UNISYS

PUBLICATIONS UPDATE

System 80

OS/3
Job Control
Programming
Reference Manual

UP-9984 Rev. 1-A

This Library Memo announces the release and availability of Update A to the *System 80 OS/3 Job Control Programming Reference Manual*, UP-9984 Rev. 1.

This manual is a standard library item (SLI). It is part of the standard library provided automatically with the purchase of the product.

Job control consists of a series of transient routines and is activated by a job control statement or an operator command that is input from the system console. These transient routines allow job control to manage system resources and prepare jobs for execution.

This manual describes how to use job control to direct and control the flow of jobs through the system.

This update provides an index and a set of tab breakers. The content of the manual is unchanged.

Copies of Update A are now available. You can order the update only, or the complete manual with the update, through your local Unisys representative. To receive only the update, order UP-9984 Rev. 1-A. To receive the complete manual, order UP-9984 Rev. 1.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists MBZ, MCZ, MMZ, MGZ, and MHA	Mailing Lists MBW, MB00, and MB01 (21 pages plus Memo)	Library Memo for UP-9984 Rev. 1-A
		RELEASE DATE: February 1989

UNISYS

PUBLICATIONS REVISION

System 80

OS/3
Job Control
Programming
Reference Manual

UP-9984 Rev. 1

This Library Memo announces the release and availability of the *System 80 OS/3 Job Control Programming Reference Manual*, UP-9984 Rev. 1.

This manual is a standard library item (SLI). It is part of the standard library provided automatically with the purchase of the product.

Job control consists of a series of transient routines and is activated by a job control statement or an operator command that is input from the system console. These transient routines allow job control to manage system resources and prepare jobs for execution.

This manual describes how to use job control to direct and control the flow of jobs through the system. It contains descriptions of the following:

- The job control stream
- The function and parameters of each job control statement
- The function and parameters of each job control proc
- Each proc definition statement

For Release 12.0, this manual adds parameters to the DD, INQ JOB, LFD, OPTION, ROUTE, and SPL job control statements, adds information about the 8494 disk subsystem, and incorporates editorial changes.

Destruction Notice: This revision supersedes and replaces the *OS/3 Job Control Programming Reference Guide*, UP-9984, released on Library Memo dated February 1984. Please destroy all copies of UP-9984, all updates and Library Memos.

Additional copies may be ordered through your local Unisys representative.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists MBZ, MCZ, MMZ, MDE, MGZ, and MHA	Mailing Lists MBW, MB00, and MB01 (231 pages plus Memo)	Library Memo for UP-9984 Rev. 1
		RELEASE DATE: October 1988

UNISYS

System 80 OS/3 Job Control

Programming Reference Manual

OS/3 Release 12.0

Priced Item

October 1988

Printed in U S America
UP-9984 Rev. 1

UNISYS

System 80
OS/3
Job Control
Programming Reference Manual

OS/3 Release 12.0

October 1988

Printed in U S America
UP-9984 Rev. 1

Priced Item

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this publication should be forwarded, using the User Comments form at the back of this manual or remarks addressed directly to Unisys Corporation, to E/MSG Product Information, P.O. Box 500, M.S. E5-114, Blue Bell, PA 19424 U.S.A.

PAGE STATUS SUMMARY
ISSUE: Update A - UP-9984 Rev. 1
RELEASE LEVEL 12.0 Forward

Part/Section	Page Number	Update Level
Cover		Orig.
Title Page/Disclaimer		Orig.
PSS	iii	A
About This Manual	Tab Breaker	A*
	v thru x	Orig.
Contents	xi thru xvii	Orig.
1	Tab Breaker	A*
	1 thru 11	Orig.
2	Tab Breaker	A*
	1 thru 105	Orig.
3	Tab Breaker	A*
	1 thru 74	Orig.
4	Tab Breaker	A*
	1 thru 5	Orig.
Appendix A	Tab Breaker	A*
	1 thru 8	Orig.
Appendix B	Tab Breaker	A*
	1	Orig.
Glossary	Tab Breaker	A*
	1 thru 9	Orig.
Index	Tab Breaker	A*
	1 thru 9	A*
User Comments Form		
Back Cover		Orig.

Part/Section	Page Number	Update Level

Part/Section	Page Number	Update Level

* New page

About This Manual

Purpose

This programming manual is one in a series to be used by programmers familiar with the Unisys Operating System/3 (OS/3). This particular manual contains descriptions of the job control statements and procs used in a System 80 enhancement to communicate with job control and the proc definition statements that allow you to expand and conditionally modify the control stream when you start a job.

Audience

The intended audience for this manual is the programmer, who will use job control to direct and control the flow of jobs through the system.

How to Use This Manual

Read the entire manual to familiarize yourself with the basic concepts it presents; then use it for reference as needed.

Organization

This manual is divided into four sections and three appendixes.

Section 1. General Information

Describes the control stream concepts.

Section 2. Job Control Statements

Describes the function and parameters of each job control statement in alphabetical order by statement mnemonic.

Section 3. Job Control Procs

Describes the function and parameters of each job control proc in alphabetical order by proc mnemonic.

Section 4. Proc Definition Statements

Describes the function and parameters of each proc definition statement in alphabetical order by statement name.

Appendix A. Logical Unit Number Assignment Information

Provides the information needed to assign or change the logical unit numbers of the peripheral devices in a system.

Appendix B. Extent Specification Information

Provides the information required to establish or to extend a disk or diskette file.

Appendix C. Unisys Disk Subsystem Characteristics

Lists the physical characteristics of the applicable disk subsystems.

Related Product Information

The following Unisys documents may be useful in understanding and implementing OS/3's interactive services.

Note: *Throughout this manual, when we refer you to another manual, use the current version that applies to the software level at your site.*

Job Control Language Programming Guide **(UP-9986)**

This guide describes OS/3 job control and explains how to use it.

Interactive Services Operating Guide **(UP-9972)**

This guide describes the procedures used to communicate interactively with the operating system through a local workstation or remote terminal.

Operations Guide **(UP-8859)**

This guide describes the hardware configuration of each System 80 model and presents procedures for initializing the system. It also covers all the commands and procedures used in the OS/3 environment.

File Cataloging Technical Overview **(UP-9982)**

This overview discusses file cataloging and describes the facilities provided for it in a System 80 environment.

***Supervisor Macroinstructions Programming Reference Manual* (UP-8832)**

This manual describes the OS/3 supervisor macroinstructions used for program management, file space management, files access multitasking, and spooling.

***Consolidated Data Management Programming Guide* (UP-9978)**

This guide describes consolidated data management and how it moves data between peripheral devices and programs.

***Consolidated Data Management Macroinstructions Programming Guide* (UP-9979)**

This guide describes the macroinstructions used by consolidated data management, which are used in assembly language programs to define data files and manage the movement of data to and from those files.

***COBOL, 1974 American National Standard Programming Quick-Reference Guide* (UP-8613)**

This guide describes 1974 ANS COBOL for the applications programmer. It includes the COBOL character set, character strings, methods of reference, and the statements permitted in each COBOL division.

***Installation Guide* (UP-8839)**

This guide provides the system administrator with the information and procedures he or she needs to install, tailor, and maintain OS/3 software in a System 80 environment.

***Screen Format Services Technical Overview* (UP-9977)**

This overview describes how you can move information easily through your computer system using the OS/3 screen format services.

***Menu Services Technical Overview* (UP-9317)**

This guide describes the OS/3 menus and explains how to use them with assembly COBOL, RPG II, and FORTRAN IV™ programs.

***Dialog Processor User Guide/Programmer Reference* (UP-8858)**

This guide describes the dialog processor, which is the interface between the dialog (written in dialog specification language) and the application program using the dialog.

FORTRAN IV is a trademark of SuperSoft Associations.

Distributed Data Processing Programming Guide **(UP-8811)**

This guide describes distributed data processing and the corresponding products, including the DDP transfer facility and the DDP file access.

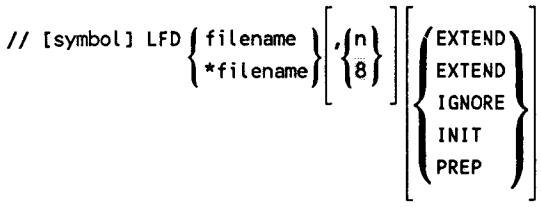
System Operations Quick-Reference Guide **(UP-9985)**

This quick-reference guide summarizes the procedures and commands used by OS/3.

Statement Conventions

The conventions used to illustrate the coding formats presented in this manual are:

- Information that must be coded exactly as shown is presented in uppercase letters. Commas, equals signs, parentheses, slashes, and percent signs also must be coded as shown.
- Information that can take various forms is presented in lowercase letters.
- Unless otherwise indicated, all parameters are positional parameters and must be coded in the order specified in the operand field and separated by commas. When a positional parameter is omitted, its associated comma must be retained to indicate the omission, except in the case of omitted trailing parameters. For example, the control statement:



could be coded in any of the following ways:

```
// LFD DISKIN,1,INIT
// LFD DISKIN,,PREP
// LFD DISKIN
```

- Keyword parameters, which take the form

```
KEYWORD=variable
```

can be coded in any sequence following the last positional parameter. Keyword parameters of the type

```
KEYWORD
```

can be coded in random sequence only when it is indicated in the text accompanying the control statement; otherwise, they must be coded as positional parameters.

- A job control statement may consist of a group of positional parameters followed by a keyword parameter (as the last parameter).

Example

```
//[symbol] EXEC program-name[,{library-name}][,[+]switch-priority][,ABNORM=label]
                             {$Y$RUN      }
                             {$Y$LOD      }
```

Since the last parameter is a keyword (not the last positional) parameter, this statement may be written as follows:

```
// EXEC program-name,ABNORM=label
// EXEC program-name,library-name,ABNORM=label
```

Commas for the omitted positional parameters may be retained if desired. For example:

```
// EXEC program-name,,,ABNORM=label
// EXEC program-name,library name,,ABNORM=label
```

The conventions for coding commas when a positional parameter is omitted and subsequent positional parameters are being specified still apply. When the second positional parameter is omitted, for example, the EXEC statement must be coded as follows:

```
// EXEC program-name,,switch-priority,ABNORM=label
```

- Information that may be coded optionally is enclosed in brackets:

```
[optional
 information]
```

- When more than one option exists for any given parameter, the options are listed within braces:

```
{option-1
 option-2
 option-n}
```

- Parameters enclosed in parentheses are called subparameters and are subject to the same rules as parameters. The parentheses enclosing subparameters must be coded to delimit a series of one or more subparameters.

- An ellipsis (. . .) is used to indicate the omission of a variable number of similar parameters:

```
lun-2,type-code,...,lun-5,type-code
```

- Optional parameters that have a default specification are shaded:

```
[{library-name}
 {$Y$RUN      }]
```

- When a portion of a parameter is underlined, only that portion need be specified. For example:

```
FORMNAME=symbol
```

can be coded as

```
FO=symbol
```

Contents

Section 3. Job Control Procs

Section 4. Proc Definition Statements

Appendix A. Logical Unit Number Assignment Information

Appendix B. Extent Specification Information

Figures

Tables

Section 1
General Information

Job Control Overview

Job control is one of the two components of the Unisys Operating System/3 (OS/3) executive; the other component is the supervisor. Job control consists of a series of transient routines and is activated by a job control statement or an operator command input from the system console. Using these transient routines, job control manages system resources and prepares a job for execution.

Specifically, job control

- Analyzes the input control stream.
- Expands job control procedure calls.
- Allocates peripheral devices and main storage.
- Schedules jobs for execution.

Job control directs and controls the flow of jobs through the system. The environment of a job and its various job steps are defined and controlled by job control statements in a control stream. As the control stream is processed, job control calls on transient routines to perform the required functions. A control stream may be read into the job control stream library (YJCS), or an alternate SAT library file for permanent storage, or into a temporary job run library file (YRUN) for scheduling and execution. Jobs are scheduled for execution based on a priority scheme and the availability of system resources. If the required resources are available for a job in the highest priority queue, the job is executed. A job is executed one job step at a time until the job is completed. Housekeeping is performed as part of the job termination process.

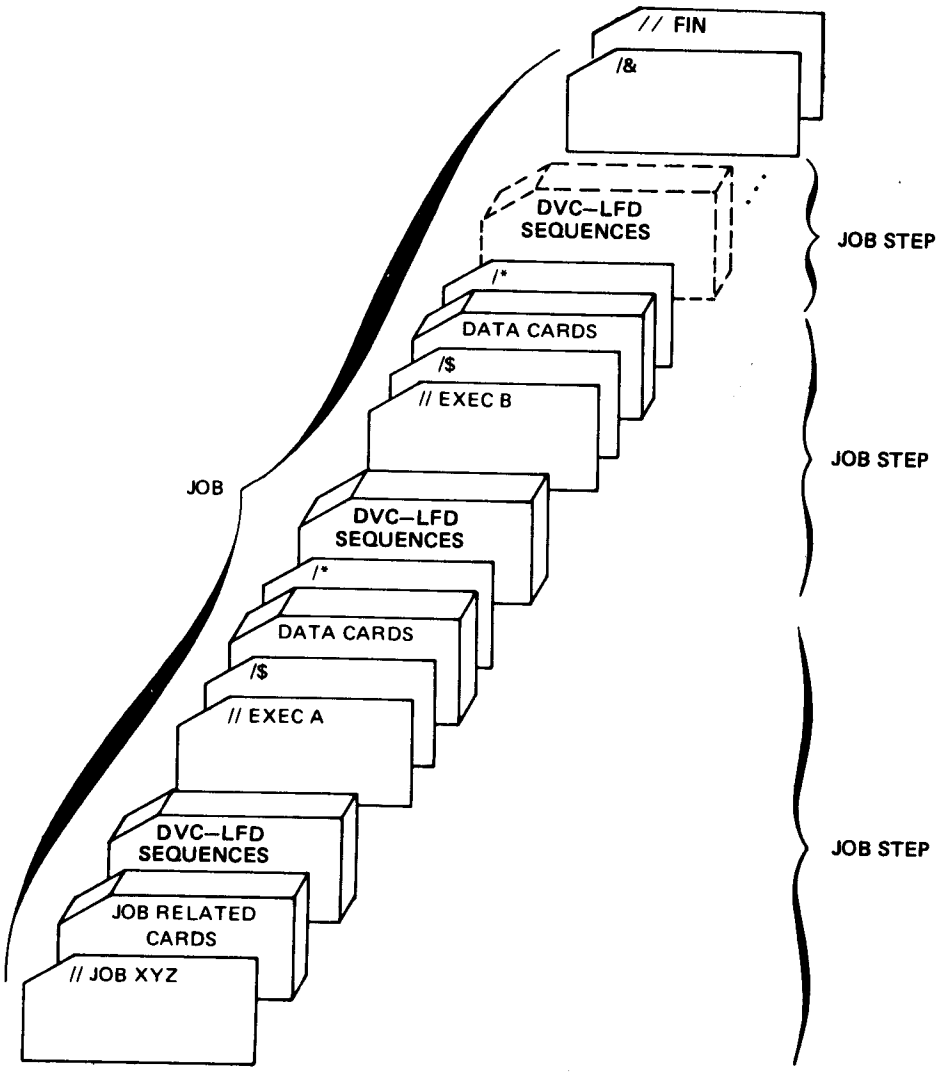
Control Stream Concepts

A control stream is a series of job control statements and job control procedures (jprocs) that define the job and direct its execution (Figure 1-1). It acts as an interface between the job and the operating system. It may also include source code or data as required by the job.

One control stream is applicable to only one job; each job is subdivided into one or more steps and a temporary YRUN file is created for each job. Any number of jobs, however, may be stored on disk for subsequent scheduling. From 7 to 47 jobs can be active concurrently, depending on the availability of system resources and the configuration of your system. Each job and, therefore, each control stream stored on disk must be given a unique name.

Job control assigns a job number to every active job and symbiont for identification purposes. When the job number reaches the maximum of 9999, job control resets it to 20.

Figure 1-1 illustrates a job control stream for a job divided into several job steps.



Note: Devices allocated (through DVC-LFD sequences) for one job step are available to all succeeding job steps.

Figure 1-1. Typical Control Stream

Job Control Statements

Each job control statement is described in alphabetical order in Section 2. The parameters associated with a statement are described in the order of their appearance when the statement format is scanned from left to right and from top to bottom within a parameter. If a default condition for a multiple-option parameter is available, it is shaded.

General Format

A job control statement has a maximum of five fields, which appear in the following order:

- Indication field

  Distinguishes job control statements from data. It is required and begins with //, /&, /$, or /*.

- Label field

  Contains an alphanumeric character symbol, of which the first character must be alphabetic. Unless this field is explained in a specific control statement, it is the target address of a SKIP control statement or the ABNORM keyword parameter of the EXEC statement. This field is not separated from the indication field by a space; it immediately follows the //.

- Operation field

  Contains the name of the function to be performed. It is required for all job control statements having an indication field of // and must be preceded by a blank.

- Operand field

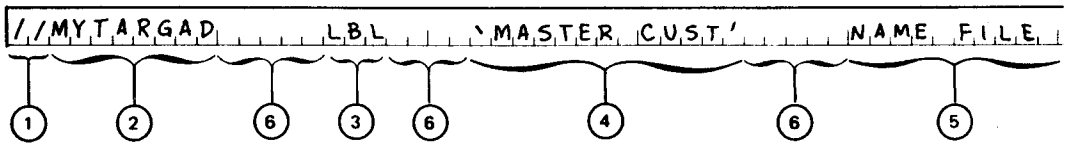
  Contains the specific information concerning the items upon which a job control function is to operate or the manner in which the function is to be performed. This field must be preceded by a blank.

- Comments field

  Contains any descriptive information desired, but not processed. The field must not contain a slash character. For those job control statements in which an operand is not permitted, such as the FIN control statement, all information beyond the operation field is treated as the comments field. This field must be preceded by a blank.

Excluding the indication and label fields, consecutive fields must be separated by one or more spaces. A space may not appear in a field except within apostrophes (hexadecimal code 7D) or parentheses in an operand field.

Example



1. Indication field
2. Label field
3. Operation field
4. Operand field
   Note that spaces are allowable due to the use of apostrophes
5. Comments field
6. Field separation spaces

Coding Conventions

- Data cannot be contained in a job control statement.
- Embedded data is normally assumed to be 80 characters long; input from diskette can be 80 or 128 characters long.
- Comments must not contain a slash.
- Job control does not scan past position 72; however, embedded data of up to 128 bytes is passed through. Positions 72 through 80 may also be used for a continuation character and sequencing.
- For a multiple-statement card:
  - Each statement, except the last, is terminated by one or more blanks followed by the first slash of the next statement.
  - Statements are processed from columns 1 through 71. Column 72 is used to indicate continuation.
  - CR and procname must be the last statement on the card.
  - Any statement may be continued at any delimiter, such as the blanks following the operation or operand fields or after the comma following a parameter in the operand field. A nonblank character must be coded in column 72.

- JOB, FIN, PROC, NAME, or END must be the only statement on a card.
- /$, /*, or /& must be the only statement on a card.
- Numbers required for particular parameters can be expressed in decimal or hexadecimal form. Numbers preceded by D and enclosed in single quotes are considered decimal. Numbers preceded by X and enclosed in single quotes are considered hexadecimal. (A trailing quote may be specified optionally.) Both of the following represent the same value:

  X'FF'

  D'255'

  Numbers not preceded by X or D are automatically considered decimal except in the following cases when they default to hexadecimal:

  - Main storage sizes specified on the JOB statement (*min* and *max* parameters)
  - Main storage sizes specified on the OPTION MIN and OPTION MAX job control statements
  - Absolute disk addresses specified on the EXT statement (*addr* or *Tccc:hh parameters)*
  - Address on the ALTER statement (*address* parameter)
  - Expansion limit on the SFT statement's DLOAD option (*expansion-limit* parameter)
- Character strings on the ALTER, LCB, and SET job control statements must be specified as shown in their formats.

Statement Continuation

A continuation line is not considered a statement in itself. It is a line that contains the continuation of the immediately preceding control statement. A nonblank character must appear in column 72 of each statement being continued. Continuation statements may be used to extend any job control statement for which at least the first two fields are already coded.

A continuation line must begin with either the three-character sequence //n, or just a simple //, separated by one or more blanks from the continuation portion of the control statement. The general format of the continuation statement is:

```
//[n]p1,...,pn
```

where:

n
: Is a decimal number from 1 through 9. The numbers do not need to be consecutive; however, each successive number must be greater than or equal to the preceding number within a specific group of continuation statements. This is an optional field and may be left blank, or numbers may be used so the user can keep a visual record of the amount of continuation statements used.

p1,...,pn
: Are the parameters required to continue the immediately preceding statement.

Any statement may be continued at any delimiter. The delimiters are the blanks following the operation or operand fields, or after the comma following a parameter in the operand field. Subparameters may not be divided.

Job Control Procs

The job control procs supplied as part of OS/3 are described in alphabetical order in Section 3. The parameter descriptions are presented in the same order specified for the job control statements. The coding formats of these procs are illustrated in the proc descriptions, using the statement conventions previously described.

Proc Definition Statements

A job control procedure definition consists of a PROC definition statement, one or more NAME definition statements, and a series of job control statements; it ends with an END definition statement. The PROC definition statement is used to signal the beginning of the procedure, the NAME definition statement declares labels by which the procedure can be called for execution, and the END definition statement signals the end of the procedure. Each time a series of job control statements is needed, a procedure call statement is written; job control then inserts the necessary number of statements at the point of reference. The procedure definition specifies the coding and statements for a particular operation to job control, and the procedure call statement specifies the values of the variable parameters to be used when the call is executed.

Coding Conventions

Statements and comments used in procedure definitions are generally written in columns 1 through 71. Column 72 is used to indicate continuation. Columns 73 through 80 may contain program identification and sequencing information; job control does not process these columns.

The job control statements within the body of the procedure definition follow standard job control statement conventions with respect to multiple-statement cards and continuation. If multiple-statement images are present in the procedure definition, they are expanded and passed to the run symbiont one statement at a time. If job control statements with continuation appear in the body of the procedure definition, they are expanded and passed to the run symbiont in the same form with continuation. Job control does not attempt to verify whether continuation is allowed on the particular control statement.

Character Set

The character set used in writing procedure definition statements consists of:

Letters	A through Z
Special letters	? $ # v
Digits	0 through 9
Special characters	+ - * / , = ' ^ ( ) . > < & w : ;

Terms

The values that may be used in the operand field of a procedure definition statement may be a symbol or a character string.

- Symbols

  A symbol consists of a group of up to eight alphanumeric characters used for parameter identification and as labels. The first, or leftmost, character must be alphabetical. Special characters or blanks may not be contained within a symbol.

  The label on a NAME directive cannot exceed a length of eight characters. The operand in a NAME directive may be obtained by referencing the symbol p(0), where p is the symbol assigned in the PROC directive used to reference any positional parameter in the definition, and (0) is the number of the positional parameter. The parameter in the operand of the NAME directive is addressed as positional parameter 0.

- Character string

  A character string can represent up to 252 valid characters, all of which must be printable. Character strings containing embedded blanks or commas must be enclosed in apostrophes or parentheses, which are considered part of the character string. A character string may not contain an embedded apostrophe.

  A null character string is represented by two consecutive apostrophes.

  All parameter values are evaluated as character strings.

Parameters

Parameters pass information from the procedure call statement, through the procedure definition, to the body of the procedure. Parameters may be equated to values, symbols, or character strings and may be used to specify label information, file names, volume serial numbers, etc. There are two types of parameters: positional and keyword. Positional parameters are identified by their position within the operand field of the procedure call statement; keyword parameters are referenced or identified by the symbols assigned to them in the directive.

Both positional and keyword parameters may be sublisted. Thus, each operand of the procedure call statement may represent one value or a series of values that may be referenced independently. When a parameter is sublisted, the subparameters must be separated by commas and the entire list enclosed in parentheses.

An omitted positional parameter has the value of a null character string. An omitted keyword parameter that had a preset value specified in the PROC directive is given that preset value. An omitted keyword subparameter that had a preset value specified in the PROC directive is not given that preset value. If no preset value was given, the value of the keyword parameter is a null character string.

Parameter Referencing

Job control statements coded in the body of a procedure definition follow the same rules and conventions as all other job control statements, with one exception. The parameters in the operand field of the job control statements that require substitute values at job execution time must begin with an ampersand (&). The positional parameters in the operand field of the job control statements, whose values will be changed when the procedure definition is called, must have some sort of symbol included in the parameter specification. For example, in the body of a procedure definition, if you had a DVC statement in which you want to vary the logical unit number, it would be coded:

```
// DVC &P(2)
```

The ampersand (&) is the indicator; the P is the symbol you assigned in the PROC directive; the (2) indicates that the logical unit number to be inserted in the control statement is coded as positional parameter 2 in the procedure call statement. The parentheses are required.

Parameter substitution is performed at job execution time by the run symbiont when it expands the procedure definition.

Substitution is attempted only on your embedded procedure call statements. PROC or NAME directives are not candidates for substitution.

For each valid character string following a single ampersand, a substitution is made. If a valid substitution exists, it is used; if no valid substitution exists, the null character string is substituted.

The ampersand, period, parentheses, apostrophe, blank (delta), comma, plus, minus, asterisk, and slash (& . ( ) ' ^ , + - * /) may not be embedded in the dummy arguments on the PROC directive.

Any statement can be continued only between operands or between the command and the first operand. This limits the length of operands on the input source (primary control stream or procedure file) to 65 characters.

The length of a single operand may not exceed 252 characters.

The length of a single parameter is limited to 242 characters. For positional parameters, this is the value field; for keyword parameters, this is the value plus the key length. If a parameter is sublisted, the maximum length is decreased by 2 for each element of the sublist.

Control Stream Considerations

Procedure definitions are filed by the file symbiont in either YJCS or an alternate library file. The new values to be used are not substituted for the preset values until the procedure definition is called. Substitution takes place then and the procedure definition is expanded. A procedure definition may be called as often as necessary or until the procedure definition is deleted from YJCS or the alternate library file.

A job input directly from the card reader may include procedure definitions in its control stream. Procedure definitions must appear in the control stream before any references are made to them. They may not be embedded in a data sequence (statements /$ through /*).

A procedure definition filed in YJCS or an alternate library file may be overridden by reading in a new version from the card reader, a diskette, or the system spool file.

System Library File Names

The system library files for OS/3 are called by their file identifiers. These allow a program to access a file via its symbolic name. In this way, a file is not device dependent, and the programmer is not hampered by physical device restrictions in the program specifications. The file identifiers used for the system library files are:

- Load library file YLOD
- Object library file YOBJ
- Source library file YSRC
- Macro library file YMAC
- Job control stream library file YJCS

A temporary job run library (YRUN jobname) is also established for each job input to the system. These files exist only for the duration of the job, unless the job's YRUN file is saved through the OPTION SAVE or OPTION NOSCHED job control statement.

Section 2
Job Control Statements

ALTER

Function

Alters load modules at execution time. ALTER control statements must precede the EXEC control statement in the job step. ALTER control statements are processed by the supervisor prior to the transfer of control to the user program.

Format

```
//[symbol] ALTER [phase-name][,address][,change] [,{RESET}]
                                                 [  {ORG  }]
```

Parameters

phase-name

One to eight alphanumeric characters specifying the name of the phase to be altered. If less than eight characters are specified, the phase being altered is assumed to be an alias phase and the name is padded on the right with EBCDIC blanks to make eight characters. The names of all other phases must be specified by using the full eight characters.

If omitted, the previously named phase is altered.

address

A 1- to 5-digit address that specifies the main storage location into which the bytes specified by positional parameter 3 are to be stored; considered hexadecimal if coded *X'number'* or *number*, and decimal if coded *D'number'*. If positional parameter 4 is *RESET*, the address may be omitted.

If omitted and an address is required, zero is used.

change
Specifies the contents of the bytes to be stored at the address specified by positional parameter 2. Changes are not stored if the address is invalid. The change must be specified in hexadecimal or EBCDIC.

- Hexadecimal

  ```
  X'cccccccc...' or cccccccc...
  ```

  The number of characters must be even, and the maximum number of hexadecimal characters allowed is 16 (eight bytes).

- EBCDIC

  ```
  C'cccccccc'
  ```

  The maximum number of EBCDIC characters allowed is 8 (eight bytes).

If positional parameters 2 and 3 are omitted, this is a null ALTER control statement, and control is returned to the user program.

If positional parameter 4 is used, change may be omitted.

RESET
Resets the alter mode indicator. If all other parameters are omitted, control returns to the user program with the alter mode indicator reset.

ORG
Adds the address specified in positional parameter 2 to all addresses on succeeding ALTER control statements until the next ALTER control statement with a RESET parameter is encountered or the final ALTER control statement for the phase is processed.

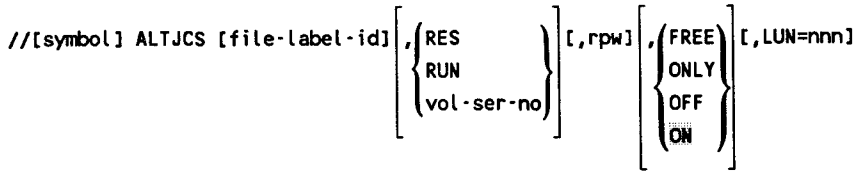
If *RESET* or *ORG* is not specified, the control stream is read until an ALTER control statement with the *RESET* or *ORG* parameter is detected or the final ALTER control statement for the phase is processed.

ALTJCS

Function

Specifies an alternate SAT library file (one other than YJCS) to be searched for jprocs. The alternate library file can be located on format-label diskette as well as disk.

Format



Parameters

`file-label-id`

A 1- to 44-alphanumeric character label name of the file to be searched for jprocs; it is optional only if you're not activating the search of a new library but changing the last parameter *(FREE, ONLY, OFF,* or *ON)* for an alternate library already defined in a previous ALTJCS statement. If the *file-label-id* is not specified, then *vol-ser-no* and *rpw* cannot be specified.

`vol-ser-no`

Specifies the volume serial number of the file to be searched for jprocs. If a *vol-ser-no* is not specified, the cataloged *vol-ser-no* is used; if it is not cataloged, *RES* is used.

`rpw`

Specifies the read password of the specified cataloged file that contains the jprocs needed for the job.

`FREE`

Specifies that only YJCS is to be searched and frees the alternate device (from the run processor) after the search is completed.

`ONLY`

Specifies that only the identified alternate library file is to be searched.

OFF

Specifies that only YJCS is to be searched. You specify this option if you no longer want an alternate library file searched for jprocs. The alternate library file remains open to the run processor and can be searched again by using the *ON* or *ONLY* options.

ON

Specifies that the identified alternate library file is to be searched first and then YJCS is to be searched; ON is the default order-of-search option.

LUN=nnn

Supplies a logical unit number indicating the device type and characteristics for the alternate library; never specified unless a volume serial number is also specified; helps to determine whether a disk or format-label diskette is required (since volume serial numbers for disk and for format-label diskette are syntactically the same).

Note: *If your job control stream is in an alternate library, you cannot use the / / ALTJCS statement to specify a different library. You can use it only to specify the options* FREE, ONLY, OFF, *or* ON. *(If the job stream is in YJCS, the / / ALTJCS statement can reference any alternate library.)*

CAT

Function

Catalogs the file defined in a previously encountered device assignment set in the job. The file is identified by the lfdname parameter in this statement and in the LFD job control statement. The read/write passwords are specified in the LBL job control statement.

Format

```
//[symbol] CAT lfdname[,catpw][,SCR][,{GEN=nn}]
                                     [{MEM  }]
```

Parameters

`lfdname`
File name which must agree with the file name of the associated LFD job control statement.

`catpw`
Password of one to six alphanumeric characters for the catalog itself; must be specified if the catalog itself is password protected.

Note: *Even if all the files in a file catalog have been decataloged (using // DECAT), the catalog password remains in effect. Catalog passwords are established, removed, and changed through the JC$CAT routine, which is explained in the* File Cataloging Technical Overview, *UP-9980. When you use // CAT or // DECAT, you must specify* catpw *if a password has been established through the* JC$CAT *routine.*

`SCR`
Indicates the file should be scratched when it is removed from the catalog. This parameter is effective only for specific references. (If several files are removed from the catalog as a result of the specification of a nonterminal node point, this parameter is not effective.)

`GEN=nn`
Number of generations to maintain for the file.

MEM

Used to fill gaps between members of a generation file. A full device assignment set must be used with the CAT statement if MEM is specified.

Note: *A CAT statement with MEM specified, and a DECAT statement cannot be used against the same member of a generation file in the same job.*

CC

Function

Allows you to issue OS/3 system commands from within a job control stream. Parallels the functions of OS/3 system console and workstation commands.

Format

```
//[symbol] CC {command                  }
             {'command and parameters'}
```

Parameters

`command`
A single system console or workstation command.

`'command and parameters'`
System console or workstation command and associated parameters; total number of characters enclosed in quotes cannot exceed 60.

Notes:

1. *When the command string contains no blanks (other than the blank separating the command from its first parameter), you can precede the first parameter with a comma instead of enclosing the command and its parameters in quotes. For example:*

   `// CC BE,JOB1` *instead of* `// CC 'BE JOB1'`

2. *The following system console commands may not be specified in the CC job control statement: MIX, SWITCH, AVR, REBUILD, SHUTDOWN, SYSDUMP, and all SET commands.*

3. *Unsolicited input messages (see the* Interactive Services Operating Guide, *UP-9972) and / / PAUSE responses cannot be specified in the / / CC statement.*

See the Operations Guide, *UP-8859, for a description of the system console commands. See the* Interactive Services Operating Guide, *UP-9972, for a description of the workstation commands.*

CR

Function

Temporarily inserts embedded data or other job control statements from cards, data-set-label diskette, or spool file into a stored control stream. This statement can be used in control streams or procedures filed in YJCS or an alternate library. The CR control statement is placed in the control stream at the point where data is to be inserted. The data from the input reader is merged with the filed control stream, and the combined control stream is examined by the job control routines. The data read from the card reader must be terminated by a FIN control statement.

Any number of CR control statements may appear in a control stream, provided the FIN control statement for a previous CR statement is encountered before the current CR control statement.

Format

```
//[symbol] CR
```

The CR statement contains no parameters.

DATA FILEID

Function

A control statement used by the input reader. It identifies and precedes any input card data that's going to be spooled as the result of the IN operator command.

