JOB SPECIFICATION LANGUAGE
REFERENCE MANUAL

TEXAS INSTRUMENTS
INCORPORATED

ASC

JOB SPECIFICATION LANGUAGE
REFERENCE MANUAL

Equipment Group
P. O. Box 2909
Austin, Texas 78767

TEXAS INSTRUMENTS
INCORPORATED

930038-2
January 1974

1948

INSERT LATEST CHANGED PAGES DESTROY SUPERSEDED PAGES

LIST OF EFFECTIVE PAGES

Note: The portion of the text affected by the changes is indicated by a vertical line in the outer margins of the page.

DATES OF ISSUE FOR ORIGINAL AND CHANGED PAGES ARE:

ORIGINAL .... 0 .... FEBRUARY 1974

Page No.	Change No.	Page No.	Change No.	Page No.	Change No.
Title	0				
A page	0				
Notice	0				
iii - xiv	0				
1-1 - 1-8	0				
2-1 - 2-15......	0				
3-1 - 3-5	0				
4-1 - 4-10	0				
5-1 - 5-32......	0				
6-1 - 6-36......	0				
7-1 - 7-82......	0				
8-1 - 8-29......	0				
9-1 - 9-34......	0				
A-1 - A-44	0				
A1-1 - A1-20....	0				
A2-1 - A2-10....	0				
B-1 - B-9......	0				
C-1 - C-6......	0				
D-1 - D-5......	0				
I-1 - I-21......	0				
Updates	0				

*The asterisk indicates pages changed, added, or deleted by the current change.

THE JSL REFERENCE MANUAL

-NOTICE-

This revision of the Job Specification Reference Manual includes new statements, macros and examples. However, the following items are not yet available for use in the ASC Operating System:

SETUP statement
SETUP parameter, FD statement
DEVICE TYPE (DTYP) parameter, FD statement

Also, DTYP parameter applies only to tape and direct secondary disc use for applicable statements.

When these features are effective in OS there will be a notice to that effect.

TABLE OF CONTENTS

SECTION I. INTRODUCTION

SECTION II. DEFINING THE JOB

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Paragraph Title Page

SECTION V. FILE DESCRIPTION

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

SECTION VII. CATALOG MANAGEMENT

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Paragraph Title Page

TABLE OF CONTENTS (Continued)

SECTION VIII. SPECIAL FUNCTIONS

TABLE OF CONTENTS (Continued)

SECTION IX. THE MACRO LANGUAGE

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

1-1 GENERAL

Processing requirements are specified to the control components of the ASC Operating System through programs written in a special purpose language. This language is known as the ASC Job Specification Language (JSL).

All Job Specification Language programs are processed by the Job Specification Language Translator. The Translator is a Central Processor step whose execution is initiated by the Command Controller component of the ASC Operating System. The Translator translates the user's job specification program into an object format compatible with the Command Controller. The execution of the job specification object program is an interpretive process.

The Job Specification Language is composed of job definition statements, program processing statements, file input and output statements, file cataloging statements, control statements, special function statements, and a macro language.

1-2 LANGUAGE ELEMENTS

The language elements for the Job Specification Language consist of lists, symbols, character strings, variables, constants, and the characters that represent them.

1-3 CHARACTER SET

The ASC recognizes the EBCDIC character set as standard notation. References in this manual are made to alphabetic characters (A through Z and $), numeric characters (0 through 9), special characters (all the rest), and alphanumeric characters which is the set of both alphabetic and numeric characters.

Table 1-1 contains a list of all the printable special characters and their names. The printable special characters are that subset of the EBCDIC set, aside from the alphanumerics, that is practical for use in coding.

1-4 VALID ASC SYMBOLS

A valid ASC symbol is a string of from one to eight characters, the first of which must be alphabetic. The remaining characters may be any sequence of alphanumeric characters, and the special character ". " (period).

Table 1-1. Printable Special Characters

CHARACTER	NAME	CARD CODE	CHARACTER	NAME	CARD CODE
	blank	blank	-	hyphen, or minus sign	11
¢	cent sign	12-8-2	/	slash	0-1
.	period	12-8-3	,	comma	0-8-3
<	less than	12-8-4	%	percent sign	0-8-4
(	left parenthesis	12-8-5	-	horizontal bar	0-8-5
+	plus sign	12-8-6	>	greater than	0-8-6
|	vertical bar	12-8-7	?	question mark	0-8-7
&	ampersand	12	↑	vertical arrow	
!	exclamation point	11-8-2	:	colon	8-2
*	asterisk	11-8-4	#	number	8-3
)	right parenthesis	11-8-5	@	at	8-4
			'	apostrophe	8-5
;	semicolon	11-8-6	=	equals	8-6
¬	not sign	11-8-7	"	quotation mark	8-7

1-5 CONSTANTS

In the Job Specification Language, a constant is a signed or unsigned decimal or hexadecimal integer. It is represented as a string of numeric characters. Hexadecimal integers must be preceded by a number sign (#).

The limits of the value of a constant are dictated by the specific parameter for which the constant is supplied.

1-6 CHARACTER STRINGS

A character string is a variable number of printable characters that are translated into their literal EBCDIC representations.

A character string is delimited either by being enclosed in quotation marks or is delimited by the Job Specification Language delimiters of the operands field; viz., commas or slashes.

If a character string is enclosed in quotation marks, any blanks in the string will be retained in the internal representation of the string; if a character string is not enclosed in quotation marks, any blanks in the string will be removed and will not appear in the internal representation of the string.

Restrictions: A pair of adjacent quotation marks appearing within a quoted string will not terminate the string; the result is one quotation mark (") embedded in the string.

A character string that is not enclosed in quotation marks cannot contain an embedded comma or slash; the comma or slash will terminate the parameter or subparameter, respectively.

Limitations: The number of characters permitted in any given character string is a function of the parameter for which the character string is supplied.

1-7 JOB SPECIFICATION VARIABLES

A Job Specification Variable is a value represented in the Job Specification Language by a symbol.

The Job Specification Language provides 27 variables. Each variable is represented symbolically by a single alphabetic character (viz., A through Z and $) and retains that symbolic identity for the duration of a job.

Variables may be set to positive or negative integer values either within job specification processing or within a job step.

Variables may be used in the programming of job execution flow with the Job Specification Language control statements, Section IV.

Variables may provide communication between job steps or between a job step and job control. Communication between job steps is achieved by assigning the same variable to both steps as run time parameters. Communication between job steps and job control is achieved by assigning the same variable to both job step(s) and job control statement(s). The value of a variable acquired during processing is available to job steps through service calls designed for that purpose.

Limitations: The range of values that a variable may have is:
$-2^{31} \leq v \leq 2^{31} - 1$.

Default: The initial values of the variables at the beginning of a job are zero and are changed only by programmed specification.

NOTE

Do not confuse the Job Specification variables with the options of the various JSL statements that are also represented by single alphabetic characters.

1-8 JOB SPECIFICATION INDICATORS

There are two Job Specification Indicators represented in the Job Specification Language by the character strings TERM and STATUS.

The purpose of the TERM indicator is to communicate TERMination data from a central processor step to the Job Specification. The value of TERM can only be modified by the Operating System. TERM is set to zero at the beginning of each step, to the value specified by the step at step termination, or upon abnormal termination it is set to 10 by the operating system. TERM is set to zero at the beginning of each step and is set to the value specified by the step at step termination.

The purpose of STATUS is to communicate information concerning the status of the Operating System to the Job Specification and to central processor steps.

Job Specification Indicators can be used in the programming of job execution flow with the Job Specification Language control statements, Section IV.

1-9 LISTS

A list is a string of elements separated by commas. In the trivial case, a list may be a single element.

A sublist is a list enclosed in parentheses. A sublist may be a single element or it may be an element in a list.

The operands field of a Job Specification Language statement is a list.

1-10 STATEMENT FORMAT

The Job Specification Language (JSL) is coded in the form of statements.

A statement consists of three fields: (1) an identifier/label field, (2) an operation field, and (3) an operands field.

The fields are of variable length and are terminated by at least one blank. Each field may be divided into subfields and the subfields may be further subdivided.

The general form of a Job Specification Language statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	ƀ	verb	ƀ	[positional params][, keyword params]

NOTE

No remark field is permitted in Job Specification Language statements.

1-11 IDENTIFIER/LABEL FIELD

The identifier/label field comprises two subfields: (1) the Job Specification Language statement identifier, and (2) an optional label.

1-12 JOB SPECIFICATION IDENTIFIER. All job specification statements are identified by placing a slash, /, in the first column of the first line of the statement. This identifier subfield is mandatory to identify the statement as a Job Specification Language statement.

More than one consecutive slash may be used in the identifier subfield, if desired. Only one, in column one, is required.

1-13 LABEL. The label, if one is provided, serves as a name to which Job Specification Language control statements may refer. The label begins immediately in the second column of the statement and terminates with the first encountered blank.

A label must be a valid ASC symbol. A symbol consists of from one to eight characters, the first of which must be alphabetic. The remaining characters may be any combination of alphabetic and numeric characters.

The presence of a label is optional for those statements that permit them.

Delimiters: A blank delimits the identifier/label field. No other delimiters are valid in this field group.

Restrictions: The identifier/label field cannot contain an embedded blank. It will be terminated by the first blank encountered.

Labels are not permitted on all Job Specification Language statements; whether a label is permitted is specified in the descriptions of the individual statements.

1-14 OPERATION FIELD

The operation field consists of the verb. The verb is mandatory; it specifies the operation that is to be performed by the system. The various verbs are identified and defined individually in this manual.

Delimiters: The operation field is delimited by blanks.

1-15 OPERANDS FIELD

The operands field consists of a list of positional and/or tagged parameters separated by commas.

1-16 POSITIONAL PARAMETERS. Positional parameters are operation data that must be provided by the user and that must be provided in a specified sequence. The sequence of positional parameters in each of the Job Specification Language statements is specified in its general form.

If both positional and keyword parameters are allowed, the positional parameters are listed first.

Positional parameters may have subfields which specify different characteristics of the given parameter. The subfields are delimited by slashes, /.

Certain positional parameters are concatenated symbols in which the catenation is indicated by a slash.

Restriction: Positional parameters are not optional; where the general form of a statement indicates a positional parameter, that parameter must be coded.

1-17 TAGGED PARAMETERS. Tagged parameters are operation data that is optional and cannot conveniently be forced to occupy a fixed coding position.

All tagged parameters are initiated by an identifying keyword followed by an equal sign, =.

There are three types of tagged parameters used in the Job Specification Language: (1) those in which the parameter is a keyword which is specified and defined internally to the system, (2) those in which the parameter is a constant, a character string, or a Job Specification Language variable assigned by the user, and (3) those in which the parameter comprises subparameters, separated by slashes, each of which may be a keyword, a constant, a character string, or a Job Specification Language variable.

Tagged parameters may be coded in any convenient sequence since they provide their own identification.

The tags and keyword parameters are specified in the general forms of the specific Job Specification Language statements to which they apply.

Delimiters: The operand field is initiated by the blank that terminates the operation field. Its termination is indicated by the end of a line wherein the last character in the line is not a semicolon.

The parameters within the operand field are delimited by commas.

Parameters having more than one element use slashes, /, as delimiters of the elements.

Certain subparameters may be conjuncted symbols in which the conjunction is indicated by an asterisk, *.

Restrictions: Blanks may be embedded anywhere between parameters in the operands field, but the blanks will not be processed as information to the system.

The last non-blank character of the operands field cannot be a semicolon, ;. A semicolon as the last non-blank character indicates that the statement is continued on the next line.

1-18 CONTINUATION LINES

A Job Specification Language statement may be continued to a second line.

A semicolon, ;, is used to indicate that a statement is to be continued.

A parameter can be split by the continuation character on the line. The statement may continue in column one of the continuation line.

A Job Specification Language statement ends with the first line that is not terminated by a semicolon.

Restrictions: A Job Specification Language statement cannot be continued prior to completion of the operation field.

A semicolon cannot be the final character of a Job Specification Language statement.

Limitations: A Job Specification Language statement may have a maximum of 128 continuations.

1-19 CARD IMAGE FORMAT RESTRICTIONS

When the Job Control Language statements are entered in card image format, the following restrictions apply:

- Information may be coded only in card columns 1 through 72. Card columns 73 through 80 are reserved for sequence numbering information.
- If a statement is to be continued, the semicolon must appear in or before column 72.

1-20 CONVENTION FOR DESCRIBING LANGUAGE STATEMENTS

The following conventions are observed in describing language statements:

- Upper case letters and punctuation marks (except those enclosed in brackets or braces) represent information that must be coded exactly as shown.
- Lower case letters and words are generic terms that represent information that must be supplied; i.e., a substitution must be made when coding a parameter or option so represented.
- Information within brackets, [], is optional. It may be included or omitted, depending upon program requirements.

- When several choices, listed vertically, are enclosed in braces, {}, one of the enclosed alternatives must be selected by the programmer. If one of the alternatives is underlined, the parameter may be omitted and the system assumes the underlined alternative.
- Mandatory blanks are represented by a slashed, lower-case letter "b" (b̸). This symbol is not used to represent permissible blanks.

1-21 COMMENTS STATEMENT (COM)

The COM statement specifies narrative comments that are to be entered into the Job Specification Language source listing of the system output file.

The general form of the COM statement is:

		OPERATION		OPERANDS
/	b̸	COM	b̸	text

The text of a COM statement is essentially the operands field of the statement, but the text is not processed by the system. The comment is entered into the system output file for documentation purposes only.

Format: The text of a comment may contain any combination of printable EBCDIC characters.

Restrictions: For card image input, the comment must terminate or be continued with a semicolon at or before column 72. COM statements do not provide any options and labels are permitted. A comment may contain an embedded semicolon. The entire field is read and a semicolon is interpreted as a continuation symbol only if it is the last non-blank character on the line.

SECTION II

DEFINING THE JOB

2-1 GENERAL

A job is specified by a sequence of Job Specification Language statements beginning with a JOB statement and ending with an EOJ statement. Optionally, a job may end with the statement immediately preceding the next JOB statement. The job is the processing initiated by these JSL statements.

The JOB statement must be the first JSL statement of a job. It provides job identification, accounting information, catalog use, priority and execution characteristics and informs the system whether the job may be restarted if processing fails. If an EOJ statement is used, it must be the last JSL statement of a job.

A job must end with a card with 7/9 punched in column one. The 7/9 card is the absolute delimiter for the end of a job. It must be the last card in the job deck. In most cases, the 7/9 card will be provided by the installation operators.

A job may be partitioned into logical sections called blocks. The job block is the basic unit for programmed flow of job execution specified by the JSL control statements. For jobs partitioned into blocks, resources can be reserved for those blocks of the job that need them as they need them to increase system efficiency. Job blocks are specified by LIMIT statements.

2-2 THE JOB STATEMENT

The JOB statement defines the beginning of a job.

The general form of the JOB statement is:

LABEL		OPERATION		OPERANDS
/	b̸	JOB		job name, acctnum, user code [, CAT = category] [, LOC = location] [, OPT = (option codes)]

The JOB statement cannot be labeled. It must contain the verb JOB in the operation field. It must also contain a valid job name, account number and user code in the operands field. These parameters are positional - they are mandatory and must be coded in the order shown. The statement may contain optional keyword parameters specifying priority and execution characteristics, output location and restart options. Keyword parameters may be coded in any convenient order.

If a previously existing job had no end-of-job definition (EOJ) statement, the occurrence of a JOB statement in the job input stream implicitly defines the end of that job, as well as initiating the new job. All job termination processing for the previous job will be initiated.

2-3 JOB NAME

Each job must have a name. It must appear first in the operands field.

Format: Job name is any combination of from one to 16 alphanumeric characters. The job name can begin with an alphabetic character, dollar sign character ($), or a numerical digit. It is assigned by the programmer. Two jobs can have the same name since more than one job can be stacked in the job input stream and executed at a time.

Default: None. Job name is mandatory.

The following are examples of naming jobs on JOB statements:

```
/b̸JOBb̸CST13710
/b̸JOBb̸TIMEANAL31072           b̸ = mandatory blanks that must be
/b̸JOBb̸$1310AB$ESTTIME              coded
/b̸JOBb̸JOBNAMEJOE
```

The following examples show incorrect job names:

```
/b̸JOBb̸*132&T2
/b̸JOBb̸CST1371¢                b̸ = mandatory blanks that must be
/b̸JOBb̸%JOBTIME                    coded
/b̸JOBb̸PAYROLL=YRTODATE
```

These names contain invalid symbols and will cause the job to terminate.

2-4 ACCOUNT NUMBER

Accounting information is supplied by the account number parameter. Each job must have an account to which job charges are billed. Installation accounting information is accumulated and processed by a job accounting function of the Operating System. The information is available to the installation for its use for billing functions.

Format: The account number may be any combination of from one to 16 alphanumeric characters (alphabetic characters, dollar sign character and numerical digits). It is assigned to the customer by installation accounting. The account number is separated from the job name with a comma.

Default: The account number must be the second parameter - it cannot be omitted.

The following examples show how to code account numbers and the job name:

```
/b̸JOBb̸CST13710, T134276
/b̸JOBb̸TIMEANAL31072, $REMC040            b̸ = mandatory blanks
/b̸JOBb̸$1310AB$ESTTIME, 12345                 that must be coded.
/b̸JOBb̸JOBNAMEJOE, ACCTNUMBER
```

2-5 USER CODE

The user code identifies and qualifies a specific user to the catalog system.

Format: The user code may be any combination of from one to eight alphanumeric characters (alphabetic, dollar sign character and numerical digits).

Default: The user code must be the third parameter; it cannot be omitted. A mandatory comma separates the user code from the account number.

The following examples show how to code job names, account numbers and user codes:

```
/b̸JOBb̸CST13710, T134276, SMITH
/b̸JOBbTIMEANAL31072, $REMC040, BARTON     b̸ = mandatory blanks
/b̸JOBb̸$1310AB$ESTTIME, 12345, 98765           that must be coded.
/b̸JOBb̸JOBNAMEJOE, ACCTNUMBER, USER001
```

2-6 CATEGORY (CAT)

The category parameter, identified by the keyword CAT, specifies the priority of service and the execution characteristics of the job.

Job execution priority and characteristics specify when a job should be scheduled for execution. The installation defines the values that specify priorities and job types at system generation. The Operating System uses these values to balance the mix of jobs and to schedule job execution.

Format: The category parameter value is an unsigned constant that describes the priority and type of job.

Default: The parameter may be omitted if the installation-defined default values are satisfactory.

The installation determines the job classifications and priority of service within a framework of three types. Any combination of execution functions available in a computer system may be specified. For example, an installation may assign a class (or category) to the following types of jobs:

- I/O-bound jobs
- Computation-bound jobs
- Mixture of these.

The installation may define these jobs in any manner up to a maximum of three. The numbers are restricted by the parameter value (which is less than or equal to 256).

The CAT parameter specifies two related functions:

- Priority of service - level of importance and sequence in which job is executed.
- Execution characteristics - maximum of three job classifications that the installation defines for the Operating System to use in determining maximum utilization of system resources when scheduling job execution.

The following examples show how to code job name, account number, user code and category:

```
/ƀJOBƀCST13710, T134276, SMITH, CAT=3
/ƀJOBƀTIMEANAL31072, $REMC040, BARTON, CAT=72
```

where CAT=3 and CAT=72 mean that the jobs belong to category 3 and 72, respectively, and that these are the installation-defined categories.

2-7 LOCATION (LOC)

The location parameter (LOC) specifies the installation at which the output for the job will be produced, if there is no explicit specification on the JSL print and punch output statement (FOSYS).

Format: The location parameter is an optional, keyword parameter identified by the keyword LOC. The parameter value is a string of one to eight alphanumeric characters. The central installation assigns the location identifiers to the various terminal installations.

Default: The LOC parameter may be omitted. The default is the location where the job was submitted.

The following examples illustrate use of the LOC parameter:

```
/ƀJOBƀCST13710, TI34276, SMITH, CAT=3, LOC=T3300
```

where T3300 is the location identifier of the terminal installation where the output is produced.

```
/ƀJOBƀCST13710, TI34276, SMITH, CAT=3
```

Since the LOC parameter has been omitted, the job's output is produced at the location where the job was submitted.

2-8 OPTION (OPT)

The option parameter (OPT) specifies options used by the Operating System pertaining to job processing upon abnormal termination of a job step, or the restart of an active step after system failure. It can also specify that a PMD is not to be FOSYSed.

Format: The requested options are denoted by the appropriate letters, separated by commas, and the entire parameter enclosed in parentheses. The parameter is identified by the keyword OPT and is optional.

The following options are defined for the JOB statement:

C - Specifies job processing is to continue although the user's CP step indicated by the XQT statement may have terminated abnormally. When specified on the JOB statement this option applies to each step in the job. However, if a loader error occurs, job processing is terminated.

D - Specifies that the PMD, if requested for a step, is not to be automatically FOSYSed. This implies that the user will handle the disposition of the PMD. When present, this option implies the PMD is written to a file with an access name of SYS.PMD and with the following characteristics:

FORG = PS

LREC = 133

BKSZ = 7392

RCFM = FBSA

JSL variable D is set to a 1 when the PMD is taken so the user should avoid use of this variable when using the D option.

R - Specifies to restart an active step after system failure.

Default. Any or all of the options can be omitted. If C is omitted and a loader error occurs, no further JSL statements are processed. If a step terminates abnormally, only FOSYS statements from the remaining JSL statements will be processed. If D is omitted and a PMD is requested it will be automatically FOSYSed. If R is omitted, an active step will not be restarted after system failure. If all the options are omitted the OPT keyword should not be used.

The following examples illustrate the use of the OPT parameter:

```
/ƀJOBƀCST13710, T134276, SMITH, CAT=3, LOC=T3300, OPT=(C, D, R)
/ƀJOBƀCST13710, T134276, SMITH, CAT=3, LOC=T3300, OPT=(C, D)
/ƀJOBƀCST13710, T134276, SMITH, CAT=3, LOC=T3300
```

In the first example all options are specified and each option causes the actions previously defined for each option. In the second example a restart will not be made for the job after system failure. In the third example all defaults, as previously defined, will occur.

2-9 JOB STATEMENT INTERACTIONS

The JOB statement is explicitly interactive with the LIMIT statement (see topic 2-14). The amount of disc reserved on the LIMIT statement is for the job specified by the JOB statement. If no disc space is reserved on LIMIT or there is no LIMIT statement, the Operating System reserves the amount defined at system generation for the entire job. In addition, installation accounting information specified on the JOB statement is overridden by LIMIT statement specifications, for the job block defined by the LIMIT statement.

The JOB statement is implicitly interactive with the EOJ statement. If a previous job has not specified an end-of-job, the JOB statement for the new job defines the end of the old job, as well as initiating the new job. All job termination processing for the previous job is activated when the new job statement is encountered.

The JOB statement must always be used to define a job. It always appears first in the sequence of JSL statements specifying the JOB to be executed.

2-10 EXAMPLE OF JOB STATEMENTS

JOB Statement - All Defaults Operative

```
/ JOB PROJECTX,S13,COMP12
```

Parameter	Interpretation
PROJECTX	Job name
S13	Account number
COMP12	User code

Default Values

CAT - installation-defines values

LOC - location where job was submitted

OPT - job cannot be restarted after a system failure

- if requested, a PMD will be automatically FOSYSed
- upon abnormal termination only. FOSYS statements from remaining JSL will be processed.

2-11 THE EOJ STATEMENT

The EOJ statement defines the end of JSL statements and data associated with a job during job input. EOJ also initiates job termination processing when it is encountered during job execution. The EOJ statement functions at two levels: (1) it defines the end of a job source at the input level and (2) it initiates termination of the job's execution at the processing level.

Normally, a job is ended with an EOJ statement and a 7/9 punched card.

The EOJ is written in the general form:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	EOJ	b̸	None

There are no operands in the EOJ statement. The verb EOJ must appear in the operation field. The statement may be labeled.

The EOJ statement must be coupled with a JOB statement; these two statements define the boundaries of a job. If an EOJ statement is used, it must be the last JSL statement in the sequence of JSL statements specifying a job. If EOJ is omitted, the job is assumed to terminate immediately after a 7/9 punched card and prior to the next JOB statement in the job input stream.

The example shows a labeled and an unlabeled EOJ statement for two jobs:

/b̸EOJ and /T10b̸EOJ

2-12 EOJ STATEMENT INTERACTIONS

The EOJ is interactive with the JOB statement and the JSL control statements. The EOJ statement is used only with a JOB statement.

EOJ communicates with JSL control statements through the label/identifier field. The label provides a name to which the control statements can refer. The label must be a valid ASC symbol.

2-13 JOB BLOCKS

A job may be partitioned into logical sections called blocks. A job block is the basic unit for programmed flow of job execution. Partitioning jobs into blocks allows resource reservation for job blocks that need specific resources at a specific time. This technique improves system efficiency.

Job blocks are defined explicitly through the JSL LIMIT statement. A block begins with a LIMIT statement and ends with the JSL statement immediately preceding the next LIMIT statement or with the end of the job, specified by the EOJ statement (see figure 2-1). The end of the job terminates the last job block defined in the job. If the JOB statement is not followed immediately by a LIMIT statement, the first job block extends to the JSL statement immediately preceding the first LIMIT statement. If a job does not have a LIMIT statement, the JOB statement starts the block and EOJ and/or the 7/9 punched card terminates it.

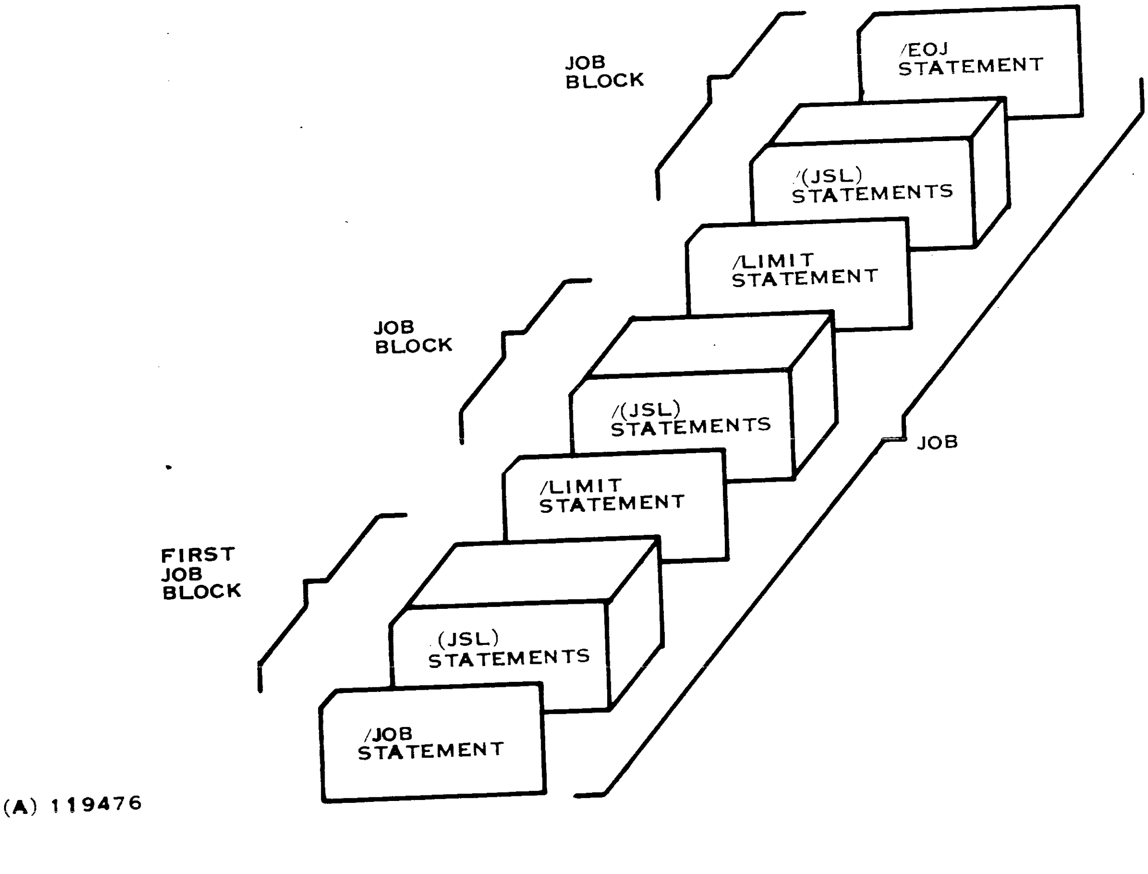


Figure 2-1. Defining Job Blocks in Job Input Stream

The following examples show different job blocks:

```
           / JOB JOBNAME, ACCTNUMBER, USERCODE
           / LIMIT
job
block      / (JSL)

           / (JSL)

           / LIMIT ACCT=ACCTNUMBER
job
block      / (JSL)

           / (JSL)

           / LIMIT ACCT=ACCTNUMBER
job
block      / (JSL)

           / EOJ
```

or

```
           / JOB JOBNAME, ACCTNUMBER, USERCODE
job
block      / (JSL)

           / EOJ
```

or

```
           / JOB JOBNAME, ACCTNUMBER, USERCODE
           / LIMIT ACCT=ACCTNUMBER
job
block      / (JSL)

           / EOJ
```

2-14 THE LIMIT STATEMENT

The LIMIT statement defines the beginning of a job block and terminates a preceding job block. LIMIT also specifies the cumulative resource allocation permitted the job steps within the job block it defines.

The general form of the LIMIT statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	LIMIT	b̸	[ACCT=account number][,USCO=user code] [, $\begin{Bmatrix}\text{SECT}\\ \text{BAND}\\ \text{WORD}\\ \text{BYTE}\end{Bmatrix}$ = disc space reservation] [, $\begin{Bmatrix}\text{SEC}\\ \text{MIN}\end{Bmatrix}$ = time]

The verb LIMIT must appear in the operation field. The parameters in the operands field are optional, keyword parameters. They may be coded in any convenient form. The LIMIT statement has no positional, mandatory parameters. The operation and operands fields must be separated by at least one blank.

The LIMIT statement may be labeled if communication with programming flow control statements is desired. The label must be separated from the verb, LIMIT, by at least one blank.

The parameters on LIMIT specify accounting information, catalog user, resource reservation, and maximum CP execution time for the block.

2-15 ACCOUNT NUMBER (ACCT)

The account number parameter, identified by the keyword ACCT, provides accounting information. Each job block may have an account number to which charges for the block are billed. This account number may be different from that specified on the JOB statement or on any other LIMIT statement. If the account number is the same as that on the JOB statement or a previous LIMIT statement, the parameter may be omitted.

Format: The account number may be any combination of one to 16 alphanumeric characters (alphabetic characters, dollar sign character and numerical digits). It is assigned to a customer by installation accounting.

Default: The last account number specified on a JOB statement or a previous LIMIT statement is billed when the account number parameter is omitted. The account number to be billed is the last specified number.

2-16 CATALOG USER (USCO)

The user code identifies a specific user to the system for the job block. User code also specifies qualified catalog access. The user code may be different from that specified on the JOB statement or on any other previous LIMIT statement.

The catalog system maintains a list of user codes, each of which is associated with a list of attributes. The list of codes is used for catalog privacy purposes. The catalog system permits a user to reference a cataloged file if his user code is identified in the access list. Other access attributes associated with the user control code shows how the user accesses the catalog once a reference is successful.

Format: The user code parameter is identified by the tag USCO. The parameter value may be any combination of from one to eight alphanumeric characters (alphabetic, dollar sign character and numerical digits).

Default: The user code last specified on either a previous LIMIT statement or on the JOB statement is used if the USCO parameter is omitted.

The following example illustrates correct coding of account numbers and user codes on LIMIT statements.

```
job
block        / JOB JOBNAME, ACCTN01, USECODE1

           ( / LIMIT ACCT=ACCTN02, USCO=USECODE1
job        <
block      (
           ( / EOJ
```

The next example illustrates a LIMIT statement with the account number and user code defaults operative:

```
/ JOB JOBNAME, ACCTN01, USECODE1

/ LIMIT

/ EOJ
```

where the default values for the job block defined by the LIMIT statement are:

ACCT the account number specified on the JOB statement: ACCTN01

USCO the user code specified on the JOB statement: USECODE1.

2-17 DISC RESERVATION (WORD OR BYTE)

Format: The disc reservation parameter is identified by the keywords SECT, BAND, WORD or BYTE. The parameter value is the maximum amount of disc space allocated to the job block. This value is an unsigned constant of one to six digits.

SECT specifies that the parameter value is expressed in sector units (1 sector=64 words). The amount reserved is in units of 64 sectors each. If the requested number of sectors is not a multiple of 64 sectors, the system rounds the specified number upward to the next higher multiple of 64.

BAND specifies that the parameter value is expressed in band units (1 band= 256 sectors).

WORD specifies that the parameter value is expressed in word units given in thousands (WORD=15 is 15,000 words). When the value specified is not an even multiple of 64 sectors it is rounded up to the next highest multiple.

BYTE specifies that the parameter value is expressed in byte units given in thousands (BYTE=12 is 12,000 bytes). When the value specified is not an even multiple of 64 sectors, it is rounded up to the next highest multiple.

During the processing by the defined job block, the disc space actually allocated to the files (including processing program files) cannot exceed the amount reserved. If it does, the job is terminated.

Default: The parameter is optional. If it is omitted, the Operating System uses the disc reservation most recently specified. If no disc reservation has been previously specified within the job, the Operating System provides a value defined by installation at system generation for the entire job.

The following examples illustrate the disc reservation parameter:

```
                  / JOB JOBNAME, ACCTN01, USECODE1

            (     / LIMIT ACCT=ACCTN02, USCO=USECODE1, SECT=192
    job     )
    block   (     / (JSL)

            (     / LIMIT ACCT=ACCTN02, USCO=USECODE1, BAND=1
    job     )
    block   (     / EOJ
```

The maximum amount of disc reserved for the first job block is 192 sectors. The maximum amount of disc reserved for the second job block is 1 band.

An example with defaults operative is:

```
                  / JOB JOBNAME, ACCTN01, USECODE1

            (     / LIMIT
    job     )
    block   (     / EOJ
```

The defaults are:

ACCT — the account number specified on the JOB statement: ACCTN01

USCO — the user code specified on the JOB statement: USECODE1

Disc Reservation — an installation-defined value.

Multiple LIMIT statements:

```
/ JOB JOBNAME, ACCTN01, USECODE1
/ LIMIT BAND=2

/ (JSL)
/ LIMIT

/ EOJ
```

where the values are

1st LIMIT statement

The maximum amount of reserved disc space is specified as two bands.

Default values:

ACCT — the account number specified on the JOB statement: ACCTN01

USCO — the user code specified on the JOB statement: USECODE1

2nd LIMIT statement

Default values:

ACCT — the account number specified on the JOB statement: ACCTN01

USCO — the user code specified on the JOB statement: USECODE1

disc reservation — the amount specified on the preceding LIMIT statement: two bands.

2-18 CENTRAL PROCESSOR TIME (SEC OR MIN)

The Central Processor time parameter specifies the maximum, cumulative amount of CP execution time allowed for the defined job block.

Format: The parameter is identified by the keywords SEC and MIN. The parameter value is an unsigned constant of one to four digits, not to exceed 30 hours.

SEC specifies that the parameter value is expressed in units of seconds. MIN specifies that the value is expressed in units of minutes. The Operating System terminates the job if the specified execution time is exceeded.

Default: If the CP execution time is omitted, the Operating System imposes an installation-defined amount of time for the block.

SECTION III

PROGRAM PROCESSING

3-1 GENERAL

The JSL XQT statement provides for the execution of any load module. System macros, however, are available to request execution of frequently used load modules such as the Language Translators. In addition, system macros also provide for multiple-step execution such as compile-linkage edit-execute.

3-2 THE XQT STATEMENT

The XQT statement is the basic load module execution statement. It initiates the execution of a load module.

The general form of the XQT statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	ƀ	XQT	ƀ	access name [, LTP = (load time parameters)] [, RTP = (run time parameters)] [, OPT = (options)] [, CPOPT = (CP options)] [, CPTIME = central processor execution time] [, ADDMEM = size] [, CHAR = character strings]

The statement is written with a slash, followed by an optional label (a valid ASC symbol), followed by the verb XQT embedded between blanks, followed by the parameter list. The parameter list begins with a mandatory, positional parameter; the remainder is a list of optional keyword parameters.

3-3 ACCESS NAME

The access name positional parameter on the XQT statement refers to the access name of a direct secondary file containing the load module or it can refer to the member name within a partitioned direct secondary file containing the load module. If the user supplies a partitioned direct secondary file containing the load module, that file must have the access name of JOBLIB. If the user wishes to access a load module in the system partitioned direct secondary file, all that is required is the specification of the member name on the XQT statement.

The load module search order is: 1) non-partitioned direct secondary load module file, 2) user's JOBLIB file, and 3) the system partitioned direct secondary file.

Format: The access name or member name must be a valid ASC symbol. It is positional and must appear as the first operand.

Default: None. Access name or member name must be specified.

3-4 LOAD-TIME PARAMETER (LTP)

The load-time parameters are the load values used to specify memory allocation, number of FORTRAN I/O files required, number of FORTRAN I/O errors allowed before aborting the step and pass initial values to the load module. For Fortran programs, this specification is provided through the Fortran PROGRAM statement. Non-Fortran load modules cannot access load-time parameters. These parameters are unsigned constants for Job Specification Language variables. They are written in a list, separated by commas, and the list must be enclosed in parentheses. The parameters are identified by the keyword LTP. The first two of these parameter values are positional and specify the number of Fortran I/O files required by the step (1-99) and the number of Fortran I/O errors allowed before the step will abort. Following these two positional parameters are the parameter values that correspond to the arguments specified in the Fortran PROGRAM statement. For example, parameter value number three of the LTP set corresponds to argument number one of the Fortran PROGRAM statement, value number four to argument number two, etc.

Default: If the XQT statement does not contain load-time parameters but the load module uses load-time parameters, the load module's load-time parameter table contains whatever was in that area of memory before the load module was loaded. Such load-time parameter values will cause unpredictable results at execution. Default values of zero can be assigned to the load-time parameters by specifying OPT=(Z) on the XQT statement, hence preloading the entire memory load area with zeros. If the XQT statement specifies load-time parameters, but the load module does not have provisions for them, the XQT statement is not executable. Default values for the first two positional values (no LTP option) will cause 99 Fortran I/O files to be allocated for the step and 10 Fortran I/O errors allowed before the step will abort.

3-5 RUN TIME PARAMETERS (RTP)

The run time parameters are the values that the load module obtains when the S$RUNV Service Call is issued.

Format: These parameters are unsigned constants or Job Specification Language variables. They are written in a list, separated by commas, and the list is enclosed in parentheses. The parameters are identified by the keyword RTP.

Default: If the parameter is unspecified, no run time parameters are passed to the program load module. If a load module issues a S$RUNV service call, but run time parameters are not specified in the XQT, the service call returns an error code in its completion code (sixth) parameter, and execution of the load module continues. If the XQT statement specifies run time parameters, but they are never accessed from the load module with a service call, load module execution occurs without reference to the run time parameter values.

3-6 OPTIONS (OPT)

The options specified by this parameter are used by the Operating System to determine whether a dump is to be taken at load module termination, and how the memory of the load module is to be preloaded, to see if processing is to continue upon abnormal termination and whether or not a page load message is desired.

Format: The requested options are denoted by the appropriate letters, separated by commas, and the entire parameter enclosed in parentheses. The parameter is identified by the keyword OPT.

The following options are defined for the XQT statement:

OPTION	FUNCTION
A	Specifies that a post-mortem dump is to be taken on abnormal termination of execution.
B	Specifies that a post-mortem dump is to be taken on normal termination of execution.
C	Specifies job processing is to continue although the step may terminate abnormally.
I	Specifies that the memory load area is to be preloaded with indefinite form data.
P	No protect.

OPTION	FUNCTION
Y	Produces page load message indicating pages of memory in which CP step was loaded.
Z	Specifies that the memory load area is to be preloaded with zeros.

Options I and Z are mutually exclusive; I overrides Z if both are present.

Default: If the option is omitted, the memory load area retains any coincidental data until it is overwritten (I or Z), a post-mortem dump is not taken, (A) (B) a page load message will not appear (Y) only FOSYS statements are processed if step terminates abnormally (C) but if there is a loader error, termination occurs immediately.

3-7 CP OPTIONS (CPOPT)

The options specified by this parameter are obtained by the load module using the S$OPTN Service Call. The option may vary from step to step depending on the particular users requirements and how the options are used in the program.

Format: The requested options are denoted by the appropriate characters (A-Z, $), separated by commas, and the entire parameter is enclosed in parentheses. The options are identified by the keyword CPOPT.

Default: If the parameter is unspecified, a CP options word filled with zeros is passed to the load module. If the XQT statement specifies CP options, but the load module never accesses them, load module execution continues without reference to the XQT statement's CP options.

3-8 CENTRAL PROCESSOR TIME (CPTIME)

The Central Processor time parameter specifies the amount of Central Processor execution time that is allowed for the step.

Format: This parameter is a positive constant which is identified by the keyword CPTIME. The value is expressed in hundredths of seconds, not to exceed 30 hours.

The range of values of the time parameter, t, is: $0 \leq t \leq 2^{24} - 1$.

Default: If the parameter is unspecified, the step is allowed an installation-defined Central Processor execution time.

3-9 ADDITIONAL MEMORY SIZE (ADDMEM)

The additional memory size parameter specifies the number of additional words of central memory that should be reserved for the step. This does not include the load module and the variable dimensioned arrays (if any), but does include space for I/O buffers. In order for the step to get to this memory, it invokes the GETMAIN procedure.

Format: This parameter is a positive constant, a constant followed by K (K denotes thousands), or a JSL variable. The parameter is identified by the keyword ADDMEM. The value is expressed in words.

Default: If this parameter is not specified, the System gives the step an installation-defined default value for additional memory.

3-10 CHARACTER STRING (CHAR)

The character string parameter allows the user to pass arbitrary character strings to a load module. These strings are obtained using the S$CHRV Service Call.

Format: This parameter is a sequence of character strings individually enclosed in quotation marks (") and separated by slashes (/). The sequence is identified by the keyword CHAR.

Default: If this parameter is not present, no character strings are passed to the load module. If the load module issues a S$CHRV service call, but the XQT statement does not specify a character string for it to access, S$CHRV returns an error code in its completion code (sixth) parameter, and execution of the load module continues. If the XQT statement specifies a character string, but the load module fails to access it with a S$CHRV service call, load module execution continues without reference to the XQT-defined string.

3-11 STATEMENT INTERACTIONS

The XQT statement interacts with other JSL statements, particularly in regard to input and output of files. The user must use RENAME statements for these files if their access names are not standard access names. In addition, if more than one step will access files via the standard access names, the user should specify REL statements for these files before the next step needs the standard names. See sections 5-2, 8-2 and 8-4.

SECTION IV

PROGRAM FLOW SPECIFICATION

4-1 GENERAL

The Job Specification Language program flow statements enable the user to direct the flow of execution within a job. These statements allow conditional bypass of statements within a job and non-sequential job termination through conditional and unconditional branching capabilities. Only forward branching is allowed. Control can be transferred to any subsequent labeled statements within a job block or to labeled job blocks.

The JSL statements that enable program flow specification are the IF, GOTO, SET, and PRINT statements. The IF and GOTO statements provide conditional and unconditional branches, respectively, to labeled statements or labeled job blocks. The SET statement provides the means of assigning and modifying the values of job specification variables. The PRINT statement provides a means of determining the current values of the job specification variables at the time the PRINT statement is processed.

4-2 JOB SPECIFICATION VARIABLES

Conditional branches depend on the values of job specification variables (JSL variables). A job specification variable is a value represented in JSL by a symbol. JSL provides 27 JSL variables, each represented symbolically by a single alphabetic character (A through Z, and $).

JSL variables provide communication between steps when the same variable is assigned to several steps as a run time parameter. Assigning the same variable to steps and program flow specification statements allows communication between steps and job control. The value of a variable acquired during processing is available to steps through the S$JSLV and S$RUNV service calls.

The Operating System initializes all 27 JSL variables to zero at the start of each job. Their values can be changed by a step within the S$JSLV service call. The JSL SET statement can also change the contents through the use of the variables. The user can display the values of the JSL variables at any point with the S$JSL PRINT statement, which gives the values of one or more variables on the job activity file.

Each job specification variable is a word of central memory. Its value can be a constant, real, or literal. JSL program flow statements, however, deal with JSL variables as if they were exclusively integers. Non-integer values for a JSL variable must originate in a step and be written to the JSL variable with the JSLV service call. The tests used in the IF conditional branch instruction are based on bit-by-bit logical comparison; hence, if integer and non-integer data are compared, the internal representation of the non-integer data influences the result.

4-3 JOB BLOCKS

Either a LIMIT statement or a JOB statement defines the beginning of a job block. A JSL program flow specification statement can transfer control to any subsequent, labeled JSL statement in the same job block. It cannot refer to any statement, labeled or not, in another job block. Job control references between blocks can only be made through the names (labels) of the blocks specified on their respective defining LIMIT statements.

Job termination can be programmed for any job in which the EOJ statement has a label to which a branch can be made. An EOJ statement might be viewed as a job block that initiates all job termination processing.

4-4 THE SET STATEMENT

The SET statement assigns a value to a JSL variable.

The general form of the SET statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	SET	b̸	variable = [±]n [±m]

The label field begins with a slash, which may be followed by the statement label (any valid ASC symbol, 1 to 8 alphanumeric characters beginning with an alphabetic [A-Z,$] character). The verb SET, preceded and followed by delimiting blanks, must follow the label field. The remaining operand field contains the name of the variable to be assigned a value, an equals sign, and an expression defining the value to be assigned to the variable.

4-5 VARIABLE

The variable parameter specifies a JSL variable and specifies the value to be assigned to that variable. The parameter is in the form of an assignment equation.

Format: The JSL variable (stated first in the field and followed by an equals sign) is one of the alphabetic characters A-Z and $. The equals sign is followed by an expression composed of:

n a signed or unsigned constant
   a signed or unsigned JSL variable
   a signed or unsigned job specification indicator (TERM or STATUS)

m a constant or null.

The value of the JSL variable is limited in range to $-2^{31} \le v \le 2^{31}-1$. The job specification indicators are read-only values. The m component is an optional modifier of the n component. The n is mandatory.

Default: If a JSL variable is not assigned a value by the SET statement or by a step, it has a value of zero.

NOTE

> The specified JSL variable acquires the SET-assigned value upon processing of the SET statement. The variable retains that value for the current job until it is redefined.

The following correctly-coded SET statements, if executed in the order shown, would produce the results indicated in table 4-1.

Table 4-1. SET Statement Values

Statement	Value of			
	A	B	C	D
/LABEL1 ƀSETƀA=1	1	0	0	0
/ƀSETƀB=A+5	1	6	0	0
/LABEL2 ƀSETƀA=A+1	2	6	0	0
/ƀSETƀB=A+5	2	7	0	0
/ƀSETƀC=-A	2	7	-2	0
/ƀSETƀD=-A-5	2	7	-2	-7

4-6 THE IF STATEMENT

The IF statement specifies a conditional forward branch of execution to a specified statement within a job block, or to another job block.

The IF statement is written:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	IF	ƀ	condition, label

The statement's label field begins with a slash, which may or may not be followed by a label. The verb IF, surrounded by single blanks, appears in the operation field. The operand field contains two parameters. The first defines a condition to be tested; the result of this comparison determines whether execution will be transferred. The second parameter is the label of the statement to which execution is transferred if the condition parameter evaluates to TRUE.

4-7 CONDITION

The condition parameter specifies the condition under which transfer of execution is to be made.

Format: The condition parameter is a mandatory positional parameter that must appear as the first operand. If the condition is TRUE, execution transfers to the statement or job block specified by the label. If the condition is FALSE, the next sequential JSL statement in the job input stream is executed.

The condition parameter is written in the format as a.op.b, where a and b can each be one of the following:

a,b	constant, JSL variable, job specification indicator (TERM or STATUS)

Op can be one of six valid relational operators, as shown in Table 4-2.

The value of the relational expression is the logical result of an algebraic comparison of a JSL variable, a constant, or a job specification indicator to another JSL variable, constant, or job specification indicator. If the condition parameter evaluates to TRUE, execution is transferred to the statement specified by the label parameter; if it evaluates to FALSE, execution passes to the next sequential JSL statement in the job input stream.

The condition parameter is restricted to one relational expression; no combination of conditions is permitted.

Default: None.

Table 4-2. IF Statement Relational Operators

Operation	Condition
a.GT.b	TRUE if a is greater than b; FALSE if a is less than or equal to b
a.GE.b	TRUE if a is greater than or equal to b; FALSE if a is less than b
a.EQ.b	TRUE if a is equal to b; FALSE if a is not equal to b
a.NE.b	TRUE if a is not equal to b; FALSE if a is equal to b
a.LE.b	TRUE if a is less than or equal to b; FALSE if a is greater than b
a.LT.b	TRUE if a is less than b; FALSE if a is greater than or equal to b

4-8 LABEL

The label parameter is the label of a JSL statement to which execution is to be transferred when the condition specified by the condition parameter evaluates to TRUE.

Format: The label must be a valid ASC symbol. The label parameter is a positional, mandatory parameter that must appear as the second operand. The label is separated from the condition parameter by a comma.

The label parameter must identify a JSL statement in the same block with the IF statement, a LIMIT statement that defines another job block, or the EOJ statement that terminates the job. A branch cannot be made from a defined job block to an initial default job block initiated by only a JOB statement; however, branches can be made within such a job block and from such a block to any labeled block. If no reachable statement has the label specified in the label parameter, the job terminates, and a message appears in the user's job activity file.

Default: None. A label must be specified.

4-9 THE GOTO STATEMENT

The GOTO statement directs an unconditional forward branch of execution to a specified statement within a job block, or to another job block.

The general form of the GOTO statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	GOTO	b̸	label

The GOTO statement's label field begins with a slash; the slash may be followed by a statement label. The operation field contains the verb GOTO, and is delimited by blanks. The single operand is the label of the statement to which execution is to be transferred when the GOTO statement is processed.

4-10 LABEL

The label parameter specifies the JSL statement to which execution is to be transferred.

Format: The label parameter is a mandatory, positional parameter. The label must be a valid ASC symbol. Only one label can be specified.

The label parameter must point to a JSL statement within the same job block as the GOTO statement, a LIMIT statement that defines another job block, or the EOJ statement that terminates the job. Forward branches can be made within an unnamed job block or they can be made from an unnamed job block to any named job block. If the label parameter points to an unreachable or undefined statement, the job terminates and a message is written on the job activity file.

Default: None. The label must be specified.

4-11 THE PRINT STATEMENT

The PRINT statement specifies that the current values of one or more or all of the 27 JSL variables or the job specification indicators (TERM or STATUS) be written to the system output file. This valuation occurs when the PRINT statement is encountered in the job input stream; the user does not become aware of the results until job termination, when part of the system output file is printed as the job activity file.

The general form of the PRINT statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	PRINT	b̸	[JSLV=variable 1[/variable 2[/... [/variable n]]]] [, MESG=message] [, OPT=(options)]

The PRINT statement begins with a slash and an optional label, followed by PRINT delimited at each end by blanks. PRINT's parameters allow specifying the variables to be printed, specifying a message to be printed with them, and indicating whether the variables are to be printed in decimal or hexadecimal.

4-12 VARIABLE (JSLV)

The variable parameter names the variables whose values are to be printed. It is a list of subparameters separated by slashes, preceded by the tag JSLV.

Format: Each subparameter may be one or all of the JSL variables denoted by the alphabetic characters A-Z and $ or may be one or both of the job specification indicators denoted by the character strings TERM or STATUS. The options parameter can affect which values are printed.

Default: The JSLV parameter is optional. If variables are specified, the values entered will be the values of the variables current at the time the PRINT statement is executed.

4-13 MESSAGE (MESG)

The message parameter specifies a user-supplied message for output in the job activity file, via the system output file.

Format: The message starts immediately after the keyword MESG and an equals sign. It is a string of up to 128 characters. Embedded blanks are dropped unless the string is enclosed in quotation marks.

Default: The message parameter is optional. If the user provides no message on the PRINT statement, no message is placed in the system output file. A message may be specified with or without the JSLV parameter.

4-14 OPTION (OPT)

The option parameter, identified by the tag OPT, enables the user to indicate whether all variables or only specified variables are printed and what notation is used.

Format: The value of the option parameter is either or both of the alphabetic codes A and H. The parameter value must be enclosed in parentheses. If both parameter values are used, they must be separated by a comma and the entire list enclosed in parentheses.

If option A is chosen, the values of all the variables are printed in decimal notation. If option H is chosen, the values of the variables specified in the JSLV parameter are printed in hexadecimal. Option A overrides the JSLV parameter. If both A and H are used, the values of all the variables are printed in hexadecimal notation.

Default: The OPT parameter is optional. If OPT is not used, the default provides that the values of only the variables specified by the JSLV parameter are printed and in decimal notation.

4-15 PARAMETER INTERACTIONS

If no parameters are specified on the PRINT statement, an error message is generated. At least one type of parameter must be used.

The JSLV parameter may be used without a MESG or OPT parameter. The values of the specified variables are printed in decimal notation.

The MESG parameter may be used without a JSLV or OPT parameter. The user-supplied message is written to the system output file.

The OPT=A parameter may be used without a JSLV or MSG parameter. OPT=H requires either a JSLV or OPT=A specification or both. OPT=H cannot appear by itself.

4-16 STATEMENT INTERACTIONS

The IF and GOTO statements may point to other JSL statements within a job block or may point to a LIMIT or EOJ statement. The IF and GOTO may not point to a default job block initiated by a JOB statement.

4-17 EXAMPLES

Directory Job Flow with a Program-Computed Variable

A sample JSL deck appears below. The object of this block of code is to execute program BRANCH, followed by UPDATE or REPORT, the choice depending on the value of JSL variable Z. Program BRANCH computes the value for Z, then uses the S$JSLV service call to store that value in Z.

JSL Statement	Explanation
/ JOB DEMO, 12345, NAME	Initiate job input stream; account number is 12345, usercode is NAME.
/BLOC1 LIMIT	Start first job block.
/ XQT BRANCH	Commence file needed to execute BRANCH (includes passing integral value to JSL variable Z).
/ PRINT JSLV=Z	Output value of Z at this point to system output file for later printing.
/ IF Z.GT.50, BLOC3	If Z is greater than 50, go to BLOC3 and execute REPORT; if less than or equal to 50, go to BLOC2 and execute UPDATE.
/BLOC2 LIMIT	Start second job block.
/ XQT UPDATE	Start file needed to execute UPDATE.
/ GOTO EXIT	Having executed UPDATE, skip execution of REPORT and branch directly to job termination (EOJ).
/BLOC3 LIMIT	Start third job block.
/ XQT REPORT	Start file needed to execute REPORT.
/EXIT EOJ	Terminate job input stream.

SECTION V

FILE DESCRIPTION

5-1 INTRODUCTION

File description is one factor to be considered in specifying input and output of files for the executing job. The FD and START/STOP statements are file description statements describing and defining the logical characteristics of the file identified by an access name. The FD statement pertains to files on the secondary disc, either as a result of creation on or transfer to the disc. The START/STOP statements define files embedded in the job input stream, i.e., card files located in a JSL program deck.

5-2 FILE SPECIFICATION

All CP steps reference files via access names. CP steps using Fortran I/O require files with standard access names for file input and output. Fortran I/O file standard access names are of the form:

FTxxFyyy

Where:

xx - corresponds to the file reference number (logical unit number) that is identified in the Fortran program unit I/O statements.

yyy - corresponds to the file number beginning with 001 and incrementing by 1 after each ENDFILE and successive writing of additional records to the next file of the appropriate logical unit.

The file reference number can be any integer value from 01-99. The Fortran programmer can read or write to any of these files on disc and, using the appropriate JSL FD, I/O, and cataloging statements, handle the files as required by the program unit. The default access name for the old Fortran II, or on-line READ statement, is FT05F001, for the PRINT statement, FT06F001, and for the PUNCH statement, FT07F001. The remainder of the discussion in this paragraph deals with CP steps that use standard access names.

If an XQT statement is used to execute any CP steps using Fortran I/O, the user must rename all his files with standard access names before the next XQT statement, a macro call generating an XQT statement, or a system use of standard-named files. An XQT statement executing a load module not using Fortran I/O does not require standard access names for its I/O files. The system macros do not require renaming of files with standard access names; they handle this function automatically. Table 5-1 summarizes the standard-named files available to CP steps and shows their characteristics.

SYS.IN is a standard access name not described in table 5-1. It is used to refer to files read in through the job input stream but not assigned an access name. Multiple files may be input in a single job specification, and the user need not give them access names. However, only one name SYS.IN is available to a single job specification, so a bookkeeping procedure is necessary to identify the unnamed file to which SYS.IN refers at any given time.

The user need not worry about SYS.IN and its uses unless he uses his own XQT statements to govern the steps in processing his program or enters two or more unnamed files through the job input stream. In any other case, the user's files are automatically made available to his CP steps.

When the System encounters a START statement that does not specify an access name, it creates a file into which it reads everything it finds after the START statement, until it finds a STOP statement; it assigns the file a default access name. When the System encounters anything other than a JSL statement in the job input stream, it assumes it is dealing with an unnamed input file; it reads the encountered non-JSL record, and everything following it until a JSL statement is found, into a file to which is assigns a default access name. Default access names are of the form

DEF.nnn

where nnn is an integer indicating the order in which the unnamed file identified was encountered in the job input stream. Hence, the first unnamed file found would be given the default access name "DEF.001," the second "DEF.002", and so forth.

Files embedded in the job input stream are removed from it when the System first encounters them. Henceforth, they are referenced by default access names or by user-assigned access names (assigned on START statements). The System replaces unnamed files embedded in the job input stream with JSL statements associating the default access name of the replaced file with the standard access name SYS.IN. The System inserts, before the statement immediately before the file, a RENAME statement replacing the file's default access name with SYS.IN. The System also inserts a statement releasing SYS.IN in the place the unnamed file was embedded. Since statements which reference unnamed files are followed by them in the job stream, and, since statements which reference unnamed files actually refer in default to SYS.IN, the statements inserted accomplish the required association.

To illustrate this process, consider the following job specification. It compiles a Fortran source program and generates a punched object deck.

Table 5-1. Standard-Access-Named Files and Their Characteristics

Standard Access Name	File Name	File Organization	Record Format	Record Length	Block Size		Other Requirements
					Allowed	Default	
SYS. AIN	Assembler Source Input File	Sequential file	Fixed, blocked (FB)	80 bytes	Integral multiple of record length	4,000 bytes (or 50 logical records)	Must exist prior to execution of Assembler
SYS. FIN	Fortran Source Input File	Sequential file	Fixed, blocked (FB)	80 bytes	Integral multiple of record length	4,000 bytes (or 50 logical records)	Must exist prior to execution of compiler
SYS. PRT	Print Output File (used by Compiler, Assembler and Linkage Editor)	Sequential file	Fixed, blocked, using USASI control characters (FBA)	136 bytes	3, 944 bytes (or 29 logical records)	3, 944 bytes (or 29 logical records)	If stacking desired, execute FD statement setting POS= MOD after last file update, before Assembler execution
SYS. USPL	User Source Program Library	Partitioned Direct Secondary file	Fixed, blocked (FB)	80 bytes	Integral multiple of record length (Efficient blocking is 4,000 bytes [or 50 logical records])	None	Must exist prior to execution of Assembler

Table 5-1. Standard-Access-Named Files and Their Characteristics (Continued)

Standard Access Name	File Name	File Organization	Record Format	Record Length	Block Size		Other Requirements
					Allowed	Default	
SYS. OMOD	Object Module File (output by Assembler, or Compiler; input to Linkage Editor)	Sequential file	Fixed, blocked (FB)	80 bytes	4,000 bytes (or 50 logical records)	4,000 bytes (or 50 logical records)	If stacking desired, execute FD statement setting POS= MOD after last file update, before execution
SYS. LEIN	Linkage Editor Control Input File	Sequential file	Fixed, blocked (FB)	80 bytes	Integral multiple of record length	4,000 bytes (or 50 logical records)	Must exist prior to execution of Linkage Editor
SYS. OLIB	System Object Library	Partitioned Direct Secondary file	Fixed, blocked (FB)	80 bytes	Integral multiple of record length (Efficient blocking is 4,000 bytes [or 50 logical records])	None	User libraries may also be used. They must have the same characteristics as SYS. OLIB. User libraries are searched before SYS. OLIB.

Table 5-1. Standard-Access-Named Files and Their Characteristics (Continued)

Standard Access Name	File Name	File Organization	Record Format	Record Length	Block Size		Other Requirements
					Allowed	Default	
SYS. LMOD	Linkage Edited Load Module	Direct Secondary file (DS)	Record, block organization not maintained in direct secondary files.				None

```
/ JOB FORTJOB, 12345, USERNAME
/ FTN OBJ=OUTDEK
        PROGRAM SIMPLE
100     FORMAT (1X, I3)
        INT = 50
        WRITE (6, 100) INT
        END
/ FOSYS OUTDEK, TYPE=PUNCH, OPT=(R)
/ EOJ
```

The CP step here is generated by the macro FTN. For the moment, consider only that it is a macro which elicits code to compile a Fortran source deck, that "OBJ=OUTDEK" identifies the object code file, and that the required input file specification defaults to the unnamed file following the FTN statement in the job input stream.

When the Job Input Stream Processor works on this file, it removes the records

```
        PROGRAM SIMPLE
100     FORMAT (1X, I3)
        INT = 50
        WRITE (6, 100) INT
        END
```

placing them in a file with a default access name of DEF.001. In their place, it inserts the required RENAME and REL statements. Macro expansion is also accomplished at this time.

5-3 THE FD STATEMENT

The FD statement, File Description, describes the characteristics of a job local file. It defines logical file characteristics during execution of the CP step and the characteristics of the file being created on or transferred to disc. More than one FD statement may be used to redefine parameter values before the CP step is executed. The FD statement also supplies the secondary disc allocation for the file when the installation default, the value supplied by a job block, or the catalog entry value are not appropriate. The FD statement also specifies which device the file is allocated to.

The characteristics of the file and the access method must be defined before any input/output requests are made to the secondary disc file by the CP step. The FD statement parameters are used during the merge actions which occur when the CP step opens the file. Merging allows parameters from various sources, i.e., File Control Block (FCB), FD statement and File Information Block (FIB), to be combined to produce a File Control Block which fully defines the file characteristics and any user processing options.

The parameter values of each FD statement are placed in a File Descriptor Block (FDB). Except in the case of concatenation, only one File Descriptor Block is generated for each access name within a job. Subsequent FD statements in the job's JSL input stream referencing the same access name, redefine the values for each parameter that is coded. Thus, the user can modify file characteristics from step to step with JSL.

The general form of the FD statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	FD	b̸	access name $[,\begin{Bmatrix}\text{SECT}\\\text{BAND}\\\text{WORD}\\\text{BYTE}\end{Bmatrix} = \text{initial/maximum/increment}]$ $[,\text{FORG} = \begin{Bmatrix}\text{PS}\\\text{DS}\end{Bmatrix}]$ [, BKSZ = number] [, LREC = number] $[,\text{EROP} = \begin{Bmatrix}\text{ACC}\\\text{SKP}\\\text{ABE}\end{Bmatrix}]$ $[,\text{RCFM} = \begin{Bmatrix}\begin{matrix}\text{F}\\ \\ \text{V}\end{matrix}\begin{bmatrix}\text{B}\\\text{BS}\\\text{S}\end{bmatrix}\left[\begin{Bmatrix}\text{A}\\\text{M}\end{Bmatrix}\right]\\ \text{U}\begin{bmatrix}\text{A}\\\text{M}\end{bmatrix}\end{Bmatrix}]$ [, BFTK = S] $[,\text{BFAL} = \begin{Bmatrix}\text{F}\\\text{D}\\\text{O}\end{Bmatrix}]$ [, BUFL = number] [, BUFN = number] $[,\text{POS} = \begin{Bmatrix}\text{NEW}\\\text{MOD}\end{Bmatrix}]$ $[,\ \text{DTYP} = \begin{Bmatrix}\underline{\text{DSEC}}\\\text{HPT}\\\text{PAD}\\\text{OPID}\end{Bmatrix}]\ /[\begin{Bmatrix}\underline{\text{NOCNTG}}\\\text{CNTG}\end{Bmatrix}]$ /[operator id] [, SETUP = set up id]

Since the merging actions of the file opening processes use parameters from sources other than the FD statement to complete the File Control Block, the FD statement may not be required to build the FCB for all files. Additionally, since other File Management statements also pertain to the file transfer between peripheral devices and the secondary disc, the FD statement may not be required to accomplish the disc file creation. Therefore, only under certain circumstances is the FD statement needed.

When the FD statement is used, the location of the FD statement in the job input stream of JSL statements determines when the File Definition Block values are initially set, or updated. Therefore, the FD statement must precede any other JSL statement which is dependent upon the values within the File Definition Block. This is particularly important when the file will be opened during the execution of a particular job step with the intention of defining certain File Control Block parameters through the FD parameter values during the OPEN merge.

The FD statement contains both positional and keyword parameters. The positional parameter is mandatory and must appear in the first position. The keyword parameters are optional and may be coded in any convenient order. The FD statement may be labeled if communication with the JSL control statements is desired.

5-4 ACCESS NAME

The file must be named to be accessible to the job. JSL statements referencing the same file within the job use the specified access name.

Format: The access name is mandatory and must appear first in the operands field. Access name is a valid ASC symbol assigned by the user.

Default: None. Access name must be specified.

The following is an example of naming a file:

```
/ɓFDɓMAINFILE                ɓ = mandatory blanks that must be coded.
```

The FD statement may be labeled:

```
/FL20ɓFDɓMAINFILE
```

5-5 DISC SPACE RESERVATION (SECT, BAND, WORD OR BYTE)

The disc space parameter specifies the management of disc space for the named file throughout the job. This parameter is optional.

Format: The disc space parameter is identified by one of the keywords SECT, BAND, WORD or BYTE. The parameter consists of three subparameters separated by slashes. Each subparameter value is an unsigned constant. The subparameters within the parameter are positional. They must be coded in the order: initial/maximum/increment.

SECT specifies the subparameters in sector units (64 words each). BAND specifies the subparameters in band units (256 sectors each). WORD specifies that the parameter value is expressed in word units given in thousands (WORD = 15 is 15,000 words). BYTE specifies that the parameter value is expressed in byte units given in thousands (BYTE = 12 is 12,000 bytes). When the value specified (WORD or BYTE) is not an even multiple of 64 sectors it is rounded up to the next even multiple.

Subparameter Set: The first subparameter, initial, specifies the initial amount of disc space allocated to the file.

The second subparameter, maximum, specifies the maximum amount of disc space that can be allocated to the file at any time during the job. The amount that can be specified by this subparameter cannot exceed the space allocated in the LIMIT statement or, by LIMIT's default, the installation-defined value. The LIMIT statement identifies the job block in which the file is defined.

The third subparameter, increment, specifies the amount of space to be added to the file's initial allocation whenever a write request to the file causes the file to use more than the initially allocated space.

Limits: The minimum disc space that can be allocated is 64 sectors (1/4 band). In this case, the amount must be coded in sector units, word units or byte units.

The maximum amount of space that can be allocated is the amount specified in the LIMIT statement, or, by LIMIT's default, is the installation-defined value.

Sector allocations specified by SECT are in units of 64 sectors for each subparameter. If any of the three subparameters is not a multiple of 64, the system rounds the value upward to the next higher multiple of 64.

Default: The entire set of subparameters must be coded or none coded. The entire set must be omitted if all disc space defaults are acceptable.

The initial amount of space allocated is not incremented if the maximum allocation is exceeded.

If a cataloged file is referenced, the default value is the file size specified in the catalog.

Examples: The following examples show a tape file with disc space allocated in sector units and band units, respectively:

```
/bFDbMAINFILE, SECT = 128/192/64
```

The file named MAINFILE will use initial disc space of 128 sectors (2 x 64 sectors). It cannot exceed 192 sectors (3 x 64 sectors) during execution of the job. If a write to the file causes the file to exceed the 128 sectors initially allocated, the file space can be incremented by 64 (1 x 64 sectors) sectors, but cannot exceed the maximum of 192 sectors.

/ƀFDƀMAINFILE, BAND = 1/2/1

The file named MAINFILE will use initial disc space of one band (1 band = 256 sectors). It cannot use more than two bands (512 sectors) during execution of the job. If a write to the file causes the file to use more than one band initially allocated, the space can be incremented by one band, but cannot exceed the maximum of two bands.

5-6 FILE ORGANIZATION (FORG)

The file organization parameter specifies the organization of the file on the secondary disc.

Format: The parameter is identified by the tag FORG. It consists of one subparameter selected from four keywords: PS and DS.

PS - the file on disc is organized sequentially

DS - the file on disc is organized as direct secondary

Default: If the file organization parameter is omitted and the value is not supplied from any other source, the system does not supply default values.

5-7 BLOCK SIZE (BKSZ)

The block size parameter specifies the maximum number of bytes in a block within the file.

Format: The parameter is identified by the keyword BKSZ and must be coded as shown. The value of the parameter is an unsigned constant.

Restrictions: (a) Direct Secondary Files FORG - all input files on tape must have a block size in multiple of sectors except for the last block. (b) Sequential FORG - block size must be specified for unlabeled input tapes.

Specifications: Block length is utilized in the user's opening merge during CP execution for files with a block organization on disc. For unlabeled tape files, the length is used to input the tape to the disc.

Limits: The subparameter value must be within the range, $1 < b < 2^{32}-1$.

Default: If block size is omitted and the value is not supplied from any other source (e.g., FCB procedure or tape labels), the system does not supply default values. Exception: Direct Secondary files are defaulted to 4096 for tape output.

5-8 RECORD LENGTH (LREC)

The record length parameter specifies the maximum length, in bytes, of any logical record in the file.

Format: The parameter is identified by the keyword LREC. The keyword LREC must be coded as shown. The value of the parameter is an unsigned constant.

Restrictions: Record length is valid only for those file organizations that support a record structure on the disc.

Specifications: Record length is used in the user's opening merge during CP execution.

Limits: The parameter value must be within the range, $0 < r < 2^{16}-1$.

Default: If the record length parameter is omitted and the value is not supplied from any other source, e.g., FCB procedure or tape labels, the system does not supply default values.

5-9 RECORD FORMAT (RCFM)

The record format parameter specifies the format of the logical records within the file on disc.

Format: The parameter is identified by the keyword RCFM and must be coded as shown. The parameter contains one to three subparameters with no separators. Each subparameter is a keyletter. The subparameters are optional. They must be coded in the order shown if specified. Omit those subparameters not required. It is not necessary to indicate omitted subparameters.

Restrictions: Record format is utilized during the user's opening merge during CP execution for those disc files with a record structure. The allowable combinations of subparameters follows:

$$F\left[\begin{Bmatrix}B\\ BS\\ S\end{Bmatrix}\right]\ \begin{matrix}[m]\\ [A]\end{matrix}\qquad V\left[\begin{Bmatrix}B\\ BS\\ S\end{Bmatrix}\right]\ \begin{matrix}[m]\\ [A]\end{matrix}\qquad \begin{matrix}[m]\\ U[A]\end{matrix}$$

Specifications: The meaning of these subparameters are:

Character	Meaning
F	The records are of fixed length.
V	The records are of variable length.
U	The records of undefined length.
B	The records are blocked.
S	For fixed-length records, the records are written as standard blocks, i.e., no truncated blocks or unfilled tracks within the file, with the exception of the last block or track. For variable-length records, a record may span more than one block.

Character	Meaning
A	The records contain USASI control characters. This subparameter specifies that the first character in each record is a carriage control character.
M	The records contain machine control characters. The subparameter specifies that the first character in each record is a carriage control character.

NOTE

See File Management Guide for a more detailed discussion.

The record format parameter may also define the records of the file on a tape volume if this information is not available to the system from any other source. This condition occurs for unlabeled tape input files only.

The control character subparameter is valid for print files only.

Default: If the record format parameter is omitted and the information is not supplied by any other source, the system supplies no defaults.

The subparameter defaults, when the record format parameter is coded, are supplied as:

Length: invalid if not coded

Blocking: no blocking

Structure: no spanning, not standard, as applicable

Control Character: none present.

Examples: The following example illustrates the named tape file that is input to the secondary disc. The amount of disc allocated is expressed in sector units.

```
/b̸FDb̸MAINFILE,SECT=128/192/64,FORG=PS,BKSZ=64,LREC=16,;
RCFM=FBA
```

The file name MAINFILE initially uses 128 sectors of disc. The space may be incremented by 64 sectors, not to exceed a maximum of 192 sectors. The file on disc is sequentially organized. The file records, 16 bytes in length, are in blocks 64 bytes in length. The records are fixed length, blocked, and contain ANSI control characters in the first character of each record.

5-10 BUFFERING TYPE (BFTK)

The buffering type parameter, identified by the keyword BFTK, defines the type of buffering to be used during CP step use of the file.

Format: The parameter is the single keyletter S, which denotes simple buffering. Only simple buffering is supported.

Restrictions: The BFTK parameter does not affect the transfer of a file to the disc, nor the structure of the file on the disc. In addition, BFTK is not applicable for processing direct secondary organized files.

Default: The parameter may be omitted. If so, no buffer type is specified. Buffer management depends on the methods that the executing CP step uses to access the file.

5-11 BUFFER ALIGNMENT (BFAL)

The buffer alignment parameter, identified by the keyword BFAL, defines the alignment desired for each buffer used by the executing CP step use of the file.

Format: The parameter value is one of the single keyletters F, D, or O. F denotes fullword buffer alignment; D, doubleword alignment; and O, octet alignment.

Restrictions: The BFAL parameter does not affect the transfer of a file to disc, nor the structure of the file on the disc. In addition, BFAL is not applicable for processing direct secondary organized files.

Default: The parameter may be omitted. If so, no buffer alignment specification is made. Buffer management depends on the methods that the executing CP step uses to access the file.

5-12 BUFFER LENGTH (BUFL)

The buffer length parameter, identified by the keyword BUFL, defines the length of each buffer used by the executing CP step.

Format: The parameter value is an unsigned constant. The range is $1 \leq n \leq 2^{16}-1$ bytes, where n is the number. The buffer length specified must be large enough to contain the maximum block length of the file being processed.

Restriction: The BUFL parameter does not affect the transfer of a file to disc, nor the structure of the file on the disc. In addition, BUFL is not applicable for processing direct secondary organized files.

Default: The parameter may be omitted. If so, no buffer length is specified. Buffer management depends on the methods that the executing CP step uses to access the file.

5-13 BUFFER NUMBER (BUFN)

The buffer number parameter specifies the number of buffers desired for processing the file during CP execution.

Format: The parameter is identified by the keyword BUFN. Its value is an unsigned constant that must be in the range $1 \leq n \leq 255$, where n is the number.

Restriction: The BUFN parameter does not affect the transfer of a file to disc, nor the structure of the file on the disc. In addition, BUFN is not applicable for processing direct secondary organized files.

Default: The parameter may be omitted. If so, no buffers are reserved for CP use of the named file. Buffer management depends on the access methods used by the executing CP step in using the file.

5-14 ERROR OPTION (EROP)

For certain access methods, the Operating System must be told what action to take if an uncorrectable I/O error occurs; for other access methods, the Operating System automatically performs error processing actions as prescribed by the access method. The FD statement provides this information for files being accessed through QSAM techniques.

Format: The error option parameter, identified by the keyword EROP, specifies the action required of the Operating System if an uncorrectable I/O error occurs. The parameter value is one of three keywords: ACC, SKP, or ABE. ACC instructs the Operating System to accept the data block causing the error and to continue processing. SKP instructs the Operating System to skip the error-causing data block and continue; ABE specifies termination of the step processing the error-causing data block.

Default: The error option parameter may be omitted. If the parameter is not used, any processing option by the Operating System should be specified by another source. The System does not supply a default value when processing options are not supplied.

5-15 POSITION (POS)

When the executing CP step opens a specified file that exists on magnetic tape, it must know the position of the tape before it can access the proper file. Files with standard labels convey the information through their labels. Files with no labels do not contain this information; thus, the position parameter on the FD statement must be used to transmit this information. If file positioning has been specified previously, the parameter changes the previous specification.

Format: The position parameter specifies the initial position of a tape file when an executing CP step opens the file. The parameter is identified by the tag POS. Its value is one of two keywords -- NEW or MOD.

NEW specifies that the file is positioned at its beginning. This condition allows the user to overwrite existing information. MOD specifies that file positioning is at the end of the file to be modified. This condition enables the executing CP step to add data to the file without destroying existing information.

It is implicit that the position parameter applies only to sequential files.

Default: The position parameter is optional and may be omitted. If the parameter is not specified, the Operating System assumes that a previously specified condition does not change. If the POS parameter has not been specified previously, the system assumes NEW as the default value.

Examples: In using the POS parameter, consider the following example:

```
/bFDbMAINFILE,SECT=128/192/64,POS=MOD
```

The file named MAINFILE initially will use 128 sectors of disc. The file cannot exceed 192 sectors, but can use an additional 64 sectors (beyond 128) if writes to the file during CP step use of the file cause the file to exceed the initial allocation. The file will be positioned to the end of file when the executing CP step accesses it.

If the position parameter is defaulted, then

```
/bFDbMAINFILE,SECT=128/192/64
```

The Operating System positions the file according to two criteria:

- If the file position has been specified previously, the Operating System assumes the former condition and positions the file accordingly.
- If the file position has not been specified previously, Operating System assumes a value of NEW and positions the file to the beginning of the specified file. Any writes to the file will destroy existing information.

5-16 DEVICE TYPE (DTYP)

The device type parameter is identified by the keyword DTYP and specifies if the file is to be allocated on the head-per-track disc, position-arm-disc, or the device specified in the Operating System default parameter table, or a specific channel or module specified by the third subparameter. It also indicates whether or not the file can be allocated without regard to physical contiguity on disc.

Format: The parameter value is a set of three subparameters separated by slashes. The first subparameter value pertains to file allocation and is one of the keywords DSEC, HPT, PAD, or OPID. The meaning of these subparameters is:

DSEC - allocate the file on the device specified in the system default parameter file (default table)

HPT - allocate the file on the head-per-track disc

PAD - allocate the file on the Positioning-arm-disc

OPID - allocate the file on the specific channel or module given by the third subparameter.

The second subparameter pertains to physical contiguity of the file on disc and its value is one of the keywords NOCNTG, or CNTG where:

NOCNTG - file can be allocated without regard to contiguity. This is the default value.

CNTG - File must be allocated on physically contiguous disc; if the file is greater than or equal to a module; allocation on each module will be contiguous and the allocation will start at the beginning of each module.

The third subparameter is used in conjunction with the OPID keyword of the first subparameter and must be specified if OPID is used. The operator ID is a character string indicating a specific device related channel or module.

Examples:

"DM0" indicates H/T module 0, channel 0

"DCH0" indicates H/T channel 0

"PAD0" indicates PAD spindle 0, 980-0, and TCC-0

"PCH0" indicates 980-0, and TCC-0.

Default: The DTYP parameter is optional. If it is omitted, the Operating System assumes DSEC and NOCNTG, which indicates the file will be allocated on the device specified in the system default parameter table and the file may be allocated without regard to contiguity.

Restrictions: If the OPID keyword is used as the first subparameter, the third subparameter, Operator ID, must be specified. Otherwise the third subparameter must not be specified.

Examples:

```
DTYP=OPID/CNTG/DM0

DTYP=HPT

DTYP=/CNTG

DTYP=OPID//DCH0
```

In the first example the file will be contiguous on the head-per-track disc module 0. In the second example the file will not necessarily be contiguous on the head-per-track disc. In the third example the file will be contiguous and on the device specified in the system default parameter table. In the fourth example the file will not necessarily be contiguous on the head-per-track disc channel 0.

5-17 SETUP IDENTIFIER (SETUP)

The SETUP parameter allows reference to a SETUP statement through use of the SETUP statement label as the SETUP parameter value. This parameter is an optional parameter identified by the keyword SETUP. The file defined by the FD statement will be allocated on the same channel as was selected by the SETUP statement. Any other location information, such as DTYP, specified on the FD statement is overridden when the SETUP parameter is present.

Format: The parameter is identified by the keyword SETUP and its value must be a valid ASC symbol that represents the SETUP statement label.

Default: The parameter can be omitted. There is no default value for this parameter. When omitted the location of the file is determined by the DTYP parameter or by default values.

5-18 LOG OF EVENTS (LOE)

The log of events parameter, identified by the keyword LOE, specifies the access name of a file to be used as log-of-error events for the file specified.

Format: The parameter value for the keyword LOE is a valid ASC symbol designating the access name of the user file that is used as a log-of-error events for the file specified.

Default: The parameter is optional and if omitted, a log is not compiled and as a result termination of file transfer can result upon error detection.

5-19 PARAMETER INTERACTIONS

The file organization parameter (FORG) is interactive with the disc file parameter on the ASG and FIT statements, the block length parameter, and the record length parameter on the FD statement. The FORG parameter is used only to specify the file organization for non-cataloged tape input files. Valid combinations are defined below.

File Organization	Disc File	Block Length	Record Length
PS	FULL	(1)	(3)
DS	FULL	(2)	(4)

NOTES

1. Block length parameter is used both for transfer of the external file and for the file's organization.
2. Block length parameter is used only for transfer of the external file.
3. Record length parameter is used only for the file's organization.
4. Record length is not required.

5-20 STATEMENT INTERACTIONS

The FD statement describes a file on the secondary disc, describes input of a file from an unlabeled tape volume, reserves a file access name, changes characteristics of a file and specifies buffering requirements for CP step use. The FD statement should not be used with the START/STOP pair in describing a file embedded in the job input stream. The FD statement is used in the following cases. It is not otherwise required.

Type Of File	FD Statement Function
All	1. The FD statement supplies, or changes from step to step, logical file characteristics.
	2. The FD statement specifies initial disc allocation parameters for the named file when the installation default, the catalog default or the File Control Block values for output files are not appropriate.
Input, non-cataloged tape file with no labels	1. FD supplies the physical block size, record length and record format of the file on tape
	2. If the organization of the file on disc is not to be sequential, FD specifies the file organization desired.
Input, non-cataloged tape file with standard labels.	1. If the organization of the file on disc is not to be sequential, FD specifies the desired organization.

5-21 RESERVING FILE ACCESS NAME FOR CP STEP EXECUTION

FD may be used to reserve an access name for CP step execution. The following conditions must be met to achieve this function:

- FD specifying the access name must be used alone
- No other File Management Input JSL statements may specify the same access name.

The results of reserving an access name are:

- No disc space is allocated for the file when FD is executed
- No transfer of a file on an external device is made to disc
- File is not defined until CP step execution
- Access name is reserved until CP step opens the file referenced by that access name or the file is released through a JSL REL statement (specifying same access name) or is renamed by a JSL RENAME statement
- If a CP step does not open the file, the file reserved by the access name is not available for processing.

5-22 INPUT OF CATALOGED FILES

The FD statement specifies use of cataloged files, both resident and non-resident, in making data available for processing by the CP step. It is applicable for all file organizations. The type of cataloged files and the functions of the FD statement in relation to them can be summarized in the following list:

Type Of File	Function Of FD
CATALOGED, standard labeled tape	1. changes logical characteristics for CP step use of file
	2. changes disc allocation if value in version entry is not appropriate
	3. provides buffering techniques for CP step execution
CATALOGED, unlabeled tape	1. provides logical characteristics of file on secondary disc
	2. provides disc allocation for file if default values are unacceptable
	3. provides buffering techniques for CP step execution

Type Of File	Function Of FD
	4. provides logical characteristics for CP step use of file
	5. provides organization of file
CATALOGED, disc resident	1. changes logical characteristics for CP step use of file
	2. provides buffering techniques for CP step execution

The FD statement interacts with the JSL ASG statement in use of cataloged files. FD provides disc space management and interfaces with CP step processing of a file; ASG handles the physical transfer of the file. The catalog information describes the file on the secondary disc. FD describes nonresident, unlabeled tape files when they are transferred to the disc. Again, ASG handles the actual transfer.

The position of the FD statement in the job input stream in relation to the ASG statement determines how FD functions (see table 5-2).

5-23 INPUT OF NON-CATALOGED FILES - VIA FIT STATEMENT

FD describes logical characteristics of a non-cataloged file transferred to disc from a non-labeled tape volume. FD also may change the characteristics specified in the tape label of standard-labeled tapes for CP use of file. It is applicable for all file organizations. FD specifies the disc space allocation required to contain the incoming file if default values are not acceptable. FD changes file organization if default of sequential is not acceptable for processing of the file. The types of non-cataloged files and the functions of the FD statement in relation to them can be summarized in the following list.

Type of File	Functions of FD
Standard-labeled tape file	1. FD specifies the amount of disc space needed to contain incoming file if defaults are not satisfactory.
	2. FD changes logical characteristics for CP step use of file.
	3. FD provides buffering techniques for CP step execution.
Unlabeled tape file	1. FD specifies the amount of disc space needed to contain incoming file if system defaults are unsatisfactory.

Type of File	Functions of FD
	2. FD provides logical characteristics and file organization of file on disc and for CP step use of file.
	3. FD provides buffering techniques for CP step execution.

The FD statement interacts with the JSL FIT statement during input of non-cataloged files. FD provides disc space management, interfaces with CP step processing of a file and specifies logical characteristics and file organization for unlabeled tape files. FIT is the peripheral utility that handles the physical transfer of the tape file to disc. For standard-labeled tape files, the characteristics of the file on disc come from the tape label. The relative placements of these statements in the job input stream and the functions of the FD statement under these conditions are summarized in table 5-3.

5-24 OUTPUT OF NON-CATALOGED FILES

The FD statement may be used in conjunction with the JSL FOT and FOSYS statements to provide logical characteristics not provided by any other source (e.g., as a result of a file OPEN). These characteristics are placed in the File Descriptor Block (FDB) to be used when the file is output. The FD should be used with caution when a file is created with certain logical characteristics. Therefore, the FD statement, if used, must specify the same logical characteristics as already exist in the files information block.

Table 5-2. FD and ASG Placement in Job Input Stream

Type of File	Position of FD and ASG Statements	Function of FD Statement	Function of ASG Statement
Non-resident standard labeled tapes	FD precedes ASG	Changes disc allocation if system default value is unacceptable and logical characteristics of the file for CP step use. Block size (BKSZ) parameter must agree with catalog entry for sequential files. File organization (FORG) parameter must agree with catalog entry on all files. Catalog entry supplies logical characteristics for file on disc.	Transfers file to secondary disc.
	ASG precedes FD	Same as above, except space allocation cannot be changed from that in catalog entry.	Transfers file to secondary disc.

Table 5-2. FD and ASG Placement in Job Input Stream (Continued)

Type of File	Position of FD and ASG Statements	Function of FD Statement	Function of ASG Statement
Unlabeled tapes	FD precedes ASG	Supplies logical characteristics of file on disc, disc allocation if system default value is unacceptable and logical characteristics for CP step use of file.	Specifies transfer of file to disc.
	ASG cannot precede FD since FD supplies logical characteristics	Changes logical characteristics for CP step use of file.	
Resident files	FD precedes ASG	Changes file organization and logical characteristics for CP step use of file. No disc allocation is required.	Specifies disc copy of file; catalog entry provides logical characteristics.
	ASG precedes FD	Same as above.	Same as above.

Table 5-3. FD and FIT Placement in Job Input Stream

Type of File	Position of FD And FIT Statements	Function Of FD Statement	Function Of FIT Statement
Standard-labeled tape file	FD precedes FIT	Changes logical characteristics for CP step use of file. Changes file organization if default of sequential is not acceptable. Supplies buffering techniques for CP step execution. Specifies disc space allocation if system default value is not acceptable. Specifies system error processing options.	Transfers tape file to secondary disc. Logical characteristics for file on disc are made available from tape label.
	FIT precedes FD	Changes logical characteristics for CP step use of file. Supplies buffering techniques for CP step execution.	Transfers tape file to secondary disc. Logical characteristics for file on disc are made available from tape label.

Table 5-3. FD and FIT Placement in Job Input Stream (Continued)

Type of File	Position of FD And FIT Statements	Function Of FD Statement	Function Of FIT Statement
		Changes file organization if default of sequential is not acceptable	Disc allocation is defaulted to system-defined value. File organization is defaulted to sequential.
	FD not specified; FIT used alone	Not applicable.	FORG default is sequential. Disc allocation is system default value. Tape file transferred to disc.
Unlabeled tape file	FD precedes FIT (at least one FD must precede FIT)	Supplies logical characteristics of file on disc and CP step use of file. Changes file organization if default of sequential is not acceptable. Supplies buffering techniques for CP step execution.	Transfers tape file to disc.

Table 5-3. FD and FIT Placement in Job Input Stream (Continued)

Type of File	Position of FD And FIT Statements	Function Of FD Statement	Function Of FIT Statement
		Specifies system error processsing options. Specifies disc space allocation if default value is not acceptable.	
	Subsequent FD following FIT	Changes logical characteristics for CP step use of file.	
	FD not specified preceding FIT	None.	Transfer not made.

5-25 START STATEMENT

The START statement initiates the input of a file within the source job specification. The file is embedded in the job input stream. START may specify any source data including a deferred job specification source file. Also, any input card not preceded with a slash is treated as a separate START file.

Files specified by the START statement cannot be nested or overlapped. These files cannot be cataloged files. The files are job local; they are available only to the current job. Their allocated disc space is released during job termination unless previously released through the REL statement.

The system creates a disc file for the file specified by START that has these characteristics:

- File organization is sequential or direct secondary.
- If sequential: record format is fixed-length, blocked standard format and record length is 80 bytes (20 words).

To reference the file in the CP program, the user opens the file using the access name and supplying those File Control Block values required for CP step use of the file. The logical characteristics of the disc file (created by the System) are contained in the disc file's File Information Block and may be merged into the File Control Block when the file is opened.

The general form of the START statement is:

LABEL		OPERATION		OPERANDS
/	b̸	START	b̸	[ACNM = access name] [, BKSZ = number] $\left[,\begin{Bmatrix}\text{SECT}\\\text{BAND}\\\text{WORD}\\\text{BYTE}\end{Bmatrix} = \text{initial/maximum/ increment}\right]$ $\left[, \text{FORG} = \begin{Bmatrix}\underline{\text{PS}}\\\text{DS}\end{Bmatrix}\right]$

START must contain the verb START in the operations field. It cannot be labeled. The parameters in the operands field are optional keyword parameters. The parameters may be coded in any convenient order. The keywords must be coded as shown.

No blanks beyond the first parameter are permitted in the START statement. The first blank encountered after the first parameter terminates processing of the statement.

5-26 ACCESS NAME (ACNM)

The access name parameter, ACNM, specifies the local name by which the file is known to the job.

Format: The parameter is identified by the keyword, ACNM. Access name must be a valid ASC symbol. It cannot be SYS.IN.

Default: The access name parameter is optional. If the parameter is omitted, the file is available only as input to the step defined by the step JSL statement immediately preceding the START statement. The System renames the file SYS.IN prior to execution of the preceding JSL statement. The unnamed file is released after execution of that step and is no longer available to the remainder of the job.

5-27 BLOCK SIZE (BKSZ)

The block size parameter, BKSZ, specifies the maximum number of bytes in a block within the file on disc.

Format: The parameter is identified by the keyword, BKSZ. Its value is an unsigned constant.

Limits: The value of the block size must be within the range: $1 \leq b \leq 2^{16}-1$. The value must be a multiple of the logical record length of 80 bytes.

Default: If the block size parameter is not specified, the default value is an installation defined size.

5-28 DISC RESERVATION (SECT, BAND, WORD OR BYTE)

The disc reservation parameter specifies the management of disc space for the named file for the duration of the job. This parameter is optional.

Format: The disc reservation parameter is identified by one of the keywords SECT, BAND, WORD or BYTE. The parameter contains a set of three subparameters, each separated by a slash. Each subparameter value is an unsigned constant. The subparameters in the set are positional. They must be coded in the order: initial/maximum/increment.

SECT specifies the subparameters in sector units. A sector is 64 words. BAND specifies the subparameters in band units. A band is 256 sectors. WORD specifies the subparameters in word units. BYTE specifies the subparameters in byte units.

Subparameter Set: The first subparameter, initial, specifies the initial amount of disc space allocated to the file.

The second subparameter, maximum, specifies the maximum amount of disc space that can be allocated to the file at any time during the job. The amount that can be specified by this subparameter cannot exceed the space specified in the LIMIT statement or, by LIMIT's default, the installation-defined value. The LIMIT statement identifies the job block in which the file is defined.

The third subparameter, increment, specifies the amount of space to be added to the file's initial allocation whenever a write request to the file causes the file to use more than the initially allocated space.

Specifications: Sixty-four sectors (1/4 band) is the minimum disc space that can be allocated for the file. In this case, the amount must be expressed in sector units, word or byte units.

The maximum allocation is the amount specified in the LIMIT statement, or by LIMIT's default, is an installation-defined value.

Default: If the disc reservation parameter is omitted, the default value is the installation-defined value expressed in sector units.

5-29 FILE ORGANIZATION (FORG)

The file organization parameter, FORG, specifies the organization of the file on the secondary disc.

Format: The parameter is identified by the keyword, FORG. The parameter consists of one subparameter. The subparameter is one of two keywords. The keyword PS specifies the file is sequentially organized. The keyword DS specifies the file is direct secondary organized.

Default: If FORG is omitted, the file is sequentially organized.

5-30 PARAMETER INTERACTIONS

The file organization and block size parameters are interactive.

If FORG = PS, block size (BKSZ) must be coded, unless the default installation-defined value is acceptable. If FORG is omitted, the file organization default is sequential. BKSZ must be coded unless the default value is used.

If FORG = DS, block size (BKSZ) may be used. It is treated only as an OPEN-merge characteristic.

5-31 STATEMENT INTERACTIONS

The START statement may be used in conjunction with a STOP statement; however, a file initiated by a START statement may also be terminated by an EOJ statement or another START statement. Both statements must not be embedded in a JOBX/EOJX statement pair.

The START statement is also interactive with JSL step-execution statements in the case where an access name is not specified. Unnamed embedded files are associated with an immediately preceding execution statement and must follow that statement. Named files may appear anywhere in the job input stream.

The START statement is not interactive with the FD statement. The results are unpredictable if an FD statement specifies the same file as the START statement.

5-32 EXAMPLES

The following examples illustrate how to use the START statement.

1. Job Local File - No Defaults Operative

```
/ JOB JOBNAMEJOE,ACCTNUMBER, USER001
  (JSL)
/ START ACNM=FILEONE, BKSZ=2400, BAND=1/2/1,;
FORG=PS
  (source input data)
/ STOP
  (JSL)
/ EOJ
```

where

ACNM=FILEONE	identifies the file by the name FILEONE and specifies the file as job local
BKSZ=2400	specifies the block size of file on disc as 2400 bytes
BAND=1/2/1	allocates one band of initial disc space that may be incremented by one band not to exceed a maximum of two bands. This parameter specifies the disc space required by the file on disc.
FORG=PS	specifies that the disc file is sequentially organized.

2. Job Local File - All Defaults Operative

```
/ JOB JOBNAMEJOE,ACCTNUMBER, USER001
  (JSL)
/ START ACNM=FILEONE
  (source input data)
/ STOP
  (JSL)
/ EOJ
```

where

ACNM=FILEONE   identifies the file named FILEONE and specifies the file as job local

The default values are

BKSZ = default value of 4000 bytes

SECT
BAND
WORD
BYTE = installation-defined value expressed in sector units

FORG = file organization is sequential

3. Unnamed Source File in Job Input Stream

```
/ JOB JOBNAMEJOE, ACCTNUMBER, USER001
/ START BKSZ=2400, BAND1/2/1, FORG=PS
/ STOP
```

The absence of the ACNM parameter specifies the file as an unnamed file. Other parameter values are as specified or are defaulted.

4. Job Local File - Direct Secondary Organization

```
/ START ACNM=FILEONE, BAND=1/2/1, FORG=DS
```

BKSZ may be omitted for direct secondary organized files.

5-33 THE STOP STATEMENT

The STOP statement is the ending delimiter of a file contained within the source job specification. STOP is used only in conjunction with the START statement. It always appears at the end of the file.

The general form of the STOP statement is:

LABEL		OPERATION		OPERANDS
/	b̸	STOP	b̸	None

The STOP statement must have the verb STOP in the operations field. It does not have any parameters in the operands field. STOP cannot be labeled.

STOP is coded as:

/b̸STOP where b̸ = a mandatory blank that must be coded.

SECTION VI

FILE INPUT AND OUTPUT

6-1 GENERAL

The basic structure of the file system is independent of the machine and devices so that the user need only be aware of symbolic addresses. All physical addressing of external devices (e.g., tape) is performed by the Operating System.

Before a file can be used within a job, the file must be specified as being available to the job and must be given an access name by which it will be known to that job.

Cataloged files are specified as a pathname to the job and are given an access name with the ASG statement. A detailed discussion of cataloged files and catalog management is presented in Section VII.

A file is a non-cataloged file with respect to a given job if, before the start of that job, the file is contained on some external medium (such as magnetic tape or cards) or is contained in the job specification source file; i.e., is in the job input stream.

At the end of the job, all non-cataloged files are released and are not recoverable for subsequent jobs unless they have been cataloged or they still exist on an external medium.

Since a program step can only access data on the disc, a non-cataloged file must be transferred to disc before initiation of the step that refers to the file. The file input statement (FIT) initiates the various disc input/output tasks that transfer the file from the external device and specifies the access name the file is to have on disc.

Non-cataloged files contained in the job specification source file are given access names in the operands fields of the START statements that initiate them. A data file in the job specification source file is delimited by a START/STOP statement pair. Note that non-cataloged files of this nature are actually on the disc at the time of initiation of the job, but they are still from the viewpoint of the job - on the logical job input device.

Non-cataloged file transfers between disc and peripheral devices are specified by the file input/output Job Specification Language statements. The devices supported are: card reader, card punch, line printer, and 7 and 9-track half-inch tape drives with 200, 556, 800, or 1600 bits per inch.

The ASC Operating System supports both labeled and unlabeled files for half-inch tape file input/output (either 200, 556, 800, or 1600 bits per inch tapes).

Unlabeled tape files may have either fixed, undefined or variable length records. The user must specify the format of the tape physical records for both input and output of unlabeled files. Unlabeled tapes must have a file mark specifying the end of the data.

Labeled files are those that are provided by the ASC Operating System with standard ASC labels. The label of a labeled file is the first record of the file and specifies the Logical Input/Output characteristics, if any, of the file.

On input, before data is transferred, file marks may be skipped to position the tape to the desired data. When labeled tapes are used, the skipping of file marks produces a skip of an entire file. File mark skips with unlabeled tapes result in the skipping of the specified number of file marks.

On output, files may be added or appended to existing tape files, or may be started on new tapes.

Refer to the START and STOP statements for non-cataloged files included in the job input stream.

The card punch output supports the punching into cards of a file of logical records formatted as card images. Both variable and fixed length images are supported but are truncated or blank-filled (as required) to 80 bytes of data.

Requests to punch a file into cards are not honored at the time of processing of the FOSYS statement but are recorded and are activated during the system output for the specific job at job termination. This aids in identification of output card files.

The printer file output supports the printing of a file of logical records formatted as print line images. Both variable and fixed length images are supported, but are truncated or blank-filled (as required) to 132 data bytes.

Requests to print a file are not honored at the time of processing of the FOSYS statement but are recorded and are activated during the system output for the specific job at job termination. This aids in identification of print files.

6-2 CATALOGED FILE ASSIGNMENT (INPUT)

Before a cataloged file can be accessed by a CP step, it must be made available to the job. This process, known as catalog file assignment, is specified by the JSL ASG statement. Any special conditions for the transfer from an external device to the secondary disc are also specified through the assignment statement.

6-3 THE ASG STATEMENT

The ASG statement performs these functions:

- Specifies the file as cataloged and as input
- Transfers a file from an external device to the secondary disc
- Links the user to the catalog system's security system.

The position of ASG in the job input stream determines when a file is made available. The only stipulation for assigning cataloged files is that the node containing the requested file must exist for the job when the ASG statement is encountered in the job input stream. Any type of cataloged file -- sequential, or direct secondary -- can be assigned through the ASG statement. A file may be assigned several times during a job through multiple ASG statements.

The general form of the ASG statement is:

	LABEL		OPERATION		OPERANDS
/	Symbol	ƀ	ASG	ƀ	access name, pathname [, VERS = version] [, USE = $\left\{\begin{matrix}\underline{\text{EXC}}\\ \text{SHR}\end{matrix}\right\}$]

The verb ASG must appear in the operation field. The operation field must be separated from the operands field with at least one blank. The operands field contains both mandatory positional parameters and optional keyword parameters.

6-4 ACCESS NAME

The requested file must be identified to the job. The access name provides the link with the job.

Format: The access name is a valid ASC symbol. The name is valid only for the duration of the job and originates when the ASG statement is encountered. Thus, the ASG statement specifying the name must precede (in the job input stream) any statements that specify intended use of the file.

Default: None. Access name must appear in the first position in the operands field; it is mandatory and cannot be omitted.

6-5 PATHNAME

The pathname represents the file name in the catalog. The pathname parameter is mandatory and must appear as the second parameter. The pathname is separated from the access name by the comma.

Format: The pathname is a concatenation of from one to ten edgenames, each separated by a slash. The first edgename may be a synonym defined by a PD statement.

All edgenames constituting the pathname must exist in the catalog prior to using the ASG statement. The creator/owner of the catalog supplies the pathname. However, a synonym for the first edgename may be defined by the current user through the PD statement.

Default: None. Pathname must be specified.

6-6 VERSION (VERS)

A cataloged file may have versions at its node. All versions of the file are associated with the same catalog node. The name of a version is the pathname of the file's node with an index which identifies the version. The user may specify that a particular version of a file be assigned at the job. The version parameter specifies which version of the file defined by the pathname is to be assigned.

Format: The version parameter, identified by the keyword VERS, identifies the desired version. The parameter value is either a signed or unsigned decimal integer, or a signed or unsigned JSL variable.

An unsigned parameter value references the absolute version number of the file. The absolute version number is determined modulo N, where N is the maximum number of versions allowed the file. A signed value references a version of the file relative to the flagged version.

Limits: The absolute value of a signed integer, i, must be within the range $0 \leq |i| \leq 63$.

The value of an unsigned integer cannot exceed the maximum number of versions, N, minus one. The range is therefore, $0 \leq i \leq N-1$.

An unsigned JSL variable refers to an absolute version. Its value has the same limits as an unsigned integer; $0 \leq v \leq N-1$.

The maximum number of file versions and the absolute number of the current flagged version are provided by the creator/owner of the catalog.

Default: The version parameter may be omitted. The flagged version of the file is assigned if no specific version is named. The flagged version is generally the version most recently cataloged. If automatic sequencing of versions has been deactivated, the version last numbered at deactivation is the constant flagged version that is assigned.

6-7 FILE USE (USE)

When a file (or one of its versions) is assigned to a job, the user may determine whether to share the disc file with other jobs or reserve it for exclusive use by his job. File use applies only to file sharing at the job level. It does not apply to step level sharing.

Sharing a file is most likely if a file is used for read only purposes. Writing to a file in shared mode produces unpredictable results because users may be updating the file while other users are accessing the data contained in the file. In addition, a resident cataloged file may be unpredictably and permanently affected if used in a shared mode. In these cases, exclusive use is recommended.

Format: The file use parameter, identified by the keyword USE, specifies whether the file is assigned in the shared or exclusive mode. The parameter value is either the keyword SHR or the keyword EXC. SHR specifies shared use; EXC exclusive use.

Default: The parameter may be omitted. Exclusive use, EXC value, is the default.

If the user assigns the same file more than once within the same job under different access names and declares conflicting file uses, the Operating System assumes the more restrictive EXC mode.

6-8 STATEMENT INTERACTIONS

When used in conjunction with the assignment of other cataloged files, the FD statement specifies disc space allocation if the value in the catalog or the system default value is unsatisfactory. In addition, FD may change file logical characteristics for CP step use of the file. FD may also specify buffering techniques required during CP execution and error option processing.

The following examples illustrate the relative positions of the FD and ASG statements in the job input stream when assigning various types of files:

FD preceding ASG

```
/ƀFDƀANAME, BAND=4/4/0
/ƀASGƀANAME, A/B/C/D, USE=SHR
```

The parameter values supply the following information:

ASG	FD
Access name of file is ANAME. Pathname of file is A/B/C/D. File will be shared among jobs.	Access name of file is ANAME. Four bands of disc space are allocated for the file on disc.

ASG preceding FD

```
/b̸ASGb̸ALPHA, A/B/C, USE=SHR
/b̸FDb̸ALPHA, RCFM=FM, BKSZ=1600, LREC=80
```

ASG	FD
Access name of file is ALPHA. Pathname of file is A/B/C. File will be shared among jobs.	Access name of file is ALPHA. Changes logical characteristics of file for CP executing program use of the file.

ASG Without FD

```
/b̸ASGb̸ANAME, A/B/C
```

Access name is ANAME.

Pathname is A/B/C.

Disc allocation sufficient to contain length of file last cataloged - an FD statement is not required.

File is used in exclusive mode by default to EXC of omitted USE parameter.

6-9 THE PD (PATH DEFINITION) STATEMENT

The PD statement defines a synonym for the pathname of a catalog node that exists at the time the synonym is referenced. The function of the PD statement is to permit access to catalog nodes that are greater than ten edgenames deep. An unlimited number of levels can be accessed by defining synonyms in terms of preceding synonyms.

A pathname that defines a synonym does not have to be valid at the time the PD statement is encountered. It can reference nodes that are not currently defined. The only exception is the first edgename; it must either be a currently defined synonym name, or a catalog name which will exist at the time the synonym is used by another JSL statement. However, at the time the synonym is used with other JSL statements, all edgenames must be valid nodes.

The general form of the PD statement is:

	OPERATION		OPERANDS
ƀ	PD	ƀ	synonym, pathname

There cannot be a label in the label field. The verb PD must appear in the operation field. The operation field must be separated from the operands field by at least one blank. The operands field contains the synonym parameter and the pathname parameter.

6-10 SYNONYM PARAMETER

The synonym parameter specifies the symbolic name by which the pathname is to be known to the remainder of the job. See paragraph 7-6.

Format: The synonym is a valid ASC symbol.

Source: The synonym is created by the user and originates with this statement for the remainder of the job.

Restrictions: The synonym is valid only for the current job.

6-11 PATHNAME PARAMETER

The pathname parameter specifies the pathname that is to be given the synonymous name.

Format: The pathname consists of one to ten edgenames catenated by slashes (/). The first, and only the first, edgename in the pathname can be a synonym that has been defined in a preceding PD statement. Otherwise, it must be a catalog name.

Source: The pathname is a valid pathname in a catalog that exists at the time the synonym is referenced.

Restrictions: The edgenames of the pathname must have been created in the catalog by the time the synonym is used in another statement.

If another synonym is used as the first edgename in this pathname, that synonym must have been defined in a preceding PD statement in the current job. The synonym must also be unique with respect to other synonym and catalog names referenced within the job.

Note: Security controls on nodes referenced by the PD statement must be passed by the user at the time the synonym is referenced by another JSL statement. An intermediate node having full reference control requires the user to have the reference attribute at that node. The use of the terminal node and its control state determines any other attribute requirements.

6-12 NON-CATALOGED FILE INPUT FROM TAPE

Peripheral utilities handle the physical transfer of a file from tape to secondary disc. The File In Tape utilities specify input of a non-cataloged tape file when the transfer request is made. There is one File In Tape utility: FIT reads a file from a half-inch magnetic tape and places the file on disc.

6-13 THE FIT STATEMENT

The File In Tape statement (FIT) initiates the utility that reads a non-cataloged file from half-inch magnetic tape and places the file on secondary disc. This statement provides through its parameters necessary information to input a file. The position of the statement in the job input stream indicates when the transfer takes place. All files input under control of the FIT statement are job local files and may be accessed by any step in the job until the file is released.

The general form of the FIT statement is:

	LABEL		OPERATION		OPERANDS
/	Symbol	b̸	FIT	b̸	access name [EFID=tape id1[/tape id2[/... [/tape id n]]] $\left[,\text{LABL=}\left[\frac{1}{\text{position}}\right]\right] \quad \left[/\left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\\ \text{BLP}\end{matrix}\right\}\right]$ $[,\text{RTRY=number}] \quad \left[,\text{PRTY=}\left\{\begin{matrix}\underline{\text{ABE}}\\ \text{ACC}\end{matrix}\right\}\right]$ $\left[,\text{DEN=}\left\{\begin{matrix}1600\\ 800\\ 556\\ 200\end{matrix}\right\}\right]$ $\left[,\text{TRKS=}\left\{\begin{matrix}\underline{9}\\ 7\end{matrix}\right\}\right] \quad \left[,\text{RCTL=}\left\{\begin{matrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{matrix}\right\}\right]$ [, LOC=dest code]

The verb FIT must appear in the operation field.

The operands field contains both positional and keyword parameters. The positional parameters are mandatory and must be coded in the first and second positions. The keyword parameters are optional. They may be coded in any convenient order. The FIT statement may be labeled if communication with JSL control statements is desired.

6-14 ACCESS NAME

The file being transferred to disc must be named so that it is accessible to the job. JSL statements and the CP step reference the file through the specified access name.

Format: The access name is mandatory and must appear first in the operands field. Access name must be a valid ASC symbol. It is assigned to the file by the user. The file retains the access name until the file is renamed through the RENAME statement or released from the job.

Default: None. Access name must be supplied.

Example: The following example illustrates a named file:

```
/bFITbTAPEFILE   initiate the transfer of the tape file named
                 TAPEFILE.
```

The FIT statement may be labeled:

```
/ALPHAbFITbTAPEFILE
```

6-15 EXTERNAL FILE IDENTIFICATION (EFID)

The external file identifiers must be specified so that the system knows the location of the specified file. The external file identification (EFID) parameter supplies the identifiers. This parameter is identified by the keyword EFID.

The identifiers of all tapes containing the file must be declared. They must be listed in the sequence that the files are to be read.

External tape identification labels are assigned to reels by the installation at the time the tapes are submitted or when they are generated. The system informs the user of the external file labels at that time.

Format: The tape identifiers are separated by slashes (/). Each tape identifier is a string of from one to six characters. If the identifier is enclosed in quotation marks, blanks in the string are retained. If the string is not enclosed, embedded blanks are eliminated.

Restriction: When the FIT statement occurs in an MFR/MFRE group, this parameter is overridden by the specifications in the MFR statement that begins the group.

Default: At least one tape identifier must be specified. There is no default for the parameter.

Examples: The following illustrations show the use of the EFID parameter:

JSL	Meaning
/ƀFITƀTAPEFILE, EFID=2340	initiate the transfer of the file named TAPEFILE. The file resides on tape volume 2340.
/ƀFITƀTAPEFIL1, EFID=13/14/17	initiate the transfer of the file named TAPEFIL1. The file resides on three tape volumes identified by the numbers 13, 14 and 17. The files will be input in the order shown.

6-16 LABEL (LABL)

The label parameter, identified by the keyword LABL, has two functions. It describes the type of label on the tape volume and specifies the position of the file to be read.

Format: The parameter consists of two subparameters separated by a slash (/). The subparameters are positional within the label parameter. The first subparameter specifies the desired initial position of the tape file. The label type subparameter describes the type of label on the tape.

The value for the position subparameter is an unsigned constant. Its range is $1 \leq n \leq 65,535$. The default value is $n = 1$; the first file on the tape volume will be read initially. Other files on the volume are then input sequentially.

The value for the label type subparameter is one of two keywords: SL specifies the tape volume has standard labels; NL specifies that the tape volume is unlabeled. The default value is standard label (SL).

The following list illustrates the interaction between the subparameters.

Position	Label Type	Meaning
n	SL	n=file sequence number. The number of physical tape marks skipped during positioning is a function of the standard label's format.
n	NL	n=file sequence number. The number of physical tape marks skipped during positioning is n-1.

If SL is specified explicitly or by default, the System requests that the correct tape volume be supplied. If a tape volume cannot be verified as standard labeled, the operator must cancel the request. If NL is specified and the tape volume cannot be verified as non-labeled, the operator must cancel the request.

Restrictions: If SL is specified explicitly or by default or, if an EFID is specified, the Operating System requests that the correct tape volume be supplied. If a tape volume cannot be verified as standard labeled, the operator must cancel the request.

If NL is specified and the tape cannot be verified as non-labeled, the operator must cancel the request.

When the FIT statement occurs in an MFR/MFRE group, this parameter (including all three subparameters) overrides the specifications in the MFR statement that begins the group.

Default: The entire label parameter may be omitted. If so, the defaults are n=1 and SL. The tape volume will be positioned so that the first file on the table will be input; other files follow sequentially. The tape volume has standard labels.

Either one or both subparameters may be omitted. The defaults are n=1 and SL. When omitting the first subparameter, the slash must be coded in front of the second subparameter. When omitting the second subparameter, the slash is not necessary.

6-17 RETRY (RTRY)

The retry parameter, identified by the keyword RTRY, specifies the number of times the System attempts to read a record from tape if a parity error has occurred. If the number of retries is exceeded, the System takes:corrective action specified by the parity error option parameter.

Format: The parameter value is an unsigned constant representing the number of read retries to be attempted. This value is limited to the range: $0 \leq r \leq 15$.

Default: If the parameter is omitted, the System tries five times to read the record.

6-18 PARITY ERROR OPTION (PRTY)

The parity error option parameter, identified by PRTY, specifies whether the input operation continues if the permissible number of read retries is exceeded.

Format: The parameter value is one of the keyword: ABE or ACC. The keyword, ABE, means that the input operation is aborted if a permanent parity error occurs. The keyword ACC specifies that the input operation continues if the error occurs.

Default: If the parameter is omitted, the default value is ABE; the input operation aborts.

6-19 TAPE DENSITY (DEN)

The tape density parameter, identified by the keyword DEN, specifies the recording density in bits-per-track for the volumes identified by the EFID parameter.

Format: The parameter value is one of the keynumbers 1600, 800, 556 or 200. The key 1600 specifies a tape density of 1600 bpi. The key 800 specifies a tape density of 800 bpi. The densities 200 and 556 are reserved exclusively for 7-track tapes.

Restrictions: When the FIT statement occurs in an MFR/MFRE group, this parameter is overridden by the specification of parameters in the MFR statement that begins the group.

Default: The tape density parameter may be omitted. The default is 1600. If the tape density is other than specified or defaulted, the system aborts the input operation.

6-20 TAPE RECORDING TRACKS (TRKS)

The tape recording track parameter, identified by the keyword TRKS, specifies if the given tape is a 7-track or a 9-track tape.

Format: The parameter value is one of the keynumbers 9 or 7 to designate either the 9-track or the 7-track tape respectively.

Default: The tape recording track parameter can be omitted. The default is to 9-track tape. If the tape recording track parameter specification or default is different than the actual tape configuration the system aborts the input operation.

6-21 RECORDING CONTROL (RCTL)

The recording control parameter allows for the specification of a parity, translation and data conversion mode. This parameter is applicable only if seven-track tape is being processed. It is identified by the keyword RCTL and the keyword values are:

O	- odd parity without translation or conversion
E	- even parity without translation or conversion

T - odd parity with translation but no conversion

C - odd parity with conversion but no translation

ET - even parity with translation but no conversion.

Default: If the RCTL parameter is omitted, and seven-track tape is designated, the default is ET or even parity with translation but no conversion.

Data Translation: Data translation refers to the ability to transform data recorded in BCD interchange to EBCDIC and vice versa. Since each BCD code is a six-bit value, upon input the six bits are translated into the corresponding eight-bit EBCDIC value. For output operations the eight-bit EBCDIC value is translated into its six-bit BCD representation.

Data Conversion: The data conversion option makes it possible to write eight-bit binary data on seven-track tape. For output operations, every three bytes of data are written as four tape characters. Figure 6-1 illustrates this data conversion.

6-22 LOCATION (LOC)

The location parameter, identified by the keyword LOC, specifies the site at which the tape file is to be input.

Format: The location parameter is identified by the keyword LOC and is a string of from one to eight alphanumeric characters. The location identifiers are assigned the various terminal installations by the central site.

Restriction: When the FIT statement occurs in an MFR/MFRE group, this parameter is overridden by the specification in the MFR statement that begins the group.

Default: If a location is not specified, the default value is the identifier of the central site.

6-23 PARAMETER INTERACTIONS

The LABL and EFID parameters are interactive. The LABL parameter describes the first tape volume of those specified by EFID. All EFID's must

DATA BYTE 1								DATA BYTE 2								DATA BYTE 3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
B	A	8	4	2	1	B	A	8	4	2	1	B	A	8	4	2	1	B	A	8	4	2	1
TAPE CHAR 1						TAPE CHAR 2						TAPE CHAR 3						TAPE CHAR 4					

Figure 6-1. Data Conversion

be specified; however, the LABL parameter may be defaulted. The LABL parameter may not be specified without an EFID parameter.

The RTRY and PRTY parameters are interactive. A parity error is permanent when all retry attempts specified by RTRY fail. The System takes the corrective action specified by the parity error option (PRTY) parameter when the error becomes permanent.

To illustrate how to use these parameters, examples follow:

```
/ƀFITƀTAPEFILE, EFID=2340, RTRY=2, PRTY=ABE
```

Transfer of file named TAPEFILE is initiated. The file resides on tape volume 2340 (EFID=2340). If a parity error occurs, the System attempts to read the record two times (RTRY=2). If the record cannot be read in two attempts, the input operation (PRTY=ABE) aborts.

```
/ƀFITƀTAPEFILE, EFID=2340, PRTY=ACC
```

Transfer of file named TAPEFILE is initiated. The file resides on tape volume 2340 (EFID=2340). If a parity error occurs, the System attempts to read the record five times (default is five attempts). If the record cannot be read in five attempts, the input operation (PRTY=ACC) is continued.

6-24 STATEMENT INTERACTIONS

The FIT statement may interact with the MFR statement when a multi-file request encloses FIT statements. If an MFR statement encloses FIT statements, the parameters specified on or defaulted by the MFR statement override any parameter specifications on the enclosed FIT statements with the exception of the LABL parameter. See MFR, MFRE discussion in topic 8-11.

The FIT statement must be used to input a requested file. The statement's parameters explicitly describe the tape volume containing the requested file. It also specifies any special transfer conditions desired by the user. The statement indirectly specifies the organization of the file contained on the volume through a system default to sequential, if the volume has standard labels. However, the FIT statement does not describe the file's logical characteristics when it has been transferred and is resident on the disc. This information is supplied by the volume's label or by an FD statement. In addition, the FIT statement does not supply information for unlabeled volumes. For unlabeled volumes, the FD statement must be used in conjunction with the FIT statement and always precedes it.

The following conditions determine when an FD statement must accompany the FIT statement.

Standard-labeled tapes — FD statement not required. The file's organization is sequential by system default. If the user wishes to change any of the file's logical characteristics or its organization for CP step use of the file, then the FD statement is used to make changes in logical characteristics and file organization. The FD statement may precede or follow the FIT statement in the job input stream.

Unlabeled tapes — FD statement is required to specify the logical characteristics of the file. The FD statement must precede the FIT statement. The transfer is terminated if the FD is not used in this fashion. The file organization must also be specified.

The FD statement may also be used to specify changes in the file's logical characteristics or its organization for CP step use of the file. Then, the FD statement may precede or follow the FIT statement.

6-25 EXAMPLES

The following examples illustrate various coding techniques for the FIT statement:

1. File Residing on One Tape Volume.

```
/ FIT TAPEFILE, EFID=2340, LABL=2/SL,DEN=1600
```

TAPEFILE — The file named TAPEFILE is to be transferred to disc.

EFID=2340 — TAPEFILE resides on one tape volume identified as 2340.

LABL=2/SL — The tape volume has standard labels. It is positioned to the second file on the tape.

DEN=1600 — The tape volume density is 1600 bpi.

Default Values

RTRY — default is five attempts to read a record

PRTY — system aborts the transfer when five attempts to read a record have failed.

TRKS — default is nine-track tape

RCTL — does not apply to nine-track tape

2. File Residing on Multiple Volumes.

```
/ FIT TAPEFILE=2340/3546, LABEL=2/NL, DEN=800, ;
TRKS=7, RCTL=0
```

TAPEFILE — The file named TAPEFILE is to be transferred to disc.

EFID=2340/2435 — TAPEFILE resides on two tape volumes identified as 2340 and 2435. The files on tape volume 2340 are read first; those on 2435 next.

LABL=2/NL — The tape volume has no labels. It is positioned to the second file on the volume.

DEN=800 — Both tape volumes have densities of 800 bpi.

TRKS=7 — Both tape volumes are 7-track

RCTL=0 — Both tape volumes have odd parity without translation or conversion

The defaults RTRY and PRTY are the same as the first example.

3. All Defaults Operative.

```
/ FIT TAPEFILE, EFID=2340
```

TAPEFILE — The file name TAPEFILE is to be transferred. Access name, TAPEFILE, must be specified.

EFID=2340 — TAPEFILE resides on the tape volume identified as 2340. All identifiers for the volumes containing the file to be transferred must be declared.

<u>Default Values</u>

LABL — The tape volume, 2340, has standard labels, The tape volume is positioned to the first file on the volume.

DEN — The tape volume density is 1600 bpi.

RETRY — The system tries to read a record five times.

PRTY — The system aborts the input operation when five attempts to read a record have failed.

TRKS — The volume is a nine-track tape.

RCTL — Not applicable for nine-track tape.

4. Fit and FD Statement - Standard Labeled Tapes

```
/ FD ANAME, SECT=64/0/64, BKSZ=1024
/ FIT ANAME, EFID=1234, LABL=3, DENS=800
```

File name — For both FD and FIT, the access name is ANAME.

Disc Reservation — Both the FD and FIT statements specify an input file. The FD statement specifies 64 sectors to be used for this file on the secondary disc. If this proves to be insufficient, the input transfer is aborted.

Volume — For FIT, EFID parameter = 1234

Position — FIT LABL first subparameter=3; therefore position tape to beginning of third file on volume 1234.

Label Type — FIT LABL second subparameter is not coded; therefore, the default value, LABL=SL, is used to indicate standard labels. This means that the file characteristics are taken from the tape volume file label and not the FD statement.

Density — FIT; DENS=800, 800 bpi.

Tracks — Since TRKS and RCTL are not supplied, the default is to nine-track tape and RCTL is not applicable.

Format — FD RCFM is not coded. The file label information from the tape file is used. If this parameter is specified on the FD statement, it is ignored.

Block size — FD BKSZ=1024 (bytes). The file label information from the tape file is used. The value specified on the FD statement is not used since the volume is SL.

Organization

FD FORG is not coded; therefore, the default value (for input utility only) of FORG=PS is used. This organization applies only to the secondary disc file created (the tape file is always considered as sequential) and results in adding DDWs to each block on disc.

Parity

Since FIT PRTY and RTRY parameters are not coded, the default values are used. Before a parity error is considered permanent, five retries are made (RTRY=5). If a permanent parity error occurs, the input request is aborted (PRTY=ABE).

5. FIT and FD Statements - Unlabeled Tapes

```
/FD BNAME, BAND=1/1/0, FORG=DS, BKSZ=16384, RCMF=F
/FIT BNAME, EFID=5678, LABL=1/NL, PRTY=ACC, DEN=1600
```

File name

Both FD and FIT access names are BNAME.

Disc reservation

Since the file is an input file, the FD statement has specified that one band be used to contain the partial file on the secondary disc. If this proves to be insufficient for any set of triplets, the request is terminated. The FD statement must precede the FIT statement.

Volume

FIT EFID=5678

Position

FIT LABL first subparameter=1 and second subparameter is NL; therefore, the tape volume is positioned to the first data block on the tape volume (i. e., no tape marks are skipped).

Label Type

FIT LABL second parameter has LABL=/NL. This specifies that the volume has no labels. Since the file characteristics (organization, format, block size, record length) are not defined by the tape volume, they must be explicitly defined on the FD statement.

Density

FIT, DENS=1600 bpi.

Tracks

Since TRKS and RCTL are not specified, the default is nine-track tape and RCTL is not applicable.

Format

FD RCFM=F specifies that the record format is fixed. This parameter is used by the input utility to indicate that the block length is fixed on the tape volume.

Block size

FD BKSZ=16384 bytes specifies that the block length is 4096 words. This is used by the input utility to limit the tape block size to a maximum of 4096 words.

Organization

FD FORG=DS specifies the file organization is direct secondary. This is used by the input utility to transfer the sequential blocks on tape to the secondary disc file without adding DDWs. The block length must be a multiple of the disc sector for this format.

Parity

FIT PRTY=ACC specifies that a permanent parity error or a triplet truncated at an end-of-file (EOF) should be accepted. The RTRY parameter is not coded; therefore, the default value, RTRY=5, is used to determine the number of retry attempts before a parity error is classed as a permanent parity error.

6-26 NON-CATALOGED FILE OUTPUT

Peripheral utilities handle the physical transfer of a file from secondary disc to magnetic tape. The File Out Tape utilities specify output of a noncataloged tape file when the request is made via JSL. There is one File Out Tape utility: FOT writes a specified disc file to a half-inch magnetic tape volume.

6-27 THE FOT STATEMENT

The JSL File Out Tape (FOT) statement initiates the utility that writes a noncataloged disc file to half-inch magnetic tape volumes. This statement provides, through its parameters, information necessary to achieve the transfer. The position of the statement in the job input stream indicates when the transfer takes place. All files output by FOT statements are job local and may be accessed by any step in the job until the file is released.

The general form of the FOT statement is:

	LABEL		OPERATION	OPERANDS
/	Symbol	␢	FOT	access name $\left[\text{, EFID=tape id1}\left[\text{/tape id2}\left[\text{/...}\right]\right]\right]$ $\left[\text{, LABL=}\left[\begin{Bmatrix}\underline{1}\\ \text{position}\end{Bmatrix}\right]\left[/\left[\begin{Bmatrix}\underline{\text{SL}}\\ \text{NL}\end{Bmatrix}\right]\left[/\left[\begin{Bmatrix}\underline{\text{NEW}}\\ \text{ADD}\end{Bmatrix}\right]\right]\right]\right]$ APND $\left[\text{, SVC=scratch volume count}\right]$ $\left[\text{, OPT=}(\frac{\text{N}}{\text{R}})\right]$ $\left[\text{, RETP=retention}\right]\left[\text{, RTRY=}\begin{Bmatrix}\underline{5}\\ \text{number}\end{Bmatrix}\right]$ $\left[\text{, PRTY=}\begin{Bmatrix}\underline{\text{ABE}}\\ \text{ACC}\end{Bmatrix}\right]\left[\text{, LOC=location id}\right]$ $\left[\text{, DEST=destination code}\right]$ $\left[\text{, DEN=}\begin{Bmatrix}1600\\ 800\\ 556\\ 200\end{Bmatrix}\right]\left[\text{, TRKS=}\begin{Bmatrix}\underline{9}\\ 7\end{Bmatrix}\right]$ $\left[\text{, RCTL=}\begin{Bmatrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{Bmatrix}\right]$

The verb FOT must appear in the operation field. The operation and operands fields must be separated with at least one blank. The FOT statement may be labeled if communication with JSL program flow statements and/or the macro language is desired. If the statement is labeled, a blank must separate the label from the verb. No blank between the slash and the label is allowed.

The parameters in the operands field are both positional and keyword parameters. The positional parameter is mandatory and must be coded in the order shown. The keyword parameters are optional and may be coded in any convenient order.

6-28 ACCESS NAME

The file must be named to be accessible to the job. The access name parameter specifies the name by which the file is known to the job. The name must have been defined within the current job.

Format: The parameter value must be a valid ASC symbol. The parameter is positional and must be the first parameter.

Default: None. Access name must be specified.

6-29 EXTERNAL FILE IDENTIFICATION (EFID)

The external file identification parameter specifies the identification numbers of the tape volumes that are to contain the file.

Format: The keyword EFID designates the external file identification parameter. The value of the parameter is a character string or a series of character strings separated by slashes.

Each central site identification number is a character string of one to six characters. If the string is enclosed in apostrophes, blanks in the string will be retained; if the string is not enclosed in apostrophes, any blanks embedded in the string will be eliminated.

External labels are affixed to the tape volumes to provide visual identification of the volume and its contents. Normal tape volume control requires two types of external tape labels. One is a permanent label that identifies the volume; the other is a temporary label that identifies the contents.

The permanent label is affixed when the tape is first received. It should contain the sequential volume serial number assigned to it. The volume serial numbers are used to identify the tape volume by a unique number and to file the tapes in the tape rack.

The contents label is used to identify the current contents of a particular volume. This temporary label is applied when data is written on the volume and contains identifying information to ensure that the contents of the volume can be easily distinguishable. The information entered in the label is usually furnished partly by the programmer and partly by the operator.

Restriction: When the FOT statement occurs in an MFR/MFRE group, this parameter is overridden by the specifications in the MFR statement that begins the group.

Default: Should no EFID be given, the file is written to a scratch volume. The user is informed of the external volume identifier in his job output if the tape file was successfully written.

6-30 LABEL (LABL)

The label parameter, identified by the keyword LABL, describes three characteristics of the tape volume that will contain the file. The parameter describes the type of label the volume will have, the position of the file to be written on the tape volume, and the placement of the new file on the volume.

Format: The label parameter is comprised of three subparameters. All three subparameters are positional and are separated by slashes.

Examples

```
LABL=2/SL
LABL=2/SL/APND
LABL=2//ADD
LABL=/NL
LABL=2
```

The first subparameter specifies the desired position of the file on the tape volume. The value of the position subparameter is limited to the range: $1 \leq n \leq 255$.

The second subparameter specifies the label type of the tape volume. The keyword SL specifies the tape volume has standard labels. The keyword NL specifies the tape volume has no labels.

The third subparameter specifies whether the output file is to be placed on a new tape, added to an existing tape, or appended to an existing file on tape.

The keyword NEW specifies that a scratch tape is to be made available to receive the output file. The keyword ADD specifies that the output file is to be written on an existing tape as an additional file. The keyword APND specifies that the file is to be written on an existing file on tape as an appendage to the existing file (by erasing the current file mark terminating the existing file). Since there is no means of preventing files of inconsistent formats from being appended, the user must exert caution with the APND option. A mismatch of density, block size or record format will cause transfer to be aborted.

This position subparameter is interactive with the label type subparameter of the LABL parameter.

Label Type	Position	Interpretation
SL	n	n=file sequence number. The number of physical tape marks to be skipped during positioning is a function of the standard label format.
NL	n	n=file sequence number. The number of physical tape marks to be skipped during positioning is n-1.

Restrictions: If SL is specified explicitly or by default or, if an EFID is specified, the System requests that the correct tape volume be supplied. If a tape volume cannot be verified as standard labeled, the operator must cancel the request.

If NL is specified and the tape cannot be verified as non-labeled, the operator must cancel the request.

When the FOT statement occurs in an MFR/MFRE group, this parameter (including all three subparmeters) is overridden by the specifications in the MFR statement that begins the group.

CAUTION

If APND is specified, the System will not cross volumes to do the appends; the file to be appended must be on the first volume specified by the first EFID.

Default: If position is not specified, the default is 1. If label type is not specified, the default is SL. If placement is not specified, the default is NEW. NEW specifies that the file is placed on a system scratch volume. If an EFID parameter has been specified, the user cannot default to NEW; he must specify ADD or APND.

6-31 SCRATCH VOLUME COUNT (SVC)

The scratch volume count parameter specifies a maximum number of system scratch tape volumes that are to contain files or parts of files.

Format: The parameter is identified by the tag, SVC. The parameter value is an unsigned constant of from one to six digits.

System scratch volumes are provided by the system. The volume identifiers have been assigned by the system and registered in the Tape Volume Catalog as scratch volumes.

The parameter specifies the maximum number of scratch volumes required. It can be used with EFID parameter if both scratch volumes and user-specified volumes will be used to contain the files. It can also be used with the LABL parameter if non-labeled scratch volumes are desired.

Restrictions: When the FOT statement occurs in an MFR/MFRE group, this parameter is overridden by the specification of parameters in the MFR statement that begins the group.

Default: If the scratch volume count parameter is omitted, the default value of twelve scratch volumes is assumed.

6-32 OPTION (OPT)

The option parameter specifies the disposition of the file upon successful output of the file.

Format: The option parameter is identified by the keyword OPT. The parameter value is one of the alphabetic characters N or R enclosed in parentheses.

The option code N specifies that the file is not to be released at this time. The option code R specifies that the file is to be released at this time. For the job to use the same access name again, new disc space must be allocated through the normal system open process.

Default: If no option parameter is specified, the default is option code N.

6-33 RETENTION PERIOD (RETP)

The retention period parameter specifies the number of days that the output file will be retained.

Format: The retention period parameter is identified by the keyword RETP. Retention period is an unsigned constant.

Retention period is converted to an expiration date and is recorded in the Tape Volume Catalog for both SL and NL tape volumes. It is not recorded on the tape for NL volumes. The value indicates the number of days the file is to be retained.

Restriction: When the FOT statement occurs in an MFR/MFRE group, this parameter is overridden by the specification in the MFR statement that begins the group.

Default: If the retention period parameter is not specified, the default value is the installation default retention period.

6-34 RETRY (RTRY)

The retry parameter specifies the number of times the system is to attempt to write a block to the tape if a parity error occurs. If any write fails due to a parity error, and the number of retries is not exceeded, several inches of tape are erased and the write operation is retried following the erased portion of tape. Thus, a RTRY=10 may try to write the block at ten different locations on the tape. For each different parity error block of the file, up to ten locations are tried, if necessary. When the specified number of retries is exceeded, corrective action is taken, as stated in the PRTY parameter.

Format: The retry parameter is identified by the keyword RTRY. The parameter is an unsigned constant. The value of the retry parameter is limited to the range: $0 \le r \le 15$.

Default: If no retry parameter is specified, the system will make five attempts to write the record.

6-35 PARITY ERROR OPTION (PRTY)

The parity error option parameter indicates the action to be taken if the number of permissible retry attempts is exceeded on output to tape.

Format: The parity error option parameter is identified by the keyword PRTY. The parameter is one of two keywords ABE or ACC. The keyword ABE specifies that the output process should be aborted for a permanent parity error. The keyword ACC specifies that the output process should be continued for a permanent parity error. Any other permanent error will cause an abort of the output process.

Default: If no error option is specified, the system will use the ABE option.

NOTE

> The retry and parity error option parameters are interactive. When a parity error occurs, it is retried the number of times given. When all retry attempts fail, the action specified by the error option is taken.

6-36 LOCATION (LOC)

The location parameter specifies the site at which the tape file is to be output.

Format: The location parameter is identified by the keyword LOC. The parameter is a string of from one to eight alphanumeric characters.

The location identifiers are assigned the various terminal installations by the central site.

Restriction: When the FOT statement occurs in an MFR/MFRE group, this parameter is overridden by the specification in the MFR statement that begins the group.

Default: If no location is specified, the default value is the identifier for the central site.

6-37 DESTINATION (DEST)

The destination parameter is transmitted to the external tape label for use in routing the tape after the tape has been output.

Format: The destination parameter is identified by the keyword DEST. The parameter is a string of two alphabetic characters. The destination parameter is assigned by the central site.

Restriction: When the FOT statement occurs in an MFR/MFRE group, this parameter is overridden by the specification in the MFR statement that begins the group.

Default: If the destination parameter is omitted, the default is the central site.

6-38 TAPE DENSITY (DEN)

The tape density parameter specifies the recording density in bits-per-inch-per-track for the tape volumes shown in the EFID parameter or the scratch volumes specified or defaulted by the SVC parameter.

Format: The tape density parameter is preceded by the keyword DEN. The density parameter allowed is either 1600, 800, 556, or 200 (bpi).

The key 1600 specifies the tape density is 1600 bpi, the key 800 specifies the tape density is 800 bpi, etc.

Restrictions: When the FOT statement occurs in an MFR/MFRE group, this parameter is overridden by the specification of parameters in the MFR statement that begins the group.

Default: If no tape density parameter is coded, the parameter value is set to 1600 for 9-track tape or 800 for 7-track tape.

6-39 RECORDING TRACKS (TRKS)

The tape recording track parameter, identified by the keyword TRKS, specifies if the given tape is a seven-track or a nine-track tape.

Format: The parameter value is one of the key numbers 9 or 7 to designate either the nine-track or the seven-track tape respectively.

Default: The tape recording track parameter can be omitted. The default is to nine-track tape. If the tape recording track parameter specification or default is different than the actual tape configuration the system aborts the input operation.

6-40 RECORDING CONTROL (RCTL)

The recording control parameter allows for the specification of a parity, translation and data conversion mode. This parameter is applicable only if seven-track tape is being processed. It is identified by the keyword RCTL and the keyword values are:

O	- odd parity without translation or conversion
E	- even parity without translation or conversion
T	- odd parity with translation but no conversion
C	- odd parity with conversion but no translation
ET	- even parity with translation but no conversion.

Default: If the RCTL parameter is omitted, and seven-track tape is designated, the default is ET or even parity with translation but no conversion.

Data Translation: Data translation refers to the ability to transform data recorded in BCD interchange to EBCDIC and vice versa. Since each BCD code is a six-bit value, upon input the six bits are translated into the corresponding eight-bit EBCDIC value. For output operations the eight-bit EBCDIC value is translated into its six-bit BCD representation.

Data Conversion: The data conversion option makes it possible to write eight-bit binary data on seven-track tape. For output operations, every three bytes of data are written as four tape characters. Figure 6-1 illustrates this data conversion.

6-41 PARAMETER INTERACTIONS

The EFID, LABL, DEN and SVC parameters may be interactive, depending on the type of tape volumes desired. The parameters may be used in various combinations to achieve a specific request.

Specifying Private Volumes Only: To output files on private volumes only, the user should specify three parameters.

$$\text{LABEL} = \left\{\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right\} / \left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\} \ / \ \left\{\begin{matrix}\text{ADD}\\ \text{APND}\end{matrix}\right\} \quad \text{NEW is not allowed.}$$

EFID must be specified
DEN is optional
The SVC parameter cannot be used

Specifying Scratch Volumes Only. In specifying system scratch volumes, the user may specify the following combination.

$$\text{LABL} = \left\{\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right\} / \left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\} / \text{NEW}$$

DEN is optional.

SVC is optional. It may be used if a maximum number of scratch volumes is required.

The EFID parameter cannot be used.

In the above case, all parameters may be omitted. The default values provide the proper conditions.

Specifying Both Scratch and Private Volumes. In specifying both system scratch and private volumes, all three parameters should be used.

$$\text{LABL} = \left\{\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right\} / \left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\} / \left\{\begin{matrix}\text{ADD}\\ \text{APND}\end{matrix}\right\}$$

NEW is not permitted; it specifies Scratch volumes only.

EFID must be specified.

SVC is optional. If a maximum number of scratch volumes is desired, the parameter should be used.

DEN is optional.

The RTRY and PRTY parameters are interactive. A parity error is permanent when all retry attempts specified by RTRY fail. The System takes the corrective action specified by the parity error option (PRTY) parameter when the error becomes permanent.

6-42 STATEMENT INTERACTIONS

The FOT statement may interact with the MFR statement when a multi-file request encloses FOT statements. If an MFR statement encloses FOT statements, the parameters specified on or defaulted by the MFR statement override any parameter specifications on the enclosed FOT statement.

The following parameters are affected by MFR parameter specifications or defaults:

EFID
LABL
DEN
TRKS
RCTL
SVC
RETP
LOC
DEST

The FOT statement may also interact with the FD statement. If the user knows that logical characteristics are not supplied by any other source, he should precede the FOT statement with an FD statement describing the file's characteristics. If these characteristics already exist as a result of the file's creation, the user may use an FD statement, but must specify the same logical characteristics as already exist for the file being output by the FOT statement.

6-43 EXAMPLES. The following examples illustrate some of the FOT statement's parameters:

1. Output a file to a nine-track half-inch magnetic tape device -- all defaults operative.

   / FOT FILEOUT

Parameter	Interpretation
FILEOUT	FILEOUT is the access name given to the file being output. This name links the file to the job.

   Default Values

   RETP is defaulted to an installation-defined value. The file will be retained for that period of time.

2. Specify a Retention Period.

   / FOT FILETAPE, RETP=10

Parameter	Interpretation
FILETAPE	Access name identifies the file to the job.
RETP	The file on tape will be retained for ten days.

6-44 THE FOSYS STATEMENT

The File Output to SYStem (FOSYS) statement specifies that the file associated with the specified access name is an output file to be printed or punched. The position of the statement in the job input stream determines the position of the output file in the system print or punch output file chain.

Output for print and punch files is a two-step action. The first step places a file on an output list of print or punch files; this action is called spooling. It occurs when the FOSYS statement is encountered in the job input stream. After the file has been spooled, no successive CP step can gain access to the file's contents. The second step performs the actual physical transfer of spooled files to specified printers and punches. This action takes place during job termination processing.

Logical characteristics of the file on disc are defined by the File Information Block, not the FOSYS statement. Any valid record format can be used with sequential file organization (FORG=PS). Only F, FB, or FBS can be used with direct secondary file organization (FORG=DS), and record length for DS organization must be a multiple of four bytes. Block size and record length values are used in deblocking logical records. Each logical record contains a) a print line image for print files, plus a control character if specified by the record format; or b) one card image for punch files.

The general form of the FOSYS statement is:

	LABEL		OPERATION		OPERANDS
/	Symbol	ƀ	FOSYS		access name [, TYPE= $\left\{\begin{array}{l}\text{PRINT}\\\text{PUNCH}\end{array}\right\}$] [, EXID=external file ID] [, LOC=location ID] [, FORM= [special forms ID] [/[carriage control tape ID]]] [, PRTS=line spacing] [, COPIES=aa] [, OPT=(R)]

The verb FOSYS must appear in the operation field. The operation and operands fields must be separated with at least one blank. The FOSYS statement may be labeled if communication with JSL program flow statements and/or the macro language is desired. If the statement is labeled, the label must be separated from the operation field with at least one blank.

The parameters in the operands field are both positional and keyword parameters. They refer to either print or punch files, or both. The parameters may be used to designate print files, punch files or both types.

6-45 ACCESS NAME

The file must be named before it is accessible to the job. The access name parameter specifies the name by which the file is known to the job and to which the specified parameters refer.

Format: The parameter value is valid ASC symbol.

Default: This parameter is positional and mandatory. It must appear in the first position of the operands field and cannot be omitted.

Example: A file named ANAME is an output file destined for a printer or card punch. The access name parameter may appear as:

```
/ȸFOSYSȸANAME
/ȸFOSYSȸT1340
```

6-46 TYPE (TYPE)

The file is spooled to either the print or punch system output chain for subsequent printing or punching on designated devices. The type of file must be specified so the Operating System can spool the file to the appropriate output chain.

Format: The type parameter, identified by the keyword TYPE, indicates whether the file is to be placed on the job's print or punch output chain. The parameter value is one of the keywords, PRINT or PUNCH. PRINT specifies the file is spooled on the job's print and output chain; while PUNCH indicates spooling on the punch output chain.

Default: The parameter may be omitted in which case the System spools the named file on the print output chain (TYPE=PRINT).

Example: To illustrate use of this parameter, an example follows:

```
/ȸFOSYSȸANAME, TYPE=PUNCH
```

or

```
/ȸFOSYSȸT1340, TYPE=PRINT
```

6-47 EXTERNAL IDENTIFICATION (EXID)

If desired, the printed or punched output file may be named. The external identification parameter, known by the keyword EXID, names the file. For print files, a special line containing the output file name is printed before the print file image. For punch files, a card containing the name is punched preceding the card deck containing the file information.

Format: The parameter value is a character string of one to eight characters. Only one type of file can be named at a time--either a print or a punch file.

Default: The parameter is optional. If it is omitted, no identifying print line or card will be output preceding the file information.

6-48 LOCATION (LOC)

After the output file has been spooled to the system print or punch output chain, it is physically transferred to the designated output device during job termination processing. The location at which the file is output should be specified if other than the location specified on or defaulted to by the JOB statement.

Format: The location parameter, identified by the keyword LOC, specifies the installation at which the file is to be output. The parameter value is a string of one to eight alphanumeric characters. The character strings allowed are assigned by the central site. They may specify terminal installations.

Default: The parameter is optional. If it is omitted, the file is output at the location specified on the JOB statement. If a location has not been specified on the JOB statement, the default is the location where the job originated. The location parameter on FOSYS is not supported if its value differs from that specified by the JOB statement's location parameter.

6-49 FORM (FORM)

The form parameter, identified by the keyword FORM, serves two purposes. It is used to specify special paper and forms requirements for both print and punch files and printer carriage control tape identification for print files.

Format: The parameter consists of two subparameters separated by a slash. Each subparameter value is a character string of one to eight characters. Each subparameter is positional within the optional parameter. The values are installation-assigned.

The first subparameter, special forms identification, designates the type of cards to contain a punch file or the multi-part and paper type to use for print files. The second subparameter specifies the identification of the printer carriage control tape, used only for print files.

Default: The form parameter is optional. Each subparameter, though positional, is optional. If the form parameter is omitted, the specified file will be printed using paper and carriage control tapes in regular use at the installation. Punch files will be punched on standard cards in regular use at the installation.

For print files, either the first or second subparameter may be omitted. Omission of either causes default to the standard paper or carriage control tape in regular use.

To specify the form parameter for punch files, the first subparameter is coded; the second omitted. The slash must not be coded.

To specify the form parameter for print files, the two subparameters are separated by a slash. To omit the first subparameter, a slash must precede the second value. To omit the second subparameter, only the first value is used. The slash must be omitted.

Example: To illustrate correct use of the form parameter, note the following examples:

1. Punch File

```
/ƀFOSYSƀANAME, TYPE=PUNCH, FORM=3360
```

where

the file named ANAME is a file to be punched on cards identified by 3360.

The default values are

EXID — no identifying card precedes the card deck containing the file.

LOC — the file is punched at the location specified on or defaulted by the JOB statement.

2. Print File

```
/ƀFOSYSƀBNAME, TYPE=PRINT, FORM=2/SP12
```

where

the file name BNAME is printed on special paper identified by 2. The carriage control tape to be used is identified by SP12.

The default values are

EXID — no identifying print line precedes the printed file information.

LOC — the file is printed at the location specified on or defaulted by the JOB statement.

3. Print File with Default Values

```
/ƀFOSYSƀBNAME, FORM=2
```

The defaults are

TYPE — default is to PRINT; file is printed

EXID — same as described previously,

LOC — same as described previously,

2nd subparameter of FORM	standard carriage control tape supplied by installation.

or

/ɓFOSYSɓBNAME, FORM=/SP12

where the defaults are

TYPE	default is to PRINT; file is printed,
EXID	same as previously described,
LOC	same as previously described,
1st subparameter of FORM	standard forms of paper supplied by installation.

or

/ɓFOSYSɓBNAME

where the default values are

TYPE	default is PRINT; file is printed,
EXID	no identifying print line precedes the printed file information,
LOC	the file is printed at the location specified on or defaulted by the JOB statement,
FORM	installation supplies standard forms or paper and carriage control tape.

6-50 PRINT SPACE (PRTS)

The print space parameter, identified by the keyword PRTS, specifies the line spacing by the printer between lines of printed code, or the card stacker receiving the punched cards.

Format: The parameter value is an unsigned, constant according to the type of file being output.

TYPE=PRINT	1,2, or 3
TYPE=PUNCH	1 or 2

The values 1,2, or 3 specify the number of lines between printouts. The values 1 or 2 specify the identification of the stacker receiving cards.

Only one type of value may be used. It must correspond to the type of file specified by the TYPE parameter. For TYPE=PRINT, use 1, 2, or 3. For TYPE=PUNCH, use either 1 or 2.

Default: The print space parameter is optional. If PRTS is not specified, the value for either type of file may be specified in the file. If the file does not contain carriage control/stacker select characters, PRTS defaults to 1 for either type of file.

Example: The following example illustrates the difference in using this parameter:

```
/ɓFOSYSɓANAME, TYPE=PUNCH, PRTS=2
```

where the stacker identified by 2 receives the cards.

```
/ɓFOSYSɓANAME, TYPE=PRINT, PRTS=3
```

where the printer spaces three lines between printouts.

6-51 COPIES (COPIES)

The copies parameter, identified by the keyword COPIES, specifies the number of copies of the specified file which is to be output to the device specified in the TYPE parameter.

Format: The value of the COPIES parameter is an unsigned decimal integer from 1 to 99.

Default: The COPIES parameter is optional. If it is omitted, the System supplies one copy.

6-52 OPTION (OPT)

When a file has been spooled to the system output print or punch chain, the disc space occupied by the file may be reallocated for further use. The option parameter, identified by the keyword OPT, specifies the disposition of the disc space that becomes available.

Format: The parameter value is the alphabetic character, R, enclosed in parentheses. R specifies that the original file space is placed on the appropriate print or punch system output chain. For the job to use the same access name again, new file space must be allocated through the normal system open process.

Default: The parameter is optional. Currently, R is the only option code available. If omitted, the default value is the code R. Since R is also the default value, the parameter does not necessarily have to be used. The parameter is available because current plans provide for additional options in the future.

SECTION VII

CATALOG MANAGEMENT

7-1 CATALOG SYSTEM

The catalog system in the ASC Operating System provides the following capabilities:

- Data retrieval by symbolic name independent of device type and volume numbers,
- File privacy and protection of data from unauthorized use,
- Maintenance of successive file generations relative to a specified file.

7-2 CATALOG STRUCTURE

The catalog is a tree structure with a root node from which all other nodes are directly or indirectly accessible. Figure 7-1 is a graphic representation of a catalog structure.

Every node of the catalog tree may have files, or versions, associated with it, although the files may be empty.

Each file is associated with only one node in the catalog, and any node can have several files associated with it.

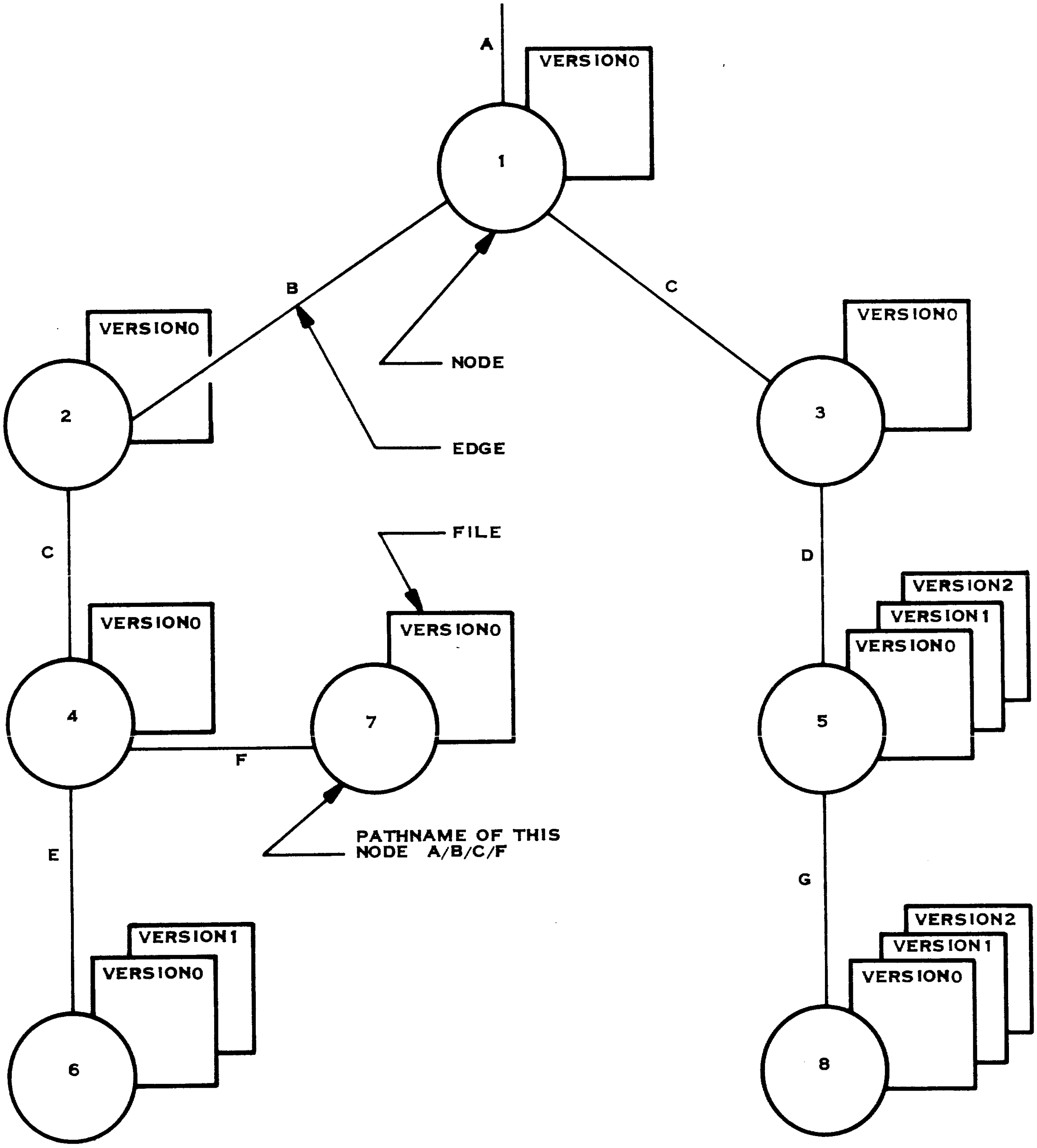
The version name is the node's pathname plus a modifier that identifies the version.

7-3 DEFINITIONS

The following terms are defined for describing the catalog system:

7-4 EDGENAME. An edgename is the name of the connector between two adjacent nodes. An edgename must be a valid ASC symbol. No two edges originating from the same node may have the same name. An edgename need not be unique within the catalog.

7-5 PATHNAME AND NODENAME. Each node in a catalog is identified by a pathname. Each file in a catalog is identified by the node's pathname plus the version number of the file at the node. The node accessed by a pathname is the terminal node of the pathname. All nodes whose pathnames could be generated by sequentially deleting one edgename from an original pathname greater than one edgename in length are reference nodes with respect to the terminal node.



(A) 107854

Figure 7-1. Graphic Representation of the Catalog Tree Structure

The pathname of the node is the sequential catenation of all edgenames from the root node to the subject node. The catenation of the edgenames is indicated by slashes. The pathname of any given node must be unique within the catalog; i.e., any given pathname must access only one mode.

A node may have only one pathname; i.e., any given node may be reached by only one path; closed loops are not permitted in the catalog.

A pathname plus version number is a valid file name only in catalog processing statements; it cannot be used as an access name.

7-6 SYNONYM. A synonym is a single valid ASC symbol that is defined as equal to a given pathname in the catalog. Synonyms are defined for pathnames by the JSL Path Definition (PD) statement.

Once a synonym is defined for a given pathname, the synonym may be used as the first edgename in the pathname of another node farther down the same path. Synonyms are required to process nodes that are more than 10 edgenames deep in a catalog. Synonyms are valid only in catalog processing; they cannot be used as access names.

A synonym must be defined from the root node. The synonym exists only for the duration of the job in which it is defined. It is a synonym only for a node, not for the file associated with the synonym's terminal node.

7-7 ACCESS NAME. An access name has two uses in catalog processing: (1) for a cataloged file to be accessible to a job, it must have a job local name that links it to that job, and (2) for a version to be cataloged by a job, the job local name of the version must be specified to cataloging procedures. An access name has no meaning to the catalog itself; it only identifies files to the cataloging procedures.

The JSL ASG statement assigns a cataloged file to a given job. The specified access name (on the ASG statement) links the file to a given job. The file may be shared concurrently with other jobs specifying the file, or it may be used exclusively by the given job.

A given cataloged file can be given more than one access name in the job. The user will have access to the same physical file through all access names. Any file with an access name defined in a current job may be cataloged by that job. The file may have been a cataloged file or an external file with respect to the specific job, or it may have been created within that job. The CAT, CATV, and RPLV statements all use access names to define files that are to become versions in a catalog.

Modification of the contents of a file specified by an access name causes the contents of the cataloged version to be modified, if the file is resident. To modify the contents of a non-resident version, the RPLV must be used after the modification; the current access name is specified as an operand and the specified pathname and version must be identical to the original ASG statement specification.

7-8 VERSIONS. There can be more than one version at a node. Every node has at least one version, although it may be a null version. The name of a cataloged file is the pathname of the file's node with a modifier that identifies the version.

The maximum number of versions permitted for a node is specified by the JSL CAT, CATN and/or CHG statements. The maximum number, N, is a positive integer, where $1 \le N \le 64$. The absolute version numbers are assigned in the sequence (0, 1, 2, . . . N-1).

Each file sequentially cataloged by a CATV statement is assigned an integer-index, beginning with zero. For example, if a node has a maximum of 5 versions, the version numbers are assigned in the sequence: 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, ... As a given index recurs, the older copy of that version is re-placed by the newer.

Every node has a version flagged, establishing a point of reference in the sequence of versions. The flagged version is the one most recently cataloged, by a CATV statement, unless specified otherwise. The automatic sequencing feature can be deactivated to maintain a constant flagged version. If no versions have been cataloged, the flagged version is a null version. When a version is not specified in catalog processing, the flagged version is accessed.

Versions implied by the maximum version's specification, but not cataloged, are null versions. Deleted versions are null versions. Versions cataloged by the CAT, CATV, or RPLV statements are cataloged versions. They remain cataloged until deleted by the DEL, DELV, RPLV statements. A cataloged version may be empty (contain no data), but not null.

7-9 CATALOG PRIVACY

The system provides for the protection of data from malicious or accidental violation. The data may be made accessible to other users on a controlled basis wherein users are identified to the system by user codes. Three control states determine node characteristics that establish catalog privacy: reference control, access control, and son-add control.

7-10 ACCESS CONTROL LIST. Each node contains an Access Control List consisting of user codes, each associated with a list of attributes. This access list is used for screening purposes, and the various attributes of the authorized users provide controlled access to the file associated with the node. A user possesses attributes assigned during catalog processing. The user's attributes are associated with his user code in the Access Control List of the respective node.

The creator of a node automatically possesses the own attribute. If the creator of the node is not listed in the access control information parameter of the statement that creates the node, the creator possesses all control attributes. If the creator of the node is listed, only those attributes listed are assigned, plus the own attribute. The creator of a node is (by definition) the user code most recently specified in the job block containing the statement creating the node.

A user may possess six attributes: own, reference, son-add, read, write, and execute.

- Possession of the own attribute gives the user authority to change any characteristics of the node. The owner may change the Access Control List and the attributes of its members, the maximum number of versions of the node, the control states of the node, the flagged version number, and the retention period of a cataloged tape. The owner may catalog files, replace files, or delete the node if it is a terminal node. The access control list is examined for any of the above operations.

- The reference attribute has significance only for those nodes in the full reference control state. For those nodes in the full reference state, any user's attempt to refer to the node will cause the system to examine the Access Control List for the user's possession of the reference attribute. If the user possesses it, he is passed through the node.

- The son-add attribute has significance only for those nodes in the full son-add control state. For those nodes in the full son-add control state, any user's attempt to add a son node will cause the system to examine the Access Control List for the user's possession of the son-add attribute. If the user possesses it, he will be permitted to add the son node. Note that a user with this attribute specifies the Access Control List of the node he creates and is automatically an owner of the node he creates.

- Possession of the read or execute attribute gives the user authority to read from the file associated with the node. For a node in the full access control state, possession of only this attribute limits the user to reading from the file.

- Possession of the write attribute gives the user authority to write to the file associated with the node. For a node in the full access control state, possession of only this attribute limits the user to writing to the file.

NOTE

Writing to a file does not affect the contents of the file in the catalog unless the file is resident. A permanent change to a non-resident file is accomplished only by cataloging a version of the file with a Job Specification Language statement.

Such a permanent change requires that the user possess the own attribute.

NOTE

These attributes are completely disjoint; possession of any one attribute does not imply possession of any other attribute. For example, possession of the own attribute does not permit the user to add a son node. The own attribute does give the user possessing it the authority to change his attributes, including giving himself additional attributes.

7-11 REFERENCE CONTROL. The concept of node reference implies only the process of catalog searching, not the intent to operate on the node or the files associated with the node. Reference control affects both access control and son-add control to the extent that a user must meet reference control requirements at every node searched for access or son-add purposes.

Each node is either in the full-reference control state or the no-reference control state. The reference control state of a node is specified during catalog processing of the node.

One user attribute is applicable to reference control: the reference attribute. Any reference to a node in the full-reference control state causes the Access Control List of the node to be examined. Reference to the node is permitted only if the user is identified in the Access Control List and if that user possesses the reference attribute. The catalog in figure 7-1 with all the nodes in the full-reference control state requires a user wishing to access node A/B/D to possess the reference attribute at nodes 1 and 2.

Any reference to a node in the no-reference control state is permitted. The Access Control List is not examined.

A successful reference to a node -- either in the full- or no-reference control state -- does not permit access to the node; only passage through the node.

Access is granted according to the access control state and the usage attributes possessed by the user.

7-12 ACCESS CONTROL. The concept of node access implies the intent to operate on the node's associated files (or versions). Node access means that the terminal node in the specified pathname is accessed.

Each node is in one of three access control states:

- Full access
- Partial access
- No access.

The node's access control state is specified during catalog processing of the node.

Three user attributes are applicable to access control:

- The read attribute
- The write attribute
- The execute attribute.

Any attempt to access a node in the full-access control state causes the Access Control List of the node to be examined. Access is permitted only to those users identified in the Access Control List; they are permitted only the type(s) of access defined by their attributes.

Any attempt to access a node in the partial-access control state causes the access control list to be examined for the user's possession of the write attribute. If the user is in the list and has the write attribute, all three access types are given; otherwise, only the read and execute attributes are given.

An attempt to access a node in the no-access control state does not cause the Access Control List of the node to be examined i.e., all users are permitted to read, write, and execute the file at the node.

For example, the catalog in Figure 7-1 with all nodes in the no-reference control state requires a user wishing to refer to node A/B/D to know the pathname, but the user is not necessarily permitted access to node A/B/D. Access is determined by the state of the node: full access, partial access or no access.

NOTE

> A user who wishes to access some given node at a catalog must pass reference control requirements at every node in the path from the root node to the object node; the user's attributes at the object node are meaningless if that user cannot refer to every node in the path.

7-13 SON-ADD CONTROL. The concept of son addition to a node implies the intent to extend an existing pathname by one edgename and create a new node in the catalog. The node is identified by the new pathname. The node associated with the existing pathname is the parent node and the new node is the son node. The parent node becomes a reference node with respect to its son.

Each node is in either the full son-add control state or the no son-add control state. The son-add control state of a node is specified during catalog processing of the node. One user attribute is applicable to son-add control: the son-add attribute.

Any attempt to add a son node in the full son-add control state causes the Access Control List to be examined. The son node is added only if the user is identified in the Access Control List and possesses the son-add attribute.

Any user may add sons to a node in the no son-add control state. The Access Control List is not examined.

7-14 THE CATBLD STATEMENT

The Job Specification Language provides interfaces with the catalog system at the structure level through the CATalog BuiLD Statement (CATBLD). The CATBLD statement defines a new catalog structure within the catalog system. When CATBLD is used, it allocates the catalog and initializes a specified root node of the catalog on the disc. The name of the root node is the catalog name. All catalog security requirements may be established for the root node. Once a user defines a catalog via the CATBLD statement, he is free to construct and manipulate the structure using other JSL catalog statements.

The general form of the CATBLD statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	CATBLD	ƀ	edgename [, MXVR=maximum number of versions] $\left[, \text{ACST=} \begin{Bmatrix} \underline{\text{NO}} \\ \text{PART} \\ \text{FULL} \end{Bmatrix}\right] \left[, \text{SON=} \begin{Bmatrix} \underline{\text{NO}} \\ \text{FULL} \end{Bmatrix}\right]$ $\left[, \text{RFST=} \begin{Bmatrix} \text{NO} \\ \text{FULL} \end{Bmatrix}\right]$ [, ACIN=access control information]

The verb CATBLD must appear in the operation field. The statement may be labeled. The label must be separated from the operation field with at least one blank. The operation field must be separated from the operands field with at least one blank.

The operands field contains both positional and keyword parameters. The positional parameter is mandatory and must be the first parameter in the field. The keyword parameters may appear in any convenient order or may be omitted.

7-15 EDGENAME

The catalog being defined must be identified to the catalog system. The edgename parameter specifies the first edgename of the catalog. The edgename is the pathname of the root node. The parameter is mandatory and positional. The edgename cannot be a synonym defined by a PD statement.

Format: The parameter value is a valid ASC symbol. The Operating System enters the specified edgename in the catalog directory maintained by the Operating System. The system identifies the catalog in the system by this name.

Default: None. The edgename must be specified.

7-16 MAXIMUM VERSIONS (MXVR)

Since each node may have more than one version associated with it simultaneously, the user can specify a maximum number of versions that can be associated with the specified root node. The maximum versions parameter, identified by the keyword MXVR, specifies this limit.

Format: The parameter value is an unsigned constant. The range is $1 \leq N \leq 64$, where N is the parameter value. This value specifies the number of versions allowed at the node. When CATBLD establishes the node, the node can be considered to contain N null versions. No version actually is cataloged at this time. The statement only creates null versions which contain no data.

Default: The maximum versions parameter is optional. If it is omitted, the Operating System assumes that only one version will be permitted at the root node.

7-17 ACCESS CONTROL STATE (ACST)

The access control state parameter, identified by the keyword ACST, specifies the degree of control the System will maintain over access to the node.

Format: The parameter value is one of the keywords NO, PART, or FULL. NO specifies no access control; i.e., all users may read, write, and execute versions at the node. PART specifies partial access control, giving all users read and execute attributes; the write attribute is only given to qualified users. FULL specifies full access control, restricting access to the versions at the node to qualified users only. Qualified users are those contained in the Access Control List established by the access control information parameter (ACIN) or the creator of the node.

Read, write and execute attributes are controlled through the access control state of the node. Changing the node's characteristics - stipulated by the own attribute - is not affected by the access control state of the node. Ownership of the node is established through the access control information parameter (ACIN) or by having created the node.

Default: The parameter may be omitted, in which case the Operating System will not maintain any access control. Users have full execute, read and write attributes. The parameter default value is NO. This condition does not imply any own attribute; users cannot change the characteristics of the root node unless they created the node or established ownership through the access control information parameter (ACIN).

7-18 REFERENCE CONTROL STATE (RFST)

The reference control state parameter, identified by the keyword RFST, specifies whether the Operating System will control user's attempts to pass through the root node during a catalog search.

Format: The parameter value is one of the keywords NO or FULL. NO specifies no reference control; any user may pass through the node during a catalog search. FULL specifies full reference control; only qualified users may pass through the node. The access control information parameter (ACIN) qualifies users by providing them with a reference attribute.

Default: The parameter is optional. If the parameter is omitted, the Operating System will not maintain reference control. Any user can pass through the node during a search. The parameter default value is NO. This condition does not affect the state of access control on the node.

7-19 SON-ADD CONTROL STATE (SON)

The son-add control state parameter establishes the degree of control exercised by the Operating System over the addition of nodes (called sons) to the root node.

Format: The parameter is identified by the keyword SON. Its value is one of the keywords NO or FULL. NO specifies a no son-add control; any user may add son nodes to the specified node. FULL specifies full son-add control. Only qualified users can add son nodes. The access control information parameter (ACIN) provides users with the proper attribute.

Default: The parameter is optional. If it is omitted, the default value is NO and any user can add son nodes.

7-20 ACCESS CONTROL INFORMATION (ACIN)

The access control information parameter, identified by the keyword ACIN, provides the most comprehensive security measures for the catalog being built. This parameter establishes valid users and determines the actions (attributes) each may take.

Format: The parameter value is a set of subparameters separated by slashes. Each subparameter is a valid user code, followed by an asterisk and one or more of the alphabetic characters E, F, O, R, S, and W, appended to the user code. Each character is also separated by an asterisk. The values are translated into an Access Control List maintained by the Operating System for the specified node.

Each user code becomes the System's identifier for a user permitted activity at the node. The alphabetic characters appended to the user code specify the action attributes being allowed the user. The attributes have the following interpretations:

E	execute
F	reference
O	own
R	read
S	son add
W	write

A maximum of ten user codes can be specified by the ACIN parameter. JSL CHanGe (CHG) statements specify additional users.

Default: The parameter is optional. If it is omitted, only the creator of the node automatically possesses all attributes. The creator is, by definition, the user identified by the user code most recently specified on a JOB or LIMIT statement in the job which creates the node. No other users are qualified.

NOTE

The creator of the node specifies the Access Control List of the node he creates and is automatically the owner of the node.

7-21 PARAMETER INTERACTIONS

The access control information parameter ACIN interacts with the remaining three security-creation parameters: access control state, ACST; reference control state, RFST; and son-add control state, SON. RFST restricts ACST and SON.

7-22 ACIN AND ACST

The Access Control List generated from the ACIN parameter specifies a valid user and the actions allowed. The Operating System examines the list when exercising the control specified by the ACST parameter. Operating System control varies according to the parameter values specified.

When ACST specifies NO (no access control), the Operating System does not examine the Access Control List.

When ACST specifies PART (partial access control), the Operating System examines the list for the write attribute. Only users having the write attribute, in the list, can read, write, and execute the node's file. All others are given read and execute only.

When ACST specifies FULL (full access control), the Operating System examines the list for all intents. Only users identified in the list may access the node. They may perform only those actions specified by the list.

7-23 ACIN AND RFST

The Access Control List generated from the ACIN parameter specifies whether a user may pass through the node during a catalog search, depending on the RFST parameter. The Operating System exercises the control specified by the RFST parameter and varies its action accordingly.

When RFST specifies NO (no reference control), any user may pass through the node. The Operating System does not examine the list and places no restrictions on passage.

When RFST specifies FULL (full reference control), the Operating System examines the list to determine if the user possesses the reference attribute. If so, he may pass; if not, he is denied passage, regardless of the node's other states.

7-24 ACIN AND SON

The Access Control List generated by ACIN specifies a valid user and his assigned attributes. The SON parameter establishes control over addition of sons to the root node. The Operating System varies its actions according to the parameter values specified.

When SON specifies NO (no son-add control) any user intending to add a son node may do so. The Operating System does not examine the list.

When SON specifies FULL (full son-add control) and the user intends to add a son node, the Operating System examines the list to determine the user's validity and his possession of the son-add attribute. The user can add a son node only if he possesses the correct attribute.

7-25 RFST, ACST AND SON

The RFST parameter restricts the ACST and SON parameters. The RFST parameter must allow passage through the root node before a file at the node may be accessed or before any son nodes can be added, regardless of the control states imposed by the ACST and SON parameters.

The ACST and SON parameters have no effect on each other, on the RFST parameter, or on ownership of the node.

7-26 THE CATN STATEMENT

The CATalog Node (CATN) statement initiates the entry of a node into an existing catalog. The statement also enables the user to construct an Access Control List and specify node characteristics. However, file version cannot be cataloged at the node. The CATN statement also enables the user to establish an additional security level in the catalog structure, particularly through reference and son-add control features. Only users with the son-add attributes, at the parent node, can use the CATN statement.

The general form of the CATN statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	CATN	b̸	pathname [, MXVR=maximum number of versions] [, ACST=$\left\{\begin{matrix}\underline{\text{NO}}\\ \text{PART}\\ \text{FULL}\end{matrix}\right\}$][, SON=$\left\{\begin{matrix}\underline{\text{NO}}\\ \text{FULL}\end{matrix}\right\}$] [, RFST=$\left\{\begin{matrix}\underline{\text{NO}}\\ \text{FULL}\end{matrix}\right\}$][, ACIN=access control information]

The verb CATN must appear in the operation field. The slash identifier, or the label symbol if the statement is labeled, is separated from the operation field by at least one blank. The operation and operands fields also must be separated by at least one blank.

Both positional and keyword parameters are contained in the operands field. The former is mandatory and must appear first in the operands field, while the latter may be listed in any order or may be omitted.

7-27 PATHNAME

The pathname consists of one to ten edgenames catenated by slashes (/). The pathname gives the location and name of the new node in the catalog. Only the first edgename in the pathname may be a synonym defined in a preceding PD statement.

Format: The new edgename must be a valid ASC symbol. The user (user code specified on the most recent LIMIT or JOB statement) must have the son-add attribute at the node to which the new node is attached, if the parent node is in the full son-add control state.

The last edgename of the pathname leads to the new node. The pathname truncated by the last edgename defines the node to which the new node will be attached.

All the edgenames of the pathname except the last must have been defined earlier to the catalog in the current or in a previous job; i.e., only one new node can be created per CATN statement.

Default: None. The pathname must be specified.

7-28 MAXIMUM VERSION (MXVR)

The maximum versions parameter specifies the maximum number of versions at this node that can exist simultaneously.

Format: The keyword MXVR precedes the maximum versions parameter value, which is an unsigned constant.

The value of the maximum versions parameter specifies the number of file versions at this node. The file is assumed to comprise N null versions, where N is the maximum number of versions.

N's value is limited to the range $1 \le N \le 64$. Versions are cataloged to this file modulo N; i.e., an attempt to catalog version N results in replacement of version 0 by the newer version.

No version is cataloged by this statement. Only the null versions are created, which contain no data.

Default: If maximum number of versions is not specified, only one version of the file is permitted at this node.

7-29 ACCESS CONTROL STATE (ACST)

The access control state parameter, identified by the keyword ACST, regulates the degree of control the Operating System maintains over access to this node.

Format: One of the keywords NO, PART, or FULL is the parameter value. No indicates there is no access control, meaning users may read, write and execute versions at the node. PART indicates partial access control, allowing all users to read and execute any file version at this node. FULL access control restricts use of versions at this node to users on the Access Control List.

Read, write and execute attributes are regulated through the access control state of the node. The user must possess the own attribute to change the node characteristics. Ownership of the node is established by creating the node or through the access control information parameter (ACIN).

Default: ACST may be omitted, thus removing all access control by the Operating System. In such a case, users have full execute, read and write attributes and the default value is NO. The default value does not imply any own attribute; users cannot change characteristics of the specified node unless they created the node or established ownership through the access control information parameter.

7-30 REFERENCE CONTROL STATE (RFST)

The reference control state parameter, identified by the keyword RFST, specifies whether the Operating System will control users' attempts to pass through the node during a catalog search.

Format: The parameter value is one of the keywords NO or FULL. NO indicates no reference control; any user may pass through the node during a catalog search. FULL specifies full reference control; only qualified users may pass through the node. The access control information parameter (ACIN) qualifies users by providing them with a reference attribute.

Default: The parameter is optional. If it is omitted, the Operating System will not maintain reference control. Any user can pass through the node during a search. The parameter default value is NO, a condition which does not affect the state of access control on the node.

7-31 SON-ADD CONTROL STATE (SON)

The son-add control state parameter establishes the degree of control exercised by the Operating System over the addition of nodes (called sons) to the node.

Format: The parameter is identified by the keyword SON. Its value is one of the keywords NO or FULL. NO specifies no son-add control; any user may add son nodes to the specified node. FULL specifies full son-add control. Only qualified users can add son nodes. The access control information parameter provides users with the proper attribute.

Default: The parameter is optional. If it is omitted, the default value is NO and any user can add son nodes.

7-32 ACCESS CONTROL INFORMATION (ACIN)

The access control information parameter, identified by the keyword ACIN, provides the most comprehensive security measures for the node being built. This parameter establishes valid users and determines the actions (attributes) each may take.

Format: The parameter value is a set of subparameters separated by slashes. Each subparameter is a valid user code, followed by an asterisk and one or more of the alphabetic characters E, F, O, R, S or W appended to the user code, i.e., user code *E*F*.... These values are translated into the Access Control List, maintained by the Operating System for the specified node.

Each user code becomes the System's identifier for a user permitted activity at the node. The alphabetic characters appended to the user code specify the attributes being allowed the user. The characters have the following interpretations:

E	execute
F	reference
O	own
R	read
S	son-add
W	write

A maximum of ten user codes can be specified by the ACIN parameter. JSL CHanGe (CHG) statements specify additional users.

Default: The parameter is optional. If it is omitted, only the creator of the node automatically possesses all attributes. The creator is, by definition, the user identified by the user code most recently specified on a JOB or LIMIT statement in the job which creates the node. No other users are qualified.

NOTE

> The creator of the node specifies the Access Control List of the node he creates and is automatically the owner of the node.

7-33 PARAMETER INTERACTIONS

The access control information parameter ACIN interacts with the remaining three security-creation parameters: access control state, ACST; reference control state, RFST; and son-add control state, SON. RFST restricts ACST and SON.

7-34 ACIN AND ACST

The Access Control List, generated from the ACIN parameter, specifies a valid user and the actions allowed. The Operating System examines the list when exercising the control specified by the ACST parameter. Operating System control varies according to the parameter values specified.

When ACST specifies NO (no access control), the Operating System does not examine the Access Control List; the user gets read, write, and execute access automatically.

When ACST specifies PART (partial access control), the Operating System examines the list for the write attribute. Write access is permitted if the user is identified by and possesses the write attribute specified by the Access Control List. The user automatically gets read and execute.

When ACST specifies FULL (full access control), the Operating System examines the list for all intents. Only users identified in the list and having read, write or execute attributes can access the node. They may perform only those actions specified by the list.

7-35 ACIN AND RFST

The Access Control List generated from the ACIN parameter specifies whether a user may pass through the node during a catalog search, depending on the RFST parameter. The Operating System exercises the control specified by the RFST parameter and varies its action accordingly.

When RFST specifies NO (no reference control), any user may pass through the node. The Operating System does not examine the list and places no restrictions on passage.

When RFST specifies FULL (full reference control), the Operating System examines the list to determine if the user possesses the reference attribute. If so, he may pass; if not, he is denied passage, regardless of the node's other states.

7-36 ACIN AND SON

The Access Control List generated by ACIN specifies a valid user and his assigned attributes. The SON parameter establishes control over addition of sons to the node. The Operating System varies its actions according to the parameter values specified.

When SON specifies NO (no son-add control), any user intending to add a son node may do so. The Operating System does not examine the list.

When SON specifies FULL (full son-add control) and the user intends to add a son node, the Operating System examines the list to determine the user's validity and his possession of the son-add attribute. The user can add a son node only if he possesses the correct attribute.

7-37 RFST, ACST AND SON

The RFST parameter restricts the ACST and SON parameters. The RFST parameter must allow passage through the node before a file at the node may be accessed or before any son nodes can be added, regardless of the control states imposed by the ACST and SON parameters.

The ACST and SON parameters have no effect on each other, on the RFST parameter, or on ownership of this node.

7-38 THE CHG STATEMENT

The CHanGe statement (CHG) specifies changes in the characteristics of a node. Only specified characteristics and user codes are changed; all others are unaffected.

This statement enables a qualified user to modify node security, the state of the automatic sequencing feature, and the versions specifications. The user attempting to implement changes through this statement must possess the own attribute at the specified node; otherwise, the operation is aborted. The user is identified by the usercode most recently specified on a LIMIT or JOB statement.

The general form of the CHG statement is:

	LABEL		OPERATION		OPERANDS
b̸	[Symbol]	b̸	CHG	b̸	pathname [, MXVR=maximum number of versions] [, FLVR=flagged version number] [, ACST=$\left\{\begin{array}{l}\underline{\text{NO}}\\ \text{FULL}\\ \text{PART}\end{array}\right\}$][, SON=$\left\{\begin{array}{l}\underline{\text{NO}}\\ \text{FULL}\end{array}\right\}$] [, RFST=$\left\{\begin{array}{l}\underline{\text{NO}}\\ \text{FULL}\end{array}\right\}$][, ACIN=access control information] [, RETP=retention[, VERS=version number]] [, ASEQ=$\left\{\begin{array}{l}\text{ON}\\ \text{OFF}\end{array}\right\}$]

The verb CHG must appear in the operation field. The label, operation and operands fields each are separated with at least one blank. The label is optional; if omitted, the slash (/) must be separated from the operation field with at least one blank.

The operands field contains both positional and keyword parameters. The positional parameter is mandatory and must appear first in the field. The keyword parameters are optional and may be coded in any convenient order.

7-39 PATHNAME

The node being changed must be identified to the catalog system. The pathname specifies the node.

Format: The pathname consists of one to ten edgenames concatenated by slashes (/). If the node is more than ten edgenames deep in the catalog, the first edgename may be defined as a synonym by a PD statement. The node's pathname must exist in the catalog by the time the CHG statement is encountered.

Default: None. The pathname parameter is both positional and mandatory. It must be specified so that the catalog system knows which node to change.

7-40 MAXIMUM VERSION (MXVR)

Since each node may have more than one version associated with it simultaneously, the user can specify a maximum number of versions that can be associated with the specified node.

Format: The maximum versions parameter, identified by the keyword MXVR, specifies this limit.

The parameter value is an unsigned constant. The range is $(M+1) \leq N \leq 64$, where N is the parameter value that specifies the number of versions allowed at the node and M is the largest existing version number or flagged version number, whichever is largest.

Default: The maximum versions parameter is optional. If it is omitted, the current maximum number of versions is retained.

7-41 FLAGGED VERSION (FLVR)

The flagged version parameter, identified by the keyword FLVR, specifies the absolute number of a new flagged version.

Format: The value of the parameter is an unsigned constant whose range is $0 \leq f \leq N-1$, where f is the flagged version value and N is the maximum number of allowed versions. The parameter value must be an absolute version reference. Also, the new flagged version can be a null version.

Default: The parameter is optional. The current flagged version of the file does not change if the parameter is omitted.

7-42 VERSION RETENTION PERIOD (RETP, VERS)

The version retention period parameter specifies the number of days that a file version output on tape will be retained and the version number to which the specified retention period applies. The Operating System converts the specified retention period to an expiration date associated with the existing version in the Tape Volume Catalog. Retention period is not a valid parameter for files on secondary disc.

Format: The version retention period parameter consists of two subparameters separated by a comma. Retention period subparameter, identified by the keyword RETP, is an unsigned constant specifying the number of days the file version is retained. The version subparameter, identified by the keyword VERS, is a signed or unsigned constant or a signed or unsigned JSL variable specifying the number of the version to which retention period pertains.

An unsigned version integer indicates an absolute version number reference. The absolute number is determined modulo N, where N is the maximum. A signed integer refers to a version number relative to the flagged version.

The absolute value is limited to the range $0 \leq |i| \leq 63$, where i is a signed integer. The value of an unsigned integer cannot exceed the maximum number of versions, N, minus one; therefore, the value's range is $0 \leq i \leq N-1$.

The value of a signed JSL variable may be any value within a variable range: $-2^{31} \leq v \leq 2^{31}-1$, where v denotes the variable. An unsigned JSL variable is a reference to an absolute version; its value is the same limits as an unsigned integer $-0 \leq v \leq N-1$, where v is the variable and N is the maximum number of versions.

Default: The two subparameters are dependent in the following manner. The RETP subparameter may be coded by itself. If the VERS subparameter is omitted, the system assumes that the retention period pertains to the flagged version. The VERS subparameter cannot be specified by itself. If RETP is omitted, the entire parameter has no meaning; VERS should not be used.

7-43 AUTOMATIC VERSION SEQUENCING (ASEQ)

When a user establishes a node in an existing catalog, the Operating System initiates an automatic sequencing feature for subsequently cataloged versions at that node. This feature is active until specifically deactivated. The CHG statement enables the user to specify the desired state of this feature.

Format: The automatic version sequencing parameter, identified by the keyword ASEQ, specifies a change in the state of the sequencing feature. The parameter value is either of the keywords ON or OFF. ON specifies activation of automatic sequencing; OFF specifies deactivation. Use of the keyword ON when the automatic sequencing feature is currently activated does not generate any action. Specifying OFF when the feature has been deactivated previously has no result.

Default: The ASEQ parameter is optional. If it is omitted, there is no change in the state of automatic sequencing.

7-44 ACCESS CONTROL STATE (ACST)

The access control state parameter, identified by the keyword ACST, specifies the degree of control the System will maintain over access to the node.

Format: The parameter value is one of the keywords NO, PART or FULL. NO specifies no access control, i.e., that all users may read, write and execute versions of the file at the node. PART specifies partial access control. This means that all users may only read and execute versions. FULL specifies full access control, restricting access to the versions at the node to qualified users only. Qualified users are those contained in the Access Control List established by the ACIN parameters or the creator of the node.

Read, write and execute attributes are controlled through the access control state of the node. Changing the node's characteristics - stipulated by the own attribute - is not permitted on the sole basis of access control. Ownership of the node is established through the access control information parameter (ACIN) or by having created the node.

Default: If the parameter is omitted, the current access control state is retained.

7-45 REFERENCE CONTROL STATE (RFST)

The reference control state parameter, identified by the keyword RFST, specifies whether the Operating System will control users' attempts to pass through the node during a catalog search.

Format: The parameter value is one of the keywords NO or FULL. NO specifies no reference control; any user may pass through the node during a catalog search. FULL specifies full reference control; only qualified users may pass through the node. The access control information parameter (ACIN) qualifies users by providing them with a reference attribute.

Default: The parameter is optional. If the parameter is omitted, the reference state of the node remains unchanged.

7-46 SON-ADD CONTROL STATE (SON)

The son-add control state parameter establishes the degree of control exercised by the Operating System over the addition of nodes (called sons) to the specified node.

Format: The parameter is identified by the keyword SON. Its value is one of the keywords NO or FULL. NO specifies no son-add control; any user may add son nodes to the specified node. FULL specifies full son-add control. Only qualified users can add son nodes. The access control information parameter (ACIN) provides users with the proper attribute.

Default: The parameter is optional. If it is omitted, the son-add control state remains unchanged.

7-47 ACCESS CONTROL INFORMATION (ACIN)

The access control information parameter, identified by the keyword ACIN, specifies additions to the Access Control List, modification of the attributes of users currently in the Access Control List, or deletion of users from the Access Control List.

Format: Each subparameter is a user code of from one to eight alphanumeric characters or is a user code followed by an asterisk and one or more of the letters E, F, O, R, S, or W each separated by asterisks. These values are translated into an Access Control List maintained by the Operating System for the specified node.

Each user code becomes the System's identifier for a user permitted activity at the node. The alphabetic characters appended to the user code specify the attributes being allowed the user. The attributes have the following interpretations:

E	execute
F	reference
O	own
R	read
S	son-add
W	write

An entry in the access control information parameter (ACIN) that consists only of a user code specifies the user code (and its attributes) to be deleted from the Access Control List of the node.

A user code with attributes specifies that a new entry is to be made in the list. If a user code with attributes is already in the list, the attributes in the Access Control list will be modified to correspond to the specification on the CHG statement; i.e., those attributes in the list but not the parameter will be deleted, and those in the parameter but not the list will be added.

Any user code currently in the Access Control List that is not listed in the access control information parameter (ACIN) remains unchanged in the Access Control List. Only those user codes in the parameter will be affected.

Each user code in the Access Control List after the CHG statement is processed in the system's identifier for a user who is to be permitted activity at this node. In subsequent jobs, any user who wants to access, reference or add sons to this node must specify one of these user codes in the job block containing any statement referencing the node. The node must, of course, have appropriate access control permitting these activities.

A maximum of ten user codes per CHG statement can be specified by the ACIN parameter. Additional CHG statements may be used to designate other users.

Default: The parameter is optional. If it is omitted, the contents of the Access Control List remain unchanged.

NOTE

> The current user cannot himself be deleted from the Access Control List and his possession of the own attribute cannot be deleted. The current user may delete any other owners. Current user is defined by the most recently specified user code on a LIMIT or JOB statement in the current job.

7-48 PARAMETER INTERACTIONS

The access control information parameter ACIN interacts with the remaining three security-creation parameters: access control state, ACST; reference control state, RFST; and son-add control state, SON. RFST restricts ACST and SON.

7-49 ACIN AND ACST

The Access Control List generated from the ACIN parameter specifies a valid user and the actions allowed. The Operating System examines the list when exercising the control specified by the ACST parameter. Operating System control varies according to the parameter values specified.

When ACST specifies NO (no access control), the Operating System does not examine the Access Control List; the user is given all access.

When ACST specifies PART (partial access control), the Operating System examines the list for the write attribute. Write access is permitted if the user is identified by and possesses the write attribute specified by the Access Control List.

When ACST specifies FULL (full access control), the Operating System examines the list for all intents. Only users identified in the list may access the node. They may perform only those actions specified by the list.

7-50 ACIN AND RFST

The Access Control List generated from the ACIN parameter specifies whether a user may pass through the node during a catalog search, depending on the RFST parameter. The Operating System exercises the control specified by the RFST parameter and varies its actions accordingly.

When RFST specifies NO (no reference control), any user may pass through the node. The Operating System does not examine the list and places no restrictions on passage.

When RFST specifies FULL (full reference control), the Operating System examines the list to determine if the user possesses the reference attribute. If so, he may pass; if not, he is denied passage, regardless of the node's other states.

7-51 ACIN AND SON

The Access Control List generated by ACIN specifies a valid user and his assigned attributes. The SON parameter establishes control over addition of sons to the node. The Operating System varies its actions according to the parameter values specified.

When SON specifies NO (no son-add control) any user intending to add a son node may do so. The Operating System does not examine the list.

When SON specifies FULL (full son-add control) and the user intends to add a son node, the Operating System examines the list to determine the user's validity and his possession of the son-add attribute. The user may add a son node only if he possesses the son-add attribute.

7-52 RFST, ACST AND SON

The RFST parameter restricts the ACST and SON parameters. The RFST parameter must allow passage through the node before a file at the node may be accessed or before any son nodes can be added, regardless of the control states imposed by the ACST and SON parameters.

The ACST and SON parameters have no effect on each other, on the RFST parameter, or on ownership of the node.

7-53 THE DEL STATEMENT

The DELete (DEL) statement initiates the deletion of a terminal node and its associated files from a catalog. It is implicit in the use of the DEL statement that the user possess the own attribute at the node specified by the pathname parameter; otherwise, the operation is aborted.

The general form of the DEL statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	DEL	b̸	pathname

The verb DEL must appear in the operation field. The slash identifier, or the label symbol if the statement is labeled, must be separated from the operation field by at least one blank. The operation and operands fields must be separated by at least one blank.

The operands field contains only the pathname parameter, which specifies the name and location of the node to be deleted.

7-54 PATHNAME

The pathname consists of one to ten edgenames catenated by slashes (/). Only the first edgename in the pathname can be a synonym defined in the PD statement.

While processing this statement, the last edgename of the pathname is deleted and this pathname no longer exists in the catalog. All files associated with the node also are deleted.

Only a terminal node can be deleted. An error terminates the statement if the node is not terminal.

7-55 THE CAT STATEMENT

The CATalog node and version statement (CAT) creates a node in an existing catalog and catalogs the first version of a specified file at that node. Since the statement deals with the catalog system on a node level, the valid user can specify node characteristics, construct an Access Control List, and provide additional security levels in the catalog structure. A qualified user must possess the son-add attribute at the parent node to use the CAT statement if the parent node is in the full son-add control state.

The CAT statement also serves a specialized function: it enables the user to catalog a tape file directly without buffering the file to secondary disc. The parameters applicable to this feature are exclusive; they are not valid for other cataloging functions.

The general form of the CAT statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	CAT	ƀ	pathname [, ACNM=access name] [, MXVR=maximum number of versions] $\left[, \text{ACST=}\begin{Bmatrix}\underline{\text{NO}}\\ \text{PART}\\ \text{FULL}\end{Bmatrix}\right]$ $\left[, \text{RFST=}\begin{Bmatrix}\underline{\text{NO}}\\ \text{FULL}\end{Bmatrix}\right]$ $\left[, \text{SON=}\begin{Bmatrix}\underline{\text{NO}}\\ \text{FULL}\end{Bmatrix}\right]$ [, ACIN=access control information] $\left[, \text{DTYP=}\begin{Bmatrix}\underline{\text{TAPE}}\\ \text{DSEC}\\ \text{HPT}\\ \text{PAD}\\ \text{OPID}\end{Bmatrix}\begin{Bmatrix}\underline{\text{NOCNTG}}\\ \text{CNTG}\end{Bmatrix} \text{operator id} \begin{Bmatrix}\underline{\text{CEOF}}\\ \text{CALL}\end{Bmatrix}\right]$ $\left[, \text{TRKS=}\begin{Bmatrix}\underline{9}\\ 7\end{Bmatrix}\right]$ $\left[, \text{RCTL=}\begin{Bmatrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{Bmatrix}\right]$ [, RETP=retention] $\left[, \text{SRC=}\begin{Bmatrix}\underline{\text{DSEC}}\\ \text{TP}\\ \text{FT}\end{Bmatrix}\right]$
					$\left[, \text{FORG=}\begin{Bmatrix}\underline{\text{PS}}\\ \text{DS}\end{Bmatrix}\right]$ $\left[, \text{LABL=}\left[\begin{Bmatrix}\underline{1}\\ \text{position}\end{Bmatrix}\right]\left[/\begin{Bmatrix}\underline{\text{SL}}\\ \text{NL}\end{Bmatrix}\right]\right]$ [, EFID= efid1/efid2/. . . /efidn] $\left[, \text{DEN=}\begin{Bmatrix}\underline{1600}\\ 800\\ 556\\ 356\\ 200\end{Bmatrix}\right]$ $\left[, \begin{Bmatrix}\text{BAND}\\ \text{SECT}\\ \text{WORD}\\ \text{BYTE}\end{Bmatrix}=\text{number}\right]$

The horizontal broken line separating parameters in the operands fields indicates special-function parameters. The parameters above the line are used for normal cataloging functions. Those below the line can be used only for special catalog processing.

The verb CAT must appear in the operand field. The label field, if the statement is labeled, must be separated from the operation field with at least one blank. If the statement is not labeled, the identifying slash (/) must be separated from the operation field with at least one blank. The operation and operands fields must be separated with at least one blank.

The parameters in the operands field are positional and keyword parameters. The positional parameter must be coded as the first parameter and cannot be omitted. The keyword parameters are optional and can be coded in any convenient order.

7-56 NORMAL NODE CREATION AND CATALOGING FUNCTIONS

Under normal circumstances, any file that is to be cataloged has been created on the secondary disc or is present there as a result of transfer actions for disc-buffering requirements of the Operating System. The CAT statement specifies the cataloging function; the Operating System handles any required physical transfers to external devices.

In the normal case, only certain parameters need be considered. The following list itemizes the applicable parameters:

pathname	
ACNM	- access name
MXVR	- maximum number of versions
ACST	- access control state
RFST	- reference control state
SON	- son-add control state
ACIN	- access control information
DTYP	- device type
RETP	- retention
SRC	- source

The remaining parameters apply only to a special function that bypasses disc-buffering requirements and should not be used in conjunction with the above parameters.

7-57 PATHNAME

The new node must be identified to the catalog system. The pathname parameter, a positional, mandatory parameter, specifies the name and location of the node being created. All nodes preceding the terminal node must exist prior to statement execution.

Format: The parameter value is one to ten edgenames concatenated by slashes (/). The first edgename may be a synonym defined in a PD statement. The pathname also specifies the location of the node in the catalog.

Default: None. Pathname must be specified.

7-58 ACCESS NAME (ACNM)

The access name parameter, identified by the keyword ACNM, identifies to the job the file that is to be cataloged as the first version at the specified node.

Format: The parameter value is the name by which the file is accessible to the job. Access name is mandatory for all normal cataloging functions. The access name must be a valid ASC symbol.

Default: The only exception to the mandatory status of the ACNM parameter is the special cataloging function. When the source parameter is set to the value designating the source device as tape (SRC = TP or FT), ACNM becomes optional. In this case, it should be omitted. ACNM has no meaning if a tape file is being cataloged under the special feature. The pathname, which is mandatory under all circumstances, identifies the file to be cataloged to the catalog system.

7-59 MAXIMUM VERSIONS (MXVR)

Since the new node may have more than one version of a file associated with it simultaneously, the user can specify a maximum number of versions that can be associated with the specified node. The maximum versions parameter, identified by the keyword MXVR, specifies this limit.

Format: The parameter value is an unsigned constant. The range is $1 \leq N \leq 64$, where N is the parameter value. This value specifies the number of versions allowed at the node. When CAT establishes the node, only the first version is cataloged. Other versions may be added later through CATV statements, up to the maximum specified by the MXVR parameter. The version cataloged is absolute version 0, the flagged version. It will remain the flagged version until another version is cataloged through the CATV statement or until a CHG statement specifies a different flagged version. Automatic sequencing at the node is activated by creating the node. This feature remains active, sequencing subsequent versions cataloged by CATV statements, until it is deactivated by a CHG statement.

Default: The maximum versions parameter is optional. If it is omitted, the operating System assumes that only one version will be permitted at the node.

7-60 ACCESS CONTROL STATE (ACST)

The access control state parameter, identified by the keyword ACST, specifies the degree of control the system will maintain over access to the node.

Format: The parameter value is one of the keywords NO, PART or FULL. NO specifies no access control; i.e., all users may read, write and execute versions of the file at the node. PART specifies partial access control. This means that all users may read and execute versions. FULL specifies full access control, restricting access to the versions at the node to qualified users only. Users are qualified by being specified in the Access Control List with the appropriate attributes.

Read, write and execute attributes are controlled through the access control state of the node. Changing the node's characteristics, stipulated by the own attribute, is not permitted on the sole basis of access control. Ownership of the node is established through the access control information parameter (ACIN) or by creation of the node.

Default: The parameter may be omitted. If the parameter is omitted, the Operating System will not maintain any access control. Users have full execute, read and write attributes. The parameter default value is NO. This condition does not imply any own attribute; users cannot change the characteristics of the node unless they created the node or have established ownership through the access control information parameter (ACIN).

7-61 REFERENCE CONTROL STATE (RFST)

The reference control state parameter, identified by the keyword RFST, specifies whether the Operating System will control users' attempts to pass through the node during a catalog search.

Format: The parameter value is one of the keywords NO or FULL. NO specifies no reference control; any user may pass through the node during a catalog search. FULL specifies full reference control; only qualified users may pass through the node. The access control information parameter (ACIN) qualifies users by providing them with a reference attribute.

Default: The parameter is optional. If the parameter is omitted, the Operating System will not maintain reference control. Any user can pass through the node during each search. The parameter default value is NO. This condition does not affect the state of access control on the node.

7-62 SON-ADD CONTROL STATE (SON)

The son-add control state parameter establishes the degree of control exercised by the Operating System over the addition of nodes (called sons) to the created node.

Format: The parameter is identified by the keyword SON. Its value is one of the keywords NO or FULL. NO specifies no son-add control; any user may add son nodes to the node. FULL specifies full son-add control. Only qualified users can add son nodes. The access control information parameter (ACIN) provides users with the proper attribute.

Default: The parameter is optional. If it is omitted, the default value is NO and any user can add son nodes.

7-63 ACCESS CONTROL INFORMATION (ACIN)

The access control information parameter, identified by the keyword ACIN, provides the most comprehensive security measures for the node being created. This parameter establishes valid users and determines the actions (attributes) each may take.

Format: The parameter value is a set of subparameters separated by slashes. Each subparameter is a valid user code, followed by an asterisk and one or more of the alphabetic characters E, F, O, R, S and W appended to the user code. Each character is also separated by an asterisk. These values are translated into an Access Control List maintained by Operating System for the specified node.

Each user code becomes the System's identifier for a user permitted activity at the node. The alphabetic characters appended to the user code specify the attributes being allowed the user. The attributes have the following interpretations:

E	execute
F	reference
O	own
R	read
S	son add
W	write

A maximum of ten user codes can be specified by the ACIN parameter. JSL CHG (CHanGe) statements specify additional users.

Default: The parameter is optional. If it is omitted, only the creator of the node automatically possesses all attributes. The creator is, by definition, the user identified by the job block user code specified in the job which creates the node. No other users are qualified.

In subsequent jobs, any user who wants to access, reference or add sons to this node must specify one of these user codes in the job block containing any statement referencing the node. The node must, of course, have appropriate access control permitting these activities.

NOTE

> The creator of a node specifies the Access Control List of the node he creates and is automatically the owner of the node.

7-64 PARAMETER INTERACTIONS

The access control information parameter (ACIN) interacts with the remaining three security-creation parameters: access control state, ACST; reference control state, RFST; and son-add control state, SON. RFST restricts ACST and SON.

7-65 ACIN AND ACST

The Access Control List generated from the ACIN parameter specifies a valid user and the actions allowed. The Operating System examines the list when exercising the control specified by the ACST parameter. Operating System control varies according to the parameter values specified.

When ACST specifies NO (no access control), the Operating System does not examine the Access Control List; the user is given all access.

When ACST specifies PART (partial access control), the Operating System examines the list for the write attribute. Write access is permitted if the user is identified by and possesses the write attribute specified by the Access Control List.

When ACST specifies FULL (full access control), the Operating System examines the list for all intents. Only users identified in the list may access the node. They may perform only those actions specified by the list.

7-66 ACIN AND RFST

The Access Control List generated from the ACIN parameter specifies whether a user may pass through the node during a catalog search, depending on the RFST parameter. The Operating System exercises the control specified by the RFST parameter and varies its action accordingly.

When RFST specifies NO (no reference control), any user may pass through the node. The Operating System does not examine the list and places no restrictions on passage.

When RFST specifies FULL (full reference control), the Operating System examines the list to determine if the user possesses the reference attribute.

If so, he may pass; if not, he is denied passage, regardless of the node's other states.

7-67 ACIN AND SON

The Access Control List generated by ACIN specifies a valid user and his assigned attributes. The SON parameter establishes control over addition of son to the node. The Operating System varies its actions according to the parameter values specified by SON.

When SON specifies NO (no son-add control), any user intending to add a son node may do so. The Operating System does not examine the list.

When SON specifies FULL (full son-add control) and the user intends to add a son node, the Operating System examines the list to determine the user's validity and his possession of the son-add attribute. The user may add a son node only if he possesses the son-add attribute.

7-68 RFST, ACST AND SON

The RFST parameter restricts the ACST and SON parameters. The RFST parameter must allow passage through the node before a file at the node may be accessed or before any son nodes can be added, regardless of the control states imposed by the ACST and SON parameters.

The ACST and SON parameters have no effect on each other, on the RFST parameter, or on ownership of the node.

7-69 DEVICE TYPE (DTYP)

The device type parameter is identified by the keyword DTYP and specifies if the file is to be allocated on the head-per-track disc, positioning-arm-disc, or the device specified in the Operating System default parameter table, or a specific channel or module specified by the third subparameter. It also indicates whether or not the file can be allocated without regard to physical contiguity on disc.

Format: The parameter value is a set of four subparameters separated by slashes. The first subparameter value pertains to file allocation and is one of the keywords TAPE, DSEC, HPT, PAD, or OPID. The meaning of these keywords is:

TAPE - allocate the file on a 9-track, 1/2-inch, 1600-bpi magnetic tape

DSEC - allocate the file on the device specified in the system default parameter file (default table)

HPT - allocate the file on the head-per-track disc

PAD - allocate the file on the positioning-arm-disc

OPID - allocate the file on the specific channel or module given by the third subparameter.

The second subparameter pertains to physical contiguity on disc of the file and its value is one of the keywords NOCNTG or CNTG where:

NOCNTG - file can be allocated without regard to contiguity. This is the default value.

CNTG - File must be allocated on physically contiguous disc; if the file is greater than or equal to a module; allocation on each module will be contiguous and the allocation will start at the beginning of each module.

The third subparameter is used in conjunction with the OPID keyword of the first subparameter and must be specified if OPID is used. The operator ID is a character string indicating a specific device related channel or module.

Examples:

"DM0" indicates H/T module 0, channel 0

"DCH0" indicates H/T channel 0

"PAD0" indicates PAD spindle 0, 980-0, and TCC-0

"PCH0" indicates 980-0, and TCC-0

The fourth subparameter pertains to the setting of the initial allocation of the cataloged copy of the file to the original files current end-of-file or current allocation. This subparameter is identified by one of the keywords CEOF or CALL where:

CEOF - means set the initial allocation of the cataloged copy of the file to the original file's current end-of-file.

CALL - means set the initial allocation of the cataloged copy of the file to the original file's current allocation.

Default: The DTYP parameter is optional. When it is omitted, the Operating System assumes TAPE and NOCNTG, which indicates the file will be allocated on 1/2-inch, 9-track, 1600 bpi magnetic tape and the file may be allocated without regard to contiguity.

Restrictions: If the OPID keyword is used as the first subparameter the third subparameter, operator ID, must be specified. Otherwise, the third subparameter must not be specified. When the first subparameter is TAPE, the second, third, and fourth subparameters are not valid.

Examples:

```
DTYP=OPID/CNTG/DM0
DTYP=HPT
DTYP=/CNTG
DTYP=OPID//DCH0
```

In the first example the file will be contiguous on the head-per-track disc module 0. In the second example the file will not necessarily be contiguous on the head-per-track disc. In the third example the file will be contiguous and on the device specified in the system default parameter table. In the fourth example the file will not necessarily be contiguous on the head-per-track disc channel 0.

7-70 RETENTION PERIOD (RETP)

When a tape file is cataloged by the CAT statement, the number of days the user wants it retained can be specified. The retention period parameter, identified by the keyword RETP, specifies the length of retention time.

Format: The parameter value is an unsigned constant.

The Operating System converts the specified number of days to an expiration date and records it in the Tape Volume Catalog for both standard labeled (SL) and unlabeled (NL) tape volumes. The Operating System also records the date on the standard labeled tapes, but not on unlabeled tapes.

Default: This parameter is optional. If omitted, the parameter is given a default value defined by the installation.

The retention period parameter is valid only for files that are output to tape volumes. DTYP=TAPE or in conjunction with SRC=TP or FT. If a file will remain on the secondary disc, RETP has no meaning. Therefore, if DTYP=DSEC, RETP should not be used.

7-71 SOURCE (SRC)

Accurate use of the CAT statement hinges on one parameter, the source parameter (SRC). The source parameter specifies the source device on which the to-be-cataloged file resides.

Format: The parameter is identified by the keyword SRC. Its value is one of the keywords: DSEC, TP or FT, one of which may be selected. The keyword DSEC specifies that the file is located on the secondary disc. TP specifies a half-inch tape volume and FT specifies a one-inch tape volume (field tape).

Default: If the source parameter is not specified, the Operating System assumes the default of DSEC (file location is secondary disc). Normal catalog functions are used.

Besides specifying the source device of the file, SRC flags the cataloging function as either normal or special processing. SRC=DSEC indicates normal processing. SRC=TP or FT indicates the special feature: the file is cataloged, but is not buffered to disc. It remains on tape.

7-72 SPECIAL CATALOGING FUNCTION

The special cataloging function of the CAT statement also creates a node in the catalog and catalogs the first version of the file. However, in this special case, the file being cataloged must be a tape file and is never buffered to disc for the cataloging activities.

All of the parameters pertinent to normal cataloging functions are required for this special feature except ACNM, DTYP. In addition, the following parameters must be considered:

LABL	- label
FORG	- file organization
EFID	- external file identification
BAND SECT WORD BYTE	- disc reservation
DEN	- tape density
TRKS	- tracks
RCTL	- recording control

7-73 FILE ORGANIZATION (FORG)

The file organization parameter specifies the organization of the file.

Format: The parameter is identified by the keyword FORG. It consists of one subparameter selected from two keywords: PS and DS.

PS - the file is organized sequentially.

DS - the file is organized as direct secondary.

Default: The default value is PS if the parameter is omitted. FORG is valid only when the source parameter (SRC) specifies that the to-be-cataloged file resides on half-inch or one-inch tape volumes. If SRC is not specified, its default value is secondary disc (DSEC); if SRC=DSEC, the parameter specifies secondary disc. If SRC=DSEC or defaults to DSEC, the FORG parameter is invalid and will be ignored if coded.

7-74 EXTERNAL FILE IDENTIFICATION (EFID)

The keyword EFID identifies the external file identification parameter. It specifies the identification numbers of the volumes that contain the cataloged file. Identifiers of all tape volumes containing the file must be declared. The EFID parameters must be coded in the sequence that the tapes are to be read.

Format: The parameter value is a character string of one to six characters. If the string is enclosed in quotation marks, blanks in the string are retained; if the string is not so enclosed, any embedded blanks are eliminated.

Tape identification labels are assigned to volumes by the installation at the time they are submitted or generated. The user is informed of the external labels of his files at that time.

Default: If SRC=DSEC or is not specified, the EFID parameter is invalid and ignored if coded.

7-75 LABEL (LABL)

The label parameter, identified by the keyword LABL, describes characteristics of the tape volume on which the file resides.

Format: The parameter is comprised of two subparameters: label type and file position. Both subparameters are positional, but not mandatory. Both must be separated by a slash (/), if used.

The first subparameter (position) specifies the desired position of the file on the tape volume. The subparameter value is either 1 or another constant, where the range is $1 \leq n \leq 255$.

The second subparameter (label type) specifies the type of label on the tape volume. The value is either the keyword SL or NL. SL specifies that the tape volume has standard labels. NL specifies no labels.

If SL is specified explicitly or by default and an EFID is specified, the System requests that the correct tape volume be supplied. If a tape volume cannot be verified as standard labeled, the operator must cancel the request.

If NL is specified and the tape volume cannot be verified as non-labeled, the operator must cancel the request.

The position and label type subparameters are interactive. These interactions are interpreted as:

Label Type	Position	Interpretation
SL	n	n = file sequence number. The number of physical tape marks to be skipped during positioning is a function of the standard label format.

Label Type	Position	Interpretation
NL	n	n = file sequence number. The number of physical tape marks to be skipped during positioning is n-1.

The LABEL parameter depends on the SRC and DTYP parameters. If SRC= DSEC or defaults to DSEC, the LABL parameter has no meaning. If DYTP= TAPE, the LABL parameter being specified will cause the operating system to abort the process.

Default: The label parameter is optional and may be omitted. In addition, each subparameter is optional. If the entire parameter is omitted, the Operating System assumes the following values:

position 1	position to first file on volume
label type SL	standard labels

If one of the subparameters is omitted, the Operating System assumes the appropriate value from those defined above. If the first subparameter is omitted, the separating slash must be used in front of the last subparameter. If the second subparameter is omitted, the separating slash should be left off. For example:

position subparameter omitted	LABL = /NL
label type subparameter omitted	LABL = 2

7-76 TAPE DENSITY (DEN)

The keyword DEN identifies the tape density parameter. DEN specifies the recording density in bits-per-inch-per-track for the tape volumes designated by the EFID parameters. All tape volumes specified must have the same density.

Format: The parameter value is 1600, 800, 712, 556, 356, or 200. The key 1600 specifies the tape density is 1600 bpi; 800 specifies 800 bpi; 712 specifies 712 bpi; etc. If the source parameter (SRC) specifies that the source device is half-inch tape (SRC=TP), either 1600, 800, 556 or 200 can be selected. If SRC specifies one-inch tape (SRC=FT), either 712 or 356 may be selected.

Default: The tape density parameter is optional. If omitted, the value is 1600 bpi.

The DEN parameter is invalid if SRC=DSEC or defaults to DSEC.

7-77 TAPE RECORDING TRACKS (TRKS)

The tape recording track parameter, identified by the keyword TRKS, specifies if the given tape is a seven-track or a nine-track tape.

Format: The parameter value is one of the keynumbers 9 or 7 to designate either the nine-track or the seven-track tape respectively.

Default: The tape recording track parameter can be omitted. The default is to nine-track tape. If the tape recording track parameter specification or default is different than the actual tape configuration the system aborts the input/output operation.

7-78 RECORDING CONTROL (RCTL)

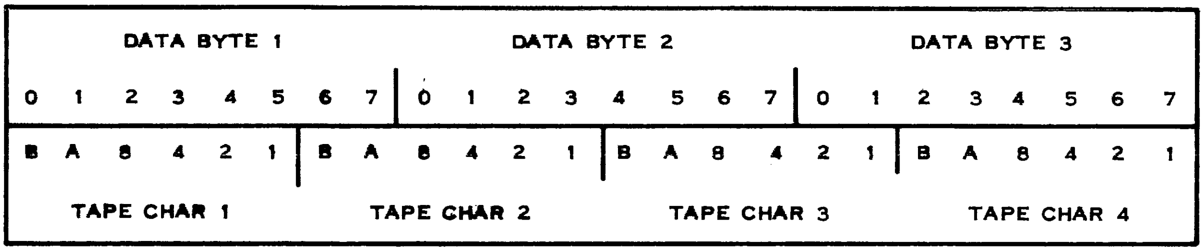
The recording control parameter allows for the specification of a parity, translation and data conversion mode. This parameter is applicable only if seven-track tape is being processed. It is identified by the keyword RCTL and the keyword values are:

O	- odd parity without translation or conversion
E	- even parity without translation or conversion
T	- odd parity with translation but no conversion
C	- odd parity with conversion but no translation
ET	- even parity with translation but no conversion.

Default: If the RCTL parameter is omitted, and seven-track tape is designated, the default is ET or even parity with translation but no conversion.

Data Translation: Data translation refers to the ability to transform data recorded in BCD interchange to EBCDIC and vice versa. Since each BCD code is a six-bit value, upon input the six bits are translated into the corresponding eight-bit EBCDIC value. For output operations the eight-bit EBCDIC value is translated into its six-bit BCD representation.

Data Conversion: The data conversion option makes it possible to write eight-bit binary data on seven-track tape. For output operations, every three bytes of data are written as four tape characters. For input operations, every four tape characters are written as three bytes of data. Figure 7-2 illustrates this data conversion.



DATA BYTE 1								DATA BYTE 2								DATA BYTE 3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
B	A	8	4	2	1	B	A	8	4	2	1	B	A	8	4	2	1	B	A	8	4	2	1
TAPE CHAR 1						TAPE CHAR 2						TAPE CHAR 3						TAPE CHAR 4					

Figure 7-2. Data Conversion

7-79 DISC RESERVATION (SECT, BAND, WORD OR BYTE)

The disc space parameter specifies the management of disc space for the named file throughout the job when it is assigned.

Format: The disc space parameter is identified by the keywords SECT, BAND, WORD or BYTE. Sector reservations are in units of 64 each. If the requested number of sectors is not a multiple of 64, the Operating System rounds it to the next higher multiple of 64.

SECT specifies the parameter in sector units (64 words each). BAND specifies the parameter in band units (256 sectors each).

WORD specifies that the parameter value is expressed in word units given in thousands (WORD=15 is 15,000 words). When the value specified is not a multiple of 64 sectors it is rounded up to the next highest multiple.

BYTE specifies that the parameter value is expressed in byte units given in thousands (BYTE=12 is 12,000 bytes). When this value as specified is not a multiple of 64 sectors it is rounded up to the next highest multiple.

The parameter value is an unsigned constant less than 999999_{10} and may be from one to six digits.

Default: The parameter is optional. If omitted, an installation-defined value is used. The parameter is valid only for special catalog processing. It is meaningless if SRC=DSEC or SRC is not specified (SRC defaults to DSEC).

7-80 PARAMETER INTERACTION SUMMARY

Parameter specification depends on the source (SRC) parameter. Certain parameters apply to normal cataloging functions; others to the special function. The following description summarizes which parameters may be used for each case, normal and special processing, respectively.

7-81 SRC = DSEC

When the source device is secondary disc, the file to be cataloged has been created on or buffered to the secondary disc. In cataloging this file, the following parameters apply:

pathname	mandatory
access name (ACNM)	mandatory
versions (MXVR)	optional
catalog security (ACST, RFST, SON, ACIN)	optional
output location (DTYP)	optional
file retention (RETP)	optional (valid only for tape files)

7-82 SRC = TP OR FT

When the source device is a half-inch or one-inch tape volume, the file to be cataloged exists but is not buffered to the secondary disc. In cataloging this file, the following parameters apply:

pathname	mandatory
versions (MXVR)	optional
catalog security (ACTS, RFST, SON, ACIN)	optional
file retention (RETP)	optional
label information (LABL)	optional
file organization (FORG)	optional
external file identifier (EFID)	mandatory
disc reservation $\begin{pmatrix}\text{BAND}\\\text{SECT}\\\text{WORD}\\\text{BYTE}\end{pmatrix}$	optional
tape density (DEN)	optional (mandatory for SRC=FT)
tracks (TRKS)	optional
recording control (RCTL)	optional

7-83 EXAMPLES

The following examples illustrate the use of the CAT statement for normal cataloging functions.

1. File Source is Secondary Disc

```
/ JSL
   ʃ
/ CAT A/B/C/D, ACNM=TFILE, MXVR=3, DTYP=DSEC;
  SRC=DSEC
   ʃ
/ JSL
```

Given Values

A/B/C/D — pathname - names file being cataloged and defines location of node being created. This parameter is mandatory.

ACNM=TFILE — TFILE identifies file to the job. This parameter is mandatory.

Given Values

MXVR=3 — Node can have three versions associated with it.

DTYP=DSEC — File will remain on disc after being cataloged.

SRC=DSEC — Specifies that the file to be cataloged is currently on disc. In this case, SRC could be omitted since its default value is DSEC.

Default Values

ACST=NO — No access control is established. Any user can access the node and read, write and/or execute versions of the file at that node.

RFST=NO — No reference control is established. Any user may pass through the node during a catalog search.

SON=NO — No son-add control is established. Any user may add son nodes to the node being created.

ACIN — Only the creator of the node automatically possesses all attributes. The creator is the user identified by the job block user code specified in the job which creates the node. No other users are qualified.

Invalid Parameters

RETP — Retention period is invalid for any file remaining on the secondary disc.

FORG
LABL
EFID
DEN
BAND
SECT
WORD
BYTE
TRKS
RCTL — These parameters apply only when SRC=TP or FT.

2. All Defaults Operative

```
/ JSL
  ʃ
/ CAT Q/R/S, ACNM=THIRSTY
```

Given Values

Q/R/S — pathname - names file being cataloged and defines location of node being created. This parameter is mandatory.

ACNM=THIRSTY — THIRSTY identifies file to the job. This parameter is mandatory.

Default Values

The default values are the same as described in the first example of file cataloging. The values are summarized in the following list:

MXVR = 1

ACST = NO

RFST = NO

SON = NO

ACIN = all attributes for creator of node

DTYP = TAPE

RETP = installation-defined value

SRC = DSEC

Invalid Parameters

FORG
LABL
EFID
DEN
TRKS
RCTL
BAND
SECT
WORD
BYTE

} apply only when SRC=TP or FT.

The following examples illustrate the use of the CAT statement for special catalog processing.

3. Half-Inch Tape Source

```
/ JSL
   ʃ
/ CAT T/R, MXVR=3, SRC=TP, FORG=DS, LABL=2/NL;
  EFID=2340
   ʃ
/ JSL
```

Given Values

T/R — pathname - names file being cataloged and defines location of node being created. This parameter is mandatory.

MXVR=3 — Node can have three versions associated with it.

SRC=TP — The source device of the file to be cataloged is a nine-track, half-inch, magnetic tape volume. This parameter also signifies that special catalog processing is used.

FORG=DS — The file being cataloged has a direct secondary organization.

LABL=2/NL — When the cataloged file is subsequently input to a job, File Management uses this information to input the tape file. The parameter specifies that positioning is to the second file and that the tape volume is unlabeled. For all unlabeled tape volumes, a JSL FD statement must be used when the file is subsequently assigned to a job to specify the file's logical characteristics.

EFID=2340 — The external volume identifier specifies that the file is contained on tape volume 2340. This parameter is mandatory when SRC=TP or FT.

Default Values

ACST=NO — No access control is established. Any user can access the node and read, write and/or execute versions at that node.

RFST=NO — No reference control is established. Any user may pass through the node during a catalog search.

SON=NO — No son-add control is established. Any user may add son nodes to the node being created.

ACIN — Only the creator of the node possesses all attributes. The creator is the user identified by the job block user code specified in the job which creates the node. No other users are qualified.

Default Values

RETP — The System assumes an installation-defined value to be converted to an expiration date. The expiration date is written to the Tape Volume Catalog, but not to the specified unlabeled tapes.

DEN — The System assumes 1600 bpi as the density value of the tape volume containing the specified file.

TRKS — The default value is nine-track tape.

RCTL — Recording control is not applicable for nine track tape.

BAND
SECT
WORD
BYTE — The System assumes an installation-defined value for disc space allocation when the cataloged file is used.

Invalid Parameters

ACNM }
DTYP } — These parameters apply only when SRC=DSEC.

4. One-Inch Tape Source

```
/ JSL
  ʃ
/ CAT S/V, SRC=FT, DEN=712, EFID=10000
  ʃ
/ JSL
```

Given Values

S/V — The pathname S/V names the cataloged file and determines the location of the node being created. This parameter is mandatory.

SRC=FT — A file existing on one-inch magnetic tape must be specified as FT on the SRC parameter. DSEC designates the source device as secondary disc; TP designates half-inch tape.

DEN=712 — The density is specified as 712 bpi; this parameter value is mandatory for one-inch tapes.

EFID=10000 — The external file identifier specifies the file is contained on tape volume 10000. This parameter is mandatory for SRC=TP or FT.

Default Values

FORG = PS

LABL = 1/SL

MXVR = 1

ACST = NO

RFST = NO

SON = NO

ACIN (creator gets all attributes)

RETP (installation defined value)

BAND
SECT {installation defined values}
WORD
BYTE

Invalid Parameters

ACNM

DTYP

TRKS

RCTL

7-84 THE CATV STATEMENT

The CATalog Version (CATV) statement initiates the entry of a version at an existing node in the catalog. The statement does not deal with the catalog system on a node level; hence, catalog protection must have been established previously when the node was constructed. The user must possess the own attribute to use the CATV statement.

The version number assigned to the file being cataloged is calculated by the Operating System by adding one to the version number last cataloged with a CAT or CATV statement. If the new number equals the current maximum number of versions at the node, it is set to zero. Also, if the automatic sequencing flag is on in the node, the new version becomes the flagged version. If the automatic sequencing flag is off, the flagged version is not changed.

Like the CAT statement, the CATV statement also serves a specialized function: it enables the user to catalog a tape file directly without buffering the file to secondary disc. The parameters applicable to this feature are exclusive; they are not valid for other cataloging functions.

The CATV statement resembles the CAT statement, except that the parameters dealing with the node structure of the catalog are not a part of CATV.

The general form of the CATV statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	CATV	ƀ	pathname [, ACNM=access name] $[,\text{DTYP=}\left\{\begin{matrix}\underline{\text{TAPE}}\\\text{DSEC}\\\text{HPT}\\\text{PAD}\\\text{OPID}\end{matrix}\right\}/\left\{\begin{matrix}\underline{\text{NOCNTG}}\\\text{CNTG}\end{matrix}\right\}/$ $\text{operator ID}/\left\{\begin{matrix}\underline{\text{CEOF}}\\\text{CALL}\end{matrix}\right\}]$ [, RETP=retention] $[,\text{SRC=}\left\{\begin{matrix}\underline{\text{DSEC}}\\\text{TP}\\\text{FT}\end{matrix}\right\}]$
					$[,\text{FORG=}\left\{\begin{matrix}\underline{\text{PS}}\\\text{DS}\end{matrix}\right\}]$ $[,\text{LABL=}[\left\{\begin{matrix}\underline{1}\\\text{position}\end{matrix}\right\}][/\left\{\begin{matrix}\underline{\text{SL}}\\\text{NL}\end{matrix}\right\}]]$ [, EFID=efid1/efid2/.../efidn] $[,\text{DEN=}\left\{\begin{matrix}\underline{1600}\\800\\712\\556\\356\\200\end{matrix}\right\}][\text{TRKS=}\left\{\begin{matrix}\underline{9}\\7\end{matrix}\right\}]$ $[,\text{RCTL=}\left\{\begin{matrix}\underline{\text{ET}}\\\text{E}\\\text{O}\\\text{T}\\\text{C}\end{matrix}\right\}]$ $[,\left\{\begin{matrix}\text{BAND}\\\text{SECT}\\\text{WORD}\\\text{BYTE}\end{matrix}\right\}=\text{number}]$

The horizontal, broken line separating parameters in the operands field indicates special-function parameters. The parameters above the line are used for normal cataloging functions. Those below the line can be used only for special catalog processing.

The verb CATV must appear in the operation field. The label field, if the statement is labeled, must be separated from the operation field with at least one blank. If the statement is not labeled, the identifying slash (/) must be separated from the operation field with at least one blank. The operation and operands fields must be separated with at least one blank.

The parameters in the operands field are positional and keyword parameters. The positional parameter must be coded as the first parameter and cannot be omitted. The keyword parameters are optional and can be coded in any convenient order.

7-85 NORMAL CATALOGING FUNCTIONS

Under normal circumstances, any file that is to be cataloged has been created on the secondary disc or is present there as a result of transfer actions for disc-buffering requirements of the Operating System. The CATV statement specifies the cataloging function; the Operating System handles any required physical transfers to external devices.

In the normal case, only certain parameters need be considered. The following list itemizes the applicable parameters:

pathname

ACNM - access name

DTYP - device type

RETP - retention

SRC - source

The remaining parameters apply only to a special function that bypasses disc-buffering requirements and should not be used in conjunction with the above parameters.

7-86 PATHNAME

The new node must be identified to the catalog system. The pathname parameter, a positional, mandatory parameter, specifies the name and location of the node in the catalog.

Format: The parameter value is one to ten edgenames concatenated by slashes (/). The first edgename may be a synonym defined in a PD statement. The pathname also specifies the location of the node in the catalog.

Default: None. Pathname must be specified.

7-87 ACCESS NAME (ACNM)

The access name parameter, identified by the keyword ACNM, identifies to the job the file that is to be cataloged as a version at the specified node.

Format: The parameter value is the name by which the file is accessible to the job. Access name is mandatory for all normal cataloging functions. The access name must be a valid ASC symbol.

Default: The only exception to the mandatory status of the ACNM parameter is the special cataloging function. When the source parameter is set to the value designating the source device as tape (SRC = TP or FT), ACNM becomes optional. In this case, it should be omitted. ACNM has no meaning if a tape file is being cataloged under the spcial feature. The pathname, which is mandatory under all circumstances, identifies the file to be cataloged to the catalog system.

7-88 DEVICE TYPE (DTYP)

The device type parameter is identified by the keyword DTYP and specifies if the file is to be allocated on the head-per-track disc, positioning-arm-disc, or the device specified in the Operating System default parameter table, or a specific channel or module specified by the third subparameter. It also indicates whether or not the file can be allocated without regard to physical contiguity on disc.

Format: The parameter value is a set of four subparameters separated by slashes. The first subparameter value pertains to file allocation and is one of the keywords TAPE, DSEC, HPT, PAD, or OPID. The meaning of these keywords is:

TAPE - allocate the file on a 9-track, 1/2-inch, 1600-bpi magnetic tape

DSEC - allocate the file on the device specified in the system default parameter file (default table)

HPT - allocate the file on the head-per-track disc

PAD - allocate the file on the positioning-arm-disc

OPID - allocate the file on the specific channel or module given by the third subparameter.

The second subparameter pertains to physical contiguity on disc of the file and its value is one of the keywords NOCNTG or CNTG where:

NOCNTG - file can be allocated without regard to contiguity. This is the default value.

CNTG - File must be allocated on physically contiguous disc; if the file is greater than or equal to a module; allocation on each module will be contiguous and the allocation will start at the beginning of each module.

The third subparameter is used in conjunction with the OPID keyword of the first subparameter and must be specified if OPID is used. The operator ID is a character string indicating a specific device related channel or module.

Examples:

"DM0" indicates H/T module 0, channel 0

"DCH0" indicates H/T channel 0

"PAD0" indicates PAD spindle 0, 980-0, and TCC-0

"PCH0" indicates 980-0, and TCC-0

The fourth subparameter pertains to the setting of the initial allocation of the cataloged copy of the file to the original files current end-of-file or current allocation. This subparameter is identified by one of the keywords CEOF or CALL where:

CEOF - means set the initial allocation of the cataloged copy of the file to the original file's current end-of-file.

CALL - means set the initial allocation of the cataloged copy of the file to the original file's current allocation.

Default: The DTYP parameter is optional. When it is omitted, the Operating System assumes TAPE and NOCNTG, which indicates the file will be allocated on 1/2-inch, 9-track, 1600 bpi magnetic tape and the file may be allocated without regard to contiguity.

Restrictions: If the OPID keyword is used as the first subparameter the third subparameter, operator ID, must be specified. Otherwise, the third subparameter must not be specified. When the first subparameter is TAPE, the second, third, and fourth subparameters are not valid.

Examples:

```
DTYP=OPID/CNTG/DM0

DTYP=HPT

DTYP=/CNTG

DTYP=OPID//DCH0
```

In the first example the file will be contiguous on the head-per-track disc module 0. In the second example the file will not necessarily be contiguous on the head-per-track disc. In the third example the file will be contiguous and on the device specified in the system default parameter table. In the fourth example the file will not necessarily be contiguous on the head-per-track disc channel 0.

7-89 RETENTION PERIOD (RETP)

When a tape file is cataloged by the CATV statement, the number of days the user wants it retained can be specified. The retention period parameter, identified by the keyword RETP, specifies the length of retention time.

Format: The parameter value is an unsigned constant.

The Operating System converts the specified number of days to an expiration date and records it in the Tape Volume Catalog for both standard labeled (SL) and unlabeled (NL) tape volumes. The Operating System also records the date on the standard labeled tapes, but not on unlabeled tapes.

Default: This parameter is optional. If omitted, the parameter is given a default value defined by the installation.

The retention period parameter is valid only for files that are output to tape volumes. DTYP=TAPE or in conjunction with SRC=TP or FT. If a file will remain on the secondary disc, RETP has no meaning. Therefore, if DTYP=DSEC, RETP should not be used.

7-90 SOURCE (SRC)

Accurate use of the CATV statement hinges on one parameter, the source parameter (SRC). The source parameter specifies the source device on which the to-be-cataloged file resides.

Format: The parameter is identified by the keyword SRC. Its value is one of the keywords: DSEC, TP or FT, one of which may be selected. The keyword DSEC specifies that the file is located on the secondary disc. TP specifies a half-inch tape volume and FT specifies a one-inch tape volume.

Default: If the source parameter is not specified, the Operating System assumes the default of DSEC (file location is secondary disc). Normal catalog functions are used.

Besides specifying the source device of the file, SRC flags the cataloging function as either normal or special processing. SRC=DSEC indicates normal processing. SRC=TP or FT indicates the special feature; the file is cataloged, but is not buffered to disc. It remains on tape.

7-91 SPECIAL CATALOGING FUNCTION

The special cataloging function of the CATV statement also catalogs a version at an existing node. However, in this special case, the file being cataloged must be a tape file and is never buffered to disc for the cataloging activities.

All of the parameters pertinent to normal cataloging functions are required for this special feature except ACNM and DTYP. In addition, the following parameters must be considered:

LABL	- label
FORG	- file organization
EFID	- external file identification
BAND SECT WORD BYTE	- disc reservation
DEN	- tape density
TRKS	- track
RCTL	recording control

7-92 FILE ORGANIZATION (FORG)

The file organization parameter specifies the organization of the file.

Format: The parameter is identified by the keyword FORG. It consists of one subparameter selected from two keywords: PS and DS.

PS - the file is organized sequentially.

DS - the file is organized as direct secondary.

The default value is PS if the parameter is omitted.

Default: FORG is valid only when the source parameter (SRC) specifies that to-be-cataloged file resides on half-inch or one-inch tape volumes. If SRC is not specified, its default value is secondary disc (DSEC); if SRC=DSEC, the parameter specifies secondary disc. If SRC=DSEC or defaults to DSEC, the FORG parameter is invalid and will be ignored if coded.

7-93 LABEL (LABL)

The label parameter, identified by the keyword LABL, describes characteristics of the tape volume on which the file resides.

Format: The parameter is comprised of two subparameters: label type and file position. Both subparameters are positional, but not mandatory. Both must be separated by a slash (/), if used.

The first subparameter (position) specifies the desired position of the file on the tape volume. The subparameter value is either 1 or another constant, where the range is $1 \leq n \leq 255$.

The second subparameter (label type) specifies the type of label on the tape volume. The value is either the keyword SL or NL. SL specifies the tape volume has standard labels. NL specifies no labels.

If SL is specified explicitly or by default and an EFID is specified, the System requests that the correct tape volume be supplied. If a tape volume cannot be verified as standard labeled, the operator must cancel the request.

If NL is specified and the tape volume cannot be verified as non-labeled, the operator must cancel the request.

The position and label type subparameters are interactive. These interactions are interpreted as:

Label Type	Position	Interpretation
SL	n	n = file sequence number. The number of physical tape marks to be skipped during positioning is a function of the standard label format.
NL	n	n = file sequence number. The number of physical tape marks to be skipped during positioning is n-1.

The LABL parameter depends on the SRC and DTYP parameters. If SRC=DSEC or defaults to DSEC, the LABL parameter has no meaning. If DTYP=TAPE, the LABL parameter being specified will cause the Operating System to abort the process.

Default: The label parameter is optional and may be omitted. In addition, each subparameter is optional. If the entire parameter is omitted, the Operating System assumes the following values:

position 1	position to first file on volume
label type SL	standard labels

If one of the subparameters is omitted, the Operating System assumes the appropriate value from those defined above. If the first subparameter is omitted, the separating slash must be used in front of the last subparameter. If the second subparameter is omitted, the separating slash should be left off. For example:

position subparameter omitted	LABL = /NL
label type subparameter omitted	LABL = 2

7-94 EXTERNAL FILE IDENTIFICATION (EFID)

The keyword EFID identifies the external file identification parameter. It specifies the identification numbers of the volumes that contain the cataloged file. Identifiers of all tape volumes containing the file must be declared. The EFID parameters must be coded in the sequence that the tapes are to be read.

Format: The parameter value is a character string of one to six characters. If the string is enclosed in quotation marks, blanks in the string are retained; if the string is not so enclosed, any embedded blanks are eliminated.

Tape identification labels are assigned to volumes by the installation at the time they are submitted or generated. The user is informed of the external labels of his files at that time.

Default: If SRC=DSEC or is not specified, the EFID parameter is invalid and ignored if coded.

7-95 TAPE DENSITY (DEN)

The keyword DEN identifies the tape density parameter. DEN specifies the recording density in bits-per-inch-per-track for the tape volumes designated by the EFID parameters. All tape volumes specified must have the same density.

Format: The parameter value is 1600, 800, 712, 556, 356, or 200. The key 1600 specifies the tape density is 1600 bpi; 800 specifies 800 bpi; 712 specifies 712 bpi; etc. If the source parameter (SRC) specifies that the source device is half-inch tape (SRC=TP), 1600, 800, 556 or 200 may be selected. If SRC specifies one-inch tape (SRC=FT), either 712 or 356 may be selected.

Default: The tape density parameter is optional. If omitted, the value is 1600 bpi for nine-track tape and 800 bpi for seven-track tapes.

The DEN parameter is invalid if SRC=DSEC or defaults to DSEC.

7-96 TAPE RECORDING TRACKS (TRKS)

The tape recording track parameter, identified by the keyword TRKS, specifies if the given tape is a seven-track or a nine-track tape.

Format: The parameter value is one of the keynumbers 9 or 7 to designate either the nine-track or the seven-track tape respectively.

Default: The tape recording track parameter can be omitted. The default is to nine-track tape. If the tape recording track parameter specification or default is different than the actual tape configuration the system aborts the input/output operation.

7-97 RECORDING CONTROL (RCTL)

The recording control parameter allows for the specification of a parity, translation and data conversion mode. This parameter is applicable only if seven-track tape is being processed. It is identified by the keyword RCTL and the keyword values are:

O - odd parity without translation or conversion

E - even parity without translation or conversion

T - odd parity with translation but no conversion

C - odd parity with conversion but no translation

ET - even parity with translation but no conversion.

Default: If the RCTL parameter is omitted, and seven-track tape is designated, the default is ET or even parity with translation but no conversion.

Data Translation: Data translation refers to the ability to transform data recorded in BCD interchange to EBCDIC and vice versa. Since each BCD code is a six-bit value, upon input the six bits are translated into the corresponding eight-bit EBCDIC value. For output operations the eight-bit EBCDIC value is translated into its six-bit BCD representation.

Data Conversion: The data conversion option makes it possible to write eight-bit binary data on seven-track tape. For output operations, every three bytes of data are written as four tape characters. For input operations, every four tape characters represent three bytes of data. Figure 7-2 illustrates this data conversion.

7-98 DISC RESERVATION (BAND, SECT, WORD OR BYTE)

The disc space parameter specifies the management of disc space for the named file throughout the job when it is assigned.

Format: The disc space parameter is identified by the keywords BAND, SECT, WORD or BYTE. Sector reservations are in units of 64 each. If the requested number of sectors is not a multiple of 64, the Operating System rounds it to the next higher multiple of 64.

BAND specifies the parameter in band units (256 sectors each). SECT specifies the parameter in sector units (64 words each).

WORD specifies that the parameter value is expressed in word units given in thousands (WORD=15 is 15,000 words). When the value specified is not a multiple of 64 sectors it is rounded up to the next highest multiple.

BYTE specifies that the parameter value is expressed in byte units given in thousands (BYTE=12 is 12,000 bytes). When this value as specified is not a multiple of 64 sectors it is rounded up to the next highest multiple.

The parameter value is an unsigned constant less than 999999_{10} and may be from one to six digits.

Default: The parameter is optional. If omitted, an installation-defined value is used. The parameter is valid only for special catalog processing. It is meaningless if SRC=DSEC or SRC is not specified (SRC defaults to DSEC).

7-99 PARAMETER INTERACTION SUMMARY

Parameter specification depends on the source (SRC) parameter. Certain parameters apply to normal cataloging functions; others to the special function. The following description summarizes which parameters may be used for each case, normal and special processing, respectively.

7-100 SRC = DSEC

When the source device is secondary disc, the file to be cataloged has been created on or buffered to the secondary disc. In cataloging this file, the following parameters apply:

pathname	mandatory
access name (ACNM)	mandatory
output location (DTYP)	optional
file retention (RETP)	optional (valid only for tape files)

7-101 SRC = TP OR FT

When the source device is a half-inch or one-inch tape volume, the file to be cataloged exists but is not buffered to the secondary disc. In cataloging this file, the following parameters apply:

pathname	mandatory
file retention (RETP)	optional
label information (LABL)	optional
file organization (FORG)	optional
external file identifiers (EFID)	mandatory
disc reservation (BAND SECT WORD BYTE)	optional
tape density (DEN)	optional (mandatory for SRC=FT)

tracks (TRKS)	optional (half-inch tape)
recording control (RCTL)	optional (seven-track tape)

7-102 EXAMPLES

The following examples illustrate the use of the CATV statement for normal cataloging functions.

1. File Source is Secondary Disc

```
/ JSL
  ∫
/ CATV A/B/C/D, ACNM=TFILE, DTYP=DSEC, SRC=DSEC
  ∫
/ JSL
```

Given Values

A/B/C/D	pathname - names file being cataloged and defines location of node. This parameter is mandatory.
ACNM=TFILE	TFILE identifies file to the job. This parameter is mandatory.
DTYP=DSEC	File will remain on disc after being cataloged.
SRC=DSEC	Specifies that the file to be cataloged is currently on disc. In this case, SRC could be omitted since its default value is DSEC.

Invalid Parameters

RETP	Retention period is invalid for any file remaining on the secondary disc.
FORG LABL EFID TRKS RCTL DEN BAND SECT WORD BYTE	These parameters apply only when SRC=TP or FT.

2. File - All Defaults Operative

```
/ JSL
  ʃ
/ CATV Q/R/S, ACNM=THIRTY
```

Given Values

Q/R/S — pathname - names file being cataloged and defines location of node. This parameter is mandatory.

ACNM=THIRTY — THIRTY identifies file to the job. This parameter is mandatory.

Default Values

The default values are the same as described in the first example of file cataloging. The values are summarized in the following list:

DTYP = TAPE

RETP = installation-defined value

SRC = DSEC

Invalid Parameters

FORG
LABL
EFID
TRKS
RCTL
DEN
BAND
SECT
WORD
BYTE

} apply *only* when SRC=TP or FT.

The following examples illustrate the user of the CATV statement for special catalog processing.

3. Half-Inch Tape Source

```
/ JSL
  ʃ
/ CATV T/R, SRC=TP, FORG=DS, LABL=2/NL, EFID=2340
  ʃ
/ JSL
```

Given Values

T/R

pathname - names file being cataloged and defines location of node. This parameter is mandatory.

SRC=TP

The source device of the file to be cataloged is half-inch, magnetic tape volume. This parameter also signifies that special catalog processing is used.

FORG=DS

The file being cataloged has a direct secondary organization.

LABL=2/NL

When the cataloged file is subsequently input to a job, File Management uses this information to input the tape file. The parameter specifies that positioning is to the second file and that the tape volume is unlabeled. For all unlabeled tape volumes, a JSL FD statement must be used when the file is subsequently assigned to a job to specify the file's logical characteristics.

EFID=2340

The external volume identifier specifies that the file is contained on tape volume 2340. The parameter is mandatory when SRC=TP or FT.

Default Values

RETP

The System assures an installation-defined value to be converted to an expiration date. The expiration date is written to the Tape Volume Catalog, but not to the specified unlabeled tapes.

DEN

The System assumes 1600 bpi as the density value of the tape volume containing the specified file.

TRKS

The default is nine track.

RCTL

Not applicable for nine track tape.

BAND
SECT
WORD
BYTE

The System assumes an installation-defined value for disc space allocation when the cataloged file is used.

Invalid Parameters

ACNM
DTYP

These parameters apply only when SRC=DSEC.

4. One-Inch Tape Source

```
/ JSL
   ∫
/ CATV S/V, SRC=FT, DEN=712, EFID=10000
   ∫
/ JSL
```

Given Values

S/V

The pathname S/V names the cataloged file and defines the location of the node. This parameter is mandatory.

SRC=FT

A file existing on one-inch magnetic tape must be specified as FT on the SRC parameter. TP designates half-inch tape.

DEN=712

The density is specified to be 712 bpi; this parameter is mandatory for one-inch tapes.

EFID=10000

The external file identifier specifies the file is contained on tape volume 10000. This parameter is mandatory for SRC=TP or FT.

Default Values

FORG = PS

LABL = 1/SL

MXVR = 1

ACST = NO

RFST = NO

SON = NO

ACIN (creator gets all attributes)

RETP
BAND
SECT installation defined value
WORD
BYTE

Invalid Parameters

ACNM
DTYP
TRKS
RCTL

7-103 THE DELV STATEMENT

The DELete Version (DELV) statement deletes a specified version of a file at an existing node. Through an optional feature, the statement also deletes all versions of a file cataloged at a node. Deletion means that the version becomes null. The process does not alter the specified maximum number of versions that can be cataloged at the node. In addition, DELV does not change the flagged version number and does not affect the version number cataloged at the node by a subsequent version-cataloging JSL statement (CATV). To use the DELV statement, the user must possess the own attribute at the named node. If not, the Operating System aborts the operation.

The general form of the DELV statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	DELV	b̸	pathname [, VERS=version number] [, OPT=(option code)]

The verb DELV must appear in the operation field. The statement may be labeled. Each field must be separated by at least one blank. The operands field contains three parameters. The positional parameter is mandatory and must be coded as the first parameter. The keyword parameters are optional and may be coded in any convenient order.

7-104 PATHNAME

The catalog node containing the file version to be deleted must be identified to the Operating System. The pathname parameter names and locates the node in the catalog.

Format: The pathname consists of one to ten edgenames concatenated by slashes (/). The first edgename may be a synonym defined by a JSL PD statement in the current job. The pathname must exist by the time the DELV statement specifies it. It may be defined in the current job or in a previous job.

Default: None. Pathname must be specified.

7-105 VERSION NUMBER (VERS)

The version parameter, identified by the keyword VERS, initiates the deletion of a specified version of an existing node.

Format: The parameter value is a signed or unsigned constant, or a signed or unsigned Job Specification Language variable. An unsigned value refers to the absolute version number of the file. The absolute version number is determined modulo N, where N is the maximum number of versions allowed the file. A signed value refers to a version relative to the flagged version.

The limits for the parameter values are:

signed integer, i	$0 \leq \|i\| \leq 63$
unsigned integer, i	$0 \leq \|i\| \leq N-1$, where N = maximum number of versions.
signed JSL variable, v	$-2^{31} \leq v \leq 2^{31}-1$
unsigned JSL variable, v	$0 \leq v \leq N-1$, where N = maximum number of versions.

The maximum number of versions has been previously specified by the catalog statements that created the node - CATBLD, CATN or CAT - or has been specified by a modifying statement - CHG. The absolute number of the current flagged version is a function of either the number of versions cataloged by CATV statements or the number fixed by a CHG statement.

Default: The version parameter may be omitted. If so, the Operating System deletes the flagged version.

7-106 OPTION (OPT)

The option parameter, identified by the keyword OPT, enables the user to delete all versions at a node.

Format: The parameter value is the alphabetic code A, enclosed in parentheses.

Default: If the option parameter is omitted, the Operating System deletes the version specified by the VERS parameter or by its default, the flagged version.

7-107 PARAMETER INTERACTIONS

The version parameter overrides the option parameter. Option A is not valid if a version number is specified in the version parameter. The specified version will be deleted.

7-108 EXAMPLES

The following examples illustrate three uses of the DELV statement.

```
/ DELV FILEONE/FILETWO/FILETHRE, VERS=3
```

The node, identified by the pathname FILEONE/FILETWO/FILETHRE, contains the version to be deleted - absolute version number 3.

```
/ DELV Q/R/S, OPT=A
```

The node, identified by the pathname Q/R/S, contains the file versions to be deleted. All versions at that node will be deleted since the version parameter has been omitted and option A specified.

```
/ DELV Q/R/S
```

All defaults are operative. The flagged version cataloged at node Q/R/S will be deleted.

7-109 THE RPLV STATEMENT

The RePLace Version statement (RPLV) replaces a specified version of a file at an existing catalog node. This statement does not deal with the catalog system on a node level; therefore, catalog privacy must have been established previously when the node was constructed. To use this statement, the user must possess the own attribute at the named mode; otherwise, the Operating System aborts the operation. Also, if a version does not exist by the given number, the new version is added anyway. If a version does exist, it is deleted and the new file is cataloged.

Like the CAT and CATV statements, the RPLV statement also serves a specialized function: it enables the user to replace a version with a tape file directly without buffering the file to secondary disc. The parameters applicable to this feature are exclusive; they are not valid for other cataloging functions.

The general form of the RPLV statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	RPLV	b̸	pathname, [, ACNM = access name] [, VERS = version number] [, DTYP= $\left\{\begin{array}{l}\underline{\text{TAPE}}\\ \text{DSEC}\\ \text{HPT}\\ \text{PAD}\\ \text{OPID}\end{array}\right\} / \left\{\begin{array}{l}\underline{\text{NOCNTG}}\\ \text{CNTG}\end{array}\right\} /$ operator ID $/\left\{\begin{array}{l}\underline{\text{CEOF}}\\ \text{CALL}\end{array}\right\}$] [, RETP = retention] [, SRC= $\left\{\begin{array}{l}\underline{\text{DSEC}}\\ \text{TP}\\ \text{FT}\end{array}\right\}$]
					[, FORG= $\left\{\begin{array}{l}\underline{\text{PS}}\\ \text{DS}\end{array}\right\}$] [, LABL= [$\left\{\begin{array}{c}\underline{1}\\ \text{position}\end{array}\right\}$][/ $\left\{\begin{array}{l}\underline{\text{SL}}\\ \text{NL}\end{array}\right\}$]] [, EFID = efid1/efid2/. . . /efidn] [, DEN= $\left\{\begin{array}{l}\underline{1600}\\ 800\\ 712\\ 556\\ 356\\ 200\end{array}\right\}$] [, TRKS= $\left\{\begin{array}{c}\underline{9}\\ 7\end{array}\right\}$] [, RCTL= $\left\{\begin{array}{l}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{array}\right\}$] [, DEN= $\left\{\begin{array}{l}\text{BAND}\\ \text{SECT}\\ \text{WORD}\\ \text{BYTE}\end{array}\right\}$ = number]

The horizontal, broken line separating parameters in the operands field indicates special-function parameters. The parameters above the line are used for normal cataloging functions. Those below the line can be used only for special catalog processing.

The verb RPLV must appear in the operation field. The label field, if the statement is labeled, must be separated from the operation field with at least one blank. If the statement is not labeled, the identifying slash (/) must be separated from the operation field with at least one blank. The operation and operands fields must be separated with at least one blank.

The parameters in the operands field are positional and keyword parameters. The positional parameter must be coded as the first parameter and cannot be omitted. The keyword parameters are optional and can be coded in any convenient order.

7-110 NORMAL CATALOGING FUNCTIONS

Under normal circumstances, any file that is to be cataloged has been created on the secondary disc or is present there as a result of transfer actions for disc-buffering requirements of the Operating System. The RPLV statement specifies the cataloging function; the Operating System handles any required physical transfers to external devices.

In the normal case, only certain parameters need be considered. The following list itemizes the applicable parameters:

pathname	
ACNM	- access name
DTYP	- device type
RETP	- retention
SRC	- source
VERS	- version number

The remaining parameters apply only to a special function that bypasses disc-buffering requirements and should not be used.

7-111 PATHNAME

The node must be identified to the catalog system. The pathname parameter, a positional, mandatory parameter, specifies the name and location of the node in the catalog.

Format: The parameter value is one to ten edgenames concatenated by slashes (/). The first edgename may be a synonym defined in a PD statement. The pathname also specifies the location of the node in the catalog.

Default: None. Pathname must be specified.

7-112 ACCESS NAME (ACNM)

The access name parameter, identified by the keyword ACNM, identifies to the job the file that is to be cataloged as a version at the specified node.

Format: The parameter value is the name by which the file is accessible to the job. Access name is mandatory for all normal cataloging functions. The access name must be a valid ASC symbol.

Default: The only exception to the mandatory status of the ACNM parameter is the special cataloging function. When the source parameter is set to the value designating the source device as tape (SRC = TP or FT), ACNM becomes optional. In this case, it should be omitted. ACNM has no meaning if a tape file is being cataloged under the special feature. The pathname, which is mandatory under all circumstances, identifies the file to be cataloged to the catalog system.

7-113 VERSION (VERS)

The version parameter specifies which version of the file at the node defined by the pathname is to be replaced.

Format: The version parameter is identified by the keyword VERS. The parameter is a signed or unsigned constant, or a signed or unsigned Job Specification Language Variable. An unsigned parameter specifies a reference to the absolute version number of the file. A signed parameter specifies a reference to a version of the file relative to the flagged version. The absolute version number is determined modulo N, where N is the maximum number of versions allowed the file.

Limits: The absolute value of a signed integer, i, must be within the range: $1 \leq |i| \leq 63$.

The value of an unsigned integer, i, cannot exceed the maximum number of versions, N, minus one; i.e., the value must be within the range: $0 \leq i \leq N-1$.

The value of a signed Job Specification Language variable may be any value within a variable's range; i.e., $-2^{31} \leq v \leq 2^{31}-1$.

An unsigned Job Specification Language variable is a reference to an absolute version and its value has the same limits as an unsigned integer; viz.:

Default: If no version parameter is specified, the flagged version of the file is replaced.

7-114 DEVICE TYPE (DTYP)

The device type parameter is identified by the keyword DTYP and specifies if the file is to be allocated on the head-per-track disc, positioning-arm-disc, or the device specified in the Operating System default parameter table, or a specific channel or module specified by the third subparameter. It also indicates whether or not the file can be allocated without regard to physical contiguity on disc.

Format: The parameter value is a set of four subparameters separated by slashes. The first subparameter value pertains to file allocation and is one of the keywords TAPE, DSEC, HPT, PAD, or OPID. The meaning of these keywords is:

TAPE - allocate the file on a 9-track, 1/2-inch, 1600-bpi magnetic tape

DSEC - allocate the file on the device specified in the system default parameter file (default table)

HPT - allocate the file on the head-per-track disc

PAD - allocate the file on the positioning-arm-disc

OPID - allocate the file on the specific channel or module given by the third subparameter.

The second subparameter pertains to physical contiguity on disc of the file and its value is one of the keywords NOCNTG or CNTG where:

NOCNTG - file can be allocated without regard to contiguity. This is the default value.

CNTG - File must be allocated on physically contiguous disc; if the file is greater than or equal to a module; allocation on each module will be contiguous and the allocation will start at the beginning of each module.

The third subparameter is used in conjunction with the OPID keyword of the first subparameter and must be specified if OPID is used. The operator ID is a character string indicating a specific device related channel or module.

Examples:

"DM0" indicates H/T module 0, channel 0

"DCH0" indicates H/T channel 0

"PAD0" indicates PAD spindle 0, 980-0, and TCC-0

"PCH0" indicates 980-0, and TCC-0

The fourth subparameter pertains to the setting of the initial allocation of the cataloged copy of the file to the original files current end-of-file or current allocation. This subparameter is identified by one of the keywords CEOF or CALL where:

CEOF - means set the initial allocation of the cataloged copy of the file to the original file's current end-of-file.

CALL - means set the initial allocation of the cataloged copy of the file to the original file's current allocation

Default: The DTYP parameter is optional. When it is omitted, the Operating System assumes TAPE and NOCNTG, which indicates the file will be allocated on 1/2-inch, 9-track, 1600 bpi magnetic tape and the file may be allocated without regard to contiguity.

Restrictions: If the OPID keyword is used as the first subparameter the third subparameter, operator ID, must be specified. Otherwise, the third subparameter must not be specified. When the first subparameter is TAPE, the second, third, and fourth subparameters are not valid.

Examples:

```
DTYP=OPID/CNTG/DM0

DTYP=HPT

DTYP=/CNTG

DTYP=OPID//DCH0
```

In the first example the file will be contiguous on the head-per-track disc module 0. In the second example the file will not necessarily be contiguous on the head-per-track disc. In the third example the file will be contiguous and on the device specified in the system default parameter table. In the fourth example the file will not necessarily be contiguous on the head-per-track disc channel 0.

7-115 RETENTION PERIOD (RETP)

When a tape file is cataloged by the RPLV statement, the number of days the user wants the new file retained can be specified. The retention period parameter, identified by the keyword RETP, specifies the length of retention time.

Format: The parameter value is an unsigned constant.

The Operating System converts the specified number of days to an expiration date and records it in the Tape Volume Catalog for both standard labeled (SL) and unlabeled (NL) tape volumes. The Operating System also records the date on the standard labeled tapes, but not on unlabeled tapes.

Default: This parameter is optional. If omitted, the parameter is given a default value defined by the installation.

The retention period parameter is valid only for files that are output to tape volumes. DTYP=TAPE or in conjunction with SRC=TP or FT. If a file will remain on the secondary disc, RETP has no meaning. Therefore, if DTYP=DSEC, RETP should not be used.

7-116 SOURCE (SRC)

Accurate use of the RPLV statement hinges on one parameter, the source parameter (SRC). The source parameter specifies the source device on which the to-be-cataloged file resides.

Format: The parameter is identified by the keyword SRC. Its value is one of the keywords: DSEC, TP or FT. The keyword DSEC specifies that the file is located on the secondary disc. TP specifies a half-inch tape volume and FT specifies a one-inch tape volume.

Default: If the source parameter is not specified, the Operating System assumes the default of DSEC (file location is secondary disc). Normal catalog functions are used.

Besides specifying the source device of the file, SRC flags the cataloging function as either normal or special processing. SRC=DSEC indicates normal processing. SRC=TP or FT indicates the special feature; the file is cataloged, but is not buffered to disc. It remains on tape.

7-117 SPECIAL CATALOGING FUNCTION

The special cataloging function of the RPLV statement also replaces a version of the specified file. However, in this special case, the file version being cataloged must be a tape file that is never buffered to disc for the cataloging activities.

All of the parameters pertinent to normal version cataloging functions are required for this feature except ACNM and DTYP. In addition, the following parameters must be considered:

LABL	- label
FORG	- file organization
EFID	- external file identification
BAND	- disc reservation
SECT	
WORD	
BYTE	
DEN	- tape density
TRKS	- tracks
RCTL	- recording control

7-118 FILE ORGANIZATION (FORG)

The file organization parameter specifies the organization of the file.

Format: The parameter is identified by the keyword FORG. It consists of one subparameter selected from two keywords: PS and DS.

PS - the file is organized sequentially.

DS - the file is organized as direct secondary.

The default value is PS if the parameter is omitted.

Default: FORG is valid only when the source parameter (SRC) specifies that to-be-cataloged file resides on half-inch or one-inch tape volumes. If SRC is not specified, its default value is secondary disc (DSEC); if SRC=DSEC, the parameter specifies secondary disc. If SRC=DSEC or defaults to DSEC, the FORG parameter is invalid and will be ignored if coded.

7-119 LABEL (LABL)

The label parameter, identified by the keyword LABL, describes characteristics of the tape volume on which the file resides.

Format: The parameter is comprised of two subparameters: label type and file position. Both must be separated by a slash (/), if used.

The first subparameter (position) specifies the desired position of the file on the tape volume. The subparameter value is either 1 or another constant, where the range is $1 \leq n \leq 255$.

The second subparameter (label type) specifies the type of label on the tape volume. The value is either the keyword SL or NL. SL specifies that the tape volume has standard labels. NL specifies no labels.

If SL is specified explicitly or by default and an EFID is specified, the System requests that the correct tape volume be supplied. If a tape volume cannot be verified as standard labeled, the operator must cancel the request.

If NL is specified and the tape volume cannot be verified as non-labeled, the operator must cancel the request.

The position and label type subparameters are interactive. These interactions are interpreted as:

Label Type	Position	Interpretation
SL	n	n = file sequence number. The number of physical tape marks to be skipped during positioning is a function of the standard label format.
NL	n	n = file sequence number. The number of physical tape marks to be skipped during positioning is n-1.

The LABL parameter depends on the SRC and DTYP parameters. If SRC= DSEC or defaults to DSEC, the LABL parameter has no meaning. If DTYP=TAPE, the LABL parameter being specified will cause the Operating System to abort the process.

Default: The label parameter is optional and may be omitted. In addition, each subparameter is optional. If the entire parameter is omitted, the Operating System assumes the following values:

position 1	position to first file on volume
label type SL	standard labels

If one of the subparameters is omitted, the Operating System assumes the appropriate value from those defined above. If the first subparameter is omitted, the separating slash must be used in front of the last subparameter. If the second subparameter is omitted, the separating slash should be left off. For example;

position subparameter omitted	LABL = /NL
label type subparameter omitted	LABL = 2

7-120 EXTERNAL FILE IDENTIFICATION (EFID)

The keyword EFID identifies the external file identification parameter. It specifies the identification numbers of the volumes that contain the cataloged file. Identifiers of all tape volumes containing the file must be declared. The EFID parameters must be coded in the sequence that the tapes are to be read.

Format: The parameter value is a character string of one to six characters. If the string is enclosed in quotation marks, blanks in the string are retained; if the string is not so enclosed, any embedded blanks are eliminated.

Tape identification labels are assigned to volumes by the installation at the time they are submitted or generated. The user is informed of the external labels of his files at that time.

Default: If SRC=DSEC or is not specified, the EFID parameter is invalid and ignored if coded.

7-121 TAPE DENSITY (DEN)

The keyword DEN identifies the tape density parameter. DEN specifies the recording density in bits-per-inch-per-track for the tape volumes designated by the EFID parameters. All tape volumes specified must have the same density.

Format: The parameter value is 1600, 800, 712, 556, 356, or 200. The key 1600 specifies the tape density is 1600 bpi; 800 specifies 800 bpi; 712 specifies 712 bpi. If the source parameter (SRC) specifies that the source device is half-inch tape (SRC=TP), 1600, 800, 556 or 200 may be selected. If SRC specifies one-inch tape (SRC=FT), either 712 or 356 may be selected.

Default: The tape density parameter is optional. If omitted, the value is 1600 bpi for nine track tape and 800 bpi for seven track tape.

The DEN parameter is invalid if SRC=DSEC or defaults to DSEC.

7-122 TAPE RECORDING TRACKS (TRKS)

The tape recording track parameter, identified by the keyword TRKS, specifies if the given tape is a seven-track or a nine-track tape.

Format: The parameter value is one of the keynumbers 9 or 7 to designate either the nine-track or the seven-track tape respectively.

Default: The tape recording track parameter can be omitted. The default is to nine-track tape. If the tape recording track parameter specification or default is different than the actual tape configuration the System aborts the input/output operation.

7-123 RECORDING CONTROL (RCTL)

The recording control parameter allows for the specification of a parity, translation and data conversion mode. This parameter is applicable only if seven-track tape is being processed. It is identified by the keyword RCTL and the keyword values are:

O - odd parity without translation or conversion

E - even parity without translation or conversion

T - odd parity with translation but no conversion

C - odd parity with conversion but no translation

ET - even parity with translation but no conversion.

Default: If the RCTL parameter is omitted, and seven-track tape is designated, the default is ET or even parity with translation but no conversion.

Data Translation: Data translation refers to the ability to transform data recorded in BCD interchange to EBCDIC and vice versa. Since each BCD code is a six-bit value, upon input the six bits are translated into the corresponding eight-bit EBCDIC value. For output operations the eight-bit EBCDIC value is translated into its six-bit BCD representation.

Data Conversion: The data conversion option makes it possible to write eight-bit binary data on seven-track tape. For output operations, every three bytes of data are written as four tape characters. For input operations, every four tape characters are written as three bytes of data.

7-124 DISC RESERVATION (SECT, BAND, WORD OR BYTE)

The disc space parameter specifies the management of disc space for the named file throughout the job when it is assigned.

Format: The disc space parameter is identified by the keywords SECT or BAND. Sector reservations are in units of 64 each. If the requested number of sectors is not a multiple of 64, the Operating System rounds it to the next higher multiple of 64.

SECT specifies the parameter in sector units (64 words each). BAND specifies the parameter in band units (256 sectors each).

WORD specifies that the parameter value is expressed in word units given in thousands (WORD=15 is 15,000 words). When the value specified is not a multiple of 64 sectors it is rounded up to the next highest multiple.

BYTE specifies that the parameter value is expressed in byte units given in thousands (BYTE=12 is 12,000 bytes). When this value as specified is not a multiple of 64 sectors it is rounded up to the next highest multiple.

The parameter value is an unsigned constant less than 999999_{10} and may be from one to six digits.

Default: The parameter is optional. If omitted, an installation-defined value is used. The parameter is valid only for special catalog processing. It is meaningless if SRC=DSEC or SRC is not specified (SRC defaults to DSEC).

7-125 PARAMETER INTERACTIONS

Parameter specification depends on the source (SRC) parameter. Certain parameters apply to normal cataloging functions; others to the special function. The following description summarizes which parameters may be used for each case, normal and special processing, respectively.

7-126 SRC = DSEC

When the source device is secondary disc, the file to be cataloged has been created on or buffered to the secondary disc. In cataloging this file, the following parameters apply:

pathname	mandatory
access name (ACNM)	mandatory
version (VERS)	optional
output location (DTYP)	optional
file retention (RETP)	optional (valid only for tape files)

7-127 SRC = TP OR FT

When the source device is a half-inch or one-inch tape volume, the file to be cataloged exists but is not buffered to the secondary disc. In cataloging this file, the following parameters apply.

pathname		mandatory
version (VERS)		optional
file retention (RETP)		optional
label information (LABL)		optional
file organization (FORG)		optional
external file identifiers (EFID)		mandatory
disc reservation	BAND SECT WORD BYTE	optional
tape density (DEN)		optional (mandatory for SRC=FT)
tracks (TRKS)		optional (half-inch tape)
recording control (RCTL)		optional (seven-track tape)

7-128 EXAMPLES

The following examples illustrate the use of the RPLV statement for normal catalog functions.

1. File - All defaults Operative

```
/ JSL
  ʃ
/ RPLV A/B/C/D, ACNM=TFILE, DTYP=DSEC, SRC=DSEC,
  VERS=6
  ʃ
/ JSL
```

Given Values

A/B/C/D — pathname - names file being cataloged and defines location of node. This parameter is mandatory.

ACNM=TFILE — TFILE identifies file to the job. This parameter is mandatory.

DTYP=DSEC — File will remain on disc after being cataloged.

Given Values

SRC=DSEC — Specifies that the file to be cataloged is currently on disc. In this case, SRC could be omitted since its default value is DSEC.

VERS — Version six is replaced.

Invalid Parameters

RETP — Retention period is invalid for any file remaining on the secondary disc.

FORG
LABL
EFID
DEN
TRKS
RCTL
BAND
SECT
WORD
BYTE
} These parameters apply <u>only</u> when SRC=TP or FT.

2. File - All defaults Operative

```
/ JSL
   ∫
/ RPLV Q/R/S, ACNM=THIRTY
```

Given Values

Q/R/S — pathname - names file being cataloged and defines location of node. This parameter is mandatory.

ACNM=THIRTY — THIRTY identifies file to the job. This parameter is mandatory.

Default Values

The default values are the same as described in the first example of file cataloging. The values are summarized in the following list:

DTYP = TAPE

RETP = installation-defined value

SRC = DSEC

VERS = flagged version

Invalid Parameters

FORG
LABL
EFID
DEN
TRKS
RCTL
BAND
SECT
WORD
BYTE

} apply only when SRC=TP or FT.

The following examples illustrate the use of the RPLV statement for special catalog processing.

3. Nine-Track Half-Inch Tape Source

```
/ JSL
   ʃ
/ RPLV T/R, SRC=TP, FORG=DS, LABL=2/NL, EFID=2340
   ʃ
/ JSL
```

Given Values

T/R

pathname - names file being cataloged and defines location of node. This parameter is mandatory.

SRC=TP

The source device of the file to be cataloged is nine-track half-inch, magnetic tape volume. This parameter also signifies that special catalog processing is used.

FORG=DS

The file being cataloged has a direct secondary organization.

LABL=2/NL

When the cataloged file is subsequently input to a job, File Management uses this information to input the tape file. The parameter specifies that positioning is to the second file and that the tape volume is unlabeled. For all unlabeled tape volumes, a JSL FD statement must be used when the file is subsequently assigned to a job to specify the file's logical characteristics.

EFID=2340

The external volume identifier specifies that the file is contained on tape volume 2340. This parameter is mandatory when SRC=TP or FT.

Default Values

RETP
The System assumes an installation-defined value to be converted to an expiration date. The expiration date is written to the Tape Volume Catalog, but not to the specified unlabeled tapes.

DEN
The System assumes 1600 bpi as the density value of the tape volume containing the specified file.

TRKS
The Systems assumes nine-track jobs as default value.

RCTL
Not applicable for nine-track tapes.

VERS
The flagged version is replaced.

BAND
SECT
WORD
BYTE
The System assumes an installation-defined value for disc space allocation when the cataloged file is used.

Invalid Parameters

ACNM
DTYP

4. One-Inch Tape Source

```
/ JSL
   ∫
/ RPLV S/V, SRC=FT, DEN=712, EFID=10000, VERS=+1
   ∫
/ JSL
```

Given Values

S/V
The pathname S/V names the cataloged file and defines the location of the node. This parameter is mandatory.

SRC=FT
A file existing on one-inch magnetic tape must be specified as FT on the SRC parameter. DSEC designates the source device as secondary disc; TP designates half-inch tape.

VERS=+1
The flagged version plus one is replaced.

DEN=712
The density is specified as 712 bpi; this parameter is mandatory for one-inch tapes.

Given Values

EFID=10000	The external file identifier specifies the byte is contained on tape volume 10000. This parameter is mandatory for SRC=TP or FT.

Default Values

RETP BAND SECT WORD BYTE	installation defined value

Invalid Parameters

ACNM
DTYP
TRKS
RCTL

SECTION VIII

SPECIAL FUNCTIONS

8-1 GENERAL

There are a variety of JSL statements that extend the capabilities of the JSL to perform special functions. A group of these JSL statements deal with file manipulation. The statements are, in the order discussed, the REL, RENAME, MFR and MFRE, and the CNT and CNTE statements. The REL statement deactivates a file's linkage with the job and releases the file's allocated disc space returning it to the job's allocation. The RENAME statement enables a user to specify new names under which the files are identified to the job. The MFR and MFRE statements provide multi-file requests to enable a file or files to be output to a group of magnetic tape volumes. The CNT and CNTE statements enable a group of files to be input and processed as a single file.

In addition, the JSL provides the capability to define and enable a job within another, and defer its execution for a later time without resubmitting it. Four statements implement this function: JOBX, EOJX, CJSL and PJSL. The JOBX and EOJX statements delimit the beginning and end of an embedded job. The CJSL statement checks the syntax of a job without executing the job. The PJSL statement initiates execution of a deferred job.

JSL also provides, through the SETUP statement, the capability of setting up information for the Operating System to use in transfer and scheduling of secondary storage files and video tape drives.

8-2 THE REL STATEMENT

The RELease statement severs a job's access to a file by the specified access name and specifies that the disc space for the named file or files is to be released and returned to the allocation for the current job block. REL physically releases only non-resident, non-cataloged files or non-resident, cataloged files if there is only one current user.

The general form of the REL statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	REL	b̸	access name [,...[,access name]]

The verb REL must appear in the operation field. The identifier slash and the label symbol, if the statement is labeled, must be separated from the verb with at least one blank. The operation and operands fields must be separated with at least one blank. If the statement is not labeled, only one blank need be used between the slash and the verb.

8-3 ACCESS NAME

The statement contains one positional, mandatory parameter. Additional parameters are optional. If specified, parameters are separated by commas. The parameter value is a valid ASC symbol which specifies the access name of the file whose disc space is being released. The number of access names that can be specified is limited only by practicality.

Access names which REL specifies must have been defined in the job before the REL statement is encountered or processed. Any access name not defined previously will be ignored. Because the specified files are released and delinked from the job when the REL statement is encountered, all named files should be ready for release at that time.

8-4 THE RENAME STATEMENT

The RENAME statement enables the user to specify a new name for a file. The new name replaces the existing name for a file. The file is no longer accessible under the former name. The RENAME statement is particularly useful in letting the user specify an access name for standard-access-named files accessed by executing CP steps. The standard-access-named files can then be accessed by other steps in the job.

The position of the RENAME statement in the job input stream determines when the file name is changed and the old access name delinked from the job. The old access name can be used to reference another file as long as the reference is made after the RENAME process is completed. The access name being replaced must exist prior to incidence of the RENAME specifying it. The new access name should not be previously defined. Only one access name can be replaced with a new name for each RENAME statement.

The format of the RENAME statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	RENAME	b̸	access name 1, access name 2

The verb RENAME must appear in the operation field. The slash identifier, or the label symbol if the statement is labeled, must be separated from the operation field by at least one blank. The operation and operand fields also must be separated by at least one blank. The operand field contains two positional, mandatory parameters separated by a comma.

8-5 ACCESS NAME 1

The first parameter is the access name by which the file is currently identified to the job. Referred to as old access name, it must be an existing access name. The old access name is the name being replaced. It is delinked from the job.

Access name 1 must be a valid ASC symbol. It is mandatory and cannot be omitted. No default is allowed.

8-6 ACCESS NAME 2

The second parameter is the new access name by which the file will subsequently be identified to the job. Referred to as the new access name, it should not be an existing name prior to its specification. The new name links the file to the job. The file is no longer accessible under the old access name. The only exception to prior existence of an access name is the name assigned to a default input file. If the new name is the default name and it exists, the existing file is released, then the rename is performed.

Access name 2 must be a valid ASC symbol. It is mandatory and cannot be omitted. No default is allowed.

8-7 MULTI-FILE REQUESTS

Multi-file requests enable users to input files from or output files to a group of magnetic tape volumes. This technique is applicable only for files input from or output to tape.

The Multi-File Request (MFR) statement is the start delimiter for the files requested. It defines the start of included files. The Multi-File Request End (MFRE) statement is the end delimiter. It defines the end of included files.

The MFR/MFRE pair may enclose two or more JSL File Management statements that define the files to input from or output to tape. All enclosed statements must specify the same function; i.e., the statements must be either all ASG, FIT, FOT or catalog statements. The allowable JSL statements are ASG, FIT, FOT, CAT, CATV and RPLV. The process outputs the files to tape volumes in the order the statements appear within the MFR/MFRE statement pair.

8-8 THE MFR STATEMENT

The general form of the MFR statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	ƀ	MFR	ƀ	id [,EFID=tape id 1[/tape id2][/...; [, tape idn]]] $[,\text{LABL}=[\begin{Bmatrix}\underline{1}\\ \text{position}\end{Bmatrix}]\ [/\begin{Bmatrix}\underline{\text{SL}}\\ \text{NL}\end{Bmatrix}];$ $[/\begin{Bmatrix}\underline{\text{NEW}}\\ \text{ADD}\\ \text{APND}\end{Bmatrix}]]]$ $[,\text{DEN}=\begin{Bmatrix}1600\\ 800\\ 556\\ 200\end{Bmatrix}]$ $[,\text{TRKS}=\begin{Bmatrix}\underline{9}\\ 7\end{Bmatrix}]\ [,\text{RCTL}=\begin{Bmatrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{Bmatrix}]$ [,SVC=scratch volume count] [,RETP=retention] [, ~~LOC=location id~~] [,DEST=destination code]

The verb MFR must appear in the operation field. The slash identifier, or the label symbol if the statement is labeled, must be separated from the operation field by at least one blank. The operation and operands field also must be separated by at least one blank.

MFR contains both positional and keyword parameters. The positional parameter is mandatory and must be coded as the first parameter in the operands field. The keyword parameters are optional and may be coded in any convenient order.

8-9 MULTI-FILE IDENTIFICATION

The ID parameter is a user-specified identity for the files included in the multi-file request. The Operating System lists the labels of tape volumes containing output files on the job activity log and identifies them with the code.

Format. The parameter is positional and mandatory. The identifier must be a valid ASC symbol.

Default. None. The parameter must be specified.

8-10 EXTERNAL FILE IDENTIFICATION (EFID)

The keyword EFID identifies the external file identification parameter which specifies the identification numbers of the volumes that contain or will contain the requested files.

Format. The parameter value is a character string of one to six characters. If the string is enclosed in quotation marks, blanks in the string are retained; if the string is not so enclosed, any embedded blanks are eliminated.

External labels are affixed to the tape volumes to provide visual identification of the volume and its contents. Normal tape volume control requires two types of external tape labels. One is a permanent label that identifies the volume; the other is a temporary label that identifies the contents.

The permanent label is affixed when the tape is first received and should contain the sequential volume serial number assigned to the tape. The volume serial numbers are used to identify tape volume unique numbers and to file the tapes in the tape rack.

The contents label is used to identify the current contents of a particular volume. This temporary label is applied when data is written on the volume and contains identifying information to ensure that the contents of the volume can be easily distinguishable. The information entered in the label is usually furnished partly by the programmer and partly by the operator.

Default. The EFID parameter is optional. If omitted, the Operating System writes the specified file to a scratch volume. After the files have been successfully written, the Operating System informs the user of the external volume identifier(s). If the series contains catalog processing statements, scratch volumes must be used.

8-11 LABEL (LABL)

The label parameter, identified by the keyword LABL, describes characteristics of the tape volume that contain or will contain the requested files.

Format. The parameter is comprised of three subparameters: label type, file position, and new file placement. All three subparameters are positional, but not mandatory. All three must be separated by slashes (/), if used.

The first subparameter (position) specifies the desired position of the file on the tape volume that contains or is to contain it. The subparameter value is either 1 or another constant, where the range is $1 \leq n \leq 255$.

The second subparameter (label type) specifies the type of label on the tape volume. The value is either the keyword SL or NL. SL specifies that the tape volume has standard labels. NL specifies no labels.

The third subparameter (new file placement) applies only to output and specifies whether the first file enclosed in an MFR/MFRE pair is placed on a new tape, added to an existing tape or appended to an existing file on tape. The remainder of the enclosed files are output serially. The placement subparameter value is one of the keywords NEW, ADD or APND. NEW specifies that a scratch tape will receive the files. Volume identifiers are returned to the user after a successful transfer to the scratch tape. ADD specifies that the files are added to an existing tape as an additional file. APND specifies that the file is written on an existing tape file as an appendage to the existing file. The Operating System appends the file by erasing the current file mark terminating the existing file.

Appending files with inconsistent formats is not prohibited. However, the user must exert caution with the APND option. A mismatch of density, block size or record format causes the Operating System to abort the transfer.

If SL is specified explicitly or by default and an EFID is specified, the System requests that the correct tape volume be supplied. If a tape volume cannot be verified as standard labeled, the operator must cancel the request.

If NL is specified and the tape cannot be verified as non-labeled, the operator must cancel the request.

The position and label type subparameters are interactive. These interactions are interpreted as:

Label Type	Position	Interpretation
SL	n	n = file sequence number. The number of physical tape marks to be skipped

Label Type	Position	Interpretation
		during positioning is a function of the standard label format.
NL	n	n = file sequence number. The number of physical tape marks to be skipped during positioning is n-1.

Default. The label parameter is optional and may be omitted. In addition, each subparameter is optional. If the entire parameter is omitted, the Operating System assumes the following values.

position 1	position to first file on volume
label type SL	standard labels
placement NEW	system scratch volume

If any of the subparameters are omitted, the Operating System assumes the appropriate value from those defined above. If the first and/or second subparameters are omitted, the separating slashes must be used in front of the last subparameter. If the second and/or third subparameters are omitted, the separating slashes must be used in front of the last subparameter. If the second and/or third subparameters are omitted, the separating slash should be left off. For example:

position subparameter omitted	LABL = /NL/ADD
label type subparameter omitted	LABL = 2//APND
placement subparameter omitted or	LABL = 2/NL

two subparameters omitted	LABL = 2
	LABL = //ADD
	LABL = /NL

8-12 TAPE DENSITY (DEN)

The keyword DEN identifies the tape density parameter. DEN specifies the recording density in bits-per-inch-per-track (bpi) for the tape volumes designated by the EFID parameter or the scratch volumes indicated or defaulted by the LABL or SVC parameters. All tape volumes specified must have the same density.

Format: The parameter value is either 1600, 800, 556, or 200. The keynumber 1600 specifies the tape density 1600 bpi, the keynumber 800 specifies the tape density 800 bpi, etc.

Default: The tape density parameter is optional. If omitted, the value is 1600 bpi for nine-track tape and 800 bpi for seven-track tape.

8-13 TAPE RECORDING TRACKS (TRKS)

The tape recording track parameter, identified by the keyword TRKS, specifies whether or not the given tape is a seven-track or a nine-track tape.

Format: The parameter value is one of the keynumbers 9 or 7 to designate either the nine-track or the seven-track tape respectively.

Default: The tape recording track parameter can be omitted. The default is to nine-track tape. If the tape recording track parameter specification or default is different than the actual tape configuration the System aborts the input/output operation.

8-14 RECORDING CONTROL (RCTL)

The recording control parameter allows for the specification of a parity, translation and data conversion mode. This parameter is applicable only if seven-track tape is being processed. It is identified by the keyword RCTL and the keyword values are:

O	- odd parity without translation or conversion
E	- even parity without translation or conversion
T	- odd parity with translation but no conversion
C	- odd parity with conversion but no translation
ET	- even parity with translation but no conversion.

Default: If the RCTL parameter is omitted, and seven-track tape is specified, the default is ET or even parity with translation but no conversion.

Data Translation: Data translation refers to the ability to transform data recorded in BCD interchange to EBCDIC and vice versa. Since each BCD code is a six-bit value, upon input the six bits are translated into the corresponding eight-bit EBCDIC value. For output operations the eight-bit EBCDIC value is translated into its six-bit BCD representation.

Data Conversion: The data conversion option makes it possible to write eight-bit binary data on seven-track tape. For output operations, every three bytes of data are written as four tape characters. For input operations, every four tape characters are written as three bytes of data.

8-15 SCRATCH VOLUME COUNT (SVC)

The scratch volume count parameter, identified by SVC, specifies a maximum number of system scratch tape volumes that will contain the requested files.

Format. The parameter value is an unsigned constant of one to six digits.

System scratch volumes are provided by the system. The volume identifiers have been assigned by the system and registered in the Tape Volume Catalog as scratch volumes. Scratch volumes may have either standard labels or no labels.

Default. The SVC parameter is optional. If omitted, the Operating System assumes twelve scratch volumes are available to contain the files output by the multi-file request.

8-16 FILE RETENTION PERIOD (RETP)

The retention period parameter, identified by the keyword RETP, specifies the number of days the file will be retained.

Format. The parameter value is an unsigned constant.

The Operating System converts the specified retention period to an expiration date which it records in the Tape Volume Catalog for both standard labeled (SL) and unlabeled (NL) tape volumes. The converted expiration date is not recorded on the tape volume for unlabeled (NL) volumes.

Default. The RETP parameter is optional. If omitted, the default value is an installation-defined value. If the series contains catalog processing statements, this parameter is ignored and the RETP coded on the specific statement is used.

8-17 LOCATION (LOC)

The location parameter specifies the site at which the tape files are output.

Format. The keyword LOC identifies the parameter. Its value is a string of one to eight alphanumeric characters. The central site assigns valid identifiers to various terminal installations.

Default. The LOC parameter is optional. If omitted, the default value is the identifier for the central site. The tapes are output at the central site.

8-18 DESTINATION (DEST)

The destination parameter, identified by the keyword DEST, specifies a code that determines where the tapes are routed after output.

Format. The parameter value is a string of two alphabetic characters defined by the central site. The code is transmitted to the external tape level.

Default. The DEST parameter is optional. If omitted, the default value is that defined for the central site. This parameter is ignored by catalog processing statements within the series.

8-19 PARAMETER INTERACTIONS

The EFID, LABL, DEN and SVC parameters may be interactive, depending on the type of tape volumes desired. The parameters may be used in various combinations to achieve a specific request for output.

- Specifying Private Volumes Only. To output files on private volumes only, the user should specify three parameters.

  LABL = $\left\{\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right\} / \left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\} / \left\{\begin{matrix}\text{ADD}\\ \text{APND}\end{matrix}\right\}$ NEW is not allowed.

  EFID must be specified

  DEN is optional.

  The SVC parameter cannot be used.

- Specifying Scratch Volumes Only. In specifying system scratch volumes, the user may specify the following combination.

  LABL = $\left\{\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right\} / \left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\}$ / NEW

  DEN is optional.

  SVC is optional. It may be used if a maximum number of scratch volumes is required.

  The EFID parameter cannot be used.

In the above case, all parameters may be omitted. The default values provide the proper conditions.

- Specifying Both Scratch and Private Volumes. In specifying both system scratch and private volumes, all three parameters should be used.

  LABL = $\left\{\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right\} / \left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\} / \left\{\begin{matrix}\text{ADD}\\ \text{APND}\end{matrix}\right\}$ NEW is not permitted; it specifies scratch volumes only

  EFID must be specified.

  SVC is optional. If a maximum number of scratch volumes is desired, the parameter should be used.

  DEN is optional.

  TRKS is optional.

  RCTL is optional.

8-20 STATEMENT INTERACTIONS

The MFR statement interacts with the ASG, FIT, FOT, CAT, CATV, and RPLV statements. The files to be input from or output to multiple tape volumes are defined and transferred through these statements. The MFR statement parameters that may be used depend on which statements are enclosed. The positional identification parameter is valid with either type of statements. The optional keyword parameters are valid only for FIT and FOT statements with one exception: the SVC parameter is valid for the FIT, FOT and the catalog statements. The values for the MFR keyword parameters (whether explicitly specified or defaulted) override the specification of corresponding parameters on the enclosed FIT and FOT statements with one exception: the file position subparameter of the LABL parameter when specified on an FIT statement will override the sequence set up by the file position subparameter specified on the MFR statement and subsequent FIT statements will be processed according to the file position designated if the LABL parameter is omitted on these statements. If keyword parameters are not specified the conditions established by MFR default values override the parameters on the FIT and FOT statements.

Table 8-1 summarizes MFR statements applicable for the FIT, FOT, and the catalog statements.

8-21 MFR EXAMPLES

The following example shows the FIT statements embedded within an MFR and an MFRE statement and the use of LABL parameter and how it changes.

```
/ MFR   FILEX,EFID=12,LABL=2

/ FIT   F1

/ FIT   F2

/ FIT   F3,LABL=5

/ FIT   F4

/ MFRE
```

Files F1, F2, F3, and F4 are input from private volume 12. Files F1 and F2 are the second and third files respectively on volume 12. Files F3, F4, and F5 are the fifth, sixth, and seventh files from the same volume. Note how the LABL subparameter on the FIT statement overrides the MFR sequence set up. Volume 12 is a standard-labeled, nine-track, 1600-bpi tape volume input from the central site and returned at the central site after input for an installation defined period of time with these values being determined by default values on the MFR statement.

Table 8-1. MFR Parameters for Enclosed JSL Statements

PARAMETER	ENCODED FIT or FOT	ENCLOSED CAT, CATV and/or RPLV
id	Required.	Required.
EFID=tape id1]/tape id2...[/...]]	MFR specified value or default value overrides FIT or FOT parameter.	Not Applicable.
LABL= $\left\{\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right\}$ / $\left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\}$ / $\begin{matrix}\underline{\text{NEW}}\\ \text{ADD}\\ \text{APND}\end{matrix}$	MFR specified value overrides the FOT parameter. The position subparameter of the FIT parameter overrides the MFR value at the point in sequence where specified.	Not Applicable.
DEN= $\left\{\begin{matrix}\underline{1600}\\ 800\end{matrix}\right\}$	MFR specified value or default value overrides FOT parameter.	Not Applicable.
SVC=scratch volume Count	MFR specified value or default overrides FIT parameter.	May be used. MFR value overrides system defaults for CAT, CATV, RPLV since this parameter is not available on them.
RETP=retention	MFR specified value or default value overrides FOT parameter.	Not Applicable.
LOC=location id	MFR specified value or default value overrides FIT or FOT parameter.	Not Applicable.
DEST=destination code	MFR specified value or default value overrides FOT parameter.	Not Applicable.
TRKS=recording tracks	MFR specified value or default value overrides FIT or FOT parameter.	Not Applicable.
RCTL=recoding control	MFR specified value or default value overrides FIT or FOT parameter.	Not Applicable.

8-22 THE MFRE STATEMENT

The general form of the MFRE statement is:

		OPERATION		OPERANDS
/	b̸	MFRE		

MFRE has no parameters. The verb MFRE must appear in the operation field. MFRE cannot be labeled. The identifier slash must be separated from the operations field by at least one blank.

MFRE must always be used with the MFR statement. It always appears at the end of the JSL statements being enclosed.

8-23 EXAMPLES

The following examples illustrate the MFR/MFRE pair enclosing FOT and catalog statements.

Enclosed Catalog Statements

```
/ MFR MULTICAT, SVC=4

/ CAT A/B/C, ACNM=ANAME, DTYP=TAPE

/ CATV A/B/C, ACNM=BNAME, DTYPE=TAPE

/ MFRE
```

Files ANAME and BNAME are output to a system scratch volume tape. The volume is a standard labeled, 1600 bpi tape. If additional volumes are required to contain both files, the system supplies a maximum of four scratch volumes. The file characteristics are obtained from the disc file File Information Blocks. ANAME is cataloged by the CAT statement; BNAME by the CATV statement. The files are output at the central site.

Enclosed FOT Statements

```
/ MFR MULTIOUT, EFID=10/12, LABL=1/NL/ADD, RETP=14

/ FOT ANAME, EFID=60, LABL=2/NL/ADD, RETP=7

/ FOT BNAME, EFID=10, LABL=1/SL/ADD, RETP=7

/ FOT CNAME, EFID=30, LABL=2/NL/ADD, RETP=10

/ MFRE
```

Files ANAME, BNAME and CNAME are output to private volumes, 10 and 12. The EFID parameter specified on the MFR statement overrides EFID specification on the FOT statements. If additional volumes are required to contain the files, the system provides scratch volumes (up to default maximum of 12) to contain any remaining parts. The files are positioned according to the LABL specifications on the MFR statement: position is 1, label type is NL and the placement is ADD.

These values override those specified on the FOT statements. Similarly, the files will be retained for fourteen days according to the MFR statement's RETP=14. This value likewise overrides specified FOT values.

The MFR statement default values apply to the files being output. These values are:

DEN = 1600 bpi tape volumes

SVC = 12, maximum number of scratch tapes

LOC = central site

DEST = central site

TRKS = nine-track tape

Enclosed FOT Statements with Defaults Operative on Both

```
/ MFR MULTISCR
/ FOT ANAME
/ FOT BNAME
/ FOT CNAME
/ MFRE
```

Files ANAME, BNAME and CNAME are output to a system scratch volume, according to the default values for the MFR statement. If additional volumes are required to contain the files, the Operating System provides scratch volumes up to the default maximum of twelve. The other default values for the MFR statement are:

LABL parameter

position = 1

label type = SL

placement = NEW

DEN = 1600 bpi

TRKS = nine track tape

RETP = installation-defined value

LOC = central site

DEST = central site

The values for the enclosed FOT statements, whether explicitly stated or defaulted, are overridden by the values for the MFR statement, either explicitly stated or defaulted.

8-24 CONCATENATION STATEMENTS (CNT and CNTE)

Concatenation allows two or more input files to be retrieved automatically by the Operating System and processed as a single file. This technique is applicable only for input files. Sequential files can be concatenated; files organized in other ways cannot.

A File Descriptor Block (FDB) is associated with each concatenated file. The access name specified by the CNT/CNTE pair links all of the associated FDBs for a set of concatenated files.

The CNT/CNTE pair may enclose two or more JSL File Management statements that define the files to be input. The ASG, FIT and FD statements are the only JSL statements that can be used in file concatenation. The process concatenates the files in the order in which the files are specified within the CNT/CNTE pair.

The CNT statement specifies the access name under which the set of concatenated files can be referenced. The transfer of non-cataloged sequential tape files is determined by the File Management statements. The access name appearing on the CNT statement can appear as an access only on the FOSYS statement for output.

The general form of the CNT statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	CNT	ƀ	access name

CNT has only one parameter - the mandatory, positional parameter specifying access name. The access name parameter identifies the set of concatenated files to the job. The access name is a valid ASC symbol.

The general form of the CNTE statement is:

		OPERATION		OPERANDS
/	ƀ	CNTE		None

CNTE has no parameters. The verb CNTE must appear in the operation field. CNTE is the ending delimiter statement for a concatenated input file request. It defines the end of the included ASG, FIT or FD statements defining the files concatenated under the CNT statement's access name.

CNTE can only be used with the CNT statement. It always appears at the end of the enclosed JSL statements.

8-25 STATEMENT INTERACTIONS

The CNT statement interacts with the FD, ASG, and FIT statements. The files concatenated by the CNT statement are defined and transferred through these statements.

The CNT statement is implicity interactive with the CNTE statement. CNT is the beginning delimiter for a concatenated input file request. CNTE is the ending delimiter for the concatenated input file request.

The CNT statement must always be used to define a concatenated set of files. It is always used with the CNTE statement.

8-26 EXAMPLE

The rules for using the FD, ASG and FIT statements within the CNT/CNTE pair are based on the same conditions for using these statements for non-concatenated files. The FD statement must never end the sequence of statements enclosed in a CNT/CNTE pair. It must always precede an associated ASG or FIT statement. The following example shows how to concatenate files.

1. Non-cataloged, Unlabeled, Tape Files. The sequence of JSL statements for specifying unlabeled tape files within the CNT/CNTE pair is essentially the same except that an FD statement explicitly specifying logical characteristics <u>must</u> precede each FT statement.

```
/ CNT TFILE
/ FD AFILE, FORG=PS, LREC=80, RCFM=FBS, BKSZ=2000
/ FIT AFILE,EFID=1240, LABL=/NL
/ FD BFILE, FORG=PS, LREC=80, RCFM=FBS, BKSZ=2000
/ FIT BFILE,EFID=1490, LABL=2/NL
/ CNTE
```

8-27 DEFERRED JOB PROCESSING

It is possible to create a JSL file, check its syntax without executing it, and catalog it or output it to an external device for future use. The deferred job specification can be retrieved later and inserted in the job input stream for execution. The four statements that implement this function are:

JOBX	- Deferred JOB statement
EOJX	- Deferred EOJ statement
CJSL	- Syntax-check deferred JSL file
PJSL	- Execute deferred JSL file

8-28 THE JOBX STATEMENT

The JOBX statement delimits the beginning of a source job specification contained in a file embedded in another source job specification. Its syntax is identical to that of the JOB statement with the exception that the verb is "JOBX" instead of "JOB".

The format of the JOBX statement is:

		OPERATION		OPERANDS
/	b̸	JOBX	b̸	jobname, acctnum, user code [, CAT=category] [, LOC=location identification] [, OPT=option]

The JOBX statement's operand field is identical to that of the JOB statement. JOBX statements are coded exactly like JOB statements; they perform similar functions for the jobs they introduce. For a discussion of parameter coding for the JOB statement and an explanation of its use internal to the job it defines, refer to section II.

8-29 THE EOJX STATEMENT

The EOJX statement delimits the end of a job specification file input embedded within the input stream of another job specification. Its syntax is identical to that of the EOJ statement with the exception that its verb is "EOJX," instead of "EOJ". EOJX is used as EOJ with respect to the job statement which it delimits; see section II for a discussion of such use of the EOJ statement.

The format of the EOJX statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	ƀ	EOJX	ƀ	none

8-30 EXAMPLES. A deferred job specification is input through the job stream. The following example illustrates the process:

```
/ JOB PORTER, 12345, USERNAME
   ʃ
JSL statement sequence
   ʃ
/ START ACNM=DEFJOB
/ JOBX PORTEE1, 12345, USERNAME
   ʃ
JSL statements for one deferred job
   ʃ
/ START ACNM=DEFJOB1
/ JOBX PROTEE2, 12345, USERNAME
   ʃ
JSL statements for another deferred job
   ʃ
/ EOJX
/ STOP
   ʃ
/ EOJX
/ STOP
JSL statement sequence
   ʃ
/ EOJ
```

When the System processes job PORTER, it treats the statements encountered embedded within the START/STOP pair as any other input file entered through the job stream. It removes the records between START and STOP statements from the job stream and stores them in a file labeled with the access name "DEFJOB." In addition, it recognizes the JOBX and EOJX statements as deferred job delimiters and replaces their verbs with "JOB" and "EOJ" respectively, as it reads those records into DEFJOB. Notice that multiple deferred job specifications can be entered in the same file, processed in the same file, and stored in the same file.

JOB and EOJ statements cannot be used as delimiters for prospective deferred jobs. Whenever a JOB statement is encountered in the job input stream without an EOJ statement immediately before it, the System assumes that the prior job has ended and that the prior job's EOJ statement was omitted. Accordingly, the System supplies an EOJ and begins a new job file with the current JOB statement. Likewise, whenever an EOJ statement is found in the job stream, the System terminates the current job specification and prepares to receive the next one. Hence, if JOB and EOJ statements were used to delimit a deferred job specification, each one, as it was encountered, would cause a premature termination of its carrier job.

Since the deferred job specification file is removed from the job stream on input and put in a file, its file can be referenced from anywhere in its parent job specification. A CJSL statement can be included to syntax-check the deferred job; the parent job can also order concurrent execution of the deferred job with a PJSL statement. However, all statements that refer to a deferred job specification do so by file access name. The START statement that introduces a deferred job specification must assign it a file access name; otherwise, the file exists but cannot be used, because it cannot be found.

To keep a job specification for future use, its carrier or parent job must preserve it in some way. It can be cataloged or output to an external device. Cataloged job specification files are executed with PJSL statements. Depending on the nature of the external medium on which it is stored, a JSL file output to an external device can be executed by reading it directly into the job input stream, or by reading it in as a file under a current job specification, which, in turn, calls for execution of the deferred job with a PJSL statement.

8-31 THE CJSL STATEMENT

The CJSL statement checks the syntax of a job defined in an external or cataloged file without executing the job. The statement is written:

	LABEL		OPERATION		OPERANDS
/	[symbol]	ƀ	CJSL	ƀ	access name

It begins with a slash, followed by an optional statement label, a blank, the verb CJSL, another blank, and the file access name of its operand field.

8-32 ACCESS NAME

The file access name is the local name for the job specification source file to be syntax-checked. The access name must be defined in the current job; the file it references must contain one or more complete job specifications. The access name is mandatory; it cannot be omitted.

8-33 EXAMPLES

The sample job specification PARENT below reads in deferred job specification SON, catalogs it, then uses CJSL to syntax-check SON as it resides in its cataloged file.

```
/ JOB PARENT, 12345, USERNAME
/ CAT JOBEDGE, ACNM=INFILE
/ CJSL INFILE
/ START ACNM=INFILE
/ JOBX SON, 12345, USERNAME
   ∫
Deferred job specification statements
   ∫
/ EOJX
/ STOP
/ EOJ
```

The System extracts INFILE from the job stream when it is first encountered, just like any other file input through the job stream. INFILE, therefore, exists as a separate file at the time the JSL code is executed; it can be referenced from any where in that code.

8-34 THE PJSL STATEMENT

The PJSL statement orders processing to begin on the deferred job specification(s) residing in the file referenced by its access name parameter. The PJSL statement appears as:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	PJSL	b̸	access name

PJSL is written with a slash, an optional label, the PJSL verb flanked by blanks, and an access name parameter.

8-35 ACCESS NAME

The access name is the local name for the source job specification file to be executed. The access name must be a valid ASC symbol defined within the current job.

8-36 EXAMPLE

The sample job specification below continues the example begun in the prior sub-section. It demonstrates the use of the PJSL statement. EXJOB accesses a cataloged file containing a deferred job specification and executes that specification.

```
/ JOB EXJOB, 12345, USERNAME

/ ASG JOBFILE, JOBEDGE

/ PJSL JOBFILE

/ EOJ
```

The PJSL statement takes the job residing in the file referenced by its access name parameter and inserts it in the system's job input queue. It does not insert the deferred job specification's statements in the current job as, for instance, a sequence of statements from a macro definition would be inserted. Instead, the current job specification continues execution after its PJSL statement is processed. The job file accessed by PJSL is added onto the job input stream; it may or may not begin execution concurrently with its parent job, depending on the availability of system resources.

8-37 THE SETUP STATEMENT

The SETUP statement instructs the Operating System to schedule a secondary storage video tape drive. It can also specify serial numbers of registered video tape reels to be used, and the number of scratch reels that can be used. The SETUP statement is used to setup, through its parameters, information necessary to achieve the transfer and scheduling of secondary storage files. The SETUP statement appears as:

	LABEL		OPERATION		OPERANDS
/	symbol	b̸	SETUP	b̸	setup id [, CHAN=operation id] [, EFID=tape id1][/tape id2]/...; [/tape idn]]]] [, SVC=$\frac{1}{\text{number}}$] [, PROTECT=$\frac{\text{YES}}{\text{NO}}$]

SETUP is written with a slash, an optional label, the SETUP verb embedded in blanks and the parameters in the operands field. The parameters in the operands field are both positional and keyword parameters and the positional parameter must be coded first with the keyword parameters, that are optional, appearing in any convenient order.

8-38 SETUP ID

The setup id parameter associates a logical name with the storage channel that be used for secondary storage transfers.

Format: This parameter is positional and mandatory. The identifier must be a valid ASC symbol.

Default: None. This parameter must be specified.

8-39 CHANNEL SPECIFICATION

The channel specification parameter, identified by the keyword CHAN, specifies which channel is to be used with the secondary storage transfer.

Format: The parameter is optional and can be coded in any order with the other optional parameters. The identifier must be a valid ASC facility "OPID" associated with a secondary storage channel.

Example:

CHAN=PCHO specifies channel zero.

Default: When CHAN is not specified the Operating System selects a PAD channel for the user.

8-40 EXTERNAL FILE IDENTIFICATION (EFID)

The external file identification parameter specifies the identification numbers of the video tape volumes that will be used in the secondary storage transfers. The order of specification is not required, but it is suggested that the user order his volumes sequentially for the most efficient tape usage. If the first operation to be performed is a read from the tape, at least one EFID must be specified.

Format: The keyword EFID designates the external file identification parameter. The tape identifiers must be separated by slashes (/). Each tape identifier is a string of from one to six characters. If the identifier is enclosed in quotation marks, blanks in the string are retained; if not, embedded blanks are removed.

Default: If this parameter is omitted, only scratch volumes will be associated with this storage channel.

8-41 SCRATCH VOLUME COUNT (SVC)

The scratch volume count parameter specifies a maximum number of system scratch tape volumes that are to contain files or members.

Format: This parameter is identified by the keyword SVC and is an unsigned constant value of from one to six digits.

System scratch volumes are provided by the System. The volume identifiers have been assigned by the System and registered in the Tape Volume Catalog as scratch volumes.

The parameter specifies the maximum number of scratch volumes required. It can be used with the EFID parameter if both scratch volumes and user-specified volumes will be used to contain the files.

Default: If the scratch volume count parameter is omitted, the default value of one scratch volume is assumed.

8-42 WRITE PROTECTION (PROTECT)

The PROTECT parameter specifies whether or not write protection is to be invoked for the VSNs specified.

Format: This parameter is identified by the keyword PROTECT and has the value of one of the keywords: YES or NO. YES instructs the System to protect the contents of the tapes whose EFIDs are listed in the EFID parameter. The System will abort any attempt by the user to modify the contents of these tapes. Scratch tapes will not be protected. NO means that protection will not be enforced for tapes scheduled by that SETUP statement.

Default: The parameter may be omitted. The default value is YES.

8-43 THE BATINT STATEMENT

The BATINT statement transforms a batch job into an interactive job, performs any preprocessing IJSL and puts the job in a suspended state by the interactive session start time. The batch job can be submitted at either a remote terminal or the central facility, but the terminal at which the interactive session is to take place must be specified. A presession start time must also be given to ensure that the job completes all processing and is in the suspended state by the session start time. There is currently a 20-minute minimum lapse required between the presession start time and the session start time, however, the interactive may be started at any time after the job enters the suspended state. In particular, this means that the session can be started before the targeted session start time if the presession processing for the job takes less than 20 minutes. It must be remembered that the BATINT statement is the last verb processed in the batch mode and therefore should be the last statement before the EOJ control card. The BATINT statement appears as:

	LABEL		OPERATION		OPERANDS
/	symbol	b̸	BATINT	b̸	PST=day:hour:min, SST=day:hour:min, SDU=min, TMID=RTn

BATINT is written with a slash, an optional label, the BATINT verb flanked by blanks, and the appropriate parameters in the operands positions. The parameters are all keyword parameters and must be supplied, however they may be given in any order.

8-44 PRESESSION START TIME (PST)

The presession start time parameter (PST) specifies the time by which the job must have started into active processing. It must be later than the time that the batch job is read in and earlier than the session start time by a system defined amount of time (currently 20 minutes).

Format: The keyword is PST and is set equal to the time given in days, hours, and minutes separated by colons. The day is the Julian day.

Default: None, the parameter must be supplied.

8-45 SESSION START TIME (SST)

The session start time parameter (SST) specifies the time by which the interactive session is to start. It must be greater than the presession start time by a system defined amount of time, (currently 20 minutes).

Format: The keyword is SST and is set equal to the time given in days, hours, and minutes separated by colons. The day is the Julian day

Default: None, the parameter must be supplied.

8-46 SESSION DURATION (SDU)

The session duration parameter (SDU) represents the number of minutes that the session is to last. It is currently not used but still must be given for the proper IJSL frame to be built.

Format: The keyword is SDU and is set equal to the number of minutes the session is to last. It is restricted to a positive number between 1 and 999999.

Default: None, the parameter must be given.

8-47 TERMINAL IDENTIFIER (TMID)

The terminal identifier parameter (TMID) is the id of the terminal at which the interactive session is to take place. It must be a terminal operating under TOS.

Format: The keyword is TMID and is set equal to an 8-character facility id of a remote terminal.

Default: None, the parameter must be given.

8-48 PARAMETER INTERACTIONS

The SST parameter must exceed the PST parameter by a system default (currently 20 minutes), and must exceed the time the job is read in by a system default value (also 20 minutes). The presession start time must be greater than the time at which the job enters the Operating System.

8-49 STATEMENT INTERACTIONS

The BATINT statement should only occur in batch jobs. Its processing causes an end-of-file marker to be placed behind the BATINT verb in the IJSL file to keep the batch job from going to termination. This causes all IJSL frames after the BATINT verb to be lost. Therefore, BATINT should be the last statement before an EOJ control card.

8-50 USER INTERFACE

Processing of the BATINT statement has several possible interfaces with the user. The normal situation is for the message PREPROCESSING COMPLETE (job - jobname) to appear at the terminal requested by the BATINT statement when all preprocessing for the batch job is completed and the job has been turned into an interactive job in the suspended state ready to be signed on as an interactive session.

Should a conflict arise with the BATINT parameters, one of the following messages will appear in the job activity file which appears at the batch job site.

BATINT: SS less than PST

BATINT: SST - PST less than default

BATINT: PST less than present time

BATINT: invalid term id or request.

In such a case, no message appears at the remote terminal requested on the BATINT statement to indicate that the preprocessing is complete.

8-51 OPERATOR INTERFACE

The processing of the BATINT verb provides for a message to the ASC operator to inform him that the presession start time for the job has past and the job has still not entered the active state. He can then take installation defined steps to ensure that the job goes active as soon as possible. The message has the form:

BATINT pretime, JID=Job, JN = Jobname

SECTION IX

THE MACRO LANGUAGE

9-1 INTRODUCTION

The JSL macro language provides the capability to reduce the amount of JSL coding in large jobs containing repeated sequences of intrinsic JSL statements. The macro language enables the user to specify a macroname (verb) and associate it with a sequence of JSL statements so that subsequent references to that name result in the sequence of JSL statements being used. This process comprises three steps: macro definition, call and expansion.

A macro may generate a sequence of intrinsic JSL statements containing exactly the same parameters each time it is called; or, each macro call may specify various parameters to be placed in the expanded code. A macro may generate the same sequence of JSL each time it is called, or it may be directed to produce optional segments of its code in response to successive calls of its name.

9-2 TERMINOLOGY

The following definitions summarize special terms used in describing the macro language.

Macro definition	-	MACRO and MEND (macro end) statement pair, BUILD, GEN, Model statements PRESET and INC, and all intrinsic JSL statements between the pair.
Macro body	-	The JSL statements in a macro between the MACRO and the MEND statements.
Macro call	-	Use of a macro name as the verb in a JSL statement.
Macro expansion	-	Sequence of JSL statements generated by the macro processor from a macro call.
Macro model statement	-	Statement in a macro body that may be expanded into an intrinsic JSL statement in a macro expansion.

Formal parameter - Parameter in a macro statement that is replaced by an actual value whenever the parameter is referenced during macro expansion. The parameter may be written as a qualified, positional, tagged or status parameter.

Actual parameter - Parameter in a macro definition statement that is unaltered in macro expansion.

Qualified parameter - A tagged or positional parameter followed by a colon and a modifier that specifies what the parameter can represent. The modifiers are V for binary value of the parameter, S for character string, A for attribute, and L for length of character string.

Intrinsic JSL - All existing JSL, including predefined macros, but not the macro language.

Macro directives - The subset of JSL that is valid for the macro language only.

Macro-local label - A label in parentheses. The label is valid for that macro only.

Positional parameter - Formal parameters defined on a MACRO statement. These parameters are positional relative to the set of positional parameters specified in the macro. The parameter is a character string preceded by an ampersand.

Tagged parameter - A formal parameter identified by a keyword followed by an equals sign. The keyword is preceded by an ampersand. The value of the parameter appears to the right of the equals sign. The value can be a character string or a quoted character string. Character strings may be any combination of alphabetic or numeric characters.

Concatenated parameter - Combinations of formal parameters, or of actual and formal parameters, that are replaced by a single character string in the corresponding statements of the expanded macro.

Status parameter - Formal parameter that is a digit or a character preceded by an exclamation point. The parameter specifies a special condition within the macro during expansion. These parameters may have predefined meanings or may be user-defined.

9-3 MACRO LANGUAGE STATEMENTS

The macro facility consists of three areas: macro file definition and manipulation, macro definition, and special functions. In support of these functions, the macro language supplies three kinds of statements:

1. Macro File Definition and Manipulation Statements

   MACBLD
   MACASG

2. Macro Definition Statements

   - Macro Delimiting Statements

     MACRO
     MEND

   - Macro Generation Statements

     BUILD
     GEN
     Model statement

   - Macro Sequence Control Statements

     JUMP

   - Parameter Modification Statements

     PRESET
     INC

3. Special Function Statements

   VERBSYN
   DMACRO

9-4 MACRO PROCESSING

When a macro definition is encountered in the processing of a job specification, it is entered in a Macro Definition Table. The macro definition is valid for subsequent macro calls within the job specification until another macro definition of the same name supersedes it, or until a deactivate macro (DMACRO) statement explicitly deactivates it. The DMACRO statement reactivates the most recent definition before the current one; i.e., same-named macro definitions constitute a stack.

Macro definitions may be entered into the job specification by specifying a macro file via the MACASG statement or by physically placing them in the source job file or by both. Macro definitions in a macro file specified on a MACASG statement are entered into the job's Macro Definition Table when the MACASG statement appears in the job input stream. Macro definitions that occur in the source job file are entered into the Macro Definition Table in the order in which they are encountered.

A macro call statement in the job specification calls the Macro Processor to search the job's Macro Definition Table for a definition having the name called.

If there are several macro definitions with the same name the one which was most recently entered into the stack is the one chosen for expansion.

The Macro Processor expands all macro model statements in the body of a macro definition into intrinsic JSL statements according to the Macro Generation and Control statements, if any, that occur in the macro body. The statements generated for the macro expansion contain all the actual parameters specified by their respective macro model statement in the macro body. All formal parameters of the respective macro model statements are replaced (in the expanded macro statements) by actual values specified in formal parameters in the macro call statement or, in default, by actual values specified in formal parameters in the beginning MACRO statement or PRESET statement.

Upon completion, the macro expansion occupies the relative position that was occupied by the macro call which initiated the expansion.

A macro name must be defined or assigned within a job specification before the first macro call of that name. Macro calls are permitted within macro definitions when the expansion satisfies this condition.

Every formal parameter processed during macro expansion must be supplied an actual value, either by parameters in the initiating macro call statement or by parameters in the MACRO, PRESET, or INC statements; otherwise, the formal parameter is undefined, and an error condition results.

9-5 MACRO PARAMETERS

In the macro language, there are two categories of parameters: formal and actual. Formal parameters are replaced by actual values when the parameters are referenced during macro expansion. Actual parameters are actual values; they are unaltered in macro expansion.

9-6 FORMAL PARAMETERS

A formal parameter may appear in a macro model statement as a single parameter, in a list of parameters or may be concatenated with other parameters. Each formal parameter must be assigned an actual value at expansion time. The value is assigned in a macro call that initiates expansion of an existing macro, in a MACRO statement that begins the definition of a macro, in a PRESET statement specified within the body of a macro, or in an INC statement within the body of a macro. Formal parameters may be positional, tagged, qualified or status parameters. The positional and tagged parameters are written as &<character string>. Positional parameters are defined as such in the MACRO statement. Positional and tagged parameters are considered to be composed of four components: the character string (or actual value) associated with the parameter; the length of the character string; the binary value associated with the parameter; and the attributes of the parameter. Each component of a parameter can be accessed by using a qualified parameter reference. The qualified parameter is written as:

$$\&\ \langle\text{character string}\rangle \left\{\begin{matrix} S \\ V \\ L \\ A \end{matrix}\right\}$$

where: S is the character string component of the parameter

V is the binary value component of the parameter

L is the length (number of characters) of the character string component

A is the attribute component of the parameter.

Notice that an unqualified reference to a parameter yields the "S" component. References to "A", "V", and "L" yield integers. If the "S" component is constant, V is the binary value of the integer; otherwise, the "V" component is zero unless modified by an INC statement. The values associated with the attribute (A) component is a bit vector. The bits are defined as:

Bit	Meaning if set
31	No S component associated with parameter
30	Defined as a positional parameter
29	Defined as a local label
28	Value for parameter appears on the macro call line
27	Parameter has a value specified in the macro definition by a MACRO or PRESET

Bit	Meaning if set
23	S component of parameter was enclosed in quotes (the string does not contain them now)
22	S component begins with a digit (i. e., is an integer constant)

The status parameter is written as:

$$!\left\{\begin{matrix}\text{character}\\ \text{digit}\end{matrix}\right\}$$

Status parameters are defined as:

Status Variable	Type	Meaning
!A	Read Only Integer	1 = default data file follows macro call line in input. 0 = JSL card follows macro call line
!B	Read Only Integer	Nesting level
!C	Read Write Integer	Next BUILD position
!D	Read Only Integer	Start position of last BUILD statement (i. e., previous !C)
!E	Read Only Integer	Number of positional parameters on macro call line
!F	Read Only Integer	Number of parameters on macro call line
!G	Read Only Integer	Number of statements generated by this macro
!H	Read Only Character String	Macro name/synonym
!0-!9	Read Write Integer	User defined

9-7 ACTUAL PARAMETERS

An actual parameter specifies a symbol, constant, or character string that is to remain unaltered in the macro expansion of the statement in which it appears.

An actual parameter may appear in a macro model statement alone, in a list, or in a concatenated parameter. It must be all or part of a basic parameter valid for the JSL statement to be generated at macro expansion.

An actual parameter terminates at the first comma encountered that is not embedded in quotation marks. An actual parameter is no more than a single operand field; it may be less than a single operand field. To avoid confusion with other sorts of parameters, an actual parameter may not contain an ampersand (&) or an at sign (@) unless they are enclosed in quotation marks.

9-8 MACRO FILE DEFINITION AND MANIPULATION

There are two files of importance to the macro user: the system macro file and the user macro file. The system macro file contains system-defined macros that are automatically made available to a job for its duration. No assignment process is required. This file is disc resident. The second file of interest to the user contains macros defined by the user, placed in the file by him, and subsequently made available to the user's job through an assignment process specified by him. This file must be a cataloged, disc-resident file. Both types of files are made available at translation time. Any manipulation and file definition is done at translation time, not when the job is in the active state.

The macro statements that define and manipulate the macro file specifically apply to the user macros since the system macro file is automatically linked to the job. The user has the capability to define a macro file via the MACBLD statement and manipulate macro files through the MACBLD and MACASG statements.

9-9 THE MACBLD STATEMENT

The MACBLD statement builds or modifies a macro file. All macros defined between the MACBLD statement and either the next MACBLD statement or the EOJ statement are written to the macro file defined by the specified access name. If the macro file is to be retained for future use, it must be cataloged through JSL statements. MACBLD-created files are opened for allocation when the MACBLD scope is terminated.

The general form of the MACBLD statement is:

	LABEL	OPERATION	OPERANDS
/	b̸	MACBLD	[access name ,BASE = access name]

The MACBLD statement contains both positional and keyword parameters. The positional parameter is mandatory and must appear as the first parameter. The keyword parameter is optional. The MACBLD statement may not be labeled since it has no execution time processing.

9-10 ACCESS NAME

The macro file being constructed must be named to be accessible to the job. JSL statements referencing the file within the job use the specified access name to access the file. The access name is mandatory and must appear first in the operands field.

Format: The access name must be a valid ASC symbol. For the job in which it is referenced, the access name originates with the MACBLD statement specifying it.

Default: None. Access name must be specified.

9-11 BASE PARAMETER

The BASE parameter enables the user to append newly defined macros to an existing macro file. The existing macro must have been linked to the job previously through a MACASG statement. This parameter provides the update of any existing macro file or the creation of a new file. Macro definitions appearing within the scope of the MACBLD are appended to the BASE file.

Format: The parameter is identified by the tag, BASE. The parameter value is an access name that must be a valid ASC symbol. The access name originates with a previously specified MACASG statement.

Specification: If names of macro definitions are repeated in both files, the system looks in both files and uses the last encountered macro definition. The other macro copies are purged from the macro directory and are no longer accessible by that name.

Default: If the BASE is omitted, a new macro file as specified by the positional parameter is created. No update is performed.

9-12 PARAMETER INTERACTION

The positional access name may be specified and the BASE parameter omitted. If so, the macro definitions following the MACBLD and up to the next MACBLD or the end of the job are placed in a newly constructed file identified by that name. Once the MACBLD has been issued, the file may be referenced by subsequent JSL statements by the access name within that job.

When the BASE parameter is used, it specifies an existing file to which the macro definitions within the scope of MACBLD are to be appended.

9-13 STATEMENT INTERACTIONS

The MACBLD statement interacts with the MACASG and catalog management statements. The conditions of this interaction follow:

- If the BASE parameter is specified on MACBLD, a MACASG or a MACBLD statement specifying the same access name must appear in the job input stream prior to the occurrence of the MACBLD statement referencing the file.
- If the BASE parameter is omitted on MACBLD, a MACASG statement is not necessary.
- The issuance of a MACBLD statement does not make the specified file a cataloged file. A JSL CAT, CATV or RPLV statement, specifying the same access name, must be used to catalog the file so that it is not lost when the job ends. The CAT, CATV or RPLV statement may be specified anywhere in the job input stream because the MACBLD statement, processed at translation time, establishes the macro file prior to execution of the JSL statements.

9-13A MACBLD EXAMPLES

```
/ƀMACBLDƀMACROS
```

All macros defined after the MACBLD statement until the next MACBLD or EOJ will be built into a macro file with an access name of macros.

```
/ƀMACBLDƀMACROS, BASE=MACFILE
```

All macros defined following the MACBLD statement until the next MACBLD or EOJ statement will be appended to the existing macro file called MACFILE.

9-14 THE MACASG STATEMENT

The MACASG statement assigns a macro file for use by the JSL translator. The statement makes the cataloged, disc-resident macro file available to the job in which it occurs. Like the MACBLD statement, the MACASG statement is processed at translation time, not at execution time. Thus, the MACASG statement must appear in the job input stream before any macro statement references a macro is the file stipulated by MACASG. Files specified by MACASG are opened immediately after being assigned. All macro files are closed, but not released, at termination of translation. They may be re-opened during subsequent translator activations if subsequently referenced. In adddition, files assigned by MACASG are available in a shared mode only.

The general form of the MACASG statement is:

	LABEL	OPERATION		OPERANDS
/	b̸	MACASG	b̸	access name, pathname [, VERS = version]

The MACASG statement contains both positional and keyword parameters. The positional parameters are mandatory and must appear as the first and second parameters, in the order shown. The keyword parameter is optional. The MACASG statement may not be labeled.

9-15 ACCESS NAME

The macro file being assigned must be named to be accessible to the job. JSL statements referencing the file within the job use the specified access name to access the file. Access name is mandatory and must appear first in the operands field.

Format: The access name must be a valid ASC symbol. For the job in which it is referenced, the access name originates with the MACASG statement.

Default: None. Access name must be specified.

9-16 PATHNAME

The pathname parameter identifies the file being accessed to the catalog system. The pathname is a positional mandatory parameter that must appear as the second parameter. The pathname must exist in the catalog prior to execution of the job containing the MACASG statement.

Format: Pathname is one to ten edgenames concatenated by slashes. The first edgename cannot be a synonym. Each edgename must be a valid ASC symbol. The pathname originates as the result of a previous cataloging activity performed before the job containing the MACASG statement is executed.

Default: None. Pathname must be specified.

9-17 VERSION NUMBER

The version number parameter, VERS, specifies which version of the file identified by the pathname is to be assigned to the job. The version parameter is optional; it may be omitted.

Format: The version parameter is identified by the tag: VERS. The parameter value is a signed or unsigned constant. An unsigned constant ranging from 0 to N-1 (N=number of versions possible) specifies an absolute version number. A signed constant ranging from -63 to +63 specifies a version relative to the flagged version.

Default: If VERS is omitted, the System assigns the flagged version to the job.

9-17A MACASG EXAMPLES

```
/bMACASGbFILE1,A/B/C,VERS=2
```

The macro file at pathname A/B/C and version 2 will be assigned to the user job, therefore, making all macros defined on that file available to the user with access name FILE1.

9-18 MACRO DEFINITION

The macro language provides four types of statements for defining a macro: macro delimiting statements, macro generation statements, macro sequence control statement, and parameter modification statements. This set of macro statements enable the user to define macros and control the environment in which the macro must exist.

9-19 MACRO DELIMITING STATEMENTS

The MACRO and MEND statements are the delimiting statements that specify the boundaries of a MACRO. A macro definition must begin with a MACRO statement and must end with a MEND statement. Macro definitions can use any JSL statements in the macro body to govern processing when the macro is called and expanded. These JSL statements are used within the MACRO/MEND pair.

9-20 MACRO GENERATION STATEMENTS

The BUILD, GEN and model statements comprise the subset of macro generation statements. The BUILD statement constructs a buffer in card image of parameter values to be input to the translator. The GEN statement clears the BUILD-generated buffer by outputting the contents to the translator. Model statements are statements in a macro body that may expand into JSL statements in a macro expansion.

9-21 MACRO SEQUENCE CONTROL STATEMENT

The JUMP statement is the macro sequence control statement. JUMP enables the user to control the sequence of statement execution in a macro through either a conditional or unconditional transfer during a macro expansion.

9-22 PARAMETER MODIFICATION STATEMENTS

There are two parameter modification statements: PRESET and INC. The PRESET statement establishes default values for positional and/or tagged formal parameters appearing in the macro being defined. The INC statement increments the specified formal parameter by one.

9-23 THE MACRO STATEMENT

The MACRO statement starts a macro definition, names the macro, defines positional parameters and default values for tagged parameters omitted on the macro call line. If a macro name has been defined previously, the current definition takes precedence at expansion time.

The general form of the MACRO statement is:

	LABEL		OPERATION		OPERANDS
/	macro name	ƀ	MACRO	ƀ	[{& positional parameter & tagged parameter=default value}] , [{& positional parameter & tagged parameter = default value}]

The MACRO statement contains a macro name and positional and/or tagged parameters.

9-24 MACRO NAME

The macro name is mandatory and must appear in the label field immediately following the slash (with no intervening blanks permitted). The macro name must be a valid ASC symbol. Any macro name associated with a valid macro definition is a valid verb for a macro call statement.

9-25 PARAMETERS

The positional and/or tagged parameters provided for the MACRO statement specify default values for any formal parameters in the macro body and not specified on a macro call statement or PRESET statements. When the macro is expanded, the values specified by these parameters appear in the expanded code in the place where the formal parameter occurs in the macro definition. These parameters are overridden by those specified on macro call statements or PRESET statements, if specified in both places.

Format: Each positional or tagged parameter must be preceded by an ampersand (&). Tagged parameters are followed by an equal sign and a value. The value may be either a character string or a quoted character string.

Specification: Positional parameters are positional relative to their appearance in the MACRO statement. The order of the positional parameters on the MACRO statement determine where the string values are placed when the macro is called and expanded. Because these parameters are relative, there is no restriction upon placing other parameters on the MACRO statement between them.

Anytime a statement in a macro is expanded, values for its formal parameters must be supplied by expansion time through the MACRO statement, PRESET statements, or the call to the macro containing the statement.

Default: All parameters on the MACRO statement are optional and may be omitted. If omitted, the parameters specified by PRESET statements establish default values for the macro definition. If PRESET statements are not used, the parameters specified on the call statement to the macro define the values for the formal parameters occurring in the macro body between the MACRO/MEND pair.

9-26 MACRO EXAMPLE

```
/COMPILEbMACRObb&ACNM, &MEM, &CIFER=*, &NPIPE=4, ;
&MACHINE=4
```

This is the beginning of a macro called COMPILE. &ACNM and &MEM are positional parameters and &CIFER, &NPIPE, and &MACHINE are keyword parameters. &CIFER will be initialized to *, &NPIPE to 4 and &MACHINE to 4.

9-27 THE MEND STATEMENT

The MEND statement delimits the end of the macro definition initiated by the MACRO statement. The MEND statement can be used only with the MACRO statement.

The general form of the MEND statement is:

	LABEL		OPERATION		OPERANDS
/	(symbol)	b̸	MEND	b̸	[macro name]

9-28 LABEL

The MEND statement may be labeled with a macro-local label. This capability enables the user to transfer expansion to the MEND statement when the macro is called and expanded. The macro-local label is valid only for the macro in which it appears.

Format: The macro-local label must be a valid ASC symbol enclosed in parentheses. No other type of label is permitted.

Specification: The macro-local label may be the same as the label specified by a call to the macro; the two labels are exclusive. The parentheses encasing the label identify it as local.

9-29 MACRO NAME

The macro name is an optional parameter that is treated as a comment concerning the macro defined by the MACRO/MEND pair. The macro name must be a valid ASC symbol. If omitted, no comment regarding the macro being defined is made.

9-30 MEND EXAMPLE

```
/(EXIT)b̸MENDbXYZ
```

This macro statement terminates the definition for the macro XYZ.

```
/b̸MEND
```

This macro statement terminates the definition of the current macro.

9-31 THE BUILD STATEMENT

The BUILD statement constructs a buffer in card image of parameter values to be input to the translator. BUILD is a macro definition statement that generates all or part of the body of the macro. Multiple BUILD statements without an intervening GEN or model statement construct a macro model statement.

The general form of the BUILD statement is:

	LABEL		OPERATION		OPERANDS
/	[(label)]	b̸	BUILD	b	[[position] <substitution string>] [, [position] <substitution string>]

The BUILD statement contains optional positional parameters. In addition, the BUILD statement may be labeled with a macro-local label. Other labels are prohibited. The verb BUILD must appear in the operations field.

9-32 POSITION PARAMETER

The position parameter specifies the column number at which the parameters defined as the substitution string are placed in the buffer.

Format: The position parameter may be either a constant or a formal parameter that is an integer. The position parameter takes the form

$$\begin{Bmatrix}\text{constant}\\ \text{formal parameter}\end{Bmatrix}$$

Default: The position parameter is optional. It may be omitted. If omitted, any specified substitution string begins at the next available position in the buffer.

9-33 SUBSTITUTION STRING

The substitution string parameter is a list of parameters to be placed in the buffer.

Format: The substitution string parameter is a string of parameters that may be a combination of an actual parameter, formal parameter and/or concatenation of formal and actual parameters. The substitution string parameter takes the form

$$[\text{character string}]\ [\text{formal parameter}]\begin{bmatrix}\text{formal parameter}\\ [@]\ \text{character string}\end{bmatrix}\ \begin{bmatrix}\text{substitution}\\ \text{string}\end{bmatrix}$$

Default: The substitution string parameter may be omitted. If so, the BUILD statement clears the buffer and resets the buffer position to column one.

9-34 BUILD EXAMPLES

```
/b̸BUILDb̸</b̸b̸b̸b̸b̸b̸b̸b̸b̸XQTb̸b̸b̸b̸SYS. LMOD, CPOPT=(A, Z)>
/(A)b̸BUILDb̸1</>, 11<XQT>, 18<SYS. LMOD, CPOPT=(A, Z)>
```

Either of the two above BUILD statements will construct the JSL statement:

```
column 1     11     18
       /     XQT    SYS. LMOD, CPOPT = (A, Z)
```

9-35 THE GEN STATEMENT

The GEN statement clears the BUILD-generated buffer by outputting the contents to the translator. A GEN statement placed at the end of a series of BUILD statements terminates the card image buffer constructed by the BUILD statements. The buffer is cleared and the buffer position reset to column one.

The general form of the GEN statement is:

	LABEL		OPERATION		OPERANDS
/	[(label)]	b	GEN	b̸	[[position]<substitution string>] [, [position]<substitution string>]

The GEN statement contains optional position parameters. In addition, the GEN statement may be labeled with a macro-local label; other labels are prohibited. The verb GEN must appear in the operations field.

9-36 POSITION PARAMETER

The position parameter specifies the point at which the parameters defined as the substitution string are placed in the buffer.

Format: The position parameter may be either a constant or a formal parameter that is an integer. The position parameter takes the form

$$\left\{\begin{matrix}\text{constant}\\ \text{formal parameter}\end{matrix}\right\}$$

Default: The position parameter is optional. It may be omitted. If omitted, any specified substitution parameters begin at the next available free position in the buffer.

9-37 SUBSTITUTION STRING

The substitution string parameter is a list of parameters to be placed in the buffer.

Format: The substitution string parameter is a string of parameters that may be a combination of an actual parameter, formal parameter and/or concatenation of formal and actual parameters. The substitution string parameter takes the form

$$[\text{character string}]\left[\text{formal parameter}\left[\begin{matrix}\text{formal parameter}\\ \text{[@] character string}\end{matrix}\right]\right]\left[\begin{matrix}\text{substitution}\\ \text{string}\end{matrix}\right]$$

Default: The substitution string parameter may be omitted. If so, the GEN statement outputs any previous information in the buffer.

9-38 GEN EXAMPLES

```
/ƀGEN
```

This statement clears the BUILD-generated buffer by outputting any previous information in the buffer and resetting the buffer position to column one.

```
/ƀGENƀ<)>
```

This statement places a close parenthesis in the next available free position in the BUILD-generated buffer, outputs the information, then resets the buffer position to column one.

9-39 MACRO MODEL STATEMENTS

A macro model statement is any statement in a macro body that may be expanded into a Job Specification Language statement in a macro expansion. The format of a macro model statement is that of an intrinsic JSL statement, with macro parameters substituted for the JSL statements' label and parameters. Macro model statements containing intrinsic JSL verbs generate single intrinsic JSL statements at macro expansion. Macro model statements which are themselves macro call statements generate multiple intrinsic JSL statements at macro expansion.

The general form of a macro model statement is:

	LABEL		OPERATION		OPERANDS
/	label (label)	ƀ	verb	ƀ	substitution string

The label of a macro model statement can be an actual label, a formal label, or a macro-local label; that is, it can be respectively a valid ASC symbol, a formal parameter or substitution string, or a valid ASC symbol enclosed in parentheses. A label parameter that is a valid ASC symbol specifies an actual label generated on the corresponding JSL statement of the macro expansion, or, if the macro model statement is a macro call statement, the label of the first statement in the resultant macro expansion. A label parameter that is a formal parameter or substitution string specifies a formal label replaced, on the corresponding JSL statement of the macro expansion, by an actual value assigned to the formal parameter. A label parameter that is a valid ASC symbol enclosed in parentheses specifies a macro-local label that is not to be generated in the macro expansion and is used only for sequence control purposes. Such labels are used only in conjunction with the JUMP statement. Notice that a macro-local label may be used in a statement modeled on intrinsic JSL statement which itself does not allow a label. Macro model statements do not require labels; if no label is specified, none is carried forward to the expanded code.

9-40 VERB

The verb of a macro model statement is any expression other than a macro definition statement. A macro model statement must have a verb. It may be a formal parameter or substitution string. If the model statement's verb is a macro name, the definition of the called macro must be entered in the job specification prior to the expansion of the model statement.

9-41 SUBSTITUTION STRING

The substitution string is a list of parameters to be placed in the expanded macro.

Format: The substitution string parameter is a string of parameters that may be a combination of an actual parameter, formal parameter and/or concatenation of formal and actual parameters. The parameter takes the form

$$[\text{character string}]\ \left[\text{formal parameter} \begin{bmatrix}\text{formal parameter}\\ [@]\ \text{character string}\end{bmatrix}\right] \begin{bmatrix}\text{substitution}\\ \text{string}\end{bmatrix}$$

If a model statement is to be expanded into a single intrinsic JSL statement, the actual, formal, and concatenated parameters in its operand field must be arranged in such a way as to construct properly the operand field of the generated statement. The macro parameters used, when expanded, must result in an operand field of valid format for the intrinsic JSL statement being built. If the intrinsic JSL statement has mandatory parameters, they must be provided; if it has positional parameters, they must appear in the correct order.

Default: None. Parameters must be specified.

9-42 MODEL EXAMPLES

```
/b̸XQTb̸&ACNM, ADDMEM=&ADDMEM, CPTIME=&CPTIME

/b̸ASGb̸&ACNM, &PATH@/M$XX, USE=SHR
```

The above statements are the same as in intrinsic translator with possible substitution.

9-43 THE JUMP STATEMENT

The JUMP statement enables the user to control the sequence of statements in a macro through either a conditional or an unconditional transfer of the generation sequence during a macro expansion.

The general form of the JUMP statement is:

	LABEL		OPERATION		OPERANDS
/	[(label)]	b̸	JUMP	b̸	(label) [, boolean expression]

The JUMP statement contains both mandatory and optional positional parameters. In addition, the JUMP statement may be labeled with a macro-local label; other labels are prohibited. The verb JUMP must appear in the operations field.

9-44 LABEL

The label parameter in the operands field specifies the label of a statement to which the generation sequence may pass.

Format: The label must be a macro-local label and must be enclosed in parentheses. The label is mandatory and must appear as the first parameter. The label must be a valid ASC symbol. It must be the macro-local label of a statement in the macro body or the macro-local label of the MEND statement.

Default: None. A macro-local label must be specified.

9-45 BOOLEAN EXPRESSION

The boolean expression is the condition parameter of the JUMP statement. It specifies the condition under which sequence control is transferred to the label specified by the label parameter. If the boolean expression is TRUE, the jump is taken; if the boolean expression is FALSE, execution proceeds to the next sequential statement.

Format: The boolean expression takes the form:

$$\left\{\begin{array}{l}\text{formal parameter}\\ \text{actual parameter}\end{array}\right\} \left\{\begin{array}{l}.\text{EQ.}\\ .\text{NE.}\\ .\text{GT.}\\ .\text{LT.}\\ .\text{GE.}\\ .\text{LE.}\end{array}\right\} \left\{\begin{array}{l}\text{formal parameter}\\ \text{actual parameter}\end{array}\right\} \left[,\left\{\begin{array}{l}\underline{\text{TRUE}}\\ \text{FALSE}\end{array}\right\}\right]$$

The meaning of the relational operators (EQ, NE, GT, LT, GE, and LE) for integers is the usual arithmetic relation.

However, for character strings and the attribute (A) component of a formal parameter, slightly different definitions apply:

Character String Comparison

Operation	Meaning
.EQ.	Two character strings have the same number of characters and identical characters.
.NE.	The character strings do not satisfy the .EQ. requirements.
.GT.	The left operand's character string has more characters than the right operand's and the leftmost substring of the left operand is identical to the right operand's character string.
.LT.	The left operand's character string has fewer characters than the right operand's and the leftmost substring of the right operand is identical to the left operand's character string.
.GE.	The character strings satisfy either the .EQ. or the .GT. requirements.
.LE.	The character strings satisfy either the .EQ. or the .LT. requirements.

Attribute (A) - Set Comparison

The two operands are considered to be algebraic sets, the members of which are the bits in the respective words. The relationships are thus in terms of set comparisons.

Operation	Meaning
.EQ.	The sets are identical
.NE.	The sets do not satisfy the .EQ. requirement.
.GT.	The left set contains the right set but .EQ. does not hold.
.LT.	The left set is contained in the right set but .EQ. does not hold.
.GE.	The sets satisfy either the .EQ. or the .GT. requirements.
.LE.	The sets satisfy either the .EQ. or the .LT. requirements.

9-46 TRUE or FALSE. The TRUE or FALSE subparameter provides a NOT capability for the boolean expression.

If TRUE is specified, the value of the boolean expression is as described. If FALSE is specified, the inverse of the boolean expression is taken; i. e., 4. LE. 3, FALSE has the boolean value TRUE.

If the subparameter is omitted, the default is TRUE.

Default: If the boolean expression is omitted, the jump is unconditional to the statement specified by the macro-local label.

4-47 JUMP EXAMPLES

```
/ȸJUMPȸ(LABEL)
```

This is an unconditional branch to the macro-local label (LABEL).

```
/ȸJUMPȸ(LABEL), &A. EQ. ABC
```

In this case if A is equal to ABC then branch to macro-local label (LABEL).

9-48 THE PRESET STATEMENT

The PRESET directive establishes default values for positional and/or tagged formal parameters appearing in the macro being defined. The PRESET statement must be used only between a MACRO/MEND pair. There is no limit on the number of PRESET statements that can be used. If more than one PRESET statement specifies values for the same parameters, the last PRESET encountered determines the values to be used during expansion. PRESET statements are not execution-time statements. They are global to the macro definition.

Parameter default values specified on a PRESET statement override the values specified on the MACRO statement for the same parameters. Parameter values specified on the call line override both MACRO and PRESET.

The PRESET statement is formatted as:

		OPERATION		OPERANDS
/	b̸	PRESET	b̸	& formal parameter=actual value [, & formal parameter=actual value] [, ...]

The PRESET statement cannot be labeled.

4-49 PARAMETERS

The parameters on the PRESET are default values for positional and/or tagged parameters. When the macro is expanded, the specified values appear in the expanded code in the place where the formal parameter occurs in the macro definition.

At least one formal parameter and its value must be specified. The remaining parameters are optional and may be omitted.

Format: Each formal parameter must be preceded by an ampersand. The parameter is followed by an equals sign and a default value. The value is a character string or a quoted character string.

Specification: Multiple PRESET statements may be used to specify the desired values. The PRESET's values for tagged parameters override the values specified by the same tagged parameters on the MACRO statement.

Restriction: PRESET statements are the subset of JSL that are valid only for the macro language. PRESET statements are definitional parts of a macro body and can be used only between a MACRO/MEND pair. If more than one PRESET specifies the same parameter value, the last encountered PRESET statement takes precedence.

9-50 PRESET EXAMPLES

```
/bPRESETb&A=A, &B=B
```

This statement sets the value of formal parameters &A to A and &B to B.

9-51 THE INC STATEMENT

The INC statement increments the binary value of the specified formal parameter by one.

The general form of the INC statement is:

	LABEL		OPERATION		OPERANDS
/	[(label)]	b̸	INC	b̸	formal parameter

The INC statement contains a mandatory parameter. In addition, the INC statement may be labeled with a macro-local label; no other labels are allowed. The verb INC must appear in the operations field.

9-52 FORMAL PARAMETER

One formal parameter may be specified; multiple parameters are prohibited. The formal parameter may take the form of a positional, tagged or status parameter. The action taken for a positional or tagged parameter and a status parameter differs. For a positional or tagged parameter, the statement increments only the V (value) component. For a status parameter, the statement increments only those status parameters that are not defined as read only.

Default: None.. A parameter must be specified.

9-53 INC EXAMPLES

```
/b̸INCb̸&A
```

The above example increments the value component of A by 1.

```
/b̸INCb̸!1
```

This example increments the user variable !1 by 1.

9-54 SPECIAL FUNCTION STATEMENTS

There are two statements that enable the user to perform special functions. The DMACRO statement deactivates the most recently entered macro definition of a specified name, and at the same time reactivates a prior definition of a macro of that name. The VERBSYN statement allows a macro, synonym or a JSL verb to be referenced by another name. These special functions provide the user with a great deal of flexibility in handling and referencing the macros he defines and uses.

9-55 THE DMACRO STATEMENT

The DMACRO statement deactivates the most recently entered macro definition of a specified name, and reactivates a prior definition of a macro of that name. The DMACRO statement works on a job's Macro Definition Table. It searches the Macro Definition Table for the most recent macro definition whose label is the same as DMACRO's operand parameter and, finding it, deletes that entry from the table. The macro by the same name that immediately precedes the deactivated macro is reactivated.

The general form of the DMACRO statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	ƀ	DMACRO	ƀ	macro name

The macro name parameter specifies the macro definition to be deactivated. If there is a preceding macro definition of the same name in the Macro Definition Table, that definition becomes the current one. If there is no prior macro of the same name in the Macro Definition Table, the current definition is destroyed, and the job can no longer call the macro specified. If the Macro Definition Table does not contain a single entry labeled like DMACRO's macro name parameter, no error results.

Since the DMACRO statement operates on a job's Macro Definition Table, its results are only valid for the job in which it occurs. To use the DMACRO statement for maintenance of a macro file, transfer the file into the Macro Definition Table by MACASGing it to the current job, operate on it within the job, then recatalog it.

9-56 DMACRO EXAMPLES

```
/ƀDMACROƀXYZ
```

This DMACRO statement deactivates the macro XYZ.

9-57 VERB SYNONYM STATEMENT (VERBSYN)

The verb synonym statement allows a macro, synonym or a JSL verb to be referenced by another name. Subsequent redefinitions of the specified operand do not change the definition of the synonym.

The general form of the VERBSYN statement is:

	LABEL		OPERATION		OPERANDS
/	synonym	b̸	VERBSYN	b̸	{macro name synonym verb}

The VERBSYN statement contains mandatory parameters in both the label and operands fields. The verb VERBSYN must appear in the operations field.

9-58 SYNONYM

The synonym is the new name by which a JSL verb, macro, or other synonym can be referenced.

Format: The synonym must appear in the label field. It follows the slash with no intervening blanks. At least one blank must separate the synonym from the verb. The synonym must be a valid ASC symbol.

Default: None. A name must be specified.

9-59 OPERAND

The operand is the macro, synonym or JSL verb that is being redefined.

Format: The operand may be the name of an existing macro, a defined synonym or a JSL verb. The operand must be specified. Only one entity may be specified.

Default: None. A macro name, a synonym or a JSL verb must be specified.

9-60 VERBSYN EXAMPLES

```
    /XYZb̸VERBSYNb̸XQT
```

The above example redesignates XYZ as XQT. Therefore, anytime the JSL translator encounters XYZ in the operation field it will do an XQT.

9-61 MACRO EXAMPLES

The following example shows a macro written to execute the Fortran compiler. Each statement is numbered and the numbers are used to explain the purpose of the individual statements.

```
 1)  /FTN    MACRO
 2)  /       PRESET   &FTNOPT=(K, M, O), &FADDMEM=4K, ;
                      &VSPACE=ANY, &FTNTIME=1000, ;
                      &SPACE=4096
 3)  /       ASG      S$FCPCH1, S$SYSCAT/S$LPPTCH/;
                      S$FCPCH1, USE=SHR
 4)  /       ASG      S$FCPCH2, S$SYSCAT/S$LPPTCH/;
                      S$FCPCH2, USE=SHR
 5)  /       ASG      S$FCPCH3, S$SYSCAT/S$LPPTCH/;
                      S$FCPCH3, USE=SHR
 6)  /       ASG      S$FCPCH4, S$SYSCAT/S$LPPTCH/;
                      S$FCPCH4, USE=SHR
 7)  /       ASG      S$FTN, S$SYSCAT/S$FTN, USE=SHR
 8)  /       JUMP     (F10), &IN:A. GE. 8
 9)  /       JUMP     (F20), !A. EQ. 0
10)  /       RENAME   SYS. IN, SYS. FIN
11)  /       JUMP     (F20)
12)  /(F10)  RENAME   &IN, SYS. FIN
13)  /(F20)  JUMP     (F30), &LIST:A. GE. 8, FALSE
14)  /       RENAME   &LIST, SYS. PRT
15)  /(F30)  FD       SYS. PRT, BAND=&LISTSIZE
16)  /       BUILD    </        XQT     S$FTN, ;
                      LTP=(1, 1, &SPACE),>
17)  /       BUILD    <ADDMEM=&FADDMEM, ;
                      CPTIME=&FTNTIME, >
18)  /       JUMP     (F40), &VSPACE. EQ. ANY
19)  /       BUILD    <RTP=(&VSPACE), >
20)  /(F40)  BUILD    <CPOPT=&FTNOPT, OPT=(A, I, P) >
21)  /       GEN
```

```
22)  /          JUMP      (F50), &LIST:A. GE. 8
23)  /          FOSYS     SYS. PRT
24)  /          JUMP      (F60)
25)  /(F50)     RENAME    SYS. PRT, &LIST
26)  /(F60)     REL       S$FCPCH1, S$FCPCH2, S$FCPCH3, ;
                          S$FCPCH4, S$FTN
27)  /          JUMP      (F70), &IN:A. GE. 8
28)  /          JUMP      (F80), !A. EG. 0
29)  /          RENAME    SYS. FIN, SYS. IN
30)  /          JUMP      (F80)
31)  /(F70)     RENAME    SYS. FIN, &IN
32)  /(F80)     MEND      FTN
```

The description of the above macros statement by statement, follows:

1) This statement begins the definition of the macro to be called FTN.

2) This statement sets the default values for certain parameters.

3-6) This group of statements assign the patch files that are used by the Fortran compiler.

7) This statement assigns the Fortran compiler.

8) When the formal parameter &IN is specified on the call line then a change of control is executed to label (F10) (statement 12); because when a parameter is specified on the call line it will have an attribute greater than or equal to 8.

9) A change of control occurs to (F20) (statement 13) if a default SYS. IN file is not available on the next macro call statement.

10) There is a default SYS. IN file behind the macro call, therefore, it must be renamed SYS. FIN, which is the standard access name for the Fortran source statement file.

11) This statement is an unconditional transfer of control to the statement with the macro local label (F20) (statement 13).

12) Control comes to this statement only if &IN was specified and must be renamed to SYS. FIN.

13) Control is changed to statement macro local label (F30) (statement 15) if &LIST was not specified by the macro call.

14) Control comes here only if &LIST was specified and will be renamed SYS. PRT, which is the standard access name for the compiler output listing.

15) This statement causes the generation of an FD statement for the SYS. PRT file specifying a band size of &LISTSIZE which is 1/8/1 unless overridden on the macro call line.

16 - 17) These statements build an XQT statement for the Fortran compiler that supplies ADDMEM, CPTIME, and LTP either from default values or macro call line override values.

18) Control is changed to macro local label (F40) (statement 20) if the formal parameter &VSPACE is equal to "ANY".

19) When &VSPACE does not equal "ANY" then add an RTP for &VSPACE to the XQT statement for the Fortran compiler.

20) This statement adds the CPOPT and OPT parameters to the XQT statement for the Fortran program.

21) This statement terminates the BUILD for the XQT statement above.

22) Control is transferred to macro local label (F50) (statement 25) if &LIST was specified on the macro call line.

23) When &LIST is not specified by the macro this statement will dispose of the SYS. PRT file by sending it to the printer.

24) This statement causes an unconditional change of control to macro local label (F60) (statement 26).

25) This statement renames SYS. PRT to &LIST, since the user will expect to find his listing under the access name &LIST.

26) This statement releases all the patch files and the Fortran compiler that were previously assigned in the macro.

27) If &IN was specified on the macro call line control is transferred to the macro local label (F70) (statement 31), otherwise, the next statement in sequence will be executed.

28) If a default SYS. IN does not exist for this macro then control is transferred to macro local label (F80) (statement 32).

29) This statement renames the Fortran input file, SYS. FIN, to SYS. IN, since the Operating System will automatically generate a release for SYS. IN.

30) This statement causes an unconditional transfer of control to macro local label (F80).

31) This statement is executed because &IN was specified on the macro call line, and it renames SYS.FIN to &IN.

32) This statement declares the end of the FTN macro definition.

The following shows another macro example that allows the printing of a file to FOSYS.

```
1)  /PRINT     MACRO     &ACNM, &TYPE
2)  /          BUILD     </b̸b̸b̸b̸FOSYSb̸b̸b̸b̸&ACNM>
3)  /          JUMP      (PRT1), &TYPE:A.GE.8, FALSE
4)  /          BUILD     <,TYPE=&TYPE>
5)  /(PRT1)    GEN
6)  /          MEND
```

The following paragraphs describe or explain the macro statements correspondingly numbered above:

1) This statement begins the definition of the macro PRINT specifying two positional parameters ACNM and TYPE.

2) This statement begins to build the FOSYS statement with a "/" in column one and FOSYS beginning in column six and the access name beginning in column fifteen.

3) This statement checks to see if TYPE was specified on the macro call line, and, if it was not, control is transferred to macro local label (PRT1) (statement 5).

4) This statement receives control only if TYPE was specified and it adds it to the FOSYS statement.

5) This statement terminates the BUILD statement(s).

6) This statement signifies the end of the PRINT macro definition.

APPENDIX A

STATEMENT SUMMARIES

APPENDIX A

STATEMENT SUMMARIES

A-1 GENERAL

Appendix A shows the complete format for all JSL statements and includes a summary of the operands for each statement. The statements are arranged in alphabetical order.

A-2 ASSIGN STATEMENT (ASG)

The general form of the ASG statement is:

	LABEL		OPERATION		OPERANDS
/	Symbol	b̸	ASG	b̸	access name, pathname $\left[\text{, VERS = version}\right] \left[\text{, USE} = \left\{\begin{matrix}\underline{\text{EXC}} \\ \text{SHR}\end{matrix}\right\}\right]$

A-3 ASG STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File identity	Access name	Identifies the assigned file to the job. Access name exists only for the duration of the job.	P	None. Always mandatory.
	Pathname	Specifies the name of the file in the catalog system.	P	None. Always mandatory.
Version specification	Version VERS=version	Specifies the number of the desired version of the file being assigned.	K	The flagged version is assigned.
File Use	Use parameter USE $=\frac{\underline{\text{EXC}}}{\text{SHR}}$	Specifies whether the assigned file on disc is to be shared among jobs or is exclusively reserved for the current job.	K	Exclusive use mode is assumed.

P = POSITIONAL K = KEYWORD

A-3A BATCH JOB INTO INTERACTIVE JOB STATEMENT (BATINT)

The general form of the BATINT statement is:

	LABEL		OPERATION		OPERANDS
/	symbol	b̸	BATINT	b̸	PST=day:hour:min, SST=day: hour:min, SDU=min, TMID=RTn

A-3B BATINT STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Session information	Presession start PST = day: hour: min	Specifies the time by which the job must have started into active processing. It must be later than the time that the batch job is read in and earlier than the session start time by a system defined amount of time.	P	None. It must be given.
	Session start SST = day: hour: min	Specifies the time by which the interactive session is to start. It must be greater than the presession start time by a system defined amount of time.	P	None. It must be given.
	Session duration SDU = min	Specifies the number of minutes that the session is to last (a positive value between 1 and 999999).	P	None. It must be given.
Terminal identification	Terminal id TMID = RTn	Specifies the terminal ID of the terminal at which the interactive session is to take place. It must be a terminal operating under TOS.	P	None. It must be specified.

A-4 CATALOG NODE AND VERSION STATEMENT (CAT)

The general form of the CAT statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	CAT	ƀ	pathname [, ACNM=access name] [, MXVR=maximum number of versions] $\left[\text{, ACST=}\begin{Bmatrix}\underline{\text{NO}}\\ \text{PART}\\ \text{FULL}\end{Bmatrix}\right]\left[\text{, RFST=}\begin{Bmatrix}\underline{\text{NO}}\\ \text{FULL}\end{Bmatrix}\right]$ $\left[\text{, SON=}\begin{Bmatrix}\underline{\text{NO}}\\ \text{FULL}\end{Bmatrix}\right]$ [, ACIN=access control information] $\left[\text{, DTYP=}\begin{Bmatrix}\underline{\text{TAPE}}\\ \text{DSEC}\\ \text{HPT}\\ \text{PAD}\\ \text{OPID}\end{Bmatrix}\begin{Bmatrix}\underline{\text{NOCNTG}}\\ \text{CNTG}\end{Bmatrix}\ \text{operator id}\ \begin{Bmatrix}\underline{\text{CEOF}}\\ \text{CALL}\end{Bmatrix}\right]\left[\text{, TRKS=}\begin{Bmatrix}\underline{9}\\ 7\end{Bmatrix}\right]$ $\left[\text{, RCTL=}\begin{Bmatrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{Bmatrix}\right]$ [, RETP=retention] $\left[\text{, SRC=}\begin{Bmatrix}\underline{\text{DSEC}}\\ \text{TP}\\ \text{FT}\end{Bmatrix}\right]$
					$\left[\text{, FORG=}\begin{Bmatrix}\underline{\text{PS}}\\ \text{DS}\end{Bmatrix}\right]$ $\left[\text{, LABL=}\left[\begin{Bmatrix}\underline{1}\\ \text{position}\end{Bmatrix}\right]\left[/\begin{Bmatrix}\underline{\text{SL}}\\ \text{NL}\end{Bmatrix}\right]\right]$ [, EFID= efid1/efid2/. . . /efidn] $\left[\text{, DEN=}\begin{Bmatrix}\underline{1600}\\ 800\\ 556\\ 356\\ 200\end{Bmatrix}\right]\left[\text{, }\begin{Bmatrix}\text{BAND}\\ \text{SECT}\\ \text{WORD}\\ \text{BYTE}\end{Bmatrix}\text{=number}\right]$

A-5 CAT STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Catalog name	Pathname	Identifies the catalog to contain the new node and its associated file. Specifies the identity and location of the node being created.	P	None. Always mandatory.
File name	ACNM = access name	Identifies the file to the job. Access name exists only for the duration of the job. Valid only for normal catalog processing.	K	Mandatory when SRC = DSEC or when SRC defaults to DSEC. Not valid for SRC = TP or FT.
Maximum number of versions	MXVR = maximum number of versions	Specifies the maximum number of file versions that exist simultaneously at the node being created.	K	One version.
Creating catalog security	Access control state ACST = $\left\{\begin{array}{l}\underline{\text{NO}}\\ \text{PART}\\ \text{FULL}\end{array}\right\}$	Specifies the degree of control the Operating System will maintain over access to the node.	K	NO; no access control.
	Reference control state RFST = $\left\{\begin{array}{l}\underline{\text{NO}}\\ \text{FULL}\end{array}\right\}$	Specifies whether the Operating System will control users' attempt to pass through the node during a catalog search.	K	NO; any user may pass through the node.
	Son-add control state SON = $\left\{\begin{array}{l}\underline{\text{NO}}\\ \text{FULL}\end{array}\right\}$	Specifies the degree of control exercised by the Operating System over the addition of son nodes to the node.	K	NO; any user can add son nodes.
	Access control information ACIN = access control information	Identifies valid users and specifies the actions each may take (attributes). The parameter is a set of subparameters (separated by slashes) each of which consists of a user code appended by a series of alphabetic characters denoting activity attributes. The user code is separated from attributes by an asterisk; each character also is separated by an asterisk. Allowable characters are E, F, O, R, S and/or W. Only ten user codes can be specified in this parameter.	K	Creation of the node as defined by user code specified most recently on a JOB or LIMIT statement in the job creating the node. Creator automatically possesses all attributes.
Destination device	DTYP = $\left\{\begin{array}{l}\underline{\text{TAPE}}\\ \text{DSEC}\\ \text{HPT}\\ \text{PAD}\\ \text{OPID}\end{array}\right\}$	Specifies whether the file being cataloged will be output to 1/2" magnetic tape (1600 bpi) or remain on secondary disc, allocated on head-per-track disc, position-arm-disc or specific channel or module given by the third subparameter.	K	TAPE. The operating system will place the cataloged file on magnetic tape. For SRC = TP or FT, the parameter is invalid.
	/ $\left\{\begin{array}{l}\underline{\text{NOCNTG}}\\ \text{CNTG}\end{array}\right\}$ /	Specifies whether the file may be allocated with or without regard to physical contiguity on disc.		No contiguity required.
	/operator id/	Specifies specific device related channel or module used in conjunction with OPID in first subparameter.		None.
	/ $\begin{array}{l}\underline{\text{CEOF}}\\ \text{CALL}\end{array}$	Specifies whether initial allocation of cataloged copy of file is set to the original file's current end of file or to the original file's current allocation.		Initial allocation set to file's current end of file.

P = POSITIONAL K = KEYWORD

A-5 CAT STATEMENT SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Retention period	RETP = retention	Specifies the length of time a cataloged file is to be retained. The number of days is converted to an expiration date that is written in the Tape Volume Catalog and the label for standard-labeled tapes. The parameter is not meaningful for files on secondary disc.	K	An installation-defined value. Not valid for DTYP = DSEC.
Source location of file	$SRC = \begin{Bmatrix} \underline{DSEC} \\ TP \\ FT \end{Bmatrix}$	Specifies the source device on which to-be-cataloged file resides. DSEC specifies secondary disc; TP half-inch magnetic tape volume; and FT one-inch tape volume. SRC also indicates whether normal cataloging functions or special processing will occur. SRC=DSEC indicates normal processing. SRC=TP or FT means special catalog processing.	K	DSEC. Normal catalog processing occurs.
The following parameters apply <u>only</u> to special catalog processing in which the file to be cataloged exists on tape and is <u>not</u> buffered to secondary disc for processing.				
File organization	$FORG = \begin{Bmatrix} \underline{PS} \\ DS \end{Bmatrix}$	Specifies the organization of the cataloged file. Organization can be one of two types: PS, physical sequential; DS, direct secondary. The source parameter must be set to TP or FT.	K	PS. If SRC is not specified or is set to DSEC, the parameter is meaningless.
Volume specification	External file identification EFID = tape id1 /tape id2 . . . /tape idn	Specifies the identification numbers of the source volumes containing the file to be cataloged. All identifiers must be declared and must appear in the sequence that the tape volumes are to be read.	K	Mandatory if SRC=TP or FT. If SRC is not specified or is set to DSEC, the parameter is meaningless.
	Label $LABL = \begin{Bmatrix} \underline{1} \\ position \end{Bmatrix} \begin{Bmatrix} \underline{SL} \\ NL \end{Bmatrix} /$	Describes characteristics of the tape volume that will contain the files specified. The parameter is comprised of a set of two subparameters separated by slashes, each of which is positional and optional:	K	If omitted, the default values are 1 and SL. If SRC is not specified or is set to DSEC, the parameter is not meaningful.
		Position	P	1
		Label type	P	SL
	Density $DEN = \begin{Bmatrix} \underline{1600} \\ 800 \\ 712 \\ 556 \\ 356 \\ 200 \end{Bmatrix}$	Specifies the recording density in bits-per-inch-per-track for the tape volumes designated by the EFID parameter. The parameter is valid only for special catalog processing. 1600 and 800 bpi pertain to half-inch, 9 track tape. 800, 556 and 200 bpi pertain to half-inch 7 track tape. 712 and 356 bpi pertain to one-inch tape.	K	1600 bpi for SRC=TP and TRKS=9 or defaulted. 800 bpi for SRC=TP and TRKS=7. If SRC is not specified or is set to DSEC the DEN parameter is meaningless.
	Track parameter $TRKS = \begin{Bmatrix} \underline{9} \\ 7 \end{Bmatrix}$	Specifies whether the half-inch tape is 9 track or 7 track.	K	If omitted, 9 track is assumed.

A-5 CAT STATEMENT SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
	Recording control parameter $\text{RCTL} = \left\{\begin{matrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{matrix}\right\}$	Specifies recording control for parity, translation, and data conversion for 7 track tape O - odd parity, no translation, no conversion. E - even parity, no translation, no conversion. T - odd parity, translation, no conversion. C - odd parity, no translation, no conversion. ET - even parity, no translation, conversion.	K	If omitted and TRKS=7 ET is assumed. If TRKS=9 or defaulted this parameter does not apply.
Managing secondary disc space	$\left\{\begin{matrix}\text{BAND}\\ \text{SECT}\\ \text{WORD}\\ \text{BYTE}\end{matrix}\right\}$ = number	Manages disc space for the to-be-cataloged file during subsequent assignment to a job. The parameter value is an unsigned constant less than 999999_{10}.	K	An installation-defined value. If SRC is not specified or is set to DSEC, the disc parameter is invalid.

A-6 CATBLD STATEMENT

The general form of the CATBLD statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	CATBLD	ƀ	edgename [, MXVR=maximum number of versions]
					$\left[, \text{ACST}=\left\{\begin{matrix}\underline{\text{NO}}\\ \text{PART}\\ \text{FULL}\end{matrix}\right\}\right] \left[, \text{SON}=\left\{\begin{matrix}\underline{\text{NO}}\\ \text{FULL}\end{matrix}\right\}\right]$
					$\left[, \text{RFST}=\left\{\begin{matrix}\text{NO}\\ \text{FULL}\end{matrix}\right\}\right]$
					[, ACIN=access control information]

A-7 CATBLD SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Catalog name	Edgename	Identifies the catalog being defined to the catalog system.	P	None. It must be specified.
Maximum number of versions	MXVR = maximum number of versions	Specifies the maximum number of file versions to exist simultaneously at the node.	K	If omitted OS assumes only 1 version.
Access control state	$\text{ACST} = \left\{\begin{array}{l}\underline{\text{NO}}\\ \text{PART}\\ \text{FULL}\end{array}\right\}$	Specifies the degree of control system will maintain over access to node NO - all users may read, write or execute PART - all users read and execute FULL - access to qualified users only	K	If omitted NO access control will be maintained.
Son-add control state	$\text{SON} = \left\{\begin{array}{l}\underline{\text{NO}}\\ \text{FULL}\end{array}\right\}$	Specifies degree of control exercised by OS over the addition of nodes to the root node. NO - any user can add sons to the specified node FULL - only qualified users can add son nodes	K	If omitted NO son-add control will be maintained.
Reference control state	$\text{RFST} = \left\{\begin{array}{l}\underline{\text{NO}}\\ \text{FULL}\end{array}\right\}$	Specifies whether OS will control users' attempt to pass through node during a catalog search. NO - any user can pass through node FULL - only qualified users can pass through node	K	If omitted NO reference control will be maintained.
Access control information	ACIN = access control information	Provides security measures for catalog being built. Establishes valid users and actions each may take through attributes given: E - execute F - reference O - own R - read S - son add W - write	K	If omitted only creator of node possesses all attributes.

A-8 CATN STATEMENT

The general form of the CATN statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	CATN	ƀ	pathname [, MXVR=maximum number of versions] [, ACST=$\left\{\begin{matrix}\underline{\text{NO}}\\ \text{PART}\\ \text{FULL}\end{matrix}\right\}$][, SON= $\left\{\begin{matrix}\underline{\text{NO}}\\ \text{FULL}\end{matrix}\right\}$] [, RFST=$\left\{\begin{matrix}\underline{\text{NO}}\\ \text{FULL}\end{matrix}\right\}$][, ACIN=access control information]

A-9 CATN SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Catalog node name	pathname	Specifies the location and name of the new node in the catalog.	P	None. It must be specified.
Maximum number of versions	MXVR = maxiimum number of versions	Specifies the maximum number of versions at this node.	K	If omitted only one version is allowed.
Access control state	$\text{ACST} = \left\{\begin{array}{l}\underline{\text{NO}}\\ \text{PART}\\ \text{FULL}\end{array}\right\}$	Specifies the degree of control system will maintain over access to node NO - all users can read, write or execute PART - all users can read and execute FULL - access only to qualified users	K	If omitted default is to NO.
Son-add control state	$\text{SON} = \left\{\begin{array}{l}\underline{\text{NO}}\\ \text{FULL}\end{array}\right\}$	Specifies degree of control exercised by OS over the addition of nodes to this node. NO - any user can add to the specified node FULL - only qualified users can add son nodes	K	If omitted default is to NO.
Reference control state	$\text{RFST} = \left\{\begin{array}{l}\underline{\text{NO}}\\ \text{FULL}\end{array}\right\}$	Specifies whether OS will control users' attempt to pass through this node during a catalog search NO - any user can pass through node FULL - only qualified users can pass through node	K	If omitted default is to NO.
Access control information	ACIN = access control information	Provides security measures for node being built. Establishes valid users and actions each may take through attributes given. E - execute F - reference O - own R - read W - write S - son add	K	If omitted only creator of node possesses all attributes.

P = POSITIONAL K = KEYWORD

A-10 THE CATALOG VERSION STATEMENT (CATV)

The general form of the CATV statement is:

```
  | LABEL    |   | OPERATION |   | OPERANDS
--+----------+---+-----------+---+--------------------------------------------------
/ | [Symbol] | ƀ |   CATV    | ƀ | pathname [, ACNM=access name]
  |          |   |           |   |
  |          |   |           |   |          ( TAPE )
  |          |   |           |   |          ( DSEC )       ( NOCNTG )
  |          |   |           |   | [, DTYP= ( HPT  )   /    ( CNTG   )   /
  |          |   |           |   |          ( PAD  )
  |          |   |           |   |          ( OPID )
  |          |   |           |   |
  |          |   |           |   |    operator ID   / ( CEOF ) ]
  |          |   |           |   |                    ( CALL ) ]
  |          |   |           |   |
  |          |   |           |   | [, RETP=retention]
  |          |   |           |   |
  |          |   |           |   |         ( DSEC )
  |          |   |           |   | [, SRC= ( TP   )    ]
  |          |   |           |   |         ( FT   )
  |          |   |           |   |---------------------------------------------
  |          |   |           |   |          ( PS )
  |          |   |           |   | [, FORG= ( DS )   ]
  |          |   |           |   |
  |          |   |           |   |             (    1     )      ( SL )
  |          |   |           |   | [, LABL=[   ( position )  ][/ ( NL )  ]]
  |          |   |           |   |
  |          |   |           |   | [, EFID=efid1/efid2/. . . /efidn]
  |          |   |           |   |
  |          |   |           |   |         ( 1600 )
  |          |   |           |   |         ( 800  )
  |          |   |           |   | [, DEN= ( 712  )  ][TRKS= ( 9 ) ]
  |          |   |           |   |         ( 556  )          ( 7 )
  |          |   |           |   |         ( 356  )
  |          |   |           |   |         ( 200  )
  |          |   |           |   |
  |          |   |           |   |          ( ET )
  |          |   |           |   |          ( E  )
  |          |   |           |   | [, RCTL= ( O  )   ]
  |          |   |           |   |          ( T  )
  |          |   |           |   |          ( C  )
  |          |   |           |   |
  |          |   |           |   |     ( BAND )
  |          |   |           |   |     ( SECT )
  |          |   |           |   | [,  ( WORD )  =  number ]
  |          |   |           |   |     ( BYTE )
```

A-11 CATV STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Catalog name	Pathname	Identifies the catalog to contain the new file. Specifies the identity and location of the node containing the file.	P	None. Always mandatory.
File name	ACNM = access name	Identifies the file to the job. Access name exists only for the duration of the job. Valid only for normal catalog processing.	K	Mandatory when SRC=DSEC or when SRC defaults to DSEC. Not valid for SRC=TP or FT.
Destination device	DTYP = {TAPE DSEC HPT PAD OPID}	Specifies whether the file being cataloged will be output to half-inch magnetic tape (1600 bpi) or remains on secondary disc, allocated on head-per-track disc, position-arm-disc, or specific channel or module given by the third subparameter.	K	TAPE. The Operating System will place the cataloged file on magnetic tape. For SRC = TP or FT, the parameter is invalid.
	/{NOCNTG CNTG}/	Specifies whether the file may be allocated with or without regard to physical contiguity on disc.		No contiguity required.
	/operator id/	Specifies specific device related channel or module used in conjunction with OPID in first subparameter.		None.
	/CEOF CALL/	Specifies whether initial allocation of cataloged copy of file is set to the original files current end of file or to the original files current allocation.		Initial allocation set to file's current end of file.
Retention period	RETP = retention	Specifies the length of time a cataloged file is to be retained. The number of days is converted to an expiration date that is written in the Tape Volume Catalog and the label for standard-labeled tapes. The parameter is not meaningful for files on secondary disc.	K	An installation-defined value. Not valid for DTYP=DSEC.

Source location of file	SRC = {DSEC
	TP
	FT}
		Specifies the source device on which to-be-cataloged file resides. DSEC specifies secondary disc; TP half-inch magnetic tape volume; and FT one-inch tape volume. SRC also indicates whether normal cataloging functions or special processing will occur. SRC=DSEC indicates normal processing. SRC=TP or FT means special catalog processing.	K	DSEC. Normal catalog processing occurs.

The following parameters apply only to special catalog processing in which the file to be cataloged exists on tape and is not buffered to secondary disc for processing.				

File organization	FORG = {PS
	DS}
		Specifies the organization of the cataloged file. Organization can be one of two types: PS, physical sequential; DS, direct secondary. The source parameter must be set to TP or FT.	K	PS. If SRC is not specified or is set to DSEC, the parameter is meaningless.

P = POSITIONAL K = KEYWORD

A-11 CATV (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Volume specification	External file identification EFID = tape id1 /tape id2 ... /tape idn	Specifies the identification numbers of the source volumes containing the file to be cataloged. All identifiers must be declared and must appear in the sequence that the tape volumes are to be read.	K	Mandatory if SRC=TP or FT. If SRC is not specified or is set to DSEC, the parameter is meaningless.
	Label $LABL=\left\{\begin{array}{c}\underline{1}\\ \text{position}\end{array}\right\}/\left\{\begin{array}{c}\underline{SL}\\ NL\end{array}\right\}$	Describes characteristics of the tape volume that will contain the files specified. The parameter is comprised of a set of two subparameters separated by slashes, each of which is positional and optional:	K	If omitted, the default values are 1 and SL. If SRC is not specified or is set to DSEC, the parameter is not meaningful.
		Position value is 1 ≤ n ≤ 255	P	1
		Label type value is either SL or NL	P	SL
	Density $DEN=\left\{\begin{array}{c}\underline{1600}\\ 800\\ 712\\ 556\\ 356\\ 200\end{array}\right\}$	Specifies the recording density in bits-per-inch-per-track for the tape volumes designated by the EFID parameter. The parameter is valid only for special catalog processing. 1600 and 800 bpi pertain to half-inch 9-track tape; 800, 556 and 200 bpi pertain to half-inch 7-track tape; 712 and 356 to one-inch tape.	K	1600 bpi for SRC=TP and TRKS=9 or defaulted. 800 bpi for SRC=TP and TRKS=7.
	Track parameter $TRKS=\left\{\begin{array}{c}\underline{9}\\ 7\end{array}\right\}$	Specifies whether the half-inch tape is 9 track or 7 track.	K	If omitted, 9 track is assumed.
	Recording control parameter $RCTL=\left\{\begin{array}{c}\underline{ET}\\ E\\ O\\ T\\ C\end{array}\right\}$	Specifies recording control for parity, translation, and data conversion for 7 track tape O - odd parity, no translation, no conversion. E - even parity, no translation, no conversion. T - odd parity, translation, no conversion. C - odd parity, no translation, conversion. ET - even parity, translation, no conversion.	K	If omitted and TRKS=7 ET is assumed. If TRKS=9 or defaulted this parameter does not apply.
Managing secondary disc space	$\left\{\begin{array}{l}BAND\\ SECT\\ WORD\\ BYTE\end{array}\right\}$ = number	Manages disc space for the to-be-cataloged file during subsequent assignment to a job. The parameter value is an unsigned constant less than 999999_{10}.	K	An installation-defined value. If SRC is not specified or is set to DSEC, the disc parameter is invalid.

P = POSITIONAL K = KEYWORD

A-12 CHG STATEMENT

The general form of the CHG statement is:

	LABEL		OPERATION		OPERANDS
ƀ	[Symbol]	ƀ	CHG	ƀ	pathname [, MXVR=maximum number of versions]
					[, FLVR=flagged version number]
					[, ACST=$\left\{\begin{matrix}\underline{\text{NO}}\\\text{FULL}\\\text{PART}\end{matrix}\right\}$][, SON=$\left\{\begin{matrix}\underline{\text{NO}}\\\text{FULL}\end{matrix}\right\}$]
					[, RFST=$\left\{\begin{matrix}\underline{\text{NO}}\\\text{FULL}\end{matrix}\right\}$][, ACIN=access control information]
					[, RETP=retention[, VERS=version number]]
					[, ASEQ=$\left\{\begin{matrix}\text{ON}\\\text{OFF}\end{matrix}\right\}$]

A-13 CHG SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Catalog node name	Pathname	Specifies to the catalog system the node being changed.	P	None. It must be specified.
Maximum number of versions	MXVR = maximum number of versions	Specifies the maximum number of versions at the node.	K	If omitted the current maximum number is retained.
Flagged version	FLVR = flagged version number	Specifies the absolute number of a new flagged version.	K	If omitted the current flagged version does not change.

Access control state	ACST = { NO
	FULL
	PART }
		Specifies the degree of control the OS maintains over access to this node
		NO - all users can read, write or execute
		PART - all users can read and execute
		FULL - only qualified users have access
			K	If omitted the current access control is retained.

Son-add control state	SON = { NO
	FULL }
		Specifies degree of control exercised by OS over addition of nodes to this node
		NO - any user can add to the specified node
		FULL - only qualified users can add son nodes
			K	If omitted, the son-add state remains unchanged.

Reference control state	RFST = { NO
	FULL }
		Specifies whether OS will control users' attempts to pass through this node
		NO - any user can pass through node
		FULL - only qualified users can pass through node
			K	If omitted, the reference state remains unchanged.

Access control information	ACIN = access control information	Specifies additions to the access control List, modification of attributes of users currently in list or deletions of users from the Access Control List E - execute F - reference O - own R - read S - son add W - write	K	If omitted, the contents of the access control list remain unchanged.

Tape file retention period	RETP = retention
	VERS = version number
		Specifies number of days the tape file version indicated by the VERS subparameter will be retained. Not applicable to disc files.	K	If VERS omitted, the flagged version is retained the number of days specified.

Automatic sequencing	ASEQ = { ON
	OFF }
		Specifies change in sequencing
		ON - activates
		OFF - deactivates
			K	If omitted sequencing remains unchanged.

P = POSITIONAL K = KEYWORD

A-14 CJSL STATEMENT

The general form of the CJSL statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	CJSL	b̸	access name

A-15 CJSL STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Job specification source file	Access name	Specifies the name of the file containing job specification source to be syntax checked.	P	None. It must be specified.

P = POSITIONAL K = KEYWORD

A-16 CNT STATEMENT

The general form of the CNT statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	CNT	b̸	access name

A-17 CNT STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Access name of concatenated file	Access name	Identifies the set of concatenated files to the job. ASG, FIT and FD statements are the only statements that can be used in file concatenation. These appear between the CNT and CNTE statements.	P	None. It must be specified.

P = POSITIONAL K = KEYWORD

A-18 CNTE STATEMENT

The general form of the CNTE statement is:

		OPERATION		OPERANDS
/	b̸	CNTE		None

A-19 DEL STATEMENT

The general form of the DEL statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	DEL	b̸	pathname

A-20 DEL STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Catalog node name	Pathname	Initiates deletion of terminal node and associated files from a catalog.	P	None. It must be specified

P = POSITIONAL K = KEYWORD

A-21 DELETE VERSION STATEMENT (DELV)

The general form of the DELV statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	DELV	b̸	pathname [, VERS = version number] [, OPT = (option)]

A-22 DELV STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Catalog node name	Pathname	Identifies the node in the catalog at which the specified version is to be deleted.	P	None. Always mandatory.
Specifying the version	VERS = version number	Specifies the number of the version to be deleted. Overrides option parameter.	K	Flagged version will be deleted.
	OPT = (option)	Permits the deletion of all versions cataloged at the specified node. Only option code A may be used at the present time. This parameter is valid only if the VERS parameter is not specified.	K	Version specified by VERS parameter or its default, the flagged version.

P = POSITIONAL K = KEYWORD

A-23 END-OF-JOB STATEMENT (EOJ)

The general form of the EOJ statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	EOJ	b̸	None

A-24 EOJX STATEMENT

The general form of the EOJX statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	EOJX	b̸	none

A-25 FILE DESCRIPTION STATEMENT (FD)

The general form of the FD statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	FD	b̸	access name
					$[,\left\{\begin{matrix}\text{SECT}\\\text{BAND}\\\text{WORD}\\\text{BYTE}\end{matrix}\right\} = \text{initial/maximum/ increment}]$
					$[,\text{FORG} = \left\{\begin{matrix}\text{PS}\\\text{DS}\end{matrix}\right\}]\ [,\text{BKSZ} = \text{number}]$
					[, LREC = number]
					$[,\text{EROP} = \left\{\begin{matrix}\text{ACC}\\\text{SKP}\\\text{ABE}\end{matrix}\right\}]$
					$[,\text{RCFM} = \left\{\begin{matrix}\text{F} & \begin{bmatrix}\text{B}\\\text{BS}\\\text{S}\end{bmatrix} & \left[\left\{\begin{matrix}\text{A}\\\text{M}\end{matrix}\right\}\right]\\\text{V} & & \\\text{U} & \begin{bmatrix}\text{A}\\\text{M}\end{bmatrix} & \end{matrix}\right\}]$
					$[,\text{BFTK} = \text{S}]\ [,\text{BFAL} = \left\{\begin{matrix}\text{F}\\\text{D}\\\text{O}\end{matrix}\right\}]$
					[, BUFL = number]
					[, BUFN = number]
					$[,\text{POS} = \left\{\begin{matrix}\text{NEW}\\\text{MOD}\end{matrix}\right\}]$
					$[,\ \text{DTYP} = \left\{\begin{matrix}\underline{\text{DSEC}}\\\text{HPT}\\\text{PAD}\\\text{OPID}\end{matrix}\right\}]\ /[\left\{\begin{matrix}\underline{\text{NOCNTG}}\\\text{CNTG}\end{matrix}\right\}]$
					/[operator id]
					[, SETUP = set up id]

A-26 FD STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File information	Access name	Identifies the file to the job. Used by other JSL statements referencing the file.	P	None. Always mandatory.
	Disc reservation $\left\{\begin{matrix}\text{SECT}\\\text{BAND}\\\text{WORD}\\\text{BYTE}\end{matrix}\right\}$ = initial/maximum/increment	Manages disc space for the specified file for the duration of the job. There are three subparameters:	K	Installation-defined values.
		Initial	P	The entire set of subparameters must be coded or the set defaulted.
		Maximum	P	
		Increment	P	
	Position $\text{POS} = \left\{\begin{matrix}\text{NEW}\\\text{MOD}\end{matrix}\right\}$	Specifies the initial position of the file during file opening. NEW specifies that the file is positioned at its beginning. MOD specifies that the file is positioned at its end.	K	NEW if value not previously specified. If previously specified, the condition does not change.
File logical characteristics	File organization $\text{FORG} = \left\{\begin{matrix}\text{PS}\\\text{DS}\end{matrix}\right\}$	Specifies the organization of the file on disc: PS sequential DS direct secondary	K	No system supplied values. May be supplied by catalog entry, default values of FIT, or through the file open process.
	Block size BKSZ = number	Specifies the maximum number of bytes in a block within the specified file.	K	No system-supplied values. May be supplied by tape labels.
	Record length LREC = number	Specifies the maximum length in bytes of any logical record within the specified file.	K	No system supplied values. May be supplied by tape labels.
	Record format $\text{RCFM=F}\left\{\begin{matrix}\text{B}\\\text{BS}\\\text{S}\end{matrix}\right\}\left\{\begin{matrix}\text{A}\\\text{M}\end{matrix}\right\}$ $\text{V}\left\{\begin{matrix}\text{B}\\\text{BS}\\\text{S}\end{matrix}\right\}\left\{\begin{matrix}\text{A}\\\text{M}\end{matrix}\right\}$ $\text{U}\left\{\begin{matrix}\text{A}\\\text{M}\end{matrix}\right\}$	Specifies the format of the logical records within the specified file. Allowable combinations: $\text{F}\left\{\begin{matrix}\text{B}\\\text{BS}\\\text{S}\end{matrix}\right\}\left\{\begin{matrix}\text{A}\\\text{M}\end{matrix}\right\}$ $\text{V}\left\{\begin{matrix}\text{B}\\\text{BS}\\\text{S}\end{matrix}\right\}\left\{\begin{matrix}\text{A}\\\text{M}\end{matrix}\right\}$ $\text{U}\left\{\begin{matrix}\text{A}\\\text{M}\end{matrix}\right\}$	K	No system supplied values. May be supplied by tape labels. If RCFM is specified, subparameter defaults are: <u>length</u>: invalid if not coded. <u>blocking</u>: none <u>structure</u>: no spanning or not standard. <u>control character</u>: none present.
Buffer management	Buffering type BFTK = S	Defines the type of buffering to be used during CP step usage of specified file. Only simple buffering can be specified.	K	Depends on access method.

	Buffer alignment
	$\text{BFAL} = \left\{\begin{matrix}\text{F}\\\text{D}\\\text{O}\end{matrix}\right\}$
		Defines the alignment wanted for each buffer used during CP step access for the specified file.
		F = fullword alignment
		D = doubleword alignment
		O = octet alignment
			K	Depends on access method.

	Buffer length BUFL = number	Defines the length in bytes of each buffer used during CP step processing of the specified file.	K	Depends on access method.
	Buffer number BUFN = number	Defines the number of buffers desired for processing the specified file during CP step execution.	K	Depends on access method.

P = POSITIONAL K = KEYWORD

A-26 FD STATEMENT SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
System specifications	Error option EROP = $\begin{Bmatrix}\text{ACC}\\\text{SKP}\\\text{ABE}\end{Bmatrix}$	Specifies the action to be taken by the System if an uncorrectable input/output error occurs. ACC - accept the block causing the error. SKP - skip the block causing the error. ABE - terminate the step.	K	No system supplied values. May be supplied by tape labels.
Destination device	DTYP = $\begin{Bmatrix}\underline{\text{DSEC}}\\\text{HPT}\\\text{PAD}\\\text{OPID}\end{Bmatrix}$/	Specifies if file is to be allocated on head-per-track disc, position-arm-disc device specified in OS default parameter table or a specific channel or module specified by the third subparameter.	K	If omitted OS assumes DSEC and file will be allocated on device specified in the system default parameter table.
	/$\begin{matrix}\underline{\text{NOCNTG}}\\\text{CNTG}\end{matrix}$/	Specifies whether the file may be allocated with or without regard to physical contiguity on disc.		No contiguity required.
	/operator id/	Specifies specific device related		None
	/$\begin{matrix}\underline{\text{CEOF}}\\\text{CALL}\end{matrix}$	Specifies whether initial allocation of file is set to the original file's current end of file or to the original file's current allocation		Initial allocation set to file's current end of file.
	SETUP = setup id	Specifies reference to a SETUP statement through use of the SETUP statement label and the file defined by the FD statement will be allocated on the same channel selected by the SETUP statement.	K	If omitted there is no default value. The location of the file is determined by the DTYP parameter or its default value.
Log of events	LOE = access name	Specifies the access name of a file to be used as a log-of-error events for the file specified.		If omitted no log is compiled. As a result, termination of file transfer can result upon error detection.

P = POSITIONAL K = KEYWORD

A-27 FILE INPUT FROM A HALF-INCH MAGNETIC TAPE (FIT)

The general form of the FIT statement is:

	LABEL		OPERATION		OPERANDS
/	Symbol	ƀ	FIT	ƀ	access name [EFID=tape id1[/tape id2[/... [/tape id n]]] $\left[,\text{LABL=}\left[\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right]\quad\left[/\left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\\ \text{BLP}\end{matrix}\right\}\right]\right]$ [,RTRY=number] $\left[,\text{PRTY=}\left\{\begin{matrix}\underline{\text{ABE}}\\ \text{ACC}\end{matrix}\right\}\right]$ $\left[,\text{DEN=}\left\{\begin{matrix}1600\\ 800\\ 556\\ 200\end{matrix}\right\}\right]$ $\left[,\text{TRKS=}\left\{\begin{matrix}\underline{9}\\ 7\end{matrix}\right\}\right]\quad\left[,\text{RCTL=}\left\{\begin{matrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{matrix}\right\}\right]$ [, LOC=dest code]

A-28 FIT STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File access name	Access name	Specifies the name associated with the file to be transferred.	P	None. Always mandatory.
Tape volume description	External file identification EFID = tape id1 [/tape id 2]	Specifies the identifiers of volumes containing the file. When the FIT statement occurs in an MFR/MFRE group this parameter is overridden by the MFR statement specifications.	K	None. At least one identifier must be specified.
	Label parameter $\text{LABL} = \left\{\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right\}/\left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\}$	Specifies the type of labels on volumes containing the specified file and designates the initial positioning of the first volume before the data is transferred. The two subparameters are positional but not mandatory. When the FIT statement occurs in an MFR/MFRE group and the LABL parameter is specified on the FIT statement the file position subparameter will override the sequence set up by the file position subparameter specified on the MFR statement or its default and subsequent FIT statements will be processed according to the file position designated if the LABL parameter is omitted on these statements.	K	LABL = 1/SL
	Tape density parameter $\text{DEN} = \left(\begin{matrix}\underline{1600}\\ 800\\ 712\\ 556\\ 356\\ 200\end{matrix}\right)$	Specifies the recording density in bits-per-inch per track for the tape volume designated by the EFID parameter. 1600 and 800 bpi pertain to half-inch, 9 track tape. 800, 556 and 200 bpi pertain to half-inch, 7 track tape. 712 and 356 pertain to one-inch tape. When the FIT statement occurs in an MFR/MFRE group this parameter is overridden by the MFR statement specification.		1600 bpi for SRC=TP and TRKS=9 or defaulted 800 bpi for SRC=TP and TRKS=7. If SRC is not specified or is set to DSEC the DEN parameter is meaningless.
	Track parameter $\text{TRKS} = \left\{\begin{matrix}\underline{9}\\ 7\end{matrix}\right\}$	Specifies whether the half-inch tape is 9 track or 7 track. When the FIT statement occurs in an MFR/MFRE group this parameter is overridden by the MFR statement specification	K	If omitted, 9 track is assumed.
	Recording control parameter $\text{RCTL} = \left\{\begin{matrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{matrix}\right\}$	Specifies recording control for parity, translation and data conversion for 7 track tape O odd parity, no translation, no conversion E even parity, no translation, no conversion T odd parity, translation, no conversion C odd parity, no translation, conversion ET even parity, translation, no conversion. When the FIT statement occurs in an MFR/MFRE group this parameter is overridden by the MFR statement specification.	K	If omitted and TRKS=7 ET is assumed. If TRKS=9 or defaulted this parameter does not apply.

P = POSITIONAL K = KEYWORD

A-28 FIT STATEMENT SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Error processing	Retry parameter RETRY = number	Specifies the maximum number of times the System attempts to read a record that has a parity error before the error becomes permanent.	K	RTRY = 5
	Parity error option parameter PRTY = $\left\{\begin{matrix}\text{ABE}\\\text{ACC}\end{matrix}\right\}$	Specifies the action taken if a permanent error occurs. One of two actions can be specified: ABE - abort the file transfer at this point. ACC - accept the error and continue processing.	K	PRTY = ABE
Destination specifications	LOC = destination code	Specifies the site at which the tape file is to be input. The parameter value is in string of one to eight alphanumeric characters designating the site. When the FIT statement occurs in an MFR/MFRE group, the LOC parameter is overridden by the MFR specifications.	K	Central site code

P = POSITIONAL K = KEYWORD

A-29 FILE OUTPUT TO SYSTEM STATEMENT (FOSYS)

The general form of the FOSYS statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	FOSYS	b̸	access name [, TYPE = $\left\{\begin{matrix}\text{PRINT}\\\text{PUNCH}\end{matrix}\right\}$] [, EXID = external file ID] [, LOC = location ID] [, FORM = [special forms ID] / [carriage control tape ID]] [, PRTS = line spacing] [, COPIES = aa] [, OPT = (R)]

A-30 FOSYS STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File identification	Access name	Identifies the file to the job.	P	None. Always mandatory.
File type	$\text{TYPE} = \left\{\begin{matrix}\underline{\text{PRINT}}\\ \text{PUNCH}\end{matrix}\right\}$	Specifies whether the file is to be placed on the print or punch output chain.	K	PRINT. The file is spooled on the print output chain.
Output location	LOC = location ID	Specifies the installation at which the file is output.	K	The location specified on the JOB statement. If JOB does not specify a location identifier, the default is the installation where the job was submitted.
Output name	EXID = external file identifier	Specifies the name of the output file on printouts or cards. For print files, the print image is preceded by a special line with the external file identifier. For punch files, a card containing the external file identifier precedes each deck.	K	No identifying message on printout or punched card will be output preceding the output file information.
Special forms	FORM = special forms ID/carriage control tape ID	The parameter consists of two subparameters. The special forms identifier specifies special paper and forms for both print and punch files. The carriage control tape identifier specifies a carriage control tape for print files.	K P P	Installation-assigned identifiers.
Printer spacing/ stacker selection	PRTS = line spacing or stacker selection characters	Specifies the line spacing by the printer between lines of code or the card stacker receiving the punched cards.	K	For print or punch files, PRTS = 1.
Disposition of disc space	OPT = (R)	Specifies that the file has been spooled to the system print or punch output chains and that the disc space previously occupied may be reassigned for other use.	K	OPT = R
Number of copies	COPIES = aa	Specifies the number of copies of the specified file to be output at the designated device. The parameter value is an unsigned constant from 1 to 99.	K	One copy is output.
NOTE: The OPT parameter has been designed so that future options may be added.				

P = POSITIONAL K = KEYWORD

A-31 FILE OUTPUT TO HALF-INCH MAGNETIC TAPE (FOT)

The general form of the FOT statement is:

	LABEL		OPERATION		OPERANDS
/	Symbol	b̸	FOT		access name $\left[,\text{EFID=tape id1}\left[\text{/tape id2}\left[/\ldots\right]\right]\right]$ $\left[,\text{LABL=}\left[\begin{Bmatrix}\underline{1}\\ \text{position}\end{Bmatrix}\right]\left[/\left[\begin{Bmatrix}\underline{\text{SL}}\\ \text{NL}\end{Bmatrix}\right]\left[/\left[\begin{Bmatrix}\underline{\text{NEW}}\\ \text{ADD}\\ \text{APND}\end{Bmatrix}\right]\right]\right]\right]$ $\left[,\text{SVC=scratch volume count}\right]$ $\left[,\text{OPT=}(\begin{smallmatrix}\underline{\text{N}}\\ \text{R}\end{smallmatrix})\right]$ $\left[,\text{RETP=retention}\right]\left[,\text{RTRY=}\begin{Bmatrix}\underline{5}\\ \text{number}\end{Bmatrix}\right]$ $\left[,\text{PRTY=}\begin{Bmatrix}\underline{\text{ABE}}\\ \text{ACC}\end{Bmatrix}\right]\left[,\text{LOC=location id}\right]$ $\left[,\text{DEST=destination code}\right]$ $\left[,\text{DEN=}\begin{Bmatrix}1600\\ 800\\ 556\\ 200\end{Bmatrix}\right]\left[,\text{TRKS=}\begin{Bmatrix}\underline{9}\\ 7\end{Bmatrix}\right]$ $\left[,\text{RCTL=}\begin{Bmatrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{Bmatrix}\right]$

A-32 FOT STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File access name	Access name	Specifies the name associated with the file to be transferred. The name must be a valid ASC symbol.	P	None. Always mandatory.

Error processing specification	RTRY = {5
	number}
		Specifies the number of times the Operating System is to attempt to write a block to the tape if a parity error occurs. The value of the parameter is an unsigned constant ranging from 0 to 15.	K	RTRY = 5
				Five attempts will be made.

	PRTY = {ABE
	ACC}
		Specifies the action the Operating System takes if the number of write-error retries is exceeded. One of two actions can be specified:
		ABE - abort the file transfer at this point.
		ACC - accept the error and continue processing.
			K	PRTY = ABE

Tape volume description	EFID = tape id1 [/tape id2 [/... [/tape idn]]]	Specifies the identification numbers of the tape volumes that are to contain the file. When the FOT statement occurs in an MFR/MFRE group, this parameter is overridden by the MFR statement specifications.	K	The file is written to a system scratch volume.

	LABL = {1
	position}
	/{SL
	NL}
	/{NEW
	ADD
	APND}
		Specifies the type of labels on the volumes to contain the specified file, designates the position of the file on the volume, and specifies the placement of the new file.
		The parameter consists of three subparameters, each separated by a slash. The three subparameters are positional, but not mandatory.
		When the FOT statement appears in an MFR/MFRE group, the LABL parameter is overriden by the MFR specification.
			K	LABL = 1/SL/NEW

	DEN = {1600 800 712 556 356 200}	Specifies the recording density in bits-per-inch-per-track for the tape volumes designated by the EFID parameter. 1600 and 800 bpi pertains to half-inch 9-track tape. 800, 556 and 200 bpi pertain to half-inch, 7-track tape. 712 and 356 bpi pertain to one-inch tape. When the FOT statement occurs in an MFR/MFRE group the parameter is overriden by the MFR statement specification.	K	DEN = 1600 1600 bpi for SRC = TP and TRKS = 9 or defaulted. 800 bpi for SRC = TP and TRKS = 7. If SRC is not specified or is set to DSEC, the DEN parameter is meaningless.

A-32 FOT STATEMENT SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
	Track parameter TRKS = $\left\{\begin{matrix}9\\7\end{matrix}\right\}$	Specifies whether the half-inch tape is 9-track or 7-track. When the FOT statement occurs in an MFR/MFRE group, this parameter is overriden by the MFR statement specification.	K	If omitted, 9-track is assumed.
	Recording control parameter RCTL = $\left\{\begin{matrix}ET\\E\\O\\T\\C\end{matrix}\right\}$	Specifies recording control for parity, translation and data conversion for 7 track tape. O - odd parity, no translation, no conversion. E - even parity, no translation, no conversion. T - odd parity, translation, no conversion. C - odd parity, no translation, conversion. ET - even parity, translation, no conversion. When the FOT statement occurs in an MFR/MFRE group, this parameter is overriden by the MFR statement specification.	K	If omitted and TRKS = 7, ET is assumed. If TRKS = 9 or defaulted, this parameter does not apply.
	SVC = scratch volume count	Specifies a maximum number of system scratch tape volumes that are to contain the file or its parts. The parameter value is an unsigned constant of one to six digits. If the FOT statement occurs in an MFR/MFRE group, the SVC parameter is overriden by the MFR specification.	K	SVC = 12
Disc space management	OPT = $\left\{\begin{matrix}N\\R\end{matrix}\right\}$	Specifies the disposition of the disc space upon successful completion of the file transfer. N specifies that the disc file is not to be released at this time. R specifies that the disc file is to be released and the disc space returned to the job's disc allocation.	K	OPT = N
File retention	RETP = retention	Specifies the number of days the tape file will be retained. The parameter value is an unsigned constant denoting days. When the FOT statement occurs in an MFR/MFRE group, the RETP parameter is overriden by the MFR specifications.	K	Installation-defined value.
Destination specifications	LOC = location id	Specifies the site at which the tape file is to be output. The parameter value is a string of one to eight alphanumeric characters designating the site. When the FOT statement occurs in an MFR/MFRE group, the LOC parameter is overriden by the MFR specifications.	K	Central site identifier.

A-32 FOT STATEMENT SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
	DEST = destination code	Specifies the code that is transmitted to the external tape label for use in routing the tape volume after the transfer has been completed. The parameter value is a character string of two alphabetic characters defined by the central site. When the FOT statement occurs in an MFR/MFRE group, the DEST parameter is overriden by the MFR specifications.	K	Central site code.

P = POSITIONAL K = KEYWORD

A-33 GOTO STATEMENT

The general form of the GOTO statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	GOTO	ƀ	label

A-34 GOTO STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Transfer label	Label	Specifies JSL statement to which execution is to be transferred unconditionally.	P	None. It must be specified.

P = POSITIONAL K = KEYWORD

A-35 IF STATEMENT

The general form of the IF statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	ƀ	IF	ƀ	condition, label

A-36 IF STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Transfer condition	Condition	Specifies condition under which transfer of execution is to be made. a. op. b where: a and b may be a constant, JSL variable or job specification indicator and . op. is one of the six relational operators.	P	None. It must be specified.
Transfer label	Label	Specifies label of JSL statement to transfer to when condition parameter is . TRUE. (If condition parameter is . FALSE., transfer is to next JSL statement in line)	P	None. It must be specified.

P = POSITIONAL K = KEYWORD

A-37 JOB STATEMENT (JOB)

The general form of the JOB statement is:

		OPERATION		OPERANDS
/	b̸	JOB		job name, acctnum, user code
				[, CAT = category]
				[, LOC = location identification]
				[, OPT = (option codes)]

A-38 JOB STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Job name	Job name	Identifies the name of the job.	P	None. Always mandatory.
Installation accounting information	Acctnum	Supplies the account to which charges for the job are to be billed.	P	None. Always mandatory.
Catalog access security information	User code	Specifies a valid user and legal uses of the catalog system.	P	None. Always mandatory.
Execution information	CAT = category	Specifies the priority of service and execution characteristics of the job within framework defined by installation.	K	Default category defined by the installation is assumed if parameter is omitted.
Output location	LOC = location identification	Specifies the installation at which the output for the job will be produced. Location identifiers are assigned by the central installation.	K	If parameter is omitted, the job output is produced at installation where job is submitted.
Restart option Job processing option	OPT = (option codes)	R - specifies whether to restart an active step after a system failure. C - specifies whether job processing is to continue normally following abnormal termination of any CP step in the job.	K	If the parameter is omitted, the step cannot be restarted and only FOSYS statements are processed following a CP step terminated abnormally unless option C is specified on XQT control card for that step.

P = POSITIONAL K = KEYWORD

A-39 JOBX STATEMENT

The general form of the JOBX statement is:

		OPERATION		OPERANDS
/	ƀ	JOBX	ƀ	jobname, acctnum, user code [, CAT=category] [, LOC=location identification] [, OPT=option]

A-40 JOBX STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Job identification	Jobname	Specifies the name of the job. Each job must have a name.	P	None. It is required.
Billing information	Acctnum	Specifies account number to which job charges are billed.	P	None. It is required.
Catalog use	User code	Identifies and qualifies a user to the catalog system.	P	None. It is required.
Priority	CAT = category	Specifies priority of service and execution characteristics of job.	K	If omitted, an installation defined priority is assigned.
Output location	LOC = loc id	Specifies installation at which output will be produced for the job.	K	If omitted, the location is where the job was submitted.
Processing control options	OPT = (options)	Specifies the options that are available to the user for processing control for the job. C - specifies job processing is to continue normally following abnormal termination of step. R - specifies an active step is to be restarted after a system failure.	K	If C is omitted, only FOSYS statements processed after CP step that terminate abnormally. If R is omitted - an active step will not be restarted following a system failure.

P = POSITIONAL K = KEYWORD

A-41 LIMIT STATEMENT

The general form of the LIMIT statement is:

```
  | LABEL    |   | OPERATION |   | OPERANDS
/ | [Symbol] | b̸ | LIMIT     | b̸ | [ACCT=account number][,USCO=user
  |          |   |           |   |                              code]
  |          |   |           |   |     (SECT)
  |          |   |           |   | [,  )BAND(= disc space reservation]
  |          |   |           |   |     )WORD(
  |          |   |           |   |     (BYTE)
  |          |   |           |   |
  |          |   |           |   | [,  {SEC} = time]
  |          |   |           |   |     {MIN}
```

A-42 LIMIT STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Account number	ACCT = acct no.	Provides accounting information for the block. It may be different from that specified on the JOB statement or previous LIMIT statement.	K	If omitted, the last account number specified on a JOB statement or previous LIMIT statement is billed.
User code	USCO = user code	Identifies a specific user to the system for the job block. It may be different from that specified on the JOB statement or previous LIMIT statement.	K	If omitted, the last user code specified on a previous LIMIT statement or the JOB statement is used.
Managing secondary disc space	$\left\{\begin{matrix}\text{BAND}\\\text{SECT}\\\text{WORD}\\\text{BYTE}\end{matrix}\right\}$ = disc space reservation	Manages disc space for job block. Specifies maximum space allocated. The parameter value is an unsigned constant less than 999999_{10}	K	An installation defined value if no disc reservation has been previously specified.
Central processor time	$\left\{\begin{matrix}\text{SEC}\\\text{MIN}\end{matrix}\right\}$ = time	Specifies maximum cumulative amount of CP execution time allowed for the job block. It is an unsigned constant of 1 to 4 digits not to exceed 30 hours.	K	Installation defined amount of time for the block.

P = POSITIONAL K = KEYWORD

A-43 MFR STATEMENT

The general form of the MFR statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	ƀ	MFR	ƀ	id [, EFID=tape id 1[/tape id2][/. . . ; [, tape idn]]] [, LABL=[$\left\{\begin{matrix}\underline{1}\\\text{position}\end{matrix}\right\}$] [/$\left\{\begin{matrix}\underline{\text{SL}}\\\text{NL}\end{matrix}\right\}$]; [/$\left\{\begin{matrix}\underline{\text{NEW}}\\\text{ADD}\\\text{APND}\end{matrix}\right\}$]]] [, DEN=$\left\{\begin{matrix}\underline{1600}\\800\\556\\200\end{matrix}\right\}$] [, TRKS=$\left\{\begin{matrix}\underline{9}\\7\end{matrix}\right\}$] [, RCTL=$\left\{\begin{matrix}\underline{\text{ET}}\\\text{E}\\\text{O}\\\text{T}\\\text{C}\end{matrix}\right\}$] [, SVC=scratch volume count] [, RETP=retention] [, LOC=location id] [, DEST=destination code]

A-44 MFR STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Multi-file identification	id	Specifies identity for files included in the multi-file request.	P	None. It must be specified.
Tape volume description	EFID = tape id1 /tape id2 /...	Specifies the identification numbers of the volumes that will contain the requested files.	K	If omitted, the OS writes the specified files to a scratch volume.
	label parameter LABL = $\left\{\begin{matrix}1\\ \text{position}\end{matrix}\right\}$ / $\left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\}$ / $\left\{\begin{matrix}\text{NEW}\\ \text{ADD}\\ \text{APND}\end{matrix}\right\}$	Describes characteristics of tape volume that will contain the files. The subparameters are positional but not mandatory. The third subparameter applies only to output.	K	If omitted, LABL = 1/SL/NEW
	Tape density parameter DEN = $\left\{\begin{matrix}\underline{1600}\\ 800\\ 556\\ 200\end{matrix}\right\}$	Specifies the recording density in bits per in. per track for tape volumes on EFID parameter or scratch volumes indicated or defaulted.	K	If omitted, 1600 for TRKS = 9 or default. 800 if TRKS = 7.
	Track parameter TRKS = $\left\{\begin{matrix}\underline{9}\\ 7\end{matrix}\right\}$	Specifies whether the tape volumes are 9 track or 7 track.	K	If omitted, 9 track is assumed.
	Recording control RCTL = $\left\{\begin{matrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{matrix}\right\}$	Specifies recording control when 7 track tape is specified (for parity, translation and data conversion) O - odd parity, no translation, no conversion. E - even parity, no translation, no conversion. T - odd parity, translation, no conversion. C - odd parity, no translation, conversion. ET - even parity, translation, no conversion.	K	If omitted and TRKS = 7, then ET is assumed. If TRKS = 9 or defaulted, this parameter does not apply.
	Scratch volumes SVC = scratch volume count	Specifies a maximum number of scratch tape volumes that will contain the requested files.	K	If omitted, OS assumes 12 scratch volumes.
	Retention period RETP = retention	Specifies the number of days the file will be retained.	K	If omitted, an installation defined value will apply.
	Location parameter LOC = loc.id	Specifies where the tapes are output.	K	If omitted, central site is assumed.
	Destination code DEST = dest. code	Specifies where tapes are routed after output.	K	If omitted, the central site is assumed.

A-45 MFRE STATEMENT

The general form of the MFRE statement is:

		OPERATION		OPERANDS
/	b̸	MFRE		

A-46 PATH DEFINITION STATEMENT (PD)

The general form of the PD statement is:

		OPERATION		OPERANDS
/	b̸	PD	b̸	synonym, pathname

A-47 PD STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Synonym definition	Synonym	Specifies the symbolic name by which the pathname is known to the job.	P	None. Always mandatory.
Catalog name	Pathname	Specifies an existing pathname to be given the synonymous name. Only the first edgename may be previously-defined synonym.	P	None. Always mandatory.

P = POSITIONAL K = KEYWORD

A-48 PJSL STATEMENT

The general form of the PJSL statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	PJSL	b̸	access name

A-49 PJSL STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Job specification source file	Access name	Specifies the job specification source file to be executed. Processing is to begin on the deferred job(s) in the file.		None. It must be specified.

A-50 PRINT STATEMENT (PRINT)

The general form of the PRINT statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	PRINT	b̸	[JSLV=variable 1[/variable 2[/... [/variable n]]]] [, MESG=message] [, OPT=(options)]

A-51 PRINT STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Variable names to print	JSLV = variable1 /variable2/... /variablen	Specifies variables whose values are to be printed in decimal.	K	If omitted, none will be printed unless OPT = (A) or OPT = (A, H) is used.
Message to print	MESG = message	Specifies user-supplied message for output in job activity file.		If omitted, no message is output.
Options	OPT = (Options)	Specifies whether all variables or only specified variables are printed and what notation is used. A - all variables printed in decimal. H - all JSLV parameter values in hexadecimal. A, H - all variables in hexadecimal.		If omitted, only JSLV parameter values are printed in decimal.

P = POSITIONAL K = KEYWORD

A-52 RELEASE STATEMENT (REL)

The general form of the REL statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	REL	b̸	access name [, . . . [, access name]]

A-53 REL STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File identification	Access name	Identifies to the job the file whose disc space is to be released. The job's access to the identified file is severed. Only non-resident files can be released. There is no limit on number of files specified. Note that the release will occur at the same time for all specified files.	P	None. One access name is always mandatory.

P = POSITIONAL K = KEYWORD

A-54 RENAME FILE STATEMENT (RENAME)

The general form of the RENAME statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	RENAME	b̸	access name$_1$, access name$_2$

A-55 RENAME STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Old file name	Access name$_1$	Specifies the existing name of the file to be renamed. The file is accessible to the job by this name.	P	None. Always mandatory.
New file name	Access name$_2$	Specifies the name by which the file will be accessed. This name replaces the existing name. Access by the old name is severed and is replaced by the new name.	P	None. Always mandatory.

P = POSITIONAL K = KEYWORD

A-56 REPLACE VERSION STATEMENT (RPLV)

The general form of the RPLV statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	RPLV	b̸	pathname, [, ACNM = access name] [, VERS = version number] [, DTYP= $\left\{\begin{matrix}\underline{\text{TAPE}}\\ \text{DSEC}\\ \text{HPT}\\ \text{PAD}\\ \text{OPID}\end{matrix}\right\}$ / $\left\{\begin{matrix}\underline{\text{NOCNTG}}\\ \text{CNTG}\end{matrix}\right\}$ / operator ID /$\left\{\begin{matrix}\underline{\text{CEOF}}\\ \text{CALL}\end{matrix}\right\}$] [, RETP = retention] [, SRC= $\left\{\begin{matrix}\underline{\text{DSEC}}\\ \text{TP}\\ \text{FT}\end{matrix}\right\}$]
					[, FORG= $\left\{\begin{matrix}\underline{\text{PS}}\\ \text{DS}\end{matrix}\right\}$] [, LABL= [$\left\{\begin{matrix}\underline{1}\\ \text{position}\end{matrix}\right\}$][/ $\left\{\begin{matrix}\underline{\text{SL}}\\ \text{NL}\end{matrix}\right\}$]] [, EFID = efid1/efid2/. . . /efidn] [, DEN= $\left\{\begin{matrix}\underline{1600}\\ 800\\ 712\\ 556\\ 356\\ 200\end{matrix}\right\}$] [, TRKS= $\left\{\begin{matrix}\underline{9}\\ 7\end{matrix}\right\}$] [, RCTL= $\left\{\begin{matrix}\underline{\text{ET}}\\ \text{E}\\ \text{O}\\ \text{T}\\ \text{C}\end{matrix}\right\}$] [, DEN= $\left\{\begin{matrix}\text{BAND}\\ \text{SECT}\\ \text{WORD}\\ \text{BYTE}\end{matrix}\right\}$ = number]

A-57 RPLV STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Catalog name	Pathname	Identifies the catalog containing the version to be replaced. Specifies the identity and location of the node being accessed.	P	None. Always mandatory.
File name	ACNM = access name	Identifies the file to the job. Access name exists only for the duration of the job. Valid only for normal catalog processing.	K	Mandatory when SRC = DSEC or when SRC defaults to DSEC. Not valid for SRC = TP or FT.
Version specification	VERS = version number	Specifies which version of the file at the specified node is to be replaced. Valid only for normal catalog processing.	K	The flagged version is replaced if version number is not specified.
Destination device	DTYP = { TAPE DSEC HPT PAD OPID }	Specifies whether the file being cataloged will be output to 1/2" magnetic tape (1600 BPI), or remain on secondary disc, allocated on head-per-track disc, position-arm-disc or specific channel or module given by the third subparameter.	K	TAPE. The Operating System will place the cataloged file on magnetic tape. For SRC = TP or FT, the parameter is invalid.
	/ { NOCNTG CNTG } /	Specifies whether the file may be allocated with or without regard to physical contiguity on disc.		No contiguity required.
	/ op id /	Specifies specific device related channel or module. Used in conjunction with OPID, first subparameter.		None.
	/ CEOF CALL	Specifies whether initial allocation of cataloged copy of file is set to the original files current end of file or to the original file's current allocation.		Initial allocation set to original file's current end of file.
Retention period	RETP = retention	Specifies the length of time a cataloged file is to be retained. The number of days is converted to an expiration date that is written in the Tape Volume Catalog and the label for standard-labeled tapes. The parameter is not meaningful for files on secondary disc.	K	An installation-defined value. Not valid for DTYP = DSEC.

Source location of file	SRC = { DSEC
	TP
	FT }
		Specifies the source device on which to-be-cataloged file resides. DSEC specifies secondary disc; TP half-inch magnetic tape volume; and FT one-inch tape volume. SRC also indicates whether normal cataloging functions or special processing will occur. SRC = DSEC indicates normal processing. SRC = TP or FT means special catalog processing.	K	DSEC. Normal catalog processing occurs.

The following parameters apply only to special catalog processing in which the file to be cataloged exists on tape and is not buffered to secondary disc for processing.

A-57 RPLV STATEMENT SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File organization	FORG = $\left\{\begin{matrix}\underline{\text{PS}}\\\text{DS}\end{matrix}\right\}$	Specifies the organization of the cataloged file. Organization can be one of two types: PS, physical sequential; DS, direct secondary. The source parameter must be set to TP or FT.	K	PS. If SRC is not specified or is set to DSEC, the parameter is meaningless.
Volume specification	External file identification EFID ⇒ tape id1 /tape id2 ... /tape idn	Specifies the identification numbers of the source volumes containing the file to be cataloged. All identifiers must be declared and must appear in the sequence that the tape volumes are to be read.	K	Mandatory if SRC = TP or FT. If SRC is not specified or is set to DSEC, the parameter is meaningless.
	Label LABL = $\left\{\begin{matrix}\underline{1}\\\text{posi-}\\\text{tion}\end{matrix}\right\}$/ $\left\{\begin{matrix}\underline{\text{SL}}\\\text{NL}\end{matrix}\right\}$	Describes characteristics of the tape volume that contains the files specified. The parameter is comprised of a set of two subparameters separated by slashes, each of which is positional and optional:	K	If omitted, the default values are 1 and SL. If SRC is not specified or is set to DSEC, the parameter is not meaningful.
		Position value is $1 \le n \le 255$	P	1
		Label type value is either SL or NL	P	SL
	Tape density parameter DEN = $\left\{\begin{matrix}\underline{1600}\\800\\712\\556\\356\\200\end{matrix}\right\}$	Specifies the recording density in bits-per-inch-per-track for the tape volumes designated by the EFID parameter. The parameter is valid only for special catalog processing. 1600 and 800 bpi pertain to 1/2-inch, 9-track tape. 800, 556 and 200 bpi pertain to 1/2-inch, 7-track tape. 712 and 356 bpi pertain to one-inch tape.	K	1600 bpi for SRC = TP and TRKS = 9 or defaulted. 800 bpi for SRC = TP and TRKS = 7. If SRC is not specified or is set to DSEC the DEN parameter is meaningless.
	Track parameter TRKS = $\left\{\begin{matrix}\underline{9}\\7\end{matrix}\right\}$	Specifies whether the 1/2" tape is 9-track or 7-track.	K	If omitted, 9-track is assumed.
	Recording control parameter RCTL = $\left\{\begin{matrix}\underline{\text{ET}}\\\text{E}\\\text{O}\\\text{T}\\\text{C}\end{matrix}\right\}$	Specifies recording control for parity, translation and data conversion for 7-track tape. O - odd parity, no translation; no conversion. E - even parity, no translation, no conversion. T - odd parity, translation, no conversion. C - odd parity, no translation, conversion. ET - even parity, translation, no conversion.	K	If omitted and TRKS = 7, ET is assumed. If TRKS = 9 or defaulted, this parameter does not apply.
Managing secondary disc space	$\left\{\begin{matrix}\text{BAND}\\\text{SECT}\\\text{WORD}\\\text{BYTE}\end{matrix}\right\}$ = number	Manages disc space for the to-be-cataloged file during subsequent assignment to a job. Specifies maximum space allocated. The parameter value is an unsigned constant less than 999999_{10}.	K	An installation-defined value. If SRC is not specified or is set to DSEC, the disc parameter is invalid.

P = POSITIONAL K = KEYWORD

A-58 SET STATEMENT (SET)

The general form of the SET statement is:

	LABEL		OPERATION		OPERANDS
/	[Symbol]	b̸	SET	b̸	variable = [±]n [±m]

A-59 SET STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
JSL variable assignment	variable = [±]n [±m]	Specifies a JSL variable and the value to be assigned that variable. n - a signed or unsigned constant. a signed or unsigned JSL variable. a signed or unsigned job specification indicator (TERM or STATUS). m - a constant or null.	P	If a JSL variable is not assigned a value by a SET statement or step, it has a zero value.

P = POSITIONAL K = KEYWORD

A-60 SETUP STATEMENT (SETUP)

The general form of the SETUP statement is:

	LABEL		OPERATION		OPERANDS
/	symbol	b̸	SETUP	b̸	setup id [, CHAN=operation id] [, EFID=tape id1][/tape id2]/. . . ; [/tape idn]]]] [, SVC=$\frac{1}{\text{number}}$] [, PROTECT=$\frac{\text{YES}}{\text{NO}}$]

A-61 SETUP STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Storage channel association	Setup id	Associates a logical name with the storage channel that will be used for secondary storage transfer.	P	None. It must be specified.
	Channel CHAN = operator id	Describes which channel is to be used with the secondary storage transfer. Must be a valid ASC facility "OPID" associated with a secondary storage channel.	K	If not specified the system will select a PAD channel for the user.
Tape Volume description	EFID = tape id1 [/tape id2 [...[/tape idn]]]	Specifies the identification numbers of the tape volumes that will be used in the secondary storage transfers.	K	If omitted only scratch tape volumes will be associated with this storage channel.
	SVC = scratch volume count	Specifies a maximum number of system scratch volumes that are to contain files or members.	K	If omitted only one scratch volume is assumed.
Protection parameter	$\text{PROTECT} = \begin{Bmatrix}\text{YES}\\ \text{NO}\end{Bmatrix}$	Specifies whether or not write protection is to be invoked for the VSN's specified.	K	YES. Write protection is invoked.

P = POSITIONAL K = KEYWORD

A-62 JOB INPUT FILE STATEMENT (START)

The general form of the START statement is:

LABEL		OPERATION		OPERANDS
/	b̸	START	b̸	[ACNM = access name] [, BKSZ = number] $\left[,\begin{Bmatrix}\text{SECT}\\ \text{BAND}\\ \text{WORD}\\ \text{BYTE}\end{Bmatrix} = \text{initial/maximum/increment}\right]$ $\left[,\text{FORG} = \begin{Bmatrix}\text{PS}\\ \text{DS}\end{Bmatrix}\right]$

A-63 START STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File name	ACNM = access name	Identifies the file to the job and specifies the file as job local.	K	If ACNM is omitted, an unnamed file embedded in the job input stream immediately following a JSL execution step statement is assumed as input for that step.
Disc reservation	{SECT BAND WORD BYTE} = initial/ maximum/ increment	Manages disc space for the embedded file for the duration of the job. There are three subparameters: Initial Maximum Increment	K P P P	Value specified on LIMIT statement or, by LIMIT's default, installation-defined value. The entire set of subparameters must be coded or the set defaulted.
Logical characteristics	Block size BKSZ = number	Specifies the maximum number of bytes in a block within the specified file. Must be a multiple of record length (80 bytes). BKSZ is interactive with FORG parameter.	K	4000 bytes
	File organization FORG = PS DS	FORG = PS, BKSZ must be specified if 4000 byte default is not acceptable. FORG = DS, BKSZ has no meaning. Do not code. Specifies the organization of the file on disc.	K	Default is PS; sequential organization.

P = POSITIONAL K = KEYWORD

A-64 INPUT FILE STOP STATEMENT (STOP)

The general form of the STOP statement is:

		OPERATION		OPERANDS
/	b̸	STOP	b̸	None

A-65 EXECUTE LOAD MODULE STATEMENT (XQT)

The general form of the XQT statement is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	XQT	b̸	access name [, LTP = (load time parameters)] [, RTP = (run time parameters)] [, OPT = (options)][, CPOPT = (CP options)] [, CPTIME = central processor execution time] [, ADDMEM = size][, CHAR = character strings]

A-66 XQT STATEMENT SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Load module identification	Access name	Identifies to the job the direct secondary file, or member of partitioned direct secondary user file JOBLIB, or member of the system partitioned direct secondary file containing the load module to be executed.	P	None. Always mandatory.
Load time specification	LTP = (load time parameters)	Specifies (1) number of Fortran I/O files required, (2) number of Fortran I/O errors allowed before termination, (3) memory allocation and passes any initial values to the load module. Values for number (3) are provided through the Fortran PROGRAM statement.	K	No load time parameters are passed to the loader 99 Fortran I/O files are allocated and 10 Fortran I/O errors are allowed before termination.
Run time specifications	RTP = (run time parameters)	Specifies the values that the load module is to obtain when a CP S$RUNV service call is issued.	K	No run time parameters are passed to the load module.
	CPOPT = (cp options)	Specifies the values to be accessed by the load module with the CP S$OPTN service call.	K	Cp options word is zero filled and passed to load module.
	CHAR = character strings	Specifies arbitrary character strings to be passed to the load module. The module obtains these strings through the S$CHRV service call.	K	No character strings are passed.
Operating System specifications	OPT = options	Specifies 1) whether OS should take dump at termination of load module, 2) how to preload the memory module, 3) if a page load message is desired, and 4) whether job is to continue upon abnormal step termination.	K	No post mortem dump is taken. Memory retains coincidental data until overwritten. Upon abnormal termination of step, only FOSYS statements following the CP step are executed. No page load message is given.
	CPTIME = t	Specifies the amount of CP execution time allowed for the step. $0 \le t \le 2^{31}-1$, expressed in hundredths of seconds, not to exceed 30 hours.	K	An installation-defined value.
Memory specifications	ADDMEM = size	Specifies the number of additional words of central memory to be reserved for the step. This size should include any required space for I/O buffers; however, it does not pertain to the size of the load module and variable-dimensioned arrays. This space is accessed by the step through a GETMAIN procedure. NOTE: If a size is specified, the user cannot assume that the default value for buffers is automatically added to the amount being specified. The amount specified must make allowance for buffers - the default value is exclusive and is never included in the specification.	K	An installation-defined value for additional memory.

APPENDIX A1

SYSTEM MACRO(S) SUMMARY

A1-1 EXECUTING THE ASC ASSEMBLER WITH THE ASM MACRO (ASM)

The general form of the ASM macro is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	ASM	b̸	[IN=access name] [, LIST=access name] [, OBJ=access name] [, USPL=access name] [, ASMOPT= cp option] [, AADDMEM=address] [, ASMTIME= cp time] [, OBJFILE=$\left\{\begin{matrix}\text{NEW}\\\text{MOD}\end{matrix}\right\}$] [, OBJSIZE= band allocation] [, PRTSIZE= band allocation] [, MEMOBJ= member name] [, UOBJLIB= access name]

A1-2 ASM MACRO SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File specification	IN = access name	Specifies the source input file to the Assembler. The access name SYS. AIN must not exist for another file in the same job. The source input file must be previously assigned to the job prior to execution of the Assembler. The access name must be a valid ASC symbol.	K	An unnamed file embedded in the input stream immediately following the ASM macro call.
	LIST = access name	Specifies the print output file for Assembler output. The access name must be a valid ASC symbol. Automatic stacking of output is provided. The print file is opened with POS=MOD. The access name SYS. PRT must not exist for another file in the same job.	K	None. If LIST is omitted, the content of SYS. PRT are output on the printer.
	OBJ = access name	Specifies the output file that contains the object module output by the Assembler. This file must be capable of being renamed SYS. OMOD.	K	None. If OBJ is omitted, the object module appears on SYS. OMOD.
	USPL = access name	Specifies the user source program library that is to be searched for definitions of procedures referred in the program being assembled.	K	None. If USPL is omitted, no user libraries are searched.

	OBJFILE = {NEW
	MOD}
		Specifies the initial position of the object module output file when files are to be appended to it. NEW specifies that the output file is positioned to its beginning; MOD specifies that appendages begin at the end of the current SYS. OMOD.	K	New

	PRTSIZE = band allocation	Specifies the band allocation for the print file output by the Assembler. The parameter must be coded in the form: initial/maximum/increment where the values are positive constants specifying the number of bands in each case.	K	1/8/1
	OBJSIZE = band allocation	Specifies the band allocation for the object module file output by the Assembler. The parameter must be coded in the form: initial/maximum/increment where the values are positive constants specifying the number of bands in each case.	K	1/8/1
Assembler specification	ASMOPT = cp option	Specifies the Assembler options available to the user. The parameter value is a string of alphabetic characters separated by commas with the entire string enclosed in parentheses. The characters have the following meanings: C assemble ASC CP program D produce no object module O list object module on assembly listing P assemble ASC PP program T produce symbol table as part of object module X do not produce a cross-reference as part of assembly listing	K	A source listing, cross-reference listing and object module are produced; no symbol table or object module listing is produced.

A1-2 ASM MACRO SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
	ASMTIME = cp time	Specifies the cp time parameter for the execution of the compiler. The value is a positive constant coded in thousandths of seconds.	K	18000 (180 seconds)
	AADDMEM = additional memory size	Specifies the amount of additional memory required by the Assembler. This space is used as an input buffer area when reading the source input file. The remainder is used for a procedure and symbol table. The parameter value is a positive constant, a positive constant followed by K (K denotes thousands), or a JSL variable.	K	30K
PDS specification	MEMOBJ = member name	Specifies that the object module output by the Assembler is to be put into a Partitioned Direct Secondary file with the member name specified. The member name must be a valid ASC symbol.	K	None. If omitted, the PDS update is bypassed.
	UOBJLIB = access name	Specifies the access name of the PDS library into which the object module will be placed if MEMOBJ was explicitly coded. The access name must be a valid ASC symbol.	K	OBJLIB

A1-3 ASSEMBLE AND LINKAGE EDIT WITH THE ASML MACRO (ASML)

	LABEL		OPERATION		OPERANDS
/	[symbol]	ƀ	ASML	ƀ	[IN=access name] [, LIST= access name][, USPL=access name] [, ASMOPT=cp options][, ASMTIME= cp time][, AADDMEM=addmem] [, CONTROL=access name][, LOAD= access name][, LNKOPT=cp option] [, LNKTIME=cp time] [, LADDMEM=addmem]

A1-4 ASML MACRO SUMMARY

The following parameters which are used in the assembly step are the same as in the ASM macro. For descriptions of these parameters, see the ASM macro statement. The parameters are the following:

IN
LIST
USPL
ASMOPT
ASMTIME
AADDMEM

The following parameters which are used in the link edit step are the same as in the LNK macro. For descriptions of these parameters, see the LNK macro statement. These parameters are the following:

CONTROL
LOAD
LNKOPT
LNKTIME
LADDMEM

A1-5 ASSEMBLE, LINKAGE EDIT, AND EXECUTE USING THE ASMLX MACRO (ASMLX)

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	ASMLX	b̸	[IN=access name] [, LIST=access name] [, USPL=access name] [, ASMOPT=cp options] [, ASMTIME=cp time] [, AADDMEM=addmem] [, CONTROL=access name] [, LOAD=access name] [, LNKOPT=cp options] [, LNKTIME=cp time] [, LADDMEM=addmem] [, any tagged parameters which are legal on the XQT statement

A1-6 ASMLX MACRO SUMMARY

The following parameters which are used in the assembly step are the same as in the ASM macro. For descriptions of these parameters, see the ASM macro statement. These parameters are the following:

IN
LIST
USPL
ASMOPT
ASMTIME
AADDMEM

The following parameters which are used in the link edit step are the same as in the LNK macro. For descriptions of these parameters, see the LNK macro statement. These parameters are the following:

CONTROL
LIST
LOAD
LNKOPT
LNKTIME
LADDMEM

The following parameters are used in the execution of the load module produced by the previous link edit. They are any tagged parameter which is legal on the XQT statement. For descriptions of these parameters, see the XQT statement. These parameters are the following:

LTP
RTP
OPT
CPOPT
CPTIME
ADDMEM
CHAR
MEMRY

A1-7 EXECUTING THE ASC FORTRAN COMPILER WITH THE FTN MACRO (FTN)

The general form of the FTN macro is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	ƀ	FTN	ƀ	[, IN=access name] [, LIST=access name] [, OBJ=access name] [, FTNOPT=cp options] [, FTNTIME=cp time] [, FADDMEM=addmem] [, SPACE=space] [, VSPACE= $\left\{\begin{matrix}\text{ANY}\\ \text{vspace}\end{matrix}\right\}$] [, OBJFILE=$\left\{\begin{matrix}\text{NEW}\\ \text{MOD}\end{matrix}\right\}$] [, OBJSIZE=band allocation] [, PRTSIZE=band allocation] [, MEMOBJ=number name] [, UOBJLIB=access name]

A1-8 FTN MACRO SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File specification	IN=access name	Specifies the source input file to the FORTRAN compiler. The access name must be a valid ASC symbol.	K	An unnamed file embedded in the input stream immediately following the FTN macro call.
	LIST = access name	Specifies the print output file for Compiler output. The access name must be a valid ASC symbol. Automatic stacking is provided for print output files. The print file is opened with POS=MOD.	K	None. If LIST is omitted, the contents of SYS. PRT are output on the printer.
	OBJ = access name	Specifies the output file to contain the object module output by the FORTRAN compiler. The access name must be a valid ASC symbol.	K	None. If OBJ is omitted, the object module appears on SYS. OMOD
	OBJFILE = $\begin{Bmatrix}\text{NEW}\\ \text{MOD}\end{Bmatrix}$	Specifies the initial position of the object module output file. NEW specifies that the initial position of the file is at its beginning. MOD specifies the initial position at the end of the file.	K	NEW
	OBJSIZE = band allocation	Specifies the band allocation for the object module file output by the compiler. The parameter must be coded in the form: initial/maximum/increment where the values are positive constants specifying the number of bands.	K	1/8/1
	PRTSIZE = band allocation	Specifies the band allocation for the print file output by the compiler. The parameter must be coded in the form: initial/maximum/increment where the values are positive constants specifying the number of bands in each case.	K	1/8/1
Fortran specification	FTNOPT = CP options	Specifies the Fortran compiler options available to the user. The parameters value is a string of alphabetic characters separated by commas with the entire string enclosed in parentheses. The characters have the following meanings: A syntax check only; do not produce object code B terminate compilation for irrecoverable errors during source scan C do not interrupt program execution for arithmetic exceptions D produce subprogram references for debugging E include only statements in error or containing warning messages in source listing F suppress warning messages H do not generate implicit calls to FIO I compile at level 1 J compile at level 2	K	(K, M, O)

A1-8 FTN MACRO SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
		K compile at level 3 L compile at level 4 M produce a program map N object module is not needed O produce an object listing P instructions are for execution only Q instructions are provided execute and read protection R instructions are provided execute, read and write protection S override all protection, the entire module may be executed, read and written. T suppress source listing		
	FTNTIME = CP time	Specifies the CP time for the execution of the compiler. The value is a positive constant coded in thousandths of seconds.	K	30000 (300 seconds)
	FADDMEM = addmem	Specifies the amount of additional memory required by the ASC Fortran compiler. The parameter value is a positive constant, or a constant followed by K (K denotes thousands), or a JSL variable.	K	4K
	SPACE = space	Specifies the load time parameter passed to the compiler determining the amount of internal list space in words to be allocated with the compiler load module. It is an unsigned integer ranging from 0 to 2^{24}-1.	K	Space = 4096

	VSPACE = {ANY
	vspace}
		Specifies the maximum number of words available for allocation as vector temporary space within the object modules produced by the Fortran compiler. The value is either an unsigned integer constant or the keyword ANY. ANY specifies that vector temporary space can be increased as required during compilation. VSPACE can be an unsigned integer ranging from 0 to 2^{24}-1.	K	ANY

Partitioned Direct Secondary specifications	MEMOBJ = member name	Specifies the access name of the Partitioned Direct Secondary library into which the object module will be placed if MEMOBJ was explicitly coded. The access name must be a valid ASC symbol.	K	None. If omitted, the Partitioned Direct Secondary update is bypassed.
	UOBJLIB = access name	Specifies the access name of the Partitioned Direct Secondary library into which the object module will be placed if MEMOBJ was explicitly coded. The access name must be a valid ASC symbol.	K	OBJLIB

A1-9 COMPILE AND LINKAGE EDITOR WITH THE FTNL MACRO (FTNL)

The general form of the FTNL macro is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	FTNL	b̸	[IN=access name] [, LIST=access name][, FTNOPT=cp options] [, FTNTIME=cp time] [, FADDMEM= addmem] [, SPACE=space] [, VSPACE=vspace] [, CONTROL= access name] [, LNKOPT=cp options] [, LNKTIME=cp time] [, LADDMEM=addmem] [, LOAD=access name]

A1-10 FTNL MACRO SUMMARY

The following parameters which are used in the compile step are the same as in the FTN macro. For descriptions of these parameters, see the FTN macro statement. The parameters are the following:

IN
LIST
FTNOPT
FTNTIME
FADDMEM
SPACE
VSPACE

The following parameters which are used in the link edit step are the same as in the LNK macro. For descriptions of these parameters, see the LNK macro statement. The parameters are the following:

CONTROL
LOAD
LNKOPT
LNKTIME
LADDMEM

A1-11 COMPILE, LINKAGE EDIT AND EXECUTE WITH THE FTNLX MACRO (FTNLX)

The general form of the FTNLX macro is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	FTNLX	b̸	[IN = access name][, LIST = access name]
					[, FTNOPT = cp options]
					[, FTNTIME = cp time]
					[, FADDMEM = addmem]
					[, SPACE = space]
					[, VSPACE = vspace]
					[, CONTROL = access name]
					[, LNKTIME = cp time]
					[, LNKOPT = cp options]
					[, LADDMEM = addmem]
					[, DATA = access name]
					[, GO = access name][, Any tagged parameter legal on XQT statement]

A1-12 FTNLX MACRO SUMMARY

The following parameters which are used in the compile step are the same as in the FTN macro. For description of these parameters, see the FTN macro statement. These parameters are the following:

IN
LIST
FTNOPT
FTNTIME
FADDMEM
SPACE
VSPACE

The following parameters which are used in the link edit step are the same as in the LNK macro. For description of these parameters, see the LNK macro statement. These parameters are the following:

CONTROL
LNKTIME
LNKOPT
LADDMEM

The following parameters are used in the execution of the FORTRAN load module produced. For descriptions of these parameters, see the FXQT macro statement. These parameters are the following:

GO
DATA
LTP
RTP
OPT
CPOPT
CPTIME
ADDMEM
CHAR
MEMRY

A1-13 EXECUTING A FORTRAN LOAD MODULE WITH THE FXQT MACRO (FXQT)

The general form of the FXQT macro is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	FXQT	b̸	[GO = access name][, DATA = access name][, LIST = access name] [, Any tagged parameters which are legal on the XQT statement]

A1-14 FXQT MACRO SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File specifications	GO = access name	Specifies the access name that refers to a direct secondary organized file which contains the load module to be executed. The access name must be a valid ASC symbol. The load module must use Fortran I/O for reading its job stream card file and for printing to output file. The input file must be referenced by use of a READ or READ(5) statement; the output file must be referenced by use of a PRINT or WRITE(6) statement.	K	GO = SYS.LMOD
	DATA = access name	Specifies the access name that refers to the input file of the Fortran load module. The access name must be a valid ASC symbol. The file so named is the file that is accessed when the Fortran load module issues a READ.	K	An unnamed file embedded in the job input stream immediately following the FXQT macro call.
	LIST = access name	Specifies the access name of the print output file to contain the load modules output. The access name must be a valid ASC symbol. Automatic stacking of output is provided. FT06F001 is specified with POS = MOD.	K	None. If no file is specified the contents of FT06F001 are output to the printer.

Any tagged parameter that is legal on the basic XQT statement may be used on the FXQT. Their use is governed by the same restrictions carried by the XQT statement. For descriptions of these parameters, see the XQT statement. These parameters are the following:

LTP	OPT	CPTIME	CHAR
RTP	CPOPT	ADDMEM	MEMRY

Note: No default values are supplied for these parameters, except ADDMEM which has a default of 2K.

A1-15 EXECUTING THE ASC LINKAGE EDITOR WITH THE LNK MACRO (LNK)

The general form of the LNK macro is:

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	LNK	b̸	[CONTROL = access name]
					[, LIST = access name]
					[, OBJ = access name]
					[, LOAD = access name]
					[, LNKOPT = cp options]
					[, LNKTIME = cp time]
					[, LADDMEM = addmem]
					[, MEMLOAD = member name]
					[, LOADLIB = access name]

A1-16 LNK MACRO SUMMARY

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
File specification	CONTROL = access name	Specifies the access name of the ASC Linkage Editor's input file. The access name must be a valid ASC symbol.	K	An unnamed file embedded in the job input stream immediately following the LNK macro call.
	LIST = access name	Specifies the access name of the file to be used as the Linkage Editor's print output file. The access name must be a valid ASC symbol. Automatic stacking is provided. The print file is opened with POS = MOD.	K	The ASC Linkage Editor uses the standard system print file (SYS. PRT) as its print output file.
	OBJ = access name	Specifies the access name of the input file from which the ASC Linkage Editor accesses object modules after it has read the input file. The access name must be a valid ASC symbol.	K	None. If OBJ is not specified, the Linkage Editor does not input any object modules in its input file (CONTROL).
Link	LOAD = access name	Specifies the file to be used by the Linkage Editor to write the output load module. This file must have direct secondary organization.	K	None. If LOAD is omitted, the Linkage Editor writes the load module to SYS. LMOD.
Link Editor specification	LNKOPT = CP option	Specifies the Linkage Editor options available to the user. The parameter value is a string of alphabetic characters separated by commas with the entire string enclosed in parentheses. The characters have the following meanings: M produce a memory allocation listing. N do not produce a memory allocation listing. X produce an external symbol cross-reference with memory allocation listing. Y do not produce an external symbol cross-reference A search user-specified libraries to resolve external references. B do not search user-specified libraries. S search system object module library to resolve external references. T do not search system object module library. E mark the produced load module as executable even if an error occurs during linkage editing. F mark the load module as not executable if an error occurs during linkage editing. K do not list the load module produced. L list the load module produced. Listing appears in SYS. PRT.	K	(M, X, A, S, F, L, Q)

A1-16 LNK MACRO SUMMARY (CONTINUED)

TYPE	PARAMETER	FUNCTION	P/K	DEFAULTS
Link Editor specification (continued)		P promote common blocks automatically to the highest level necessary to consolidate all occurrences. Q. do not automatically promote common blocks		
	LNKTIME = CP time	Specifies the cp time parameter for the execution of the ASC Linkage Editor. The value is a positive constant coded in thousandths of seconds.	K	30000
	LADDMEM = addmem	Specifies the amount of additional memory required by the ASC Linkage Editor. The parameter value is a positive constant, a constant followed by K (K denotes thousands), or a JSL variable.	K	8K
Partitioned Direct Secondary specification	MEMLOAD = member name	Specifies that the load module output by the Linkage Editor is to be put into a Partitioned Direct Secondary file with the member name specified. The member name must be a valid ASC symbol.	K	None. If omitted, the Partitioned Direct Secondary update is bypassed.
	LOADLIB = access name	Specifies the access name of the Partitioned Direct Secondary library in which the load module will be placed if MEMLOAD was explicitly coded. The access name must be a valid ASC symbol.	K	LOADLIB

A1-17 EXECUTING THE ASC LINKAGE EDITOR AND THE RESULTING LOAD MODULE WITH THE LNKX MACRO (LNKX)

	LABEL		OPERATION		OPERANDS
/	[symbol]	b̸	LNKX	b̸	[, CONTROL = access name]
					[, LIST = access name]
					[, OBJ = access name]
					[, LOAD = access name]
					[, LNKOPT = cp options]
					[, LNKTIME = cp time]
					[, LADDMEM = addmem][, Any tagged parameters which are legal on the XQT statement]

A1-18 LNKX MACRO SUMMARY

The following parameters, which are used in the link edit step, are the same as in the LNK macro. For descriptions of these parameters, see the LNK macro statement. These parameters are the following:

CONTROL
LIST
OBJ
LOAD
LNKOPT
LNKTIME
LADDMEM

The following parameters are used in the execution of the load module produced by the previous link edit. They are any tagged parameter which is legal on the XQT statement. For descriptions of these parameters, see the XQT statement. These parameters are the following:

LTP
RTP
OPT
CPOPT
CPTIME
ADDMEM
CHAR
MEMRY

APPENDIX A2
MACRO EXAMPLES

APPENDIX A2

MACRO EXAMPLES

A2-1 ASM MACRO EXAMPLES:

1. Card Input (Unnamed file all defaults operative)

```
/ JOB
/ ASM
  ≀
  Assembler source code
  ≀
/ FOSYS SYS.OMOD, TYPE=PUNCH
/ EOJ
```

2. Card Input (Named File)

```
/ JOB
/ START ACNM=INFILE
  ≀
  Assembler Source Code
  ≀
/ STOP
/ ASM IN=INFILE
/ FOSYS SYS.OMOD, TYPE=PUNCH
/ EOJ
```

3. Cataloged File Input

```
/ JOB
/ ASG INFILE, A/B/C/D, USE=SHR
/ ASM IN=INFILE
/ FOSYS SYS.OMOD, TYPE=PUNCH
/ EOJ
```

4. Card Input (Unnamed File - Specify Parameters)

```
/ JOB
/ ASG MODFILE, PATH/NAME
/ ASM LIST=PRINTOUT, OBJ=MODFILE, OBJFILE=MOD
  ʃ
  Assembler source code
  ʃ
/ FOSYS MODFILE, TYPE=PUNCH
/ FOSYS PRINTOUT
/ EOJ
```

5. Partitioned Direct Secondary File Update

```
/ JOB
/ ASG MYLIB, P/D/S/LIBRARY
/ START ACNM=INPUT
  ʃ
  Assembler source code
  ʃ
/ STOP
/ ASM IN=INPUT, MEMOBJ=MOD1, UOBJLIB=MYLIB
/ CATV P/D/S/LIBRARY, ACNM=MYLIB
/ EOJ
```

A2-2 ASML MACRO EXAMPLES:

1. Card Input

```
/ JOB
/ START ACNM=SRCFILE
  ≀
  Assembler source code
  ≀
/ STOP
/ ASML IN=SRCFILE, AADDMEM=15K, ASMOPT=(P), ;
           CONTROL=SYS. LEIN, LOAD=LMOD
/ START ACNM=SYS. LEIN
  ≀
  Linkage Editor control directives
/ STOP
/ FOSYS LMOD, TYPE=PUNCH
/ EOJ
```

2. Alternate Card Sequence for Example 1

```
/ JOB
/ START ACNM=SRCFILE
  ≀
  Assembler source code
  ≀
/ STOP
/ ASM IN=SRCFILE, AADDMEM=15K, ASMOPT=(P)
/ START ACNM=SYS. LEIN
  ≀
  Linkage Editor control directives
  ≀
/ STOP
/ LINK CONTROL=SYS. LEIN, LOAD=LMOD
/ FOSYS LMOD, TYPE=PUNCH
/ EOJ
```

A2-3 ASMLX MACRO EXAMPLES:

1. Card Input (Parameter Specified)

```
/ JOB
/ ASG MYSRC,A/B/C/D, USE=SHR
/ START ACNM=SYS.AIN
  ʃ
  Assembler source code
  ʃ
/ STOP
/ ASMLX IN=SYS.AIN, USPL=MYSRC, CPTIME=3000, ADDMEM=65K
/ FOSYS SYS.LMOD, TYPE=PUNCH
/ EOJ
```

A2-4 FTN MACRO EXAMPLES:

1. Card Input (All defaults)

```
/ JOB

/ FTN
  ≀
  FORTRAN Source program

/ FOSYS SYS.OMOD, TYPE=PUNCH

/ EOJ
```

2. Card Input (Parameters Specified)

```
/ JOB

/ START ACNM=FORTPROG
  ≀
  FORTRAN source program

/ STOP

/ FTN IN=FORTPROG, LIST=PRTFILE, FTNOP=(I, M), ;
       SPACE=4096, PRTSIZE=10/30/5

/ FOSYS PRTFILE, COPIES=5

/ FOSYS SYS.OMOD, TYPE=PUNCH

/ EOJ
```

3. Partitioned Direct Secondary File Update

```
/ JOB

/ ASG OBJLIB, MY/PATH/NAME

/ FTN MEMOBJ=SUB45, OBJ=OBJMOD, VSPACE=6000

/ CATV MY/PATH/NAME, ACNM=OBJLIB, DTYP=DSEC

/ CATV MY/OBJECTS, ACNM=OBJMOD

/ START ACNM=SYS.FIN
  ≀
  FORTRAN source program
  ≀
/ STOP

/ EOJ
```

A2-5 FTNL MACRO EXAMPLES:

1. Card Input (All defaults operative)

```
/ JOB
/ FTNL
/ START ACNM=SYS.FIN
  {
  FORTRAN source program
  {
/ STOP
/ FOSYS SYS.LMOD,  TYPE=PUNCH
/ EOJ
```

2. Alternate Card Sequence for Example 1

```
/ JOB
/ START ACNM=SYS.FIN
  {
  FORTRAN source program
  {
/ STOP
/ FTN
/ LNK
/ FOSYS SYS.LMOD, TYPE=PUNCH
/ EOJ
```

3. All Defaults

```
/ JOB
/ FTNL FTNOPT=(I, M, O), LADDMEM=8K, LOAD=LMOD
/ CATV MY/LOAD/MODS,ACNM=LMOD
/ EOJ
```

A2-6 FTNLX MACRO EXAMPLES:

1. Card Input

```
/ JOB
/ FTNLX
  ⌇
  FORTRAN source cards
  ⌇
/ START ACNM=FT05F001
  ⌇
  data file
  ⌇
/ STOP
/ EOJ
```

2. Cataloged File Input (Parameters Specified)

```
/ JOB
/ ASG SRCPROG,ALPHA/DAD
/ ASG LINKDIR,BETA/TIGER
/ ASG DATAFIL,GAMMA/ONE/TWO
/ FTNLX IN=SRCPROG, CONTROL=LINKDIR,DATA=DATAFIL
/ EOJ
```

3. Tape Input (Parameters Specified)

```
/ JOB
/ ASG LINKDIR,BETA/TIGER
/ FIT DTFILE,EFID=10210
/ FTNLX CONTROL=LINKDIR,DATA=DTFILE
  ⌇
  FORTRAN source program
  ⌇
/ EOJ
```

A2-7 FXQT MACRO EXAMPLES:

1. Card Input (Default input file)

```
/ JOB
/ ASG SYS.LMOD,A/B/C/D
/ FXQT
  ʃ
  data file
  ʃ
/ EOJ
```

2. Tape Input (Parameters Specified)

```
/ JOB
/ FIT LOADMOD,EFID=3300
/ FIT DATAFILE,EFID=11440
/ FXQT GO=LOADMOD,DATA=DATAFILE,LIST=OUTFILE
/ FOT OUTFILE
/ EOJ
```

A2-8 LNK MACRO EXAMPLES:

1. Card Input (All Defaults Operative)

```
/ JOB
/ LNK
  ≀
  Linkage Editor control file
  ≀
/ CATV A/B/C/D, ACNM=SYSLMOD
/ EOJ
```

2. Card Input (Parameters Specified)

```
/ JOB
/ START ACNM=OBJMOD
  ≀
  object module
  ≀
/ STOP
/ LNK LNKTIME=50000, LOAD=DECK6, OBJ=OBJMOD
  ≀
  Control Directives
  ≀
/ FOT DECK6
/ EOJ
```

3. Cataloged Input (Parameters Specified)

```
/ JOB
/ ASG MYLOADS, R/S/T
/ ASG SYS.OMOD, OBJECT/LIBRARY
/ START ACNM=LINKCTL
  ≀
  Linkage Editor Control Directives
  ≀
/ STOP
/ LNK CONTROL=LINKCTL, MEMLOAD=LODMOD1, ;
        LOADLIB=MYLOADS, LOAD=LODMOD1
/ CATV R/S/T, ACNM=MYLOADS
/ FOSYS LODMOD1, TYPE=PUNCH
/ EOJ
```

A2-9 LNKX MACRO EXAMPLES:

1. Card Input (All defaults operative)

```
/ JOB

/ LNKX
  ʃ
  Linkage Editor Control Directives
  ʃ
/ FOSYS SYS.LMOD, TYPE=PUNCH

/ FOSYS OUTPUT

/ EOJ
```

2. Card Input

```
/ JOB

/ ASG OBJMOD, M/N/O, USE=SHR

/ START ACNM=GGG
  ʃ
  Linkage Editor Control Directives
  ʃ
/ STOP

/ START ACNM=DATA
  ʃ
  data for load module program
  ʃ
/ STOP

/ LNKX OBJ=OBJMOD, CONTROL=GGG, LIST=LISTING, ;
         CPTIME=4000, ADDMEM=5K

/ FOSYS LISTING

/ FOSYS OUTPUT

/ EOJ
```

APPENDIX B

GLOSSARY OF JSL TERMS

APPENDIX B

GLOSSARY OF JOB SPECIFICATION LANGUAGE (JSL) TERMS

ACCESS CONTROL

The control invoked when a node is in the access control state. This control is further categorized as full, partial, or no access control.

ACCESS CONTROL LIST

A list constructed at a catalog node containing a list of potential users of the node and their node and file access attributes.

ACCESS CONTROL STATE

One of the three possible states of a node which are defined by the user's intent at that node.

ACCESS NAME

The access name is the local name for a file. It is the name by which a file is identified within a job..

ACCOUNT NUMBER

The account number is a string of alphanumeric characters up to sixteen characters in length.

ATTRIBUTE

One of six specifiable attributes for a user in the Access Control List of a node. These are the read, write, execute, own, reference, and son-add attributes.

BAND

256 sectors on disc.

BLOCK, FILE

A set of records treated as a unit.

BLOCK, JOB SPECIFICATION LANGUAGE

Within a job, the LIMIT statement will initiate a Job Specification Language block. This block will be terminated by a new LIMIT statement or an EOJ statement. A job with no LIMIT statements is considered to be one Job Specification Language block.

CATALOG

A tree structure whose nodes contain information pertinent to the storage, retrieval, and security of user files.

COUPLED STATEMENTS

Statements which have a relationship such that they must occur in pairs, e.g., JOB/EOJ, statements which are related in that their functions are the opposite of one another, e.g., CAT/DEL.

CP

Central Processor.

DATA EXCEPTION

A mechanical malfunction on input or output operations, e.g., parity errors.

DEFAULT

A value or meaning which is assumed in the absence of an explicit specification.

EDGE NAME

A name assigned to an edge of a tree-structured catalog.

FIELD, JOB SPECIFICATION LANGUAGE

In a Job Specification Language statement, a specified area used for a particular category of data.

FILE

An organized collection of data stored in a manner to facilitate the retrieval of each individual datum.

FILE, CATALOGED

A file cataloged by a previously executed job. It is made accessible through the ASG statement. A file whose name, location, and security characteristics are stored in the system catalog.

FILE, JOB LOCAL

A file which is created by a step within a job and which exists until that job is terminated. A job local file can be cataloged during the job.

FILE, NON-CATALOGED

A file which is not cataloged in the ASC system; i.e., it is not a job local file, or a cataloged file.

IJSL

Internal Job Specification Language.

INPUT STREAM, (JOB)

The sequence of specification statements and data submitted to the Operating System on an input device especially activated for this purpose.

JOB

A total computer process (application) which consists of one or more related process programs (steps). A job is specified by a sequence of Job Specification Language statements.

JSL

Job Specification Language.

KEYWORD

A word which is selected from a prescribed list of words defined internally to the system.

LABEL, JOB SPECIFICATION LANGUAGE

A symbol that identifies a specification statement. It must immediately follow the Job Specification Language statement identifier.

LABELED TAPE

The first data block of the first file on tape is preceded by a standard label record provided by the Operating System.

LOAD MODULE

A unit of output, created by the Link Editor, from one or more object modules. (A Load Module is used as input to the ASC loader and is not relocatable within the virtual address space.)

LOAD TIME BOUND MEMORY

The load time bound memory is the Central Memory requirement calculated at load time, this includes the load time request for array sizes.

LOAD TIME PARAMETERS

Load time parameters are integer values supplied by the assembly language user to the loader to effect memory allocation and initial values in the load module.

LOADER

A component of the Operating System which processes load modules created by the Link Editor.

NAME, JOB SPECIFICATION LANGUAGE

A one-to eight-character name, beginning with an alphabetic character that identifies a file, a program, or a specification statement.

NODE

The components of the vector $\overline{n}$ which together with a set of unilateral associations define a directed graph. The vector $\overline{n}$ is called a node vector.

NODE, ROOT

The node of a catalog that is accessible through the catalog directory.

OBJECT MODULE

A unit of relocatable output, comprising one or more control sections, which can be input to the Link Editor to produce a Load Module or another Object Module. (An Object Module is usually produced by a compiler, or assembler.)

OPERAND, JOB SPECIFICATION LANGUAGE

The operand consists of a list of parameters which are determined by and follow a Job Specification Lanaguage verb.

OPERATION, (JOB SPECIFICATION LANGUAGE)

A Job Specification Language operation consists of a Job Specification Language verb possibly followed by options.

OWN

An access control list attribute which when on for a user in the access control list specifies that user as an owner of the node.

OWNER, FILE (NODE)

A user is an owner when his name appears in the access control list for that node with the own attribute. The creator of a node is automatically made an owner.

PARAMETER

An entry in the operand of a Job Specification Language statement. A variable that is given a constant value for a specific purpose or process.

PARAMETER, TAGGED

A parameter which must be preceded by an identifying word and separated from this word by an equal sign (=); e.g., ACCT = account number.

PARAMETER, POSITIONAL

A parameter which must be ordered in a prescribed way relative to other such parameters so that its function may be correctly interpreted.

PATH NAME

A D-dimension vector which specifies a node N at a level D in the file management tree. Its Ith component is the edge name of the edge entering the node at level I on the path from the root node to the node N.

POST MORTEM DUMP

A user requested dump at the end of a step.

PP

Peripheral Processor

RECORD

A user oriented logical way of breaking down a block of data.

REFERENCE CONTROL

The control invoked when a node is in the reference state with respect to a given user. This control is further categorized as full or no reference control.

REFERENCE STATE

One of the two possible states of a node which are defined by the user's intent at that node. A node is in the reference state when a user desires to utilize the son list at that node in a catalog search operation for higher level nodes in the catalog tree.

RESTART

To reestablish the execution of a job, using the data recorded at a check point.

RUN TIME PARAMETERS

Run time parameters are integer values or Job Specification Language variables which can be transmitted from the Job Specification Language statement to the user at execution time.

SECTOR

64 words on disc.

SERVICE

A component of a job which performs a service necessary for the completion of a job.

SEQUENCING VERSIONS

As versions of a file are cataloged, they are assigned sequential version numbers modulo the maximum number of versions set by a node owner.

SON

A node accessible from another node in a tree with a path length equal to one.

SOURCE LANGUAGE

A language that is the input to a given translation process.

STATEMENT, SPECIFICATION

An element of the Job Specification Language (JSL). It begins with the Job Specification Language identifer (/) in column one, and it contains at least a verb in the operation field.

STEP

A component of a job which is a processing program.

SUBFIELD

Within the operand, a finer division of a field. The delimiters used to separate subfields are the slash (/) and the asterisk (*) in the finest division; e.g., SECT = initial/maximum/limit.

SUBFILE

(1) A facility that divides a file into distinct entities. A file mark on tape causes a subfile to be defined on disc storage.

(2) One of the distinct divisions of a file.

SUBTREE

A tree consisting of a root node which can be any node of another tree and all nodes of that tree reachable from the root node.

SYMBOL, (ASC)

A symbol is a string of from one to eight characters, the first of which must be alphabetic. The remaining characters may be any sequence of alphanumeric characters.

TASK

A task is the most primitive program unit which performs a basic system service.

TREE

(1) A directed graph which contains no circuits and which has at most one edge (branch) entering each node.

(2) A finite set of one or more nodes, wherein:

(a) there is one specially designated node called the root of the tree, and

(b) the remaining nodes are partitioned into disjoint sets, and each of these sets in turn is a tree. These trees are called subtrees of the root.

USER CODE

A sequence of one to eight alphanumeric characters which identify a specific user.

VARIABLE, JOB SPECIFICATION

A user may equate a numeric value to an alphabetic character which may be referenced in Job Specification Language dependency structure or by the user program. The variable value may change dynamically to direct various sequences to be executed in the Job Specification Language or the user program.

VERB

A word or abbreviation which specifies the function of a Job Specification Language statement. The verb must always be preceded by a blank.

VERSION

One of the copies of the file cataloged at a node.

VERSION, FLAGGED

That version of the file cataloged at a node which is assigned to a user job when no other specific version number has been specified by the user.

APPENDIX C

WORDS, SECTORS, BANDS CONVERSIONS

APPENDIX C: WORDS, SECTORS, BANDS CONVERSIONS

WORD	SECT	BAND	WORD	SECT	BAND	WORD	SECT	BAND
0	0	0.0	204800	3200	12.50	409600	6400	25.00
4096	64	0.25	208896	3264	12.75	413696	6464	25.25
8192	128	0.50	212992	3328	13.00	417792	6528	25.50
12288	192	0.75	217088	3392	13.25	421888	6592	25.75
16384	256	1.00	221184	3456	13.50	425984	6656	26.00
20480	320	1.25	225280	3520	13.75	430080	6720	26.25
24576	384	1.50	229376	3584	14.00	434176	6784	26.50
28672	448	1.75	233472	3648	14.25	438272	6848	26.75
32768	512	2.00	237568	3712	14.50	442368	6912	27.00
36864	576	2.25	241664	3776	14.75	446464	6976	27.25
40960	640	2.50	245760	3840	15.00	450560	7040	27.50
45056	704	2.75	249856	3904	15.25	454656	7104	27.75
49152	768	3.00	253952	3968	15.50	458752	7168	28.00
53248	832	3.25	258048	4032	15.75	462848	7232	28.25
57344	896	3.50	262144	4096	16.00	466944	7296	28.50
61440	960	3.75	266240	4160	16.25	471040	7360	28.75
65536	1024	4.00	270336	4224	16.50	475136	7424	29.00
69632	1088	4.25	274432	4288	16.75	479232	7488	29.25
73728	1152	4.50	278528	4352	17.00	483328	7552	29.50
77824	1216	4.75	282624	4416	17.25	487424	7616	29.75
81920	1280	5.00	286720	4480	17.50	491520	7680	30.00
86016	1344	5.25	290816	4544	17.75	495616	7744	30.25
90112	1408	5.50	294912	4608	18.00	499712	7808	30.50
94208	1472	5.75	299008	4672	18.25	503808	7872	30.75
98304	1536	6.00	303104	4736	18.50	507904	7936	31.00
102400	1600	6.25	307200	4800	18.75	512000	8000	31.25
106496	1664	6.50	311296	4864	19.00	516096	8064	31.50
110592	1728	6.75	315392	4928	19.25	520192	8128	31.75
114688	1792	7.00	319488	4992	19.50	524288	8192	32.00
118784	1856	7.25	323584	5056	19.75	528384	8256	32.25
122880	1920	7.50	327680	5120	20.00	532480	8320	32.50
126976	1984	7.75	331776	5184	20.25	536576	8384	32.75
131072	2048	8.00	335872	5248	20.50	540672	8448	33.00
135168	2112	8.25	339968	5312	20.75	544768	8512	33.25
139264	2176	8.50	344064	5376	21.00	548864	8576	33.50
143360	2240	8.75	348160	5440	21.25	552960	8640	33.75
147456	2304	9.00	352256	5504	21.50	557056	8704	34.00
151552	2368	9.25	356352	5568	21.75	561152	8768	34.25
155648	2432	9.50	360448	5632	22.00	565248	8832	34.50
159744	2496	9.75	364544	5696	22.25	569344	8896	34.75
163840	2560	10.00	368640	5760	22.50	573440	8960	35.00
167936	2624	10.25	372736	5824	22.75	577536	9024	35.25
172032	2688	10.50	376832	5888	23.00	581632	9088	35.50
176128	2752	10.75	380928	5952	23.25	585728	9152	35.75
180224	2816	11.00	385024	6016	23.50	589824	9216	36.00
184320	2880	11.25	389120	6080	23.75	593920	9280	36.25
188416	2944	11.50	393216	6144	24.00	598016	9344	36.50
192512	3008	11.75	397312	6208	24.25	602112	9408	36.75
196608	3072	12.00	401408	6272	24.50	606208	9472	37.00
200704	3136	12.25	405504	6336	24.75	610304	9536	37.25

WORD	SECT	BAND	WORD	SECT	BAND	WORD	SECT	BAND
614400	9600	37.50	819200	12800	50.00	1024000	16000	62.50
618496	9664	37.75	823296	12864	50.25	1028096	16064	62.75
622592	9728	38.00	827392	12928	50.50	1032192	16128	63.00
626688	9792	38.25	831488	12992	50.75	1036288	16192	63.25
630784	9856	38.50	835584	13056	51.00	1040384	16256	63.50
634880	9920	38.75	839680	13120	51.25	1044480	16320	63.75
638976	9984	39.00	843776	13184	51.50	1048576	16384	64.00
643072	10048	39.25	847872	13248	51.75	1052672	16448	64.25
647168	10112	39.50	851968	13312	52.00	1056768	16512	64.50
651264	10176	39.75	856064	13376	52.25	1060864	16576	64.75
655360	10240	40.00	860160	13440	52.50	1064960	16640	65.00
659456	10304	40.25	864256	13504	52.75	1069056	16704	65.25
663552	10368	40.50	868352	13568	53.00	1073152	16768	65.50
667648	10432	40.75	872448	13632	53.25	1077248	16832	65.75
671744	10496	41.00	876544	13696	53.50	1081344	16896	66.00
675840	10560	41.25	880640	13760	53.75	1085440	16960	66.25
679936	10624	41.50	884736	13824	54.00	1089536	17024	66.50
684032	10688	41.75	888832	13888	54.25	1093632	17088	66.75
688128	10752	42.00	892928	13952	54.50	1097728	17152	67.00
692224	10816	42.25	897024	14016	54.75	1101824	17216	67.25
696320	10880	42.50	901120	14080	55.00	1105920	17280	67.50
700416	10944	42.75	905216	14144	55.25	1110016	17344	67.75
704512	11008	43.00	909312	14208	55.50	1114112	17408	68.00
708608	11072	43.25	913408	14272	55.75	1118208	17472	68.25
712704	11136	43.50	917504	14336	56.00	1122304	17536	68.50
716800	11200	43.75	921600	14400	56.25	1126400	17600	68.75
720896	11264	44.00	925696	14464	56.50	1130496	17664	69.00
724992	11328	44.25	929792	14528	56.75	1134592	17728	69.25
729088	11392	44.50	933888	14592	57.00	1138688	17792	69.50
733184	11456	44.75	937984	14656	57.25	1142784	17856	69.75
737280	11520	45.00	942080	14720	57.50	1146880	17920	70.00
741376	11584	45.25	946176	14784	57.75	1150976	17984	70.25
745472	11648	45.50	950272	14848	58.00	1155072	18048	70.50
749568	11712	45.75	954368	14912	58.25	1159168	18112	70.75
753664	11776	46.00	958464	14976	58.50	1163264	18176	71.00
757760	11840	46.25	962560	15040	58.75	1167360	18240	71.25
761856	11904	46.50	966656	15104	59.00	1171456	18304	71.50
765952	11968	46.75	970752	15168	59.25	1175552	18368	71.75
770048	12032	47.00	974848	15232	59.50	1179648	18432	72.00
774144	12096	47.25	978944	15296	59.75	1183744	18496	72.25
778240	12160	47.50	983040	15360	60.00	1187840	18560	72.50
782336	12224	47.75	987136	15424	60.25	1191936	18624	72.75
786432	12288	48.00	991232	15488	60.50	1196032	18688	73.00
790528	12352	48.25	995328	15552	60.75	1200128	18752	73.25
794624	12416	48.50	999424	15616	61.00	1204224	18816	73.50
798720	12480	48.75	1003520	15680	61.25	1208320	18880	73.75
802816	12544	49.00	1007616	15744	61.50	1212416	18944	74.00
806912	12608	49.25	1011712	15808	61.75	1216512	19008	74.25
811008	12672	49.50	1015808	15872	62.00	1220608	19072	74.50
815104	12736	49.75	1019904	15936	62.25	1224704	19136	74.75

WORD	SECT	BAND	WORD	SECT	BAND	WORD	SECT	BAND
1228800	19200	75.00	1433600	22400	87.50	1638400	25600	100.00
1232896	19264	75.25	1437696	22464	87.75	1642496	25664	100.25
1236992	19328	75.50	1441792	22528	88.00	1646592	25728	100.50
1241088	19392	75.75	1445888	22592	88.25	1650688	25792	100.75
1245184	19456	76.00	1449984	22656	88.50	1654784	25856	101.00
1249280	19520	76.25	1454080	22720	88.75	1658880	25920	101.25
1253376	19584	76.50	1458176	22784	89.00	1662976	25984	101.50
1257472	19648	76.75	1462272	22848	89.25	1667072	26048	101.75
1261568	19712	77.00	1466368	22912	89.50	1671168	26112	102.00
1265664	19776	77.25	1470464	22976	89.75	1675264	26176	102.25
1269760	19840	77.50	1474560	23040	90.00	1679360	26240	102.50
1273856	19904	77.75	1478656	23104	90.25	1683456	26304	102.75
1277952	19968	78.00	1482752	23168	90.50	1687552	26368	103.00
1282048	20032	78.25	1486848	23232	90.75	1691648	26432	103.25
1286144	20096	78.50	1490944	23296	91.00	1695744	26496	103.50
1290240	20160	78.75	1495040	23360	91.25	1699840	26560	103.75
1294336	20224	79.00	1499136	23424	91.50	1703936	26624	104.00
1298432	20288	79.25	1503232	23488	91.75	1708032	26688	104.25
1302528	20352	79.50	1507328	23552	92.00	1712128	26752	104.50
1306624	20416	79.75	1511424	23616	92.25	1716224	26816	104.75
1310720	20480	80.00	1515520	23680	92.50	1720320	26880	105.00
1314816	20544	80.25	1519616	23744	92.75	1724416	26944	105.25
1318912	20608	80.50	1523712	23808	93.00	1728512	27008	105.50
1323008	20672	80.75	1527808	23872	93.25	1732608	27072	105.75
1327104	20736	81.00	1531904	23936	93.50	1736704	27136	106.00
1331200	20800	81.25	1536000	24000	93.75	1740800	27200	106.25
1335296	20864	81.50	1540096	24064	94.00	1744896	27264	106.50
1339392	20928	81.75	1544192	24128	94.25	1748992	27328	106.75
1343488	20992	82.00	1548288	24192	94.50	1753088	27392	107.00
1347584	21056	82.25	1552384	24256	94.75	1757184	27456	107.25
1351680	21120	82.50	1556480	24320	95.00	1761280	27520	107.50
1355776	21184	82.75	1560576	24384	95.25	1765376	27584	107.75
1359872	21248	83.00	1564672	24448	95.50	1769472	27648	108.00
1363968	21312	83.25	1568768	24512	95.75	1773568	27712	108.25
1368064	21376	83.50	1572864	24576	96.00	1777664	27776	108.50
1372160	21440	83.75	1576960	24640	96.25	1781760	27840	108.75
1376256	21504	84.00	1581056	24704	96.50	1785856	27904	109.00
1380352	21568	84.25	1585152	24768	96.75	1789952	27968	109.25
1384448	21632	84.50	1589248	24832	97.00	1794048	28032	109.50
1388544	21696	84.75	1593344	24896	97.25	1798144	28096	109.75
1392640	21760	85.00	1597440	24960	97.50	1802240	28160	110.00
1396736	21824	85.25	1601536	25024	97.75	1806336	28224	110.25
1400832	21888	85.50	1605632	25088	98.00	1810432	28288	110.50
1404928	21952	85.75	1609728	25152	98.25	1814528	28352	110.75
1409024	22016	86.00	1613824	25216	98.50	1818624	28416	111.00
1413120	22080	86.25	1617920	25280	98.75	1822720	28480	111.25
1417216	22144	86.50	1622016	25344	99.00	1826816	28544	111.50
1421312	22208	86.75	1626112	25408	99.25	1830912	28608	111.75
1425408	22272	87.00	1630208	25472	99.50	1835008	28672	112.00
1429504	22336	87.25	1634304	25536	99.75	1839104	28736	112.25

WORD	SECT	BAND	WORD	SECT	BAND	WORD	SECT	BAND
1843200	28800	112.50	2048000	32000	125.00	2252800	35200	137.50
1847296	28864	112.75	2052096	32064	125.25	2256896	35264	137.75
1851392	28928	113.00	2056192	32128	125.50	2260992	35328	138.00
1855488	28992	113.25	2060288	32192	125.75	2265088	35392	138.25
1859584	29056	113.50	2064384	32256	126.00	2269184	35456	138.50
1863680	29120	113.75	2068480	32320	126.25	2273280	35520	138.75
1867776	29184	114.00	2072576	32384	126.50	2277376	35584	139.00
1871872	29248	114.25	2076672	32448	126.75	2281472	35648	139.25
1875968	29312	114.50	2080768	32512	127.00	2285568	35712	139.50
1880064	29376	114.75	2084864	32576	127.25	2289664	35776	139.75
1884160	29440	115.00	2088960	32640	127.50	2293760	35840	140.00
1888256	29504	115.25	2093056	32704	127.75	2297856	35904	140.25
1892352	29568	115.50	2097152	32768	128.00	2301952	35968	140.50
1896448	29632	115.75	2101248	32832	128.25	2306048	36032	140.75
1900544	29696	116.00	2105344	32896	128.50	2310144	36096	141.00
1904640	29760	116.25	2109440	32960	128.75	2314240	36160	141.25
1908736	29824	116.50	2113536	33024	129.00	2318336	36224	141.50
1912832	29888	116.75	2117632	33088	129.25	2322432	36288	141.75
1916928	29952	117.00	2121728	33152	129.50	2326528	36352	142.00
1921024	30016	117.25	2125824	33216	129.75	2330624	36416	142.25
1925120	30080	117.50	2129920	33280	130.00	2334720	36480	142.50
1929216	30144	117.75	2134016	33344	130.25	2338816	36544	142.75
1933312	30208	118.00	2138112	33408	130.50	2342912	36608	143.00
1937408	30272	118.25	2142208	33472	130.75	2347008	36672	143.25
1941504	30336	118.50	2146304	33536	131.00	2351104	36736	143.50
1945600	30400	118.75	2150400	33600	131.25	2355200	36800	143.75
1949696	30464	119.00	2154496	33664	131.50	2359296	36864	144.00
1953792	30528	119.25	2158592	33728	131.75	2363392	36928	144.25
1957888	30592	119.50	2162688	33792	132.00	2367488	36992	144.50
1961984	30656	119.75	2166784	33856	132.25	2371584	37056	144.75
1966080	30720	120.00	2170880	33920	132.50	2375680	37120	145.00
1970176	30784	120.25	2174976	33984	132.75	2379776	37184	145.25
1974272	30848	120.50	2179072	34048	133.00	2383872	37248	145.50
1978368	30912	120.75	2183168	34112	133.25	2387968	37312	145.75
1982464	30976	121.00	2187264	34176	133.50	2392064	37376	146.00
1986560	31040	121.25	2191360	34240	133.75	2396160	37440	146.25
1990656	31104	121.50	2195456	34304	134.00	2400256	37504	146.50
1994752	31168	121.75	2199552	34368	134.25	2404352	37568	146.75
1998848	31232	122.00	2203648	34432	134.50	2408448	37632	147.00
2002944	31296	122.25	2207744	34496	134.75	2412544	37696	147.25
2007040	31360	122.50	2211840	34560	135.00	2416640	37760	147.50
2011136	31424	122.75	2215936	34624	135.25	2420736	37824	147.75
2015232	31488	123.00	2220032	34688	135.50	2424832	37888	148.00
2019328	31552	123.25	2224128	34752	135.75	2428928	37952	148.25
2023424	31616	123.50	2228224	34816	136.00	2433024	38016	148.50
2027520	31680	123.75	2232320	34880	136.25	2437120	38080	148.75
2031616	31744	124.00	2236416	34944	136.50	2441216	38144	149.00
2035712	31808	124.25	2240512	35008	136.75	2445312	38208	149.25
2039808	31872	124.50	2244608	35072	137.00	2449408	38272	149.50
2043904	31936	124.75	2248704	35136	137.25	2453504	38336	149.75

WORD	SECT	BAND	WORD	SECT	BAND	WORD	SECT	BAND
2457600	38400	150.00	2662400	41600	162.50	2867200	44800	175.00
2461696	38464	150.25	2666496	41664	162.75	2871296	44864	175.25
2465792	38528	150.50	2670592	41728	163.00	2875392	44928	175.50
2469888	38592	150.75	2674688	47192	163.25	2879488	44992	175.75
2473984	38656	151.00	2678784	41856	163.50	2883584	45056	176.00
2478080	38720	151.25	2682880	41920	163.75	2887680	45120	176.25
2482176	38784	151.50	2686976	41984	164.00	2891776	45184	176.50
2486272	38848	151.75	2691072	42048	164.25	2895872	45248	176.75
2490368	38912	152.00	2695168	42112	164.50	2899968	45312	177.00
2494464	38976	152.25	2699264	42176	164.75	2904064	45376	177.25
2498560	39040	152.50	2703360	42240	165.00	2908160	45440	177.50
2502656	39104	152.75	2707456	42304	165.25	2912256	45504	177.75
2506752	39168	153.00	27711552	42368	165.50	2916352	45568	178.00
2510848	39232	153.25	2715648	42432	165.75	2920448	45632	178.25
2514944	39296	153.50	2719744	42496	166.00	2924544	45696	178.50
2519040	39360	153.75	2723840	42560	166.25	2928640	45760	178.75
2523136	39424	154.00	2727936	42624	166.50	2932736	45824	179.00
2527232	39488	154.25	2732032	42688	166.75	2936832	45888	179.25
2531328	39552	154.50	2736128	42752	167.00	2940928	45952	179.50
2535424	39616	154.75	2740224	42816	167.25	2945024	46016	179.75
2539520	39680	155.00	2744320	42880	167.50	2949120	46080	180.00
2543616	39744	155.25	2748416	42944	167.75	2953216	46144	180.25
2547712	39808	155.50	2752512	43008	168.00	2957312	46208	180.50
2551808	39872	155.75	2756608	43072	168.25	2961408	46272	180.75
2555904	39936	156.00	2760704	43136	168.50	2965504	46336	181.00
2560000	40000	156.25	2764800	43200	168.75	2969600	46400	181.25
2564096	40064	156.50	2768896	43264	169.00	2973696	46464	181.50
2568192	40128	156.75	2772992	43328	169.25	2977792	46528	181.75
2572288	40192	157.00	2777088	43392	169.50	298188	46592	182.00
2576384	40256	157.25	2781184	43456	169.75	2985984	46656	182.25
2580480	40320	157.50	2785280	43520	170.00	2990080	46720	182.50
2584576	40384	157.75	2789376	43584	170.25	2994176	46784	182.75
2588672	40448	158.00	2793472	43648	170.50	2998272	46848	183.00
2592768	40512	158.25	2797568	43712	170.75	3002368	46912	183.25
2596864	40576	158.50	2801664	43776	171.00	3006464	46976	183.50
2600960	40640	158.75	2805760	43840	171.25	3010560	47040	183.75
2605056	40704	159.00	2809856	43904	171.50	3014656	47104	184.00
2609152	40768	159.25	2813952	43968	171.75	3018752	47168	184.25
2613248	40382	159.50	2818048	44032	172.00	3022848	47232	184.50
2617344	40896	159.75	2822144	44096	172.25	3026944	47296	184.75
2621440	40960	160.00	2826240	44160	172.50	3031040	47360	185.00
2625536	41024	160.25	2830336	44224	172.75	3035136	47424	185.25
2629632	41088	160.50	2834432	44288	173.00	3039232	47488	185.50
2633728	41152	160.75	2838528	44353	173.25	3043328	47552	185.75
2637824	41216	161.00	2842624	44416	173.50	3047424	47616	186.00
2641920	41280	161.25	2846720	44480	173.75	3051520	47680	186.25
2646016	41344	161.50	2850816	44544	174.00	3055616	47744	186.50
2650112	41408	161.75	2854912	44608	174.25	3059712	47808	186.75
2654208	41472	162.00	2859008	44672	174.50	3063808	47872	187.00
2658304	41536	162.25	2863104	44736	174.75	3067904	47936	187.25

WORD	SECT	BAND	WORD	SECT	BAND	WORD	SECT	BAND
3072000	48000	187.50	3276800	51200	200.00	3481600	54400	212.50
3076096	48064	187.75	3280896	51264	200.25	3485696	54464	212.75
3080192	48128	188.00	3284992	51328	200.50	3489792	54528	213.00
3084288	48192	188.25	3289088	51392	200.75	3493888	54592	213.25
3088384	48256	188.50	3293184	51456	201.00	3497984	54656	213.50
3092480	48320	188.75	3297280	51520	201.25	3502080	54720	213.75
3096576	48384	189.00	3301376	51584	201.50	3506176	54784	214.00
3100672	48448	189.25	3305472	51648	201.75	3510272	54848	214.25
3104768	48512	189.50	3309568	51712	202.00	3514368	54912	214.50
3108864	48576	189.75	3313664	51776	202.25	3518464	54976	214.75
3112960	48640	190.00	3317760	51840	202.50	3522560	55040	215.00
3117056	48704	190.25	3321856	51904	202.75	3526656	55104	215.25
3121152	48768	190.50	3325952	51968	203.00	3530752	55168	215.50
3125248	48832	190.75	3330048	52032	203.25	3534848	55232	215.75
3129344	48896	191.00	3334144	52096	203.50	3538944	55296	216.00
3133440	48960	191.25	3338240	52160	203.75	3543040	55360	216.25
3137536	49024	191.50	3342336	52224	204.00	3547136	55424	216.50
3141632	49088	191.75	3346432	52288	204.25	3551232	55488	216.75
3145728	49152	192.00	3350528	52352	204.50	3555328	55552	217.00
3149824	49216	192.25	3354624	52416	204.75	3559424	55616	217.25
3153920	49280	192.50	3358720	52480	205.00	3563520	55680	217.50
3158016	49344	192.75	3362816	52544	205.25	3567616	55744	217.75
3162112	49408	193.00	3366912	52608	205.50	3571712	55808	218.00
3166208	49472	193.25	3371008	52672	205.75	3575808	55872	218.25
3170304	49536	193.50	3375104	52736	206.00	3579904	55936	218.50
3174400	49600	193.75	3379200	52800	206.25	3584000	56000	218.75
3178496	49664	194.00	3383296	52864	206.50	3588096	56064	219.00
3182592	49728	194.25	3387392	52928	206.75	3592192	56128	219.25
3186688	49792	194.50	3391488	52992	207.00	3596288	56192	219.50
3190784	49856	194.75	3395584	53056	207.25	3600384	56256	219.75
3194880	49920	195.00	3399680	53120	207.50	3604480	56320	220.00
3198976	49984	195.25	3403776	53184	207.75	3608576	56384	220.25
3203072	50048	195.50	3407872	53248	208.00	3612672	56448	220.50
3207168	50112	195.75	3411968	53312	208.25	3616768	56512	220.75
3211264	50176	196.00	3416064	53376	208.50	3620864	56576	221.00
3215360	50240	196.25	3420160	53440	208.75	3624960	56640	221.25
3219456	50304	196.50	3424256	53504	209.00	3629056	56704	221.50
3223552	50368	196.75	3428352	53568	209.25	3633152	56768	221.75
3227648	50432	197.00	3432448	53632	209.50	3637248	56832	222.00
3231744	50496	197.25	3436544	53696	209.75	3641344	56896	222.25
3235840	50560	197.50	3440640	53760	210.00	3645440	56960	222.50
3239936	50624	197.75	3444736	53824	210.25	3649536	57024	222.75
3244032	50688	198.00	3448832	53888	210.50	3653632	57088	223.00
3248128	50752	198.25	3452928	53952	210.75	3657728	57152	223.25
3252224	50816	198.50	3457024	54016	211.00	3661824	57216	223.50
3256320	50880	198.75	3461120	54080	211.25	3665920	57280	223.75
3260416	50944	199.00	3465216	54144	211.50	3670016	57344	224.00
3264512	51008	199.25	3469312	54208	211.75	3674112	57408	224.25
3268608	51072	199.50	3473408	54272	212.00	3678208	57472	224.50
3272704	51136	199.75	3477504	54336	212.25	3682304	57536	224.75

APPENDIX D

STATEMENT USES

APPENDIX D

STATEMENT USES

The following is a brief description of all JSL statements. The flagged statements (*) are those which are macros; the remainder are JSL statements.

Statement	Purpose
ASG	The ASG statement informs the system that the specified version of a cataloged file is to be accessible to the current job by the name specified in the statement.
*ASM	The ASM statement is a macro that specifies execution of the ASC Assembler. The ASM macro produces the XQT statement, and optional supporting JSL statements for file definitions and renaming, required to execute the ASC Assembler.
*ASML	The ASML statement is a macro that specifies execution of the ASC Assembler and the ASC Linkage Editor. The ASML macro produces the XQT statements, and optional supporting JSL statements for file definition and renaming, required to execute the ASC Assembler and Linkage Editor.
*ASMLX	The ASMLX statement is a macro that specifies execution of the ASC Assembler, the ASC Linkage Editor and the load module produced. The ASMLX macro produces the XQT statements, and optional supporting JSL statements and renaming, required to execute the ASC Assembler, Linkage Editor, and the load module produced.
BATINT	The BATINT statement specifies the change of a batch job to an interactive job.
BUILD	The BUILD statement is a macro language directive that constructs a buffer in card image of parameter values to be input to the Translator.
CAT	The CAT statement initiates concurrently the entry of a node into a catalog and the entry of the first version of the file at that node.
CATBLD	The CATBLD statement initiates the entry of a new catalog into the system.
CATN	The CATN statement initiates the entry of a node into a catalog.

Statement	Purpose
CATV	The CATV statement initiates the entry of a version of a file at an existing node.
CHG	The CHG statement initiates specified changes in the characteristics of a node.
*CJSL	The CJSL statement initiates a check of the syntax of a job that is defined in an external or cataloged file.
CNT	The CNT statement is the start delimiter for a concatenated input file request. Each concatenated input file request results in associating two or more ASG, FIT, or FD statements under the CNT's access name as concatenated files.
CNTE	The CNTE statement is the end delimiter for a concatenated input file request. CNTE defines the end of the associated ASG, FIT, or FD statements defining the files to be concatenated under the CNT's access name.
COM	The COM statement specifies narrative comments that are to be entered into the Job Specification Language source listing of the system output file.
DEL	The DEL statement initiates the deletion of a terminal node from a catalog.
DELV	The DELV statement initiates the deletion of a specified version of the file at an existing node.
DMACRO	The DMACRO statement initiates the deactivation of the most recently activated macro definition of the specified name and initiates the reactivation of the next preceding macro definition of the specified name. The deactivated definition is removed from the stack.
EOJ	The EOJ statement indicates the end of statements and data associated with a job during job input, and initiates job termination when the statement is processed during job execution.
EOJX	The EOJX statement specifies the end of statements and data associated with a deferred job.
FD	The FD statement defines the characteristics of a job local file -- logical I/O characteristics and disc allocation. The FD statement is associated with a file on secondary disc. Multiple FD statements within the same job for the same access name allow modification of logical file characteristics from step to step within the job.

Statement	Purpose
FIT	The FIT statement initiates a call to read a file from a half-inch magnetic tape and place that file on disc.
FOSYS	The FOSYS statement places the specified file into either the job's print or punch output file chain. The print or punch files are output at the termination of the associated job.
FOT	The FOT statement initiates a call to have a specified disc file written to half-inch magnetic tape.
*FTN	The FTN statement is a macro that specifies execution of the ASC Fortran Compiler. The FTN macro produces the XQT statement, and optional supporting JSL statements for file definition and renaming required to execute the ASC Fortran Compiler.
*FTNL	The FTNL statement is a macro that specifies execution of the ASC Fortran Compiler and the ASC Linkage Editor. The FTNL macro produces the XQT statements, and optional supporting JSL statements for file definition and renaming required to execute the ASC Fortran Compiler and Linkage Editor.
*FTNLX	The FTNLX statement is a macro that specifies execution of the ASC Fortran Compiler, Linkage Editor, and resulting load module. The FTNLX macro produces the XQT statements, and optional supporting JSL statements for file definition and renaming required to execute the ASC Fortran Compiler, Linkage Editor and load module.
*FXQT	The FXQT statement is a macro that specifies the execution of Fortran load modules using Fortran I/O. The FXQT eases the naming of the input file and the printing of the output print file.
GEN	The GEN statement is a macro language directive that clears the BUILD-generated buffer by outputting the buffer contents to the Translator.
GOTO	The GOTO statement specifies an unconditional forward branch of execution to a specified step within a job block, or specifies an unconditional forward branch of execution to another job block.
IF	The IF statement specifies a conditional forward branch of execution to a specified step within a job block or to another job block.
JOB	The JOB statement defines the beginning of a job.

Statement — Purpose

JOBX — The JOBX statement defines the beginning of a source job specification contained in a file embedded in another source job specification file (a source job specification in a file defined by a START/STOP statement pair).

JUMP — The JUMP statement specifies either a conditional or an unconditional transfer of the generation sequence during the expansion of a macro definition.

LIMIT — The LIMIT statement defines the beginning of a job block within a job.

*LNK — The LNK statement is a macro that specifies execution of the ASC Linkage Editor. The LNK macro produces the XQT, the optional supporting JSL statements for file definition and renaming, required to execute the ASC Linkage Editor.

*LNKX — The LNKX statement is a macro that specifies execution of the ASC Linkage Editor and resulting load module. The LNKX macro produces the XQT statements, the optional supporting JSL statements for file definition and renaming, required to execute the ASC Linkage Editor and load module produced.

MACASG — The MACASG statement defines the use of a macro file to the JSL Translator.

MACBLD — The MACBLD statement creates or modifies a macro file. All macros defined between the MACBLD and either the current end of input file, or the next MACBLD statement, are written to the specified access name.

MACRO — The MACRO statement initiates the definition of a macro, specifies the name of the macro to be defined, and specifies the default values of the formal parameters in the macro body.

MEND — The MEND statement marks the end of the macro definition with which it is associated.

MACRO CALL — A Macro Call statement initiates the expansion of a macro whose name is used as the verb of that macro call statement.

MACRO MODEL — A Macro Model statement is any statement in a macro body that may be expanded into one or more basic Job Specification Language statements in a macro expansion.

MFR — The MFR statement is the start delimiter for a multi-file reel request. All file statements included must be of one type.

Statement	Purpose
MFRE	The MFRE statement is the end delimiter for a multi-file reel request. MFRE defines the end of the included files following the MFR statement.
PD	The PD statement defines a synonym for the pathname of a catalog node that exists at the time the synonym is referenced.
PJSL	The PJSL statement initiates the entry of a job that is defined in an external or cataloged file into the system input stream to be queued for execution.
PRINT	The PRINT statement specifies that the current value(s) of one or more of the twenty-seven JSL variables and/or a message to be entered into the system output file at the time of execution of PRINT.
REL	The REL statement specifies that the disc space for the named file or files is to be released and returned to the allocation for the current job block.
RENAME	The RENAME statement specifies the correspondence between the standard access name and a user-defined access name. The new access name replaces the existing access name for the file; i.e., the new name destroys any former name in the job database.
RPLV	The RPLV statement initiates the replacement of a specified version of a file at an existing node.
SET	The SET statement assigns an integer value to a Job Specification Language variable.
SETUP	The SETUP statement instructs the system to schedule a secondary storage file on video tape.
START	The START statement initiates the input of a file within the source job specification file (a file embedded in the job input stream).
STOP	The STOP statement is the terminator of a file that is contained within the source job specification file.
VERBSYN	The VERBSYN statement is a macro language directive that enables the user to redefine an intrinsic JSL verb function.
XQT	The XQT statement initiates the execution of a load module. The ASC Fortran Compiler, ASC Assembler, and ASC Linkage Editor may be executed through an XQT statement. The JSL statements needed to define all input and output files expected by the Compiler, Assembler or Linkage Editor must be included within the JSL program preceding the XQT statement.

INDEX

JSL REFERENCE MANUAL

INDEX

JSL REFERENCE MANUAL

TEXAS INSTRUMENTS
INCORPORATED
EQUIPMENT GROUP
AUSTIN, TEXAS

PUBLICATION UPDATE

PUBLICATION

PROGRAM ASC PUBLICATION NO. 930038-2

TITLE Job Specification Language Reference Manual

DATE February 1974 JOB NO. 1948

TYPE OF CHANGE

□ IMMEDIATE
(MAY CAUSE PERSONAL INJURY OR EQUIPMENT DAMAGE/FAILURE)

□ ROUTINE
(BATCH PROCESSED)

SUBMITTED BY

NAME ______________ PHONE ______________

ADDRESS ______________

MAIL STATION ______________ DATE ______________

LIST PAGE AND PARAGRAPH OR FIGURE NUMBERS AND DESCRIBE RECOMMENDED CHANGES.

FORWARD CHANGES BY FOLDING THIS SHEET AND STAPLING. RETURN ADDRESS IS ON BACK OF SHEET.

TEXAS INSTRUMENTS INCORPORATED
EQUIPMENT GROUP
P.O. BOX 2909
AUSTIN, TEXAS 78767

ATTENTION: TECHNICAL DATA BRANCH
MAIL STATION 2146

TEXAS INSTRUMENTS
INCORPORATED
EQUIPMENT GROUP
AUSTIN, TEXAS

APPLICATION FOR AUTOMATIC UPDATE

PUBLICATION

PROGRAM ASC PUBLICATION NO. 930038

TITLE JOB SPECIFICATION LANGUAGE REFERENCE MANUAL

SUBMITTED BY

NAME __________ PHONE __________ DATE __________

ADDRESS __________

MAIL STATION __________ NO. OF COPIES ORDERED __________

ON RECEIPT OF THIS REQUEST, WE WILL ENTER YOUR NAME ON THE AUTOMATIC UPDATE LIST. IF YOUR ADDRESS IS CURRENT, YOU WILL RECEIVE ANY UPDATED MATERIAL THAT IS PUBLISHED REGARDING THIS MANUAL.

FOLD AND STAPLE THIS SHEET. RETURN ADDRESS IS ON REVERSE SIDE

TEXAS INSTRUMENTS
INCORPORATED

EQUIPMENT GROUP
P.O. BOX 2909
AUSTIN, TEXAS 78767

ATTENTION
TECHNICAL DATA BRANCH
MAIL STATION 2146