrance Requirement

This must be the first macro instruction executed for the file.

- Exit Conditions

The label block has been written on the tape unit specified by parameter 5 of the TAPE3 call.

If specified in parameter 10 of the TAPE3 call, a special label subroutine has been executed.

If the arithmetic register form of the macro instructions is used, the four least significant characters of AR1 contain the absolute address of the first character of the first record area.

4.2.2. Record Advance

E	LABEL	OPERATION	OPERANDS
INS 6	7 11	13 18 19	30 40
	label	P U T 3	file ID
	label	P U T 3	origin, file ID

- Format

Origin is specified if the transfer form was called for in TAPE3. It is the label of an area large enough to contain one record of the file.

■ Entrance Requirement

The file is open.

■ Exit Conditions

If the arithmetic register form is used, the absolute address of the next record area is in the four least significant characters of AR1.¹

If the transfer form is used, the record has been transferred from origin to an output area.

If an end-of-tape condition was detected, and parameter 11 of the TAPE3 call was blank, two end-of-reel blocks were written on the current reel, which has been rewound with interlock. A label block was written on the next reel. If specified in parameter 10 of the TAPE3 call, a special label subroutine has been executed.

If an end-of-tape condition was detected, and parameter 11 of the TAPE3 call was not blank, control has been transferred to the specified label.

4.2.3. Close File

E	LABEL		OPERATION			OPERANDS			
INS	6	7	11	13	18	19	30	40	
	label		C L O S 3	file ID, rewind option					

■ Format

Rewind option is RWD for a rewind without interlock; LOCK for a rewind with interlock; or NORWD if the current reel is not to be rewound.

■ Entrance Requirement

The file is open.

■ Exit Conditions

All records committed to output have been written onto tape, together with two end-of-file blocks.

The current reel has been rewound as specified.

No further macro instructions maybe executed for the file.

¹This also applies in the case of an end-of-tape condition.

4.2.4. Close Reel

E	LABEL		OPERATION			OPERANDS			
INS	6	7	11	13	18	19	30		40
		label		C L O S 3	file ID, rewind option, REEL				

- Format

Rewind option is RWD, for a rewind without interlock; LOCK for a rewind with interlock; or NORWD if the current reel is not to be rewind.

- Entrance Requirement

The file is open.

- Exit Conditions

All records committed to output have been written onto tape together with two end-of-reel blocks.

The current reel has been rewound as specified, and a label block has been written on the next reel.

If specified in parameter 10 of the TAPE3 call, a special label subroutine has been executed.

If the arithmetic register form of the macro instructions is used, the four least significant characters of AR1 contain the absolute address of the first character of the first record area.

4.3. TAPE3 (OPR) STOPS AND PROCEDURES

There are no program stops for TAPE3.

The following are peripheral error stops in the tape handler portion of the OPR Executive Routine. The recovery procedures are given in the OPR Executive Routine document.

- 30 1c uu 45 — write error; zero character count due to programmer error
- 30 1c uu 55 — unrecoverable error
- 30 1c uu 66 — tape parity
- 30 1c uu 76 — non-ready first block condition
- 30 1c uu 77 — unit offline or non-ready

In these stops, c = channel, and uu = unit.

APPENDIX A. TAPE1 SPECIAL INPUT ERROR ROUTINE

A closed subroutine, providing user own code in the case of label, block count, or block type errors, may be specified for an input file in parameter 16 of the FILE1 call. When an error occurs, TAPE1 will execute this subroutine, rather than performing an error typeout. The address of the appropriate FILE1 packet will be in index register XTCX1, and an error flag will be set in absolute location 0. Currently, there are three different error flags: 01 for a label error; 03 for a block count error; and 010 for block type error.

A label error occurs when the label and reel number fields from tape do not agree with the expected label and reel number values. A block count error occurs when the block count from tape does not agree with the computed block count. A block type error occurs when, following a label or data block, a block type error is detected which is not a data, end-of-reel sentinel, end-of-file sentinel, or bypass sentinel block. (Refer to the data tape convention document.)

The information necessary to log or otherwise process the error can be found in the FILE1 packet for the file. If control is to be returned to TAPE1 when the error has been processed XTCX1 must contain its original value, and an error disposition code must be set in absolute location 31. The disposition code directs further TAPE1 processing as follows:

Error 01 – Label Error

CODE	RESULT
00	(1) Current reel rewinds with interlock. (2) Label error typeout (1cuu01) is made, directing operator to mount a new reel on the same tape unit (c = channel, uu = unit). (3) When new reel is mounted, the operator types in a value other than 00 or 01. (4) TAPE1 reads and checks the label on the new reel.
01	TAPE1 ignores error and continues processing current reel.
other	TAPE1 reads the next block from the current reel and performs a label check on it.

Error 03 – Block count Error

CODE RESULT

00	(1) TAPE1 reads the next block from tape. (2) Computed block count is overlaid with block count from new tape block. (3) Processing continues with new tape block.
01	(1) Computed block count is overlaid with block count from current tape area. (2) Processing continues with current block, beginning with block type check.
other	TAPE1 loops through OPR, waiting for operator-initiated jettison.

Error 010 – Block type Error

CODE RESULT

Same as block count error.

Note: If processing of error block as a data block is to continue (disposition code = 01), the value 04 (data block) must be placed in the most significant characters of the current tape area. The only other block types valid at this time are end-of-reel (06) and end-of-file (07). Insertion of one of these will cause the current block to be treated as a sentinel block.

APPENDIX B. FILE1 PACKET (OPR)

The following table describes the FILE1 packet containing 74 characters of constants and working storages that are produced by the FILE1 call and manipulated by the TAPE1 coding. The MSC of a file packet may be addressed by the file ID specified in the label field of the call. The addresses in the fourth column of the table are relative to this character. An entry p_n in the table refers to parameter n of the FILE1 call.

DESCRIPTION	OP'N	OPERANDS	REL ADDR OF LSC	CONSTANT/VARIABLE
XF instruction	XF	function, detail, p_7, p_5	4	V
Completion storage address (minus 2), current area	+3	$\$+36$	7	V
Base address, current area	+4	p_8	11	V
Block size	+4	p_{15}	15	C
Limit address for current block	+4	0	19	V
Address of current record or record area	+4	0	23	V
Record size	+4	p_{13}	27	C
Alternate tape unit for tape	+1	p_6 (+ 020, if $p_5 = 5$)	28	V
File descriptor: binary iexxsx, where $i = 1$, if input; $e = 1$, if EOT warning on reel, or reel being closed; $s = 1$, if standby; and x is unused.	+1	binary i000s0	29	V
Entrance to worker program's special label subroutine	+3 -3	$p_{10} + 3$ (p_{10} not blank) 1 (p_{10} blank)	32	C
Completion storage address (minus 2), alternate area	+3	$\$+12$ (p_9 not blank); $\$+8$ (p_9 blank)	35	V
Base address, alternate area	+4	p_9 (p_9 not blank); p_8 (p_9 blank)	39	V
Completion storage for p_8	+4	0	43	V
Completion storage for p_9	+4	0	47	V
Address in worker program for end-of-file (input) or end-of-tape (output)	+3 -3	p_{11} (p_{11} not blank) 1 (p_{11} blank)	50	C
Data character count (output), or entrance to worker program's special error subroutine (input)	+3 +3 -3	$p_{15} - p_{14} - 6$ (if output) $p_{16} + 3$ (if input & p_{16} not blank) 1 (if input & p_{16} blank)	53	C
Block count	+4	Decimal 1 followed by three blanks	57	V
Label	+13	p_2	70	C
Reel number	+3	p_3	73	V