Format

```
// DATA FILEID=file-identifier[,RETAIN][,IGNORE]
```

Parameters

`FILEID=file-identifier`
File identifier. Agrees with either the *file-identifier* of the LBL control statement for the file or a concatenation of the *jobname* from the JOB control statement and the *filename* from the LFD control statement.

`RETAIN`
Used to maintain the spool file after the job has processed the file. The only means of deleting the reader file after RETAIN has been specified is by issuing the operator console command DE SPL,RDR.

`IGNORE`
Specifies that // RUN/RV statements in the card deck that are to be spooled are to be processed as data cards.

Note: *If the LBL control statement is used in the device assignment set for a spooled card file, then the file-identifier of the DATA control statement must be the same as the file-identifier of the LBL control statement.*

DATA STEP

Function

A control statement that identifies replacement embedded data for a saved and translated job control stream. It must precede the replacement data submitted from a card reader, a data-set-label diskette, or a input spool file.

Format

```
// DATA STEP=nnn
```

Parameters

nnn
: A decimal number in the range 1 to 255 that specifies the number of the job step in the job for which you are submitting new embedded data.

Notes:

1. *A DATA STEP statement must be submitted for each job step that contains embedded data you want to replace. You must replace the old data sets in the specified job step with an equal number of new data sets.*

2. *Because the replacement embedded data is submitted to the saved translated stream from an input device, you must use the SI command or the // CC SI job control statement to initiate the job stream run.*

3. *The embedded data is replaced for that job run only.*

DD

Function

Provides a way to change certain data definitions at run time. The DD statement must appear within the DVC-LFD sequence. If the file is a cataloged file, it must follow the LBL statement.

```
//[symbol] DD [RCFM=(FIXBLK|FIXUNB|UNDEF|VARBLK|VARUNB)] [,BKSZ=n][,RCSZ=n][,SIZE=AUTO][SIZEn=n]

[,(KLEN|KLENn)=n][,(KLOC|KLOCn)=n][,INDS=n]

[,ACCESS=(EXC|EXCR|SRDO|SRD|SADD|UCP)][,REWIND=(NORWD|UNLOAD)]

[,OPRW=NORWD][,CLRW=(NORWD|RWD)][,FILABL=(NO|NSTD|STD)]

[,TPMARK=NO][,RECV=(YES|LOAD|NO|FCE|OFF)][,VSEC=(YES|n)][,VMNT=(ONE|NO)][,DMRECV=YES]

[,RCB=(NO|YES)][,OFFSET=1][,RESTORE=(n|YES)][,CACHE=(NO|YES)][,KEYREF=m]

[,MSGSUPP=(DM36|LB05|ALL)]
```

Parameters

The *n* that is suffixed to the *KLEN* and *KLOC* keywords refers to KEYn of a multikey MIRAM disk file. The *n* that is suffixed to the *SIZE* keyword refers to the partition identifier of a multikey MIRAM disk file. The definitions for keyword parameters that are associated with each file type are found in the *Consolidated Data Management Programming Guide*, UP-9978. Definitions for the keyword parameters associated with SAT files are found in the *Supervisor Macroinstructions Programming Reference Manual*, UP-8832. Tables in the *Job Control Programming Guide*, UP-9986, provide a summary of the allowable keywords for each file type.

The SYSGEN parameter *DMRECV=YES* acts the same as *RECV=YES*. Refer to the *Installation Guide*, UP-8839, the *Consolidated Data Management Programming Guide*, UP-9978, and the *Systems Operations Quick-Reference Guide*, UP-9985, for more information.

The *RESTORE=* parameter allows you to restore a MIRAM file that was accidentally initialized. *n* specifies the number of records that data management is to use for file retrieval. *YES* means to use this parameter only to read the data partition of the file (PCA1) for the purpose of recreating the file with a copy program such as DATA or MILOAD. You cannot access the index partition (PCA2).

The *MSGSUPP* parameter optionally allows you to suppress the display of the DM36 DUPLICATE RECORD error message, the LB05 MODULE NOT FOUND error message, or all error messages from the console or log.

DECAT

Function

Removes a file from the catalog. The file is defined by the lfdname parameter in this statement and in the LFD job control statement.

Format

```
//[symbol] DECAT lfdname[,catpw][,SCR][,{GEN}]
                                       {ROL}
```

Parameters

`lfdname`
File name, which must agree with the file name of the associated LFD job control statement.

`catpw`
Password of from one to six alphanumeric characters in length for the catalog itself; it must be specified if the catalog itself is password protected.

Note: *Even if all the files in a file catalog have been decataloged (using // DECAT), the catalog password remains in effect. Catalog passwords are established, removed, and changed through the JC$CAT routine, which is explained in the* File Cataloging Technical Overview, *UP-9982. When you use // CAT or // DECAT, you must specify catpw if a password has been established through the JC$CAT routine.*

`SCR`
Scratches the file when it's removed from the catalog. (See the CAT job control statement.)

`GEN`
Removes all generations of the file from the catalog.

`ROL`
Used to reset a generation file's status to its condition *before* the current generation was added. *ROL* (rollback) removes the current member of a generation file, making the previous member current, and inserts the rolled-back file as the oldest member of the generation file.

Note: ROL *can only be used against the current member of a generation file, and it can only apply once to the current member within a single job.*

DST

Function

Sends spooled out (print or punch) to RBP (remote batch processing) terminals. This statement must appear in the device assignment set for the print or punch file.

Format

```
//[symbol] DST dest-1[,dest-2,...,dest-16]
```

Parameters

`dest-n`
Destination identifier of one to six alphanumeric charactersin length; it is defined by RBP. Specify OS3CTR or CENTRAL to specify the local site's control printer.

Note: *Although the DST statement can be used in DDP-initiated jobs, DDP or auxiliary printer output (specified by // ROUTE) and RBP output (specified by // DST) cannot be mixed for any one job. For any job, all output must be of one type or the other.*

DVC

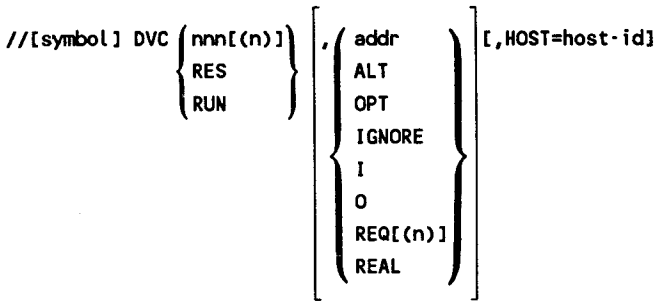
Function

Requests the assignment of a peripheral device to a job. The device must be available before the job can be scheduled and is allocated by job control at job step time. Specific devices can be requested and assigned.

This statement must be the first in a device assignment set, and one set is required for each file. A device assignment set consists of the DVC, VOL, EXT, LBL, and LFD statements. At least one DVC statement and one LFD statement are required for each file accessed by the user program. The EXT and LBL statements are optional. The VOL statement is required for tape and disk files. For catalog files, the EXT statement must follow the LBL statement.

The maximum number of unique devices allowed in a job is 255. The maximum number of unit record devices (e.g., card readers, data-set-label diskettes, printers) allowed in one job is 42.

Format



Parameters

nnn[(n)]

Decimal logical unit number specifying a device type and features from the system's logical unit table (Table A-1). When a job is scheduled, each logical unit number is assigned to a specific device. The device type and features for a logical unit number may be changed with the EQU job control statement.

When the device is a workstation, *(n)* is used to specify the number of workstations that can access a single workstation file. A maximum of 255 workstations of the type specified by *nnn* can be associated with one file.

Notes:

1. *If more than one volume is assigned the same logical unit number, the operator may be required to demount the first volume and then mount the next volume on the same device. All volumes but the last are unshareable with other jobs because they must be demounted.*

2. *If a single volume is assigned more than one logical unit number in different job steps, the operator may be required to demount the volume from the device assigned to the first logical unit number and then mount the volume on the device assigned as the next logical unit number. The volume is not shareable with other jobs until it is mounted on the last device.*

3. *Before a mount is requested, the system tests to determine if the required volumes are mounted on other devices.*

4. *You cannot use logical unit numbers 50 through 59 on the DVC statement you use to catalog a disk file. These logical unit numbers specify "any type" of disk. To ensure that cataloged files obtain a valid device when a job is scheduled, you must use logical unit numbers that specify a particular kind of device. For example, you could specify / / DVC 64 to indicate that an 8416 disk is to be used.*

RES

Use the system-resident device. The VOL statement is not required when RES is specified and will be ignored if it is present.

RUN

Use the device containing the job's YRUN file. The VOL statement is not required when RUN is specified and will be ignored if it is present.

addr

Physical address of the device, in hexadecimal. This represents the channel number, the control unit address, and the device number. The use of this parameter inhibits dynamic allocation of devices.

Notes:

1. *The* addr *parameter may be used to specify a real device in a spooling system. Real devices and virtual devices (of the same type) may not be assigned in the same job.*

2. *The* addr *parameter of the DVC job control statement was not designed for tape, disk, or diskette, but will function correctly if the specified volume is not premounted on another unit. Also, the specified unit cannot be premounted with any volume specified in the same job stream.*

ALT

same type and features may be assigned (within a single job step). If an alternate device is not available, the job will execute with a single device.

The ALT parameter of the DVC job control statement does not work correctly if it is used more than once in a job stream. A separate drive is allocated for each ALT, and if there are insufficient drives to accommodate all of the ALTs, only one drive is allocated even if two drives are available. If the ALT function is needed more than once in a job stream, the following job control statements can be used:

```
// DVC 90 // VOL A
// DVC 91 // VOL B.
```

OPT

Requested device is not essential to the running of the job.

IGNORE

More than one logical file may be assigned to a physical device. Use this parameter when different files are used on one device in a serial manner by a job step.

Note: *When the IGNORE parameter is specified in the DVC statement, the logical unit number and the volume specified in the VOL statement must stay the same if repeated in the job stream. The following job control is incorrect.*

```
// DVC 90,IGNORE // VOL A LFD A // EXEC A

// DVC 91 // VOL A // LFD B
```

DVC 91 should be DVC 90.

I

Indicates input spooling was configured for the diskettes at SYSGEN time.

O

Indicates output spooling was configured for the diskettes at SYSGEN time.

REQ[(n)]

Used for workstations. When a number is specified for (n), this parameter indicates how many workstations specified in *nnn[(n)]* must be connected before the job is run. If *(n)* is omitted in this parameter, all workstations specified in *nnn[(n)]* are required.

REAL

Used to assign a real device to a job, without having to specify the device's physical address. This parameter can be used to bypass spooling.

HOST=host-id

Indicates that a disk file is located at a remote DDP host. The host-id is from one to four alphanumeric characters in length and identical to the label-id of the LOCAP macroinstruction in your ICAM network. You can specify $HOST instead of a host-id to indicate a remote originator (the host that initiated the job). The specified host-id or $HOST must always be for a remote host; otherwise, a data management error will result.

Notes:

1. *The logical unit number of a DVC statement associated with a VOL SCRATCH statement must be unique in the job stream. A VOL SCRATCH statement cannot be specified as follows:*

```
// DVC 90 // VOL X // LFD X // EXEC A

// DVC 90 // VOL SCRATCH // LFD Y
```

The NOV parameter can be used in place of VOL SCRATCH.

The following is permitted:

```
// DVC 90 // VOL SCRATCH // LFD X // EXEC A

// DVC 90 // VOL X // LFD Y
```

Once the unique logical unit number 90 is specified for VOL SCRATCH, 90 can be used in subsequent steps.

2. *The logical unit number in the DVC statement and the volume in the VOL statement must stay the same if repeated in the job stream. The following job control statement is incorrect:*

```
// DVC 50 // VOL PACK1 // LFD A // EXEC PROGRAM

// DVC 51 // VOL PACK1 // LFD B
```

This job requires two disk drives to be allocated but the volume PACK1 is declared nonshareable. DVC 51 should be DVC 50.

In situations where a job sits on the job queue for a long period of time for no apparent reason, the DVC and VOL statements for the job should be checked for the condition shown in the preceding example.

The job control stream for a job cannot specify more than 44f unit record devices, that is, printers, readers, and punches.

DVC PROG

Function

Allows a BAL program (using consolidated data management macros) to communicate with another BAL program via DDP's program-to-program facility; it can be specified only once in any single job step. Used in place of // DVC when the device assignment set is for a program-to-program type of file; the device assignment set may include an LBL statement but must include an LFD statement.

Format

```
//[symbol] DVC PROG [,jobname][,HOST=host-id]
```

Parameters

jobname

Any 1- to 6-character job name. Identifies the other participant in the program-to-program communication. (When specified in // DVC PROG for the primary, for example, it identifies the surrogate. When specified in // DVC PROG for the surrogate, it identifies the primary.) This parameter is required in the // DVC PROG statement for the primary but is optional for the surrogate.

HOST=host-id

Specifies a particular host in a DDP network. The host-id is one to four alphanumeric characters long and is identical to the label-id of the LOCAP macroinstruction in your ICAM network. Use $HOST to indicate the originator (the host that initiated the job).

EQU

Function

Equates a logical unit number specified in the control stream with a device type in your system. This enables you to run on a system other than the one the control stream was designed and generated for. The EQU statement must be used to assign additional logical unit numbers to virtual readers, printers, and punches.

The EQU statement must precede the device assignment set in the control stream that references the devices and is effective for the entire job.

Format

```
//[symbol] EQU lun-1,type-1[,lun-2,type-2,...,lun-n,type-n]
```

Parameters

`lun-n`
Logical unit number specified in the control stream. The valid logical unit numbers are shown in Appendix A.

`type-n`
Code of four to eight hexadecimal characters that provide the exact device characteristics (Appendix A).

EXEC

Function

Identifies the program to be executed. The EXEC control statement is a job step delimiter and is the last control statement processed by job control *before* the execution of the program named in the statement. Any PARAM control statements included in the control stream must immediately follow an EXEC statement.

The EXEC control statement loads the specific program from either YLOD, the job's YRUN file, or from an alternate load library previously defined by a device assignment set.

Before actual program loading, job control completes any pending tape and disk mounting requests. The specified program is then loaded, overlaying job control, and receives control.

Format

```
//[symbol] EXEC program-name[,{library-name}][,[+]switch-priority][,ABNORM=label]
                             {$Y$RUN     }
                             {$Y$LOD     }
```

Parameters

`program-name`

1- to 6-character name identifying the program to be executed. It must be the same as the load module name specified to, or generated by, the linkage editor. The name is left-justified in the field and zero filled to the right, if necessary.

`library-name`

LFD name of the previously defined library file (on disk) containing the load module. Use this parameter when the program must be loaded from an alternate load library. If the program cannot be found, YLOD and the job's YRUN file, in that order, are searched.

`$Y$RUN`

Load user program from the job's YRUN file. If the program cannot be found, YLOD is searched for the program.

If positional parameter 2 is omitted, YLOD is searched first and if the program cannot be found, the job's YRUN file is searched.

[±]switch-priority

Decimal number specifying the program task switching priority; used by the supervisor for dispatching control during program execution. The number specified can be an absolute value ranging from 1 to 60 or a relative value such as +3 or -3. With absolute values, the lower number represents the higher priority. (The highest priority is 1.) Relative values increment or decrement the overall absolute task switching priority set for a job via the SWITCH operator command or // OPTION PRI. A positive value (+) *decrements* the overall priority resulting in a *higher* task switching priority for that program. A negative priority *increments* the overall priority resulting in a *lower* task switching priority for the program. If *switch-priority* is omitted, the lowest priority in the system is used. Task switching is explained in the *Supervisor Macroinstructions Programming Reference Manual* (UP-8832).

Notes:

1. *Program switch list priority*

   When the step processor has given control to the program, it switches the task to a priority (if any) placed in the preamble by the supervisor. This priority is specified on a new console command option - switch to priority for rest of job. If this priority is not in the preamble, the priority passed by the RUN processor is used.

   The priority passed by the run processor is one of the following:

   - *The one explicitly specified on the EXEC statement*
   - *The one specified on a new run processor option, OPTION PRI= job priority, which will be applicable from the point of encounter to the end of job or until another such OPTION is encountered in the job stream*
   - *The system default, which currently is taken from a new field in the System Information Block (SIB)*

2. *Relative switch list priority*

   The priority specified on the / /EXEC statement is an absolute value, e.g., 3. The user specifies that the priority will change relative to the overall job priority for that job step only (e.g., +3 or -3).

ABNORM=label

Used to bypass job control statements in case the program that is named terminates abnormally. The *label* corresponds to the label specified in the optional label (symbol) field of a subsequent job control statement. Should abnormal termination occur, job control skips forward to this statement. Since ABNORM=label is a keyword parameter, it may be coded in any position, for example:

```
// EXEC MYPROG,ABNORM=ERR
```

EXT

Function

Obtains disk or diskette space. Provides the information that is needed to create new files or extend existing files on disks or diskettes.

Any number of extents may be specified on a single EXT statement or on several EXT statements. The first EXT statement in a DVC-LFD sequence applies to the first volume specified on the immediately preceding VOL statement. If multiple extents are specified for disk or format-label diskette files, the specifications given for the first four positional parameters apply to all extents specified on that statement. When allocating catalog files that require an EXT statement, // EXT must follow // LBL in the device assignment set.

See Appendix B for further information regarding extent specification.

Format 1: Disk or Format-Label Diskette

```
//[symbol] EXT {MI} [,{C  }][,{inc}][,{addr    }][,{mi     }]
             {ST}  [,{CF }]  [,{0  }]  [,{Tccc:hh}]  [,{(bi,ai)}]
                   [,{F  }]  [,{1  }]  [,{BLK   }]
                                       [,{TBLK  }]
                                       [,{CYL   }]
                                       [,{TRK   }]
                                       [,{OLD   }]

[,{mj     } ,...][,OLD][,FIX][NTERM]
  {(bj,aj)}
```

Format 2: Data-Set-Label Diskette

```
//[symbol] EXT MI,C,0,BLK,(bi,ai)[,NDI]
```

Parameters

MI

MIRAM file, only valid specification for data-set-label diskette.

ST

System access technique (SAT) file.

C

Allocates contiguous space; only valid specification for a data-set-label diskette file.

F

Formats the file at allocation time. Positional parameter 4 must be BLK.

CF

Both of the preceding.

inc

Specifies the secondary increment in cylinders by which the file is to be extended if automatic extension is required. If no EXT statement is in this file, the value of the most recently specified secondary increment for this file is used, which could be the very first time this file was used.

O

File cannot be dynamically extended; must be specified for a data-set-label diskette file.

addr

Absolute cylinder address where the file is to begin; considered hexadecimal if you code *X'number' or number;* considered decimal if you code *D'number'.* The allocation is in terms of cylinders.

Tccc:hh

Absolute track address in cylinder/head format where the file is to begin; considered hexadecimal if you code *X'number* or *number;* considered decimal if you code *D'number.* Allocation is in terms of tracks.

BLK

Allocates space in blocks; actual allocation is in terms of cylinders. This is a request for a new extent; only valid specification for data-set-label diskette files.

TBLK

Allocates space in blocks; actual allocation is in tracks.

CYL

Allocates space in cylinders.

TRK

Allocates space in tracks.

OLD

Indicates that parameter 3 is being used to change the secondary increment (the automatic allocation amount for dynamic extension) of the extent for a previously allocated file.

Cannot be followed by any other parameters even if they are specified.

`mi`

Number of cylinders or tracks allocated for this file; positional parameter 4 must be *CYL, addr, TRK,* or *Tccc:hh.*

`(bi,ai)`

Used when allocation is in terms of blocks (by cylinder or track) for disk and diskette files; positional parameter 4 must be *BLK or TBLK.*

`bi`

Is the average block length.

`ai`

Is the number of blocks.

`mj` and `(bj,aj)`

Provides the same function as *mi* and *(bi,ai)* parameters but are used only for additional extents in the file; not to be specified for data-set-label diskette files.

`OLD`

Indicates the file's previously allocated extent is to be increased by the allocation amount specified (*mi*, *(bi,ai)*, etc). If *OLD* is omitted, the request is for a new extent.

`FIX`

Indicates you're allocating the extent in the fixed-head area of an 8417 disk. Note that you can specify *FIX* and *OLD* in any order so that if you omit *OLD* there's no need to retain the comma.

`NTERM`

System alerts you if the extent cannot be allocated because of insufficient disk space or because a specified absolute disk area is already in use. Your job does not terminate. Instead, the system displays a JC48 message and waits for either a retry (R) or cancel (C) reply. This allows an operator to evaluate the files currently on the disk and to clear those that are not needed so your job can continue.

`NDI`

NDI (nondata interchange) must be specified to allocate space for all data-set-label diskettes that are not basic data exchange (BDE) diskettes. If omitted, a BDE diskette (a single-sided, single-density diskette having 128-byte sectors, 26 sectors per track, and 73 tracks) is assumed.

Notes:

1. *All the parameters (except* FIX*) that apply to disk also apply to format-label diskette.*

2. *Data files must be allocated as MIRAM files.*

FIN

Function

Terminates card reader operations. Used as a sentinel card to signal the end of card input to the run symbiont. If // FIN is used to signal the termination of card reader operations when a job is being read, it must be preceded by a /& control statement.

This statement is also used to terminate card reader operations that are initiated by a FILE operator command when complete control streams or procedure definitions are filed or by an IN operator command when card input is being spooled.

Format

```
//[symbol] FIN
```

There are no parameters associated with this statement.

FREE

Function

Releases volumes and peripheral devices assigned to a job. Released volumes and devices, including related alternate devices, are returned to the pool of unallocated system resources and are available to fulfill requirements later in the job or for subsequent assignment to other jobs unless required by later job steps.

Format

```
//[symbol] FREE lfdname-1 [(DEV)],...,lfdname-n [(DEV)]
```

Parameters

`lfdname`
File name previously defined in a DVC-LFD control statement sequence.

`(DEV)`
The device and volume containing the file are released.

Note: *You should always specify the* (DEV) *parameter even though it's shown as optional. Additionally, you must specify* (DEV) *to free unit record devices, such as card readers and punches, printers, and workstations.*

GBL

Function

Global status is given to the named set symbols. This statement may appear anywhere in the job stream, and the symbols are global from the point of encounter forward. When defining a set symbol, an ampersand is not to be included on the global declaration. In the run processor, an ampersand always means substitution and is not part of the symbol name.

Format

```
//[symbol] GBL set-id-1[=init-1][,set-id-2[=init-2] ,...,set-id-n[=init-n]]
```

Parameters

`set-id-1` through `set-id-n`
Name of the set symbol that is to have global status.

`=init-1` through `=init-n`
Value given to *set-id* if the variable has not been previously defined as being global. If previously defined, *init* is ignored.

Note: *Whenever a quoted value is assigned to a set symbol (using / / GBL), the quotes are considered part of the value (for example, if / / GBL X='ABC', then &X is 'ABC').*

GO

Function

Causes an unconditional branch, in a forward direction, to another control statement. GO is acted upon during run processor time (prior to the job's execution), unlike the SKIP job control statement, which is acted upon at execution time.

Format

```
//[symbol] GO destination
```

Label

symbol
: Identifier that is used if this control statement is to be the target of another GO or IF control statement.

Parameter

destination
: Identifier of the target control statement that is to receive control; it must agree with the name in the label field of the receiving control statement.

IF

Function

Causes a conditional branch, in a forward direction, to another control statement. IF is acted upon during run processor time (prior to the job's execution), unlike the SKIP job control statement, which is acted upon at execution time.

Format

```
//[symbol] IF (a op b)destination
```

Label

symbol
: Identifier that is used if this control statement is to be the target of a preceding GO or IF control statement.

Parameters

a op b
: Conditional branch test values:

  a and b
  : Are the operands to be compared; can be either alphabetic or numeric; (a run processor error results if one is alphabetic and the other is numeric). If a numeric comparison is done and neither operand is numeric, both the greater than and less than conditions are set, resulting in all conditions except equal being allowed to branch. If a character comparison is being done and the two operands are not the same length, the comparison is made on the number of characters, rather than on the contents of the operands.

  op
  : Relational operator:

    EQ *a* equal to *b*
    NE *a* not equal to *b*
    GT *a* greater than *b*
    LT *a* less than *b*
    GE *a* greater than or equal to *b*
    LE *a* less than or equal to *b*

    The comparand and the relational operator are separated by spaces.

destination
: Identifier of the target control statement that is to receive control if the transfer condition is true. This must agree with the name in the label field of the receiving control statement.

Notes:

1. *If the operands to be compared by an IF statement contain embedded blanks, you must use / / JSET to replace the values before you can compare the operands.*

2. *Whenever you enclose an operand in quotes, the quotes are considered part of the operand. For example, the two operands in ('a' EQ a) are not considered equal.*

INQ JOB

Function

Through the use of symbols, INQ JOB allows you to examine job-related values and to determine the availability of certain facilities.

Format

```
//symbol INQ JOB,keyword
```

Label

symbol
Symbol name.

Parameter

keyword
One of the following keywords that assigns the specified value to *symbol:*

NAME	Assigns the job name.
ORI	Assigns the user-id of the originator.
HOST	Assigns the host-id of the originator (null value if none).
ORID	Assigns the device-id of the originator (if a local workstation).
WKS	Assigns 0 if a job is not initiated from a workstation.
	Assigns 1 if a job is initiated from a workstation.
DDP	Assigns 0 if remote DDP is not initiated.
	Assigns 1 if remote DDP is initiated.
JBNO	Assigns a 4-byte job number.
JUL	Assigns the Julian date YYDDD.

INQ SYS

Function

Through the use of symbols, allows you to examine system-related values and to determine the availability of certain facilities.

Format

```
//symbol INQ SYS,keyword
```

Label

symbol
Symbol name

Parameter

keyword
One of the following keywords that assigns the specified value to *symbol:*

RES	Assigns the SYSRES volume serial number.
RUN	Assigns the SYSRUN volume serial number.
DATE	Assigns the system date (YY/MM/DD).
TIME	Assigns the system time (HH.MM.SS.).
HOST	Assigns the system's own host-id.
CDM	Assigns 0 if consolidated data management is not configured. Assigns 1 if consolidated data management is configured.
DDP	Assigns 0 if DDP is not configured. Assigns 1 if DDP is configured.
WKS	Assigns 0 if workstation support is not configured. Assigns 1 if workstation support is configured.

S/80	Assigns 3 if running on model 3. Assigns 4 if running on model 4. Assigns 5 if running on model 5. Assigns 6 if running on model 6. Assigns 08 if running on model 8. Assigns 10 if running on model 10. Assigns 15 if running on model 15. Assigns 20 if running on model 20.
SPL	Assigns 0 if spooling is not configured. Assigns 1 if spooling is configured.
REL	Assigns the system release-id (vv.r.rr).
SUP	Assigns the supervisor name.

JNOTE

Function

Displays a message at the system console or specific workstations. JNOTE is acted upon by the run processor, before a job is put into execution.

Format

```
//[symbol] JNOTE comment-line [,destination-1,...,destination-n]
```

Parameter

`comment-line`
Comment or message to be displayed; may contain up to 60 characters. If the message contains embedded blanks, the slash character, or commas, the line must be enclosed in single quotes.

`destination` (where `destination=[host-id:]user-id`) `host-id`
In a DDP environment, directs the message to a particular host; is from one to four alphanumeric characters long and is identical to the label-id of the LOCAP macroinstruction in your ICAM network. You can specify $HOST to indicate the originator/master host. The host-id is optional but, if specified, must be followed by a user-id. If the host-id is omitted, the local host is assumed.

`user-id`
Directs the message to a particular workstation; can be any 1- to 6-alphanumeric character workstation user-id, the keyword OPERATOR or YCON (denoting the console workstation), or YORI (denoting the job's originator workstation). YORI is the default.

If you omit a destination, the message goes to the originator workstation. A message sent to an originator workstation that is not logged on is rerouted to the console. No other messages are rerouted.

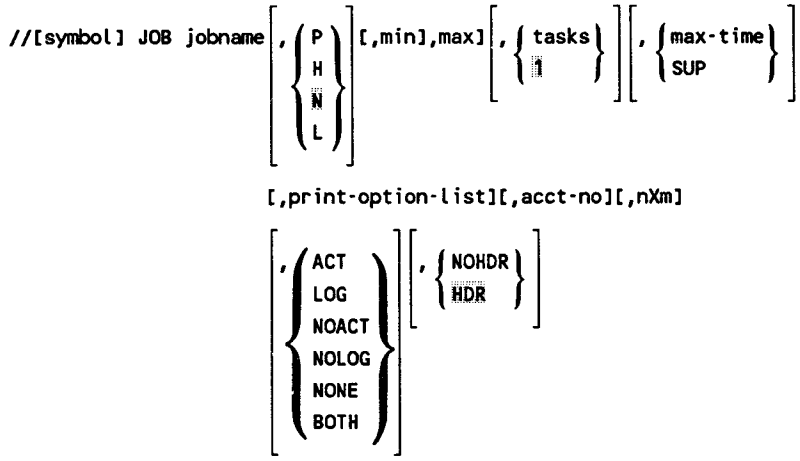
If the system does not have workstations or DDP, messages go to the system console.

JOB

Function

Identifies the job and indicates the beginning of control information for the job. The associated job's YRUN file is given the same name.

Format



Parameters

jobname

One to eight alphanumeric characters specifying the job identifier. This name must be used to reference the control stream after it has been filed in YJCS.

Note: *Do not hyphenate jobname if you plan to save the job. The save processor does not allow or recognize hyphens.*

P

Preemptive job selection priority. If specified on a system that does not support roll in/roll out, the priority is converted to high (H).

H

High job selection priority.

N

Normal job selection priority.

L

Low job selection priority.

`min`

Minimum number of main storage bytes required to execute the largest job step of the job; may be specified in hexadecimal (by coding the *number* or *X'number'*) or in decimal (by coding *D'number'*). If the number is not preceded by an X or a D and enclosed in single quotes, it is considered hexadecimal. 2000 hexadecimal (8K decimal) is the minimum amount that can be assigned; it does not include the job prologue. `min` is required when the programs being executed are not in YLOD, in an alternate load library on SYSRES, or on the volume containing the job's YRUN file.

Note: *If you are compiling (or assembling), link-editing, and executing your program in a single run, job control cannot determine how much main storage is required for the execution of your program. If your generated load module requires more main storage than the compiler (the largest known job step), insufficient main storage will be allocated. If possible, you should indicate how much main storage is needed for your load module.*

`max`

Maximum number of main storage bytes requested, but not required, to execute the largest job step on the job. May be specified in hexadecimal (by coding the number or *X'number'*) or in decimal (by coding *D'number'*). If the number is not preceded by an X or a D and enclosed in single quotes, it is considered hexadecimal.

If either *min* or *max* is omitted, the value specified for one is assumed for the other. Only that space actually needed by the largest program in the job is specified. Job control adds additional storage for the prologue, as required.

If both *min* and *max* are omitted, the length of the largest load module needed by the job is used. In most cases, omitting *min* and *max* gives the best use of main storage. Exceptions are programs that use additional main storage to improve performance (such as sort/merge), those cases where the user loads phases to locations other than addresses assigned by the linkage editor, or when programs to be executed are not on the SYSRES volume or the volume containing the job's YRUN file at run time.

`tasks`

Maximum number of tasks that can be active simultaneously in any job step. Each task specified requires 256 bytes in the job prologue so that the maximum number of tasks you can specify is limited by the size of the prologue (65,535 bytes).

max-time
: Maximum number of minutes this job should take for execution.

  If *max-time* is omitted, the default value specified at system generation (SYSGEN) time is used. If your job executes beyond this limit (specified or default), the operator is queried as to whether the job should continue or be terminated.

  If a timer service is not specified at SYSGEN time, the job continues processing until normal or abnormal termination occurs, whether or not *max-time* is specified in job control.

  The specified *max-time* limit is adjusted to allow for the following conditions: checkpoint/restart, PAUSE job control statements, SET CLOCK commands, and roll out/roll in.

SUP
: Suppresses the *max-time* function completely for a particular job.

print-option-list
: A list of one or more options used to control the printing of job control statements on the job log (in a spooling environment) or their display on the system console (in a nonspooling environment).

BASIC	Basic job control statements with substitution (This is the default in a spooling system.)
PROC	Expanded procs
EMB	Embedded data
DEBUG	Job control statements before substitution
SKIP	Statements skipped as a result of an IF or a GO directive
WARNING	Job control warning messages are not displayed on the console or workstation
ALL	All options in effect except WARNING
NONE	No options in effect (This is the default in a nonspooling system.)

  Only the first character needs to be specified for each option. BASIC (B) is automatically in effect whenever P, E, D, or S is chosen.

If more than one option is given, they must be separated by commas and enclosed in parentheses.

acct-no
: Job account number of from one to four alphanumeric characters long.

nXm
: Buffer pool size. Job log and spooled files not having reserved buffers use this pool.

  n
  : Number of buffers.

  X
  : Constant.

  m
  : The number of 256-byte blocks per buffer. The only values accepted for m are 1, 2, 4, 8, 16, and 32. Numbers larger than 32 default to 32; numbers outside the acceptable range are changed to the lower acceptable constant (e.g., 6 is changed to 4).

  If omitted, one 256-byte buffer is assumed if only job log is used. If other files are also sharing the pool, two buffers (2x2) are assumed.

ACT
: Forces the printing of accounting records, regardless of what system options are in effect.

LOG
: Forces the printing of job log records, regardless of what system options are in effect. File catalog passwords do not appear in the job log file.

NOACT
: Suppresses the printing of accounting records from the job log file.

NOLOG
: Suppresses the printing of log information from the job log file (including main storage dumps).

NONE
: Suppresses the printing of both accounting records and log information from the job log file.

BOTH
Allows the printing of accounting records and job log information.

NOHDR
Suppresses the printing of page headers in burst mode.

HDR
Allows page headers to be printed.

Note: *The user should consult the parameters used for system generation to ensure that a conflict does not arise when submitting these parameters.*

JSET

Function

Used to declare a local set symbol (local set symbols can be declared and referenced only within a jproc) or to change the value of a global set symbol without changing its status to local.

Format

```
//symbol JSET value
```

Label

symbol
: Set symbol name.

Parameter

value
: Value assigned to the symbol. May be a character string enclosed within apostrophes if it contains embedded blanks, may be a 2-term expression that is evaluated and converted to a character string before it is assigned, or may be used to establish a null character (blank). The operators allowed are +, -, *, /, ++, -| |-, **, and //. Leading zeros are maintained for multiple-digit numeric values in a JSET job control statement. If a leading zero is required when a symbol is used, it must be created by another JSET statement.

Notes:

1. *When using a JSET statement to perform an operation, the operands to be acted upon must be numeric.*

2. *When specifying a quoted value with / / JSET, one level of quotes is always removed from the value. (For example, if / /X JSET 'ABC', then &X is ABC.)*

LBL

Function

Supplies label information for files on disk, diskette, or tape volumes for use by data management. There may be only one LBL statement in a device assignment set.

Format

```
//[symbol] LBL {file-identifier  } [,{file-serial-number}] [,expiration-date]
             {'file-identifier'}     {VCHECK           }

[,creation-date] [,{file-sequence-number}] [,{generation-number}]
                   {1                    }    {1               }

[,{version-number}]
  {1             }
```

Parameters

`file-identifier`
Name that identifies the file; maximum of 17 characters for tape and data-set-label diskette files and 44 for format-label diskette and disk files, unless the disk files are temporary job or job step (scratch) files. In this case, the file-identifier can be from 1 to 39 characters long. This name corresponds to the file label. If prefixed by $SCR, the file is temporary and deleted at the completion of the job step. If prefixed by $JOB, the file is temporary and deleted at the completion of the job. If prefixed by $LOKnn (for disk files), the file is lockable.

`'file-identifier'`
Name that contains embedded blanks.

`file-serial-number`
One to six alphanumeric characters specifying the file serial number. Identical to the volume serial number of the first volume of the file or diskette.

`VCHECK`
A file serial number is to be created on output and checked on input.

If positional parameter 2 is omitted, the tape does not have a VOL1 label, or there is no volume serial number/file serial number relationship for disk.

`expiration-date`
File expiration date; may take either of two forms:

- `yyddd`

  `yy`
  Is the year.

  `ddd`
  Is the day of the year.

- `Rdddd`

  `R`
  Indicates that the file is to be retained.

  `dddd`
  Is the number of days (1-9999) that the output file is to be saved beyond the creation date.

If you omit the *expiration date* and allocate then write to the file (in the same job step), the current system date is used. If omitted and the file is only being allocated, no date is specified and zeros are inserted.

`creation-date`
File creation date; written in the form, yyddd:

`yy`
Is the year.

`ddd`
Is the day of the year.

If the *creation-date* is omitted for a tape file, the date stored in either the job prologue or the system information block is used. For a disk output file, the date stored in the job prologue is used; for a disk input file, the field is ignored.

`file-sequence-number`
At file creation, assigns a sequence number to the file indicating its position with respect to the first file within a multifile set; should be specified in subsequent // LBL statements to indicate the file's position. If omitted on output, data management assigns 1 as the sequence number regardless of the file's position. If omitted on input, data management does not check for sequence numbers but expects to use the first file encountered. This parameter applies only to standard labeled tape files.

generation-number
Unique edition of a file; applies only to tape files.

version-number
Version of a generation of a file; applies only to tape files.

LBL

File Catalog

Function

Provides information for cataloging files. Allows for the maintenance of multiple generations of a file and protects against unauthorized access or update by using read/write passwords. Only disk and tape files may be cataloged.

Format

```
//[symbol] LBL  [qual/]level-id-1[,level-id-2...[,level-id-n]][{nn}][(rpw/wpw)]
                                                              {+n}
                                                              {-n}

'[qual/]level-id-1[,level-id-2...[,level-id-n]][{nn}][(rpw/wpw)]'
                                                {+n}
                                                {-n}

[,{file-serial-number}][,expiration-date][,creation-date]
  {VCHECK            }

[,{file-sequence-number}][,{generation-number}]
  {1                    }  {1               }

[,{version-number}]
  {1             }
```

Parameters

```
[qual/]level-id-1[,level-id-2...[,level-id-n]][{nn}][(rpw/wpw)]
                                               {+n}
                                               {-n}
```

Specifies the file catalog information; limited to 17 characters for tape files and 44 characters for disk files. If the level names include blanks, the parameter must have leading and trailing apostrophes.

qual/
File qualifier of from one to eight characters long. This removes a file from general use and places it in the category of restricted use. The slash is coded as part of the qualifier.

level-id-1 through level-id-n
Level names for the file; each level must be separated by a period.

nn
Absolute reference number of a file generation.

+n
Creation number of a new file generation; the plus sign must be coded.

-n
Relative reference number of an old member of a generation in relation to the current member. The minus sign must be coded.

(rpw/wpw)
Read and write passwords of from one to six alphanumeric characters long; must be separated by a slash and enclosed in parentheses.

file-serial-number
One to six alphanumeric characters specifying the file serial number. Identical to the volume serial number of the first volume of the file.

VCHECK
The volume serial number/file serial number relationship is to be checked on input and created on output by data management.

If positional parameter 2 is omitted, the tape does not have a VOL1 label, or there is no volume serial number/file serial number relationship for disk.

expiration-date
File expiration date; may take either of two forms

- yyddd

  yy
  Is the year.

  ddd
  Is the day of the year.
- Rdddd

  R
  Indicates that the file is to be retained.

dddd
Is the number of days (1-9999) that the output file is to be saved beyond the creation date.

If you omit the *expiration-date* and allocate then write to the file (in the same job step), the current system date is used. If omitted and the file is only allocated, no date is specified and zeros are inserted.

creation-date
File creation date; written in the form, yyddd

yy
Is the year.

ddd
Is the day of the year.

If *creation-date* is omitted for a tape file, the date stored in either the job prologue or the system information block is used. For a disk output file, the date stored in the job prologue is used; for a disk input file, the field is ignored.

file-sequence-number
At file creation time, assigns a sequence number to the file indicating its position with respect to the first file within a multifile set; should be specified in subsequent // LBL statements to indicate the file's position. If omitted on output, data management assigns 1 as the sequence number regardless of the file's position. If omitted on input, data management does not check for sequence numbers but expects to use the first file encountered. This parameter applies only to standard labeled tape files.

generation-number
Unique edition of a file; applies only to tape files.

version-number
Version of a generation of a file; applies only to tape files.

Notes:

1. *The logical unit number specified on the DVC control statement when a file is cataloged must not be of the general type (50-59). This ensures that cataloged files will obtain a valid device whether or not the required volume is mounted when the job is scheduled.*

2. *The plus and minus are only used to delimit relative generation references. Using it elsewhere results in errors.*

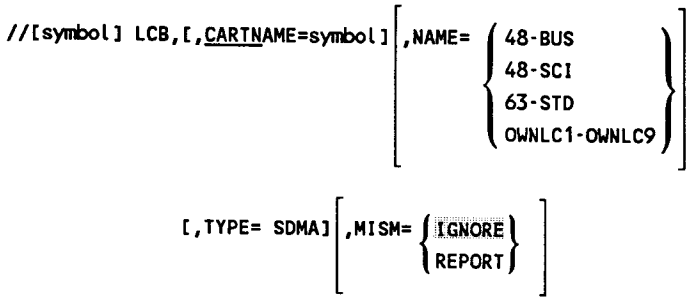
3. *When cataloging a new member to a generation that will cause the maximum number of generations to be exceeded, the device type and volume serial number will be taken from the catalog entry for the oldest member of the generation. The oldest member is also automatically decataloged. This allows for the cycling of three tapes for the three current generations.*

LCB

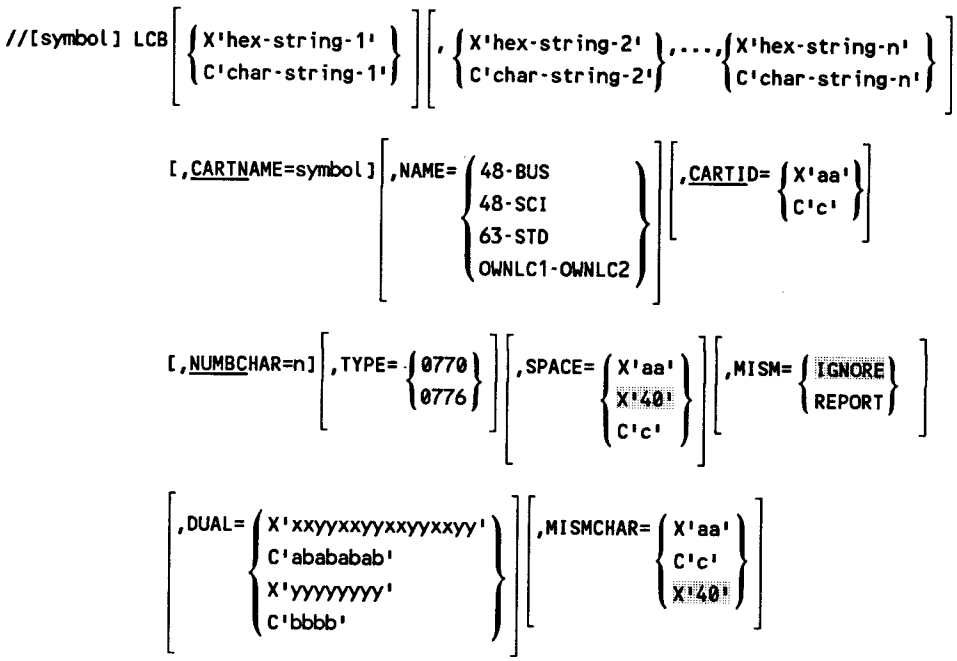
Function

Overrides the system default load code buffer for a print file; becomes effective when the file is opened. During interjob-step processing, the SYSGEN default value is used. If used, the LCB control statement must be within the DVC-LFD sequence for the print file that uses it.

Format 1: (SDMA Printers)



Format 2: (Non-SDMA Printers)



Label

symbol
: Specifies a default cartridge name when *CARTNAME* is omitted or specifies the name of a filed load code buffer (48-BUS, 48-SCI, 63-STD, OWNLCn) that you're changing via the job SG$PRB.

Parameters

{X'hex-string' / C'char-string'}
: Actual load code buffer on the cartridge for which this load code is being constructed (applicable only to non-SDMA printers). You need two hexadecimal characters or one EBCDIC character for every graphic symbol on the print cartridge. The position of each in the string of positional parameters corresponds to its position on the print cartridge. As many positional parameters as are needed to specify the entire print cartridge may be used by mixing the character and hexadecimal types, as required.

Note: *Statement continuation may be used only between parameters. A particular parameter may not be partially coded on one statement line and continued on another. In hexadecimal representation, the number of digits must be even.*

CARTNAME=symbol
: 1- to 8-character identifier of the print cartridge that the operator must mount. If specified, the system issues a mount message to the operator. If omitted, the user must take steps to stop the execution of the program, if necessary, to allow the operator to mount a required cartridge. You may use *symbol* instead of *CARTNAME* to specify a cartridge name. If you use both *symbol* and *CARTNAME* to specify a cartridge name, *CARTNAME* takes precedence.

NAME={48-BUS / 48-SCI / 63-STD / OWNLCn}
: Specifies the name of a filed load code buffer, which in turn specifies a cartridge name. When *NAME* is specified, *CARTNAME* is unnecessary. When the job SG$PRB is being used to change a filed buffer, specify the buffer name using *symbol* instead of *NAME*.

CARTID={X'aa' / C'c'}
: Load code buffer cartridge identification. May be two hexadecimal digits (X'aa') or one EBCDIC character (C'c'). This parameter is required for the Unisys 0770 and (non-SDMA) 0776 Printer Subsystems.

NUMBCHAR=n

For non-SDMA printers only. Total number of graphic symbols expected on the print cartridge. Should coincide with the number of characters specified in the positional parameters as a safety check.

If omitted, the number of characters specified by the positional parameters is assumed to be correct.

TYPE= { SDMA / 0770 / 0776 }

Type of Unisys print device. An SDMA printer is attached to the system via a paper peripheral controller and is the only valid type for models 3 through 6. The 0770 and non-SDMA 0776 printers are attached via a byte multiplexer and are available only on the model 8 system.

Note: *There are two types of 0776 printers SDMA and non-SDMA. Only the Model 8 can accommodate a non-SDMA 0776 printer.*

SPACE= { X'aa' / X'40' / C'c' }

For non-SDMA printers only. Blank character representation. May be two hexadecimal digits (X'aa') or one EBCDIC character (C'c').

MISM= { IGNORE / REPORT }

Specifies whether character mismatch error conditions are to be reported on the system log device. A mismatch occurs when you try to print a character that has not been placed in the printer's load code buffer or specified as a dualed character.

DUAL= { X'xxyyxxyyxxyyxxyy' / C'abababab' / C'bbbb' / X'yyyyyyyy' }

For the 0770 and 0776 printers only. Allows you to equate up to four nonprintable character codes with up to four printable characters. This enables you to print data with up to 52 character codes using a 48-character print band.

For the Unisys 0770 or non-SDMA 0776 printer subsystems, you can choose the replacement symbol. If you specify in hexadecimal, you would choose the *DUAL= X,xxyyxxyyxxyyxxyy'* parameter option; *xx* is a specified character in the load code buffer and *yy* is a code that is not in the buffer; *xx* and *yy* print as the same graphic symbol. If you specify in EBCDIC, you would use the *DUAL=C'abababab'* parameter option, *a* being a specified character representing a specific graphic symbol on the print cartridge and *b* being the character that is printed as the same graphic symbol as *a*. You may specify from one to four character substitutions.

MISMCHAR= { X'aa' / C'c' / X'40' }

For non-SDMA printers only. The character specified through this parameter is printed for any character detected as a mismatch (a character that is not in the load code buffer and is not dualized). The character specified through the *MISMCHAR* parameter must be specified as a graphic symbol through positional parameter 1.

Note: *Only the first five characters of the LCB control statement's keywords needs to be specified (i.e., CARTN=symbol, rather than CARTNAME=symbol). If the keyword symbol has less than five characters, you have to specify them all.*

LFD

Function

Links the file information in the control stream with the data management file definition. For each LFD statement in the control stream, a file control block is generated that contains information determined as a result of processing device assignment sets.

A device assignment set consists of at least one DVC statement and one LFD statement for each file to be accessed in the user program, and the LFD statement must follow the DVC statement. Depending on the file definition requirements, a device assignment set may also include VOL, EXT, and LBL statements inserted, in that order, between the DVC and LFD statements.

Format

```
//[symbol] LFD { filename  } [,{ n }] [,{ EXTEND }]
               { *filename }   { 8 }   { IGNORE }
                                       { INIT   }
                                       { PREP   }
                                       { ID     }
```

Parameters

`filename`
One to eight alphanumeric characters specifying the logical file name; must be the same as the name specified in the label field of the DTF macroinstruction for data management or the PIOCB macroinstruction for the physical I/O users.

`*filename`
Input-only file. The operator should verify that the write enable ring has been removed from the tape reel or that the file protect ON/OFF switch has been pressed to the ON position for the disk pack.

`n`
Maximum number of extents (up to 20) in the file; used to reserve extent table storage for use by data access methods. Space acquired by using this parameter increases the total storage requirement for the job and must be taken into consideration when allocating main storage, but not when specifying main storage space on the JOB control statement. This number is the sum of the physical extents on the disk plus the number of partitions in the file.

EXTEND

Tape file is to be extended . The information is appended to the present end of the file, provided your program makes the proper specifications.

IGNORE

Specified file is to be treated as an optional file. Allows you to decide at execution time whether accesses to the file should be processed or ignored based on the file resources that are available without having to change the program.

Notes:

1. *This parameter is intended for use with consolidated data management only. It is ignored if specified for a file that is accessed by basic data management. SAT files are also ignored.*

2. *When IGNORE is specified, the following status is returned to the executing program:*

   - *File initialization (OPEN) - successful status*
   - *Input operation - end-of-file (EOF) indication*
   - *All other operations - successful status (request is ignored)*

3. *When IGNORE is specified for an input file, the program should be performing sequential retrieval because it may not be expecting an EOF indication.*

INIT

Specified file is to be initialized starting at the first record the first time the file is opened. Previous control information in the format labels is ignored at the file open time and is overwritten by specifications contained in this DVC-LFD sequence at file close time. This parameter should not be used for a checkpoint record file.

PREP

Specifies that a cataloged tape file be prepped before it is used as an output file.

ID

One to ten characters that specify the owner's name. This information is stored in DL$ONR in the VOL1 id field.

MTC

Function

Positions tape volumes prior to the execution of a job step. Can be used to position a data file or pre-position a multifile tape volume.

Must be specified after the device assignment set for that magnetic tape unit.

Format

```
//[symbol] MTC lfdname, { BB,nn }
                        { BM,nn }
                        { FB,nn }
                        { FM,nn }
                        { WM,nn }
                        { RL    }
                        { RU    }
```

Parameters

`lfdname`
The name of a tape file that is identified in a preceding DVC-LFD control statement sequence.

`BB`
Space volume backward the specified number of blocks.

`BM`
Space volume backward the specified number of tape marks.

`FB`
Space volume forward the specified number of blocks.

`FM`
Space volume forward the specified number of tape marks.

`WM`
Write number of tape marks specified.

`RL`
Rewind volume to load point.

NOP

Function

Inserts labels to be used as targets of branch or SKIP statements.

Format

```
//[symbol] NOP [QUERY]
```

Parameter

QUERY
: Used as the target of a dynamic SKIP function, which is accomplished through the // OPTION QUERY job control statement. This parameter is for workstation users and console operators.

Note: *Comments preceded by one or more blanks and enclosed in single quotes may be specified in the NOP statement in place of the* QUERY *parameter. When used for this purpose, NOP does not have to be the target of branch or SKIP job control statements.*

OPR

Function

Displays a message at the system console or specific workstations. The statement can appear anywhere in the control stream and is displayed at interstep processing time.

Format

```
//[symbol] OPR comment-line [,destination-1,...,destination-n]
```

Parameters

`comment-line`
Comment or message to be displayed; may contain up to 60 characters and must be enclosed in single quotes if it contains embedded blanks, the slash character, or commas.

`destination` (where `destination=[host-id:]user-id`)

`user-id`
Directs the message to a particular workstation; can be any 1- to 6-alphanumeric character workstation user-id, the keyword OPERATOR or YCON (denoting the console), or YMAS (denoting the job's master workstation). YMAS is the default.

`host-id`
In a DDP environment, directs the message to a particular host; is one to four alphanumeric characters long and is identical to the label-id of the LOCAP macroinstruction in your ICAM network. You can specify $HOST to indicate the originator/master host. The host-id is optional but, if specified, must be followed by a user-id. If a host-id is omitted, the local host is assumed.

If you omit a destination, the message goes to the master workstation. A message sent to a master workstation that is not logged on is rerouted to the console. No other messages are rerouted.

If the system does not have workstations or a DDP, the message goes to the system console.

OPTION

Function

Specifies certain optional features and controls the operating environment. Some options are effective only in the job step in which they are specified; others are effective from the time the option is encountered until end of job; while others are in effect for the entire job. The options may be written in any order and must be separated by commas with no intervening spaces. Only the first three characters of each option need to be coded.

Format

```
//[symbol] OPTION p1[,...,pn]
```

Parameters

p1[,...,pn]
Represents the options to be specified:

ACN=account-number
Overrides the *acct-no* specified in the JOB control statement.

ABRDUMP
Provides abbreviated dump-only printing of selected areas in the macro storage vicinity relative to the current TCB PSW and OPEN DTFs.

BOF
User program is given control with binary overflow interrupt-enabled.

BUF=nXm
Overrides the *nXm* parameter specified in the JOB control statement.

DOF
User program is given control with decimal overflow interrupt-enabled.

DUMP
Job region dump is provided in hexadecimal at job execution time if job step termination is requested, or snapshot dumps are given in response to a SNAP macroinstruction execution.

EOD=xx
Supplies substitute characters for the /* (end-of-embedded-data) job control statement. Used when embedded data is dialog specification language source code. The first character specified must be a slash (/). The second character can be anything but a slash, an asterisk, an ampersand, or a currency symbol (/, *, &, $).

GABRDUMP
Specifies that OPTION ABRDUMP is in effect for every job step from the time GABRDUMP is encountered to end-of-job.

GDUMP
Specifies that OPTION DUMP is in effect for every job step from the time GDUMP is encountered to end-of-job.

GJOBDUMP
Specifies that OPTION JOBDUMP is in effect only for the job step.

GO
Root phase of the last load module generated during the linkage editor job step is automatically executed at the completion of the linkage editor without intervention from job control.

GSUB
Provides symbol substitution for all embedded data sets in the job stream. This is a global SUB option.

GSYSDUMP
Specifies that OPTION SYSDUMP is in effect for every job step from the time GSYSDUMP is encountered to end-of-job.

HDR= { NOHDR / HDR }
NOHDR suppresses the printing of page separators. HDR allows page separators to be printed. OPTION HDR overrides page separator specifications in the JOB control statement.

HOLD
Places a job containing it in hold status while the job is in the job queue table. A job containing this option is not released until a BEGIN operator command is issued or until a CC job control statement with BE specified is encountered in a subsequent control stream. // CC BE cannot be used to release a HOLD within the same job.

JOBDUMP
Specially edited version of the dump should be provided for a job step if a termination dump is requested.

LINK
Linkage editor is automatically executed following the termination of the program named on the EXEC statement. This allows you to compile and link job steps with no intervention from job control.

LOG= { logical-unit-number | ORIGINATOR | CENTRAL }

Directs the job log to a specific printer or magnetic tape. The keywords ORIGINATOR and CENTRAL apply only to printers. ORIGNATOR directs the log to the printer at the job's originator. (This includes an auxiliary workstation printer if the originator is a workstation.) CENTRAL directs the log to the local site's central printer. Only LOG=CENTRAL can be specified in RBP-initiated jobs. The default log destination for RBP is the originator. If the DVC statement for the output file indicates a specific printer device (e.g., DVC 24) OPTION LOG=24 should be specified so that the job log and the output file are directed to the same device.

IMMOVE

Prevents memory consolidation movable shuffle in this job step.

MASTER=destination (where destination=[host-id:]user-id)

Assigns the specified workstation (at the specified host) as the master - the workstation that has control of the job during execution. By default, the originator (see OPTION ORI) has control of the job unless you use this option. OPTION MAS becomes effective when the job name is entered in the job queue and remains in effect throughout job processing. Specify OPERATOR to designate the console as master. Specify a host-id to designate (a workstation at) a particular host. If host-id is omitted, the local host is assumed. When you specify a host-id, you must follow it with a user-id. This option in a saved translated control stream is effective when the stream is restored.

MASTER=destination (EXEC)(where destination=[host-id:]user-id)

Functions the same as MASTER=*destination* but takes effect only when the job is in execution. The originator has control when the job is in the job queue.

MAX=maximum-main-storage-size

Overrides the *max* parameter specification in the JOB control statement. The *max* value is expressed in hexadecimal (*MAX=number* or *MAX=X'number'*). To indicate that the value of *max* is to be interpreted as a decimal value, specify *MAX=D'number'*. If a maximum value is specified more than once (via the // JOB statement or multiple // OPTION statements), the largest value is used.

MERGE=NO

Used to create a separate identifer for a job's log in the spool LOG file. By including this option, you can determine if your job log is present in the accumulated LOG file.

MIN=minimum-main-storage-size

Overrides the *min* parameter specification in the JOB control statement. The *min* value is expressed in hexadecimal (*MAX=number* or *MAX=X'number'*). To indicate that the value of *min* is to be interpreted as a decimal value, specify *MIN=D'number'*. If a minimum value is specified more than once (via the // JOB statement or multiple // OPTION statements), the largest value is used.

MXT=maximum-time

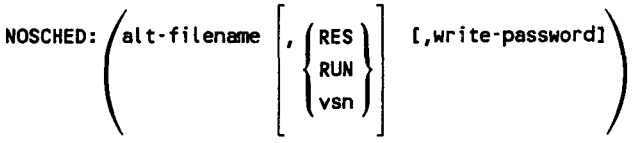
Overrides the *max-time* parameter specified in the JOB control statement. The maximum time can be specified in minutes, or you can specify SUP or DEF. MXT=SUP suppresses the max-time function. MXT=DEF specifies that the system default is to be used for the *max-time* value.

NODUMP

No dumps are taken for this job step. This option turns off snap dumps, end-of-job-step dumps, and abnormal termination dumps. This is defined for compatibility purposes and should not be specified. Job control assumes this condition by default.

NOSCHED

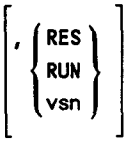
Saves a job control stream in its expanded state in YSAVE but does not schedule the job. See the description of SAVE option for information about subsequent runs of the job.



Saves the job in its expanded state (in an alternate MIRAM library) but does not schedule the job.

alt-filename

Used to specify a 1- to 44-character file identifier.



Used to specify the volume containing the alternate file. *RES* identifies the SYSRES volume, *RUN* identifies the RUN pack, and *vsn* identifies the volume serial number of a disk pack or format-label diskette. If the file is cataloged, *RES*, *RUN*, or *vsn* overrides the volume indicated in the catalog. If the file is cataloged and *RES*, *RUN*, or *vsn* is omitted, the volume indicated in the catalog is used. If the file is not cataloged and you omit *RES*, *RUN*, or *vsn*, SYSRES is assumed. (Simply code // OPTION NOSCHED *alt-filename* in this case.)

`write-password`
Used to specify a 1- to 6-character write-password when the file is cataloged with a write-password; ignored if the file has no password.

See the SAVE option for information about subsequent runs of the job.

NSCAN
Resets the SCAN facility. It should be used only within the embedded data of a job step for which SCAN has previously been specified. Subsequent job control statements normally removed by SCAN are not removed. The OPTION NSCAN statement itself is removed. When NSCAN is specified, SCAN cannot be used again in the same job step.

NSRCH
Specifies that only the library indicated by the EXEC control statement is to be searched.

NSUB
Resets the SUB facility. It should be used only within the embedded data of a job step for which both SUB and SCAN have previously been specified. Set symbols in embedded data are not substituted until another SUB is encountered.

NULL
Specifies a no-operation for the OPTION statement.

`OFT=+n`
Tells the run processor to reserve space for an additional number *(n)* of files in the open file table. The *n* parameter must be in the range 1 through 16 and must be preceded by a plus sign.

`OPL=print-option-list`
Overrides *print-option-list* specifications in the JOB control statement. Any of the options available through the *print-option-list* parameter of the JOB control statement may be specified via OPTION OPL.

ORIGINATOR=destination (where destination=[host-id:]user-id)

Assigns a workstation (at the specified host) as the originator. The workstation that physically initiates the job is considered the originator and subsequently has control of the job during processing unless this option is used. OPTION ORI takes effect (changes user-id) when it's encountered in the control stream. That is, the run processor immediately changes the user-id to the one specified by the ORI parameter. You have the option of specifying more than one OPTION ORI statement in the same job stream. In such cases, the last OPTION ORI statement encountered in the job stream designates the workstation that controls processing at execution time as the originator. Specify OPERATOR to designate the console as master. Specify a host-id to designate (a workstation at) a particular host. If host-id is omitted, the local host is assumed. When you specify a host-id, you must follow it with a user-id. This option in a saved translated control stream is effective when the stream is restored.

OUT={ORIGINATOR / CENTRAL / [host-id:]user-id}

Directs all job output (print files, punch files, and job logs) to the specified destination as follows:

- ORIGINATOR

  Directs all printed output to the printer at the job's originator. Directs all punch output to the central punch at the job's originator.

- CENTRAL

  Directs all print or punch output to the local site's central printer/punch.

- [host-id:]user-id

  Directs all printed output to the specified destination and all punch output to the central punch at the specified host.

  The host-id is from one to four alphanumeric characters long and identical to the label-id of the LOCAP macroinstruction in your ICAM network. Use $HOST to indicate the job's originator. The local host is assumed if the host-id is omitted. The host-id is optional but, if specified, must be followed by a user-id.

  A 1- to 6-alphanumeric-character workstation user-id identifies an auxiliary workstation printer. The keyword CENTRAL in place of a user-id identifies the central printer.

This option is effective for all of the job's print and punch output but can be changed for individual print or punch files by specifying // ROUTE or // DST in the device assignment set for that file.

PRI=switch-priority

Establishes an overall task switching priority that can be changed for particular job steps by specifying a relative priority (e.g., +3 or -3) or an absolute priority (e.g. 3) on the EXEC statement.

PRT = { ACT | LOG | NOACT | NOLOG | NONE | BOTH }

Overrides the print option specified in the JOB control statement.

- *ACT* forces the printing of accounting records.
- *LOG* forces the printing of job log records.
- *NOACT* suppresses the printing of accounting records.
- *NOLOG* suppresses the printing of the log file.
- *NONE* suppresses the printing of both accounting records and log information from the log file.
- *BOTH* allows the printing of accounting records and job log information.

PSYSDUMP

Terminates the job immediately if abnormal termination occurs. SYSDUMP is executed as a separate job. This allows immediate rerunning of the terminated job. You can force SYSDYMP to initiate a new job when SYSDUMP is initiated by program error.

QUERY

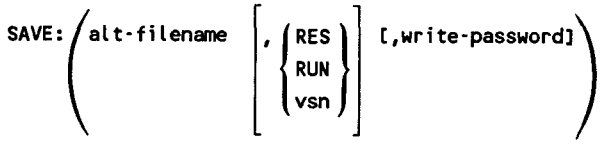
Allows the interactive user to change control stream execution from a workstation. The QUERY option is explained in the job control user guide for your system.

REPEAT

Currently executing program is automatically reinitialized upon termination until all embedded data files are exhausted. This gives you the ability to execute stacked assemblies or compilations without job control intervention.

SAVE

Saves a job control stream in its expanded state in YSAVE and schedules the job to be run. A copy of the control stream as it appears in YRUN is stored in the system file YSAVE. Subsequent runs of the job are initiated through the SC/SI workstation or system console command, or through the CC (SC/SI) job control statement. Saving an expanded job control stream that has a number of jprocs eliminates the time-consuming chore of jproc expansion by the run processor on subsequent runs. Information about the SC/SI workstation/console command is found in the workstation user guide and the operations handbook, respectively, for your system.



Functions the same as SAVE but saves the control stream in an alternate MIRAM library.

alt-filename

Used to specify a 1- to 44-character file identifier.

Used to specify the volume containing the alternate file. *RES* identifies the SYSRES volume, *RUN* identifies the RUN pack, and *vsn* identifies the volume serial number of a disk pack or format-label diskette. If the file is cataloged, *RES*, *RUN*, or *vsn* overrides the volume indicated in the catalog. If the file is cataloged and *RES*, *RUN*, or *vsn* is omitted, the volume indicated in the catalog is used. If the file is not cataloged and you omit *RES*, *RUN*, or *vsn*, SYSRES is assumed. (Simply code // OPTION NOSCHED:*alt-filename* in this case.)

write-password

Used to specify a 1- to 6-character write-password when the file is cataloged with a write-password; ignored if the file has no password.

SEVERE

Specifies that the run processor is to be terminated (the job is not to be scheduled) if warning errors are encountered. Normally, warning errors would not terminate the job.

SCAN
Acts upon and then removes selected control statements (CR, OPTION, GBL, GO, IF, JSET, and NOP) from the embedded data files.

If omitted, only the terminators (FIN, END, /$, and /*) are detected.

SIG
User program is given control with floating-point significant exception interrupt-enabled.

SUB
Scans embedded data for possible parameter or set symbol substitution.

SYSDUMP
Fully edited system dump is provided if job step termination is requested.

TEST
Specifies that the job is not to be queued or run.

TRACE
Brings in the monitor routine to record the effect of each executed instruction. Option monitor tasks are selected as described in the supervisor user guide.

TSK=number-of-tasks
Overrides the tasks parameter specified in the JOB control statement. From 1 to 255 tasks can be active within any job step.

UNDEFINED
Specifies that from the time this option is encountered to the end of job, a warning error message is to be generated whenever an undefined set symbol is detected.

UNEQUAL
Specifies that a warning error message is to be generated whenever two unequal character strings are compared.

XUF
User program is given control with exponent underflow exception interrupt-enabled.

Note:

The OPTION job control statement should not be placed between these job control statements:

- EXEC and /$
- EXEC and PARAM
- PARAM and PARAM
- /* and /$ (where they delimit two separate embedded data sets)

PARAM

Function

Used to submit information to the program during its execution. The information can be in the form of keyword parameters, control statements (other than job control statements), or any other type of information the program wants it to be. The changes are coded in the operand field. Job control verifies the format.

PARAM statements must follow the EXEC control statement. Therefore, control belongs to the program named in the EXEC control statement, not to job control.

There is no limit to the number of PARAM control statements allowed in a control stream.

Format

```
//[symbol] PARAM operand-1[,...,operand-n]
```

Parameter

operand
: Contains the information that is passed to the requesting program. If the information contains embedded blanks, it must be enclosed with single quotation marks.

PAUSE

Function

Displays a message at the system console or specific workstations but also causes the processing of the job to come to a halt until the message is acknowledged. The statement can be placed anywhere in the job step, but is not displayed at the system console or workstation until immediately before a program is executed.

Format

```
//[symbol] PAUSE comment-line [,destination-1,...,destination-n]
```

Parameter

`comment-line`
Comment or message to be displayed; may contain up to 60 characters and must be enclosed in single quotes if it contains embedded blanks, the slash character, or commas.

`destination` (where `destination=[host-id:]user-id`)

`host-id`
In a DDP environment, directs the message to a particular host; is one to four alphanumeric characters and is identical to the label-id of the LOCAP macroinstruction in your ICAM network. You can specify $HOST to indicate the originator/master host. The host-id is optional but, if specified, must be followed by a user-id. If host-id is omitted, the local host is assumed.

`user-id`
Directs the message to a particular workstation; can be any 1- to 6-alphanumeric-character workstation user-id, the keyword OPERATOR or YCON (denoting the console), or YMAS (denoting the job's master workstation). YMAS is the default.

If you omit a destination, the message goes to the master workstation. A message sent to a master workstation that is not logged on is rerouted to the console. No other messages are rerouted.

If the system does not have workstations or a DDP, the message goes to the system console.

Note: *The PAUSE statement suspends all job control activity for the control string in which it appears until the message is acknowledged, but the processing of other jobs in the system continues.*

QGBL

Function

Allows you to change the value of global set symbols at run time from the workstation.

Format

```
//[symbol] QGBL set-id-1[=init-1].,set-id-2[=init-2],...,set-id-n[=init-n].
```

Parameter

`set-id-1` through `set-id-n`
Name of the set symbols; can be a maximum of eight characters in length.

`=init-1` through `=init-n`
Initial value of set symbols; can be a maximum of 60 characters in length.

Note: *Whenever a quoted value is assigned to a set symbol (using / / QGBL), the quotes are considered part of the value. (For example, if / / QGBL X='ABC', then &X is 'ABC' unless changed.)*

QUAL

Function

Prefixes an automatic qualifier to all subsequent file identifiers in the job; remains in effect until the end of the job or until another QUAL job control statement is encountered.

Format

```
//[symbol] QUAL [qualname]
```

Parameter

qualname
: Qualifier of one to eight alphanumeric characters.

If omitted, overrides the previous QUAL job control statement and terminates the use of a qualifier.

REN

Function

Permanently changes (renames) the label of a disk or format-label diskette file.

Format

```
//[symbol] REN lfdname,{new-label  }[,NTERM]
                       {'new-label'}
```

Parameter

`lfdname`
Identifies the file to be renamed. It must match the *lfdname* in the LFD statement for the file.

`new-label`
Specifies the new name for the file; 1 to 44 alphanumeric characters in length. If *new-label* contains embedded blanks, it must be enclosed by apostrophes.

`NTERM`
Specifies that errors encountered during the renaming process are to be ignored and the job allowed to continue. If this parameter is specified and a renaming error occurs, the job continues, but the file is not renamed. If this parameter is omitted and a renaming error occurs, the job terminates at the point of error.

Notes:

1. *Subsequent references to a renamed disk file must specify* new-label *in the LBL statement of the device assignment set for the file.*

2. *If you rename a cataloged file, you must recatalog the file with the new name.*

3. *REN statements are not permitted against files on SYSRES that begin with Y or against files on SYSRUN that begin with YR.*

ROUTE

Function

Specifies up to eight destinations for non-DDP destinations or one DDP site destination. These destinations are the central printer or punch at a local or DDP site, a workstation auxiliary printer at a DDP site, or an auxiliary printer that is locally or remotely connected to your system.

Format

```
//[symbol] ROUTE destination-1, . . . ,destination-8
```

Parameter

`destination` (where `destination=[host-id:]user-id`)

`CENTRAL` or `OS3CTR`
Destination is the central printer or punch at the local site.

`host-id:CENTRAL`
Destination is the central printer or punch at the DDP site identified by the one to four character *host-id.*

`host-id:user-id`
Destination is a DDP site (identified by the one to four character *host-id*) at which the printer output will be routed to an auxiliary workstation printer identified by the one to six character *user-id.*

`user-id`
Destination is an auxiliary printer on a workstation (identified by the one to six character *user-id*) that is locally or remotely connected to your system.

`$Y$MAS`
Destination is the auxiliary printer if the master at the local site is a workstation. Otherwise, the destination is the central printer at the local site.

This destination is valid only for print files.

Notes:

1. *Output can be routed to the site central printer and up to seven auxiliary printers.*
2. *The / / ROUTE and / / DST statements cannot be mixed in the same job.*

Examples

```
// ROUTE OS3CTR
```

and

```
// ROUTE CENTRAL
```

route output to the central printer.

```
// ROUTE USERID1, . . .,USERID8
```

routes output to up to eight auxiliary printers.

```
// ROUTE OS3CTR,USERID1, . . .,USERID7
```

routes output to the central printer and up to seven auxiliary printers.

Notes:

1. *DDP or auxiliary printer output (specified by / / ROUTE) and RBP output (specified by / / DST) cannot be mixed for any one job. For any job, all output must be of one type or the other. Also, DDP destinations and local auxiliary printer destinations cannot be used for the same print file.*

2. *When a workstation or terminal initiates a job that directs printed output to an auxiliary printer connected to a local or remote workstation or terminal (one that is not the originator), the user at the other workstation or terminal must be logged on with the same user-id as specified in the / / ROUTE job control statement and must issue an RP command to initiate printing. See the* Interactive Services Operating Guide, *UP-9972, for more information on the RP command.*

RST

Function

Restarts a BAL or COBOL program from a checkpoint when a computer malfunction causes a job to terminate. Checkpoint records are established by the CHKPT macroinstruction (in BAL) or the RERUN clause (in COBOL).

When the job is on cards, the RST statement must be the first card in the deck when the job is rerun. For a prefiled job, RST can be submitted from a card reader or added to the control stream (as the first statement) via the general editor (EDT) or the librarian. When the job is rerun, the program's execution begins at or near the checkpoint reached when the job stopped.

Format

```
//[symbol] RST filename,checkpoint-id,step-number ,jobname[(rename)] [,pri]
  [,key-1=val-1,...,key-n=val-n]
```

Parameter

`filename`
Name of the checkpoint file; must agree with the filename specified on the LFD statement for the checkpoint file.

`checkpoint-id`
Checkpoint number identifying which checkpoint is used to restart the job. This number is displayed on the system console when checkpoint records are encountered in the program.

`step-number`
Specifies the number of the job step that executes the user program to be restarted.

`jobname`
Names a prefiled job to be rerun when RST is submitted from a card reader. Must be specified if the *(rename)* parameter is used to assign an alternate name to the prefiled job.

`(rename)`
Alternate name of the job. Executes a previously filed control stream and enables the use of different job names.

`pri`
Scheduling priority.

key-n=val-n
A keyword and its value that may be referenced as would any GBL control statement. The effect of these parameters is as if a GBL control statement were inserted as the first control statement of the job. The total length of this parameter cannot exceed 44 characters.

Notes:

1. *All information necessary to prepare an RST control statement is displayed at the system console each time a program encounters a checkpoint.*

2. *The LFD job control statement in the device assignment set for the checkpoint file must not contain the INIT parameter. This parameter causes the file to be written from the beginning of the file, thus overwriting any existing records.*

3. *In COBOL, the RERUN clause is equivalent to the checkpoint macroinstruction.*

4. *The following must be taken into consideration when restarting a job:*

   - *Only one RST control statement may be submitted for any one job.*
   - *If the program being executed at the time the checkpoint is recorded is in the job's YRUN file (output of the linkage editor), the job to be restarted will not run to normal completion if a program overlay is called after the job has been restarted.*
   - *If a restart is to be done after the job has terminated (normally or abnormally), the restarted job step must not have originally had requests for temporary work areas.*
   - *Tapes previously positioned via an MTC control statement are not positioned to the proper point in the restarted job.*
   - *Card files cannot be repositioned.*
   - *If a multifile tape is to be repositioned, the file sequence number must be included on the LBL job control statement.*
   - *The disk file containing checkpoint records cannot contain user data.*
   - *Scheduling may be delayed if all the resources needed by all job steps in the job are not available, even if those needed only by the job step to be restarted are available.*
   - *Mount messages to the operator may be produced for volumes that were not needed in the original run. This is due to the fact that SKIP control statements are ignored.*
   - *Workstation files cannot be repositioned; therefore, a job cannot be restarted from a checkpoint that occurs when workstation files are opened.*

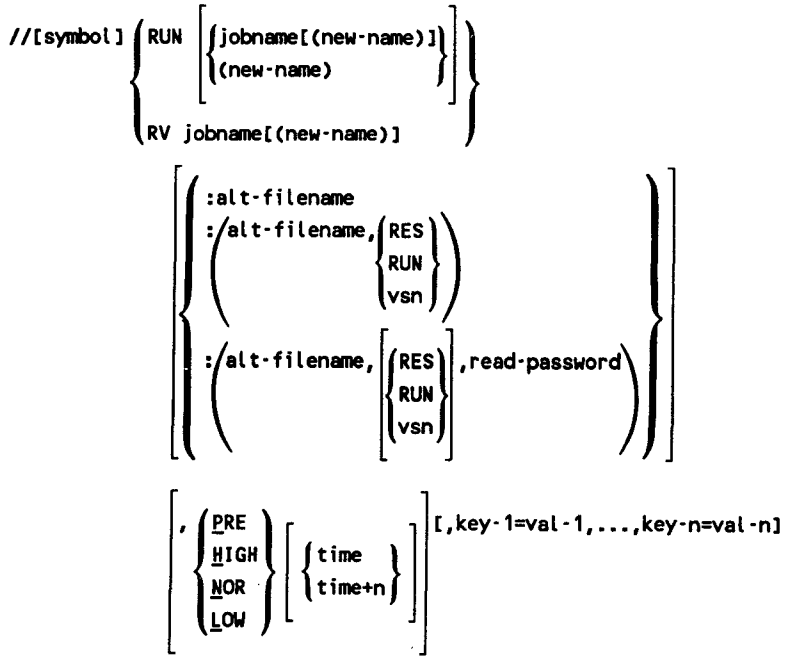
RUN/RV

Function

The RUN and RV statements permit another job to be initiated. RUN and RV control statements function in the same way, except that RUN should be used to call a control stream that is on cards or that is prefiled (in YJCS or an alternate library) and contains a CR statement because input (from a card reader) is necessary. RV is only used to call prefiled control streams that do not require a card reader. If RUN is used when a card reader is not required, the job is not initiated until the (unnecessary) reader is available. If RV is used when a card reader is required, the job is not initiated.

The RUN or RV statement may appear anywhere within the control stream, except between /$ and /* control statements or a critical series of control statements, such as the DVC through LFD sequence.

Format



Parameters

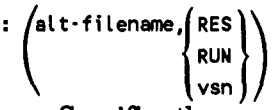
`jobname[(new-name)]`

One to eight alphanumeric characters identifying the job. A job name is required with RV, but can be omitted from RUN if the control stream is to be read from a card reader. The name used on the control stream's JOB statement is assigned in this case.

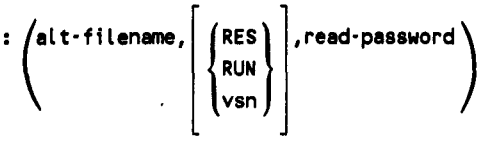
Include *(new-name)* to assign a new 1- to 8-alphanumeric character name

identifying the selected job control stream. The job assumes the new name, thus reducing the chance of duplicate job names. The new name alone may be used with RUN if the control stream is input from a card reader.

`:alt-filename`
Specifies the name of the alternate SAT library file, residing on SYSRES, that contains the job control stream. If the file name is cataloged (without a read password), the volume associated with that file name in the catalog is used.



Specifies the name of the alternate SAT library file and the volume containing the file. *RES* identifies SYSRES as the volume, *RUN* identifies the system RUN pack, and *vsn* identifies the volume serial number of a disk pack or format-label diskette. If the file name is cataloged (without a read password), the volume you specify (*RES*, *RUN*, or *vsn*) is used instead of the volume associated with the file name in the catalog.



Specifies the name of the alternate SAT library file and the volume containing the file. The *read-password* parameter must be specified if the file name is in the file catalog with a read password. RES identifies SYSRES as the volume containing the file, RUN identifies the system RUN pack, and *vsn* identifies the volume serial number of a disk pack or format-label diskette. If *RES*, *RUN*, or *vsn* is omitted, the volume associated with the file name in the catalog is used.

If no alternate file name is specified, the job control stream is read from YJCS.

PRE
Specifies preemptive priority for the job control stream.

HIGH
Specifies high priority for the job control stream.

NOR
Specifies normal priority for the job control stream.

LOW
Specifies low priority for the job control stream.

If omitted, the priority specified in the control stream's JOB statement is used.

`time`

Is a 4-digit number specifying a time of day in military format indicating when execution of the job is to begin.

`n`

Is an integer from 1 to 9 specifying the number of days after today to start execution of the job.

If you use the time with the priority parameter, you *must* use the first letter of the priority parameter, not the whole word. (For example, H1230 is acceptable, HIGH1230 is not.)

`key-1=val-1,...,key-n=val-n`

A keyword and its value that may be referenced as would any GBL control statement. The effect of these parameters is as if a GBL control statement were inserted as the first control statement of the job.

Note: *The total length of the RUN and RV statements, from the R in RUN or RV to the last character of the last specified value, cannot exceed 60 characters.*

SCR

Function

Scratches (deletes) files. Any specified file is deleted immediately upon the detection of an SCR control statement. Therefore, files may be deleted prior to the execution of a given job step. Files to be scratched should not be in use by another job. Files with a label prefix of Y cannot be deleted from the SYSRES volume. Only files on volumes currently mounted are deleted. Only one volume serial number may be specified for any SCR job control statement. The DATE and PRE parameters are not supported for diskette files.

Format

```
//[symbol] SCR lfdname[,{DATE,[yyddd]}]
                       {PRE,[aaaa]  }
```

Parameters

`lfdname`
File name to be deleted; must agree with the file name given on a previous LFD statement in a DVC-LFD sequence. If positional parameter 2 is DATE or PRE, the LBL statement may be omitted from the DVC-LFD sequence.

`DATE`
Delete all expired files from the volume. The expiration date, if different from the current date, is specified in positional parameter 3.

`PRE`
All files on the volume with the prefix specified in positional parameter 3 are to be deleted. The first three characters of the prefix cannot be Y.

If positional parameter 2 is omitted, the entire file specified by positional parameter 1 is deleted. Positional parameter 3 should also be omitted.

yyddd
: Date used in testing for expired files:

  yy
  : Is the year.

  ddd
  : Is the day of the year.

  Leading zeros must be specified. If the date is omitted, the current date from the job prologue is used.

aaaa
: The 1- to 8-character prefix of files to be deleted. The first three characters of the prefix must not be Y. If omitted, the file identifier from the associated LBL job control statement is used.

SET

Function

Sets or modifies the date fields, the user program switch indicator (UPSI), the communications region in the job prologue, or the local data area in the job prologue. Support for the user local data area is primarily for compatibility with the IBM System/34 LDA feature. It can be used as a larger communications region if so desired. The SET statement does not alter fields in the system information block (SIB); this must be done by the SET operator command.

Format

```
//[symbol] SET {DATE,yy/mm/dd[,t-date][,d-date]
              {UPSI,switch-setting
              {COMREG,character-string
              {LDA,n,n,character-string
```

Parameters

DATE
Stores the contents of positional parameters 2, 3, and 4 in the job prologue date fields. These dates are to be used rather than the system date. The date change is temporary and remains in effect until the end of the job.

yy/mm/dd
Calendar date.

t-date
5-digit date (yyddd) for a tape file:

yy
Is year.

ddd
Is day of the year.

The date is right-justified in a 6-character field in the job preamble. The leftmost character is set to an EBCDIC blank. You may, however, indicate the quarter of the year in this position.

If positional parameter 3 is omitted, the SIB date is used.

d-date
5-digit date (yyddd) for a disk file:

yy
Is the year.

ddd
Is the day of the year.

The date is converted to a 4-byte discontinuous binary format, 0ydd.

If positional parameter 4 is omitted, the positional parameter 3 value is used, if specified, or the SIB value is used if both positional parameters 3 and 4 are omitted.

UPSI
Sets the UPSI byte in the job prologue according to the bit pattern of positional parameter 2. The UPSI byte is the last byte of the 12-byte communication region in the job preamble.

switch-setting
Switch setting (one to eight bits). The bit indicators are:

0 Off
1 On
X Unchanged

Unspecified rightmost positions are assumed to be X. The byte is initially set to 0.

COMREG
Stores the contents of positional parameter 2 in the job prologue communication region.

Note: *If UPSI and COMREG are not specified, the fields in the job prologue are set to binary zeros.*

character-string
Stores 2 to 24 hexadecimal or 1 to 12 EBCDIC characters in the job preamble communications region. The data is left-justified, and unspecified rightmost characters remain unchanged. Hexadecimal characters are designated X'ccccc...' and must be even in length. EBCDIC characters are designated C'ccccc...'.

LDA

Automatically sets the OPTION LDA which sets up the LDA in the job prologue. Also allows you to store character strings in the LDA.

n

Specifies the byte at which the character string starts in the LDA. The lowest value for this parameter is 1.

m

Specifies the total number of bytes occupied by the character string. Value must be equal to or greater than the length of the string. It cannot exceed the length of the LDA or be a value that in conjunction with the *n* parameter specification extends beyond the end of the LDA.

character-string

Data stored in the LDA. The character string is stored left-justified. If it contains blanks, the character string must be enclosed in single quotes.

SFT

Function

There are three applications for the SFT statement:

- Identifying user-written shared code modules that reside in a library not on SYSRES or YRUN.
- Identifying data management shared-code modules so that those modules are loaded before job initiation and stay resident for the duration of the job. Even though you do not need to identify data management modules to load them, if you do identify them through // SFT, they are loaded *before* job initiation. This ensures that your job does not have to wait for main storage space to be allocated for the data management modules.
- Specifying the dynamic loading facility and/or overriding system generation limits for dynamic expansion of the user job region. (This feature is for ANSI 74 COBOL users.)

Format

```
//[symbol] SFT  { module-1[,...,module-n][,DLOAD=[([calls],[{expansion-limit}])]] }
                {                                          {MAX            }      }
                { DLOAD=[([calls],[{expansion-limit}])]                           }
                {                  {MAX            }      }
```

Parameters

`module`
Identifies the shared-code modules needed.

`DLOAD`
Tells the run processor that your job needs the OS/3 dynamic loading facility and/or that the job is to override the SYSGEN limits for expansion of the user job region. (For ANSI 74 COBOL users only.)

`calls`
Specifies the maximum number of dynamically loaded modules allowed for this job; if omitted, the system default (as set by the SYSGEN parameter DLOADTABLE) applies.

`expansion-limit`

Specifies the maximum number of bytes (total) that can be added to this job in support of the DLOAD facility; considered hexadecimal if you specify *X'number'*; considered decimal if you specify *D'number'*.

`MAX`

Indicates that the size of the job is limited only by the amount of main storage in the system.

If an expansion limit is omitted (*expansion-limit* or *MAX* are not specified), the system default (as set by the SYSGEN parameter DLOADBUFR) applies.

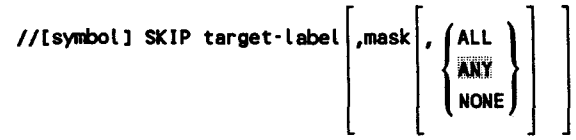
Note: *Data management shared-code load modules reside in the system library YSCLOD. The SAT librarian is used to list the modules in YSCLOD and to obtain information about them.*

SKIP

Function

Bypasses desired portions of the control stream, either conditionally, based on UPSI bit settings, or unconditionally. This skip is effective at execution time, as opposed to the IF and GO control statements, where the skip occurs when the run processor scans the control stream prior to execution. The statements that are skipped are not acted on. The skip can only be in a forward direction.

Format



Parameters

target-label
Corresponds to the label specified in the optional label field of another control statement. The label field may be on a NOP control statement.

mask
Is a binary number used to test the UPSI byte (one to eight bits). Thus, the SKIP statement is conditional. If fewer than eight bits are specified, the unspecified rightmost positions are each assumed to be 0.

The allowable characters in the mask are:

- 0 Ignore the bit.
- 1 Test the corresponding UPSI bit to determine where it is set.

If the *mask* parameter is omitted, the SKIP control statement is unconditional.

ALL
States that all the UPSI bits indicated by the mask must be set to satisfy the skip condition.

ANY
States that any UPSI bit indicated by the mask can be set to satisfy the skip condition.

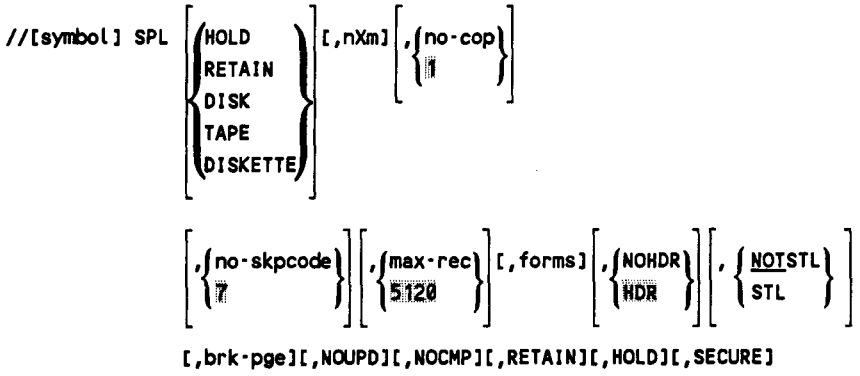
NONE
States that no UPSI bit indicated by the mask needs to be set to satisfy the skip condition.

SPL

Function

Controls output spooling. When used, SPL must occur within the DVC-LFD sequence for the file to be spooled. It is ignored in a nonspooling environment.

Format



Parameters

HOLD
Output file is to be held for later disposition. Can be released by a BEGIN SPL workstation/console command or through the CC job control statement specifying this command.

RETAIN
Output file is retained in the spool file after it is printed or punched; file may have to be reprinted or repunched at a later time.

DISK
Redirects the spooled output to another disk volume.

TAPE
Redirects the spooled output to a tape volume.

DISKETTE
Redirects the spooled output to a format-label diskette.

Note: *When using the SPL statement for a spooled data-set-label output file, only the nXm, NOUPD, NOCMP, RETAIN, and HOLD parameters are meaningful.*

nXm

Buffer size established for use by this file

n

Number of buffers. No advantage is gained by making *n* greater than 2, unless system symbionts are being used.

X

Is a constant.

m

Is the size of each buffer, in 256-byte increments.

If omitted, the file shares the buffer pool with the job log and other spooled files not having their own buffers.

{no-cop / 1}

Number of times the spooled file is to be printed or punched (output), in range of 0 through 255. Zero indicates no output.

{no-skpcode / 7}

Used when a filed vertical format buffer having more than seven skip codes is requested (via // VFB) or when the system default buffer has more than seven skip codes. Indicates the number of lines in the buffer that contain skip codes, plus three (one for forms overflow and two for home paper position). Zero indicates no skip codes.

{max-rec / 5120}

Maximum number of records (lines, including spaces and skipped lines from print files, and cards for punch files) that can be placed in the spool file before the operator is queried as to whether the job should be continued or terminated. The largest number the *max-rec* parameter will accept is 262144.

forms

A 1- to 8-alphanumeric-character name identifying a special printer form or card type to the operator.

NOHDR

Suppresses the printing of page headers in burst mode.

HDR

Allows page headers to be printed.

NOTSTL
: Suppresses the query to the operator questioning if a sample test pattern page is to be printed when a change of form name is detected.

STL
: Sends a test lines message to the operator when a forms change is required. If *STL* and *NOTSTL* are omitted, the system default is used.

brk-pge
: Indicates the specific number of pages or cards to be spooled out before a file is breakpointed and printed or punched. The largest number that can be entered in this parameter is 32,000.

NOUPD
: Indicates that the spool subdirectory entry is to be updated only when a file is closed. If NOUPD is specified and the program cancels, any output generated before cancellation is lost. If omitted (causing the spooler to update the subdirectory entry each time it crosses a logical track in the program file) and the program cancels, output generated before cancellation can be printed.

NOCMP
: Prevents the system from attempting to compress data that's directed to the output spool file. Normally, NOCMP should not be specified if data in the file contains a large number of embedded blanks or if the file contains a block size larger than 120. If specified when the block size is 121 or greater, the resulting spool file contains only one line per sector and requires that nXm be at least 2X4.

RETAIN
: Functions the same as the first *RETAIN* parameter but is independent of the other first-parameter options. If specified with *TAPE, DISK,* or *DISKETTE,* the output is redirected to the appropriate device while a copy of the file is also retained for later use.

HOLD
: Functions the same as the first *HOLD* parameter but is independent of the other first-parameter options. If specified with *RETAIN; TAPE, DISK,* or *DISKETTE*; or *RETAIN* and *TAPE, DISK,* or *DISKETTE,* the spool file is put on hold first. Other parameters are acted upon accordingly when the file is released.

SECURE
: Suppresses the printing of the output file if the workstation to which the auxiliary printer is connected is not logged on. If not specified, the file is printed at the specified auxiliary printer whether or not the workstation is logged on.

UID

Function

Allows you to specify a workstation, including the system master workstation, by user-id, device address, or both. The workstation specified is required and, when logged on, is automatically connected to the job. Used as part of the device assignment set for a workstation.

Format

```
//[symbol]UID {user-id-1          }[,...,{user-id-255          }]
              {(addr-1)             }      {(addr-255)          }
              {user-id-1(addr-1)    }      {user-id-255(addr-255)}
```

Parameters

`user-id`
Specifies a workstation by user-id. You can use YMAS as a user-id to identify the system master workstation.

`(addr)`
Specifies the physical address of the device in hexadecimal; represents the channel number, control unit address, and the device number.

`user-id(addr)`
Specifies a workstation by user-id and physical address.

USE DP

Function

Specifies that the dialog processor is needed in support of a dialog session at the workstation. Used as part of the device assignment set for a workstation.

Format

```
//[symbol]USE DP,dialog-name[,printer-lfd][,new-audit-lfd][,old-audit-lfd]
```

Parameters

`dialog-name`
A 1- to 8-alphanumeric-character name of the dialog file; must match the name specified on the LFD statement of the dialog file's device assignment set.

`printer-lfd`
A 1- to 8-alphanumeric-character name of the printer file; must match the name specified on the LFD statement of the printer's device assignment set. This parameter is specified when you want to produce a printed summary of the dialog session.

`new-audit-lfd`
A 1- to 8-alphanumeric-character name of the new audit file produced by the dialog processor; must match the name specified on the LFD statement of the new audit file's device assignment set.

`old-audit-lfd`
A 1- to 8-alphanumeric-character name of the old audit file input to the dialog processor; must match the name specified on the LFD statement of the old audit file's device assignment set.

Note: *Audit files must be previously allocated MIRAM files.*

USE LIB

Function

Allows a user program to either sequentially read or write (create) a source module. Specified in the device assignment set for a source library file.

Format

```
//[symbol] USE LIB,module-name
```

Parameters

`module-name`

A 1- to 8-alphanumeric-character name of the source module to be accessed. The first character must be alphabetic.

USE MENU

Function

Indicates that menu services are to be used at the workstation. Specified as part of the workstation device assignment set.

Format

```
//[symbol] USE MENU [,{menu-file-LFD/$Y$FMT }] [,initial-menu]
                     {$Y$FMT/menu-file-LFD}
                     {$Y$FMT              }

                  [,{nnn}] [,menu-format-1=alias-1[,...,menu-format-12=alias-12.]
                    {1  }
```

Parameters

```
[,{menu-file-LFD/$Y$FMT}]
  {$Y$FMT/menu-file-LFD}
  {$Y$FMT              }
```

Specifies LFD names for up to two menu format files; must match LFD names used in previously defined device assignment sets for the files. The menu-file-LFD name is one to eight alphanumeric characters in length and must refer to a MIRAM file.

`menu-file-LFD/$Y$FMT`
: Identifies two format files. The file identified by menu-file-LFD is examined first, then YFMT.

`$Y$FMT/menu-file-LFD`
: Identifies two format files. YFMT is examined first, then the file identified by menu-file-LFD.

If nothing is specified, it is assumed that all menu formats reside in the system format file YFMT.

`initial-menu`
: Specifies the name of the first menu to be used. It is one to eight alphanumeric characters in length.

`nnn`
: Specifies the number of menus to be resident in main storage at any one time, in the range 1 to 255. The default value is 1.

USE SFS

Function

Specifies that screen format services are needed in support of a screen format session at the workstation. Used as part of the device assignment set for a workstation.

Format

```
//[symbol] USE SFS [,{[format-file-LFD-1]/[format-file-LFD-2]}] [,initial-screen]
                   {format-file-LFD                         }
                   {$Y$FMT                                  }

[,{nnn}] ,screen-format-1=alias-1[,...,screen-format-12=alias-12].
  {1  }
```

Parameters

```
[,{[format-file-LFD-1]/[format-file-LFD-2]}]
  {format-file-LFD                         }
  {$Y$FMT                                  }
```

Specifies LFD names for up to two screen format files; must match LFD names used in previously defined device assignment sets for the screen format files. The format-file-LFD name is one to eight alphanumeric characters in length and must refer to a MIRAM file.

`[format-file-LFD-1]/[format-file-LFD-2]`

Identifies two format-file-LFD names. If *format-file-LFD-1* is coded alone, *format-file-LFD-1* is examined first then YFMT. If */format-file-LFD-2* is coded alone, YFMT is examined first then *format-file-LFD-2.*

`format-file-LFD`

Identifies one format-file-LFD name. This is the only file examined.

If nothing is specified, it is assumed that all screen formats reside in the system format file YFMT.

`initial-screen`

Specifies the name of a screen format to be used by the application program. It is one to eight alphanumeric characters in length. Use of this parameter depends on the program's language. For more information, see the *Screen Formatting Services Technical Overview* (UP-9977).

`nnn`

Specifies the number of screens to be resident in main storage at any one time, in the range 1 to 255. The default value is 1.

`screen-format=alias`

Equates a screen format name specified in an application program (alias) to the screen format name generated by the screen format generator. A maximum of twelve alias name sets may be specified. The *screen-format name* and *alias name* may each be from one to eight alphanumeric characters in length.

VFB

Function

Overrides the system default vertical format buffer for a printer; becomes effective when the file is opened. During interstep-job processing, the SYSGEN default value is used. If used, the VFB directive must be within the DVC-LFD control statement sequence for the print file that uses it.

Format

```
//[symbol] VFB [,FORMNAME=symbol][,USE={STAND1 }][LENGTH=lines][,DENSITY={6}]
                                     {OWNVF1  }                      {8}
                                     {OWNVF2-OWNVF9}
                                                                (SDMA PRINTERS ONLY)

      [,TYPE={SDMA}][,OVF=(line-1,...,line-n)][,OVF2=(line-1,...,line-n)]
             {0770}
             {0776}

      [,CD1=(line-1,...,line-n),...[,CD15=(line-1,...,line-n)]
```

Label

symbol

Specifies 1- to 8-character vertical format buffer name; used as the default form name when the *FORMNAME* parameter is omitted.

Specifies a default form name when *FORMNAME* is omitted or specifies the name of a filed vertical format buffer (STAND1 or OWNVFn) you're changing via the job SG$PRB.

Parameters

FORMNAME=symbol

Specifies a 1- to 8-character printer form name. This is the name that the operator associates with the actual form. He is asked to mount this form before you access the print file. A form name specified through this parameter takes precedence over a form name specified in *symbol*. Form names specified through VFB statements take precedence over form names specified through the SPL job control statement or the SPOOL jproc.

USE= { STAND1 / OWNVF1 / OWNVF2-OWNVF9 } (for SDMA printers only)
Name of a filed vertical format buffer established at SYSGEN time or via the job SG$PRB. If used, *FORMNAME* and *TYPE* are the only other parameters you can specify. When the job SG$PRB is being used to change a filed buffer, specify the buffer name in *symbol* instead of *USE*.

LENGTH=lines
Total length of printer form in terms of lines, in the range of 1 to 192.

DENSITY={6 / 8}
Number of print lines per inch. If specified, *LENGTH* must also be specified.

TYPE={SDMA / 0770 / 0776}
Type of printer to be used. SDMA is the only valid type for models 3, 4, 5, and 6. Only the model 8 can accommodate the 0770 and non-SDMA 0776 printers. If the VFB is designed to be used with more than one type of printer, the *TYPE* parameter should be omitted.

Note: *In the following keyword parameters,* line *is a decimal number between 1 and the number specified in the LENGTH keyword parameter. It specifies which line number is to be assigned a skip code (overflow or device control code). A particular code may be repeated if desired, or multiple lines for a given code may be specified as one parameter. If only one line is specified for a given code, the enclosing parentheses may be omitted. If multiple codes for a given line are used, the first is accepted and a diagnostic error message is given on all others.*

OVF=(line-1,...,line-n)
Overflow line indicator.

OVF2=(line-1,...,line-n)
Secondary overflow line indicator.

CD1=(line-1,...,line-n),...,CD15=(line-1,...,line-n)
Device control codes for the printer. They indicate which lines control certain functions, such as skipping and line spacing. See the *Consolidated Data Management Programming Guide* (UP-9978) for a list of the applicable codes for each printer.

Notes:

1. *In a spooling system, the number of skip codes being used must be passed to the open processors. If a VFB job control statement is present in the device assignment set for a file being spooled, the number of codes specified is automatically passed without the necessity of an SPL job control statement. However, if you request a filed vertical format buffer (STAND1 or OWNVF1) having more than seven skip codes or if you use a system default buffer having more than seven skip codes, you must specify the number of codes using the* no-skpcode *parameter in the / / SPL statement or the* SKPCODE *parameter in the / / SPOOL jproc.*

   When a / / SPL statement or / / SPOOL jproc is omitted, the default is 7 skip codes. Three skip codes are always included in this count: home position for current page, overflow for next page, and home position for next page. The four remaining codes are user-specified. The / / SPL statement and / / SPOOL jproc, therefore, specify the total count of lines on a form where a skip code is allowed, plus three.

2. *For 0776 printers, the CLASS= parameter should be used if a unique logical unit number is required.*

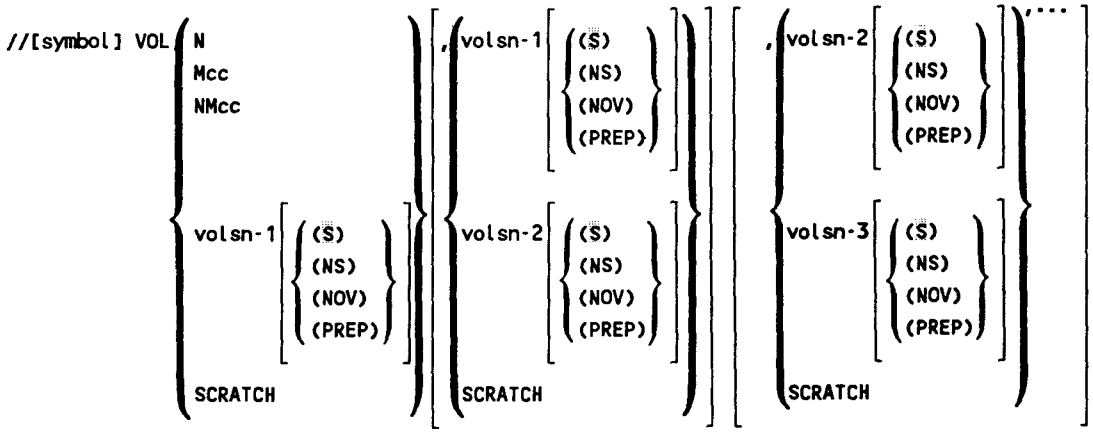
VOL

Function

Supplies the volume serial numbers used to uniquely identify tape and disk volumes.

When used, the VOL statement must immediately follow the DVC statement in a device assignment set. For a disk file, the first VOL statement must precede any EXT statements for that volume.

Format



Parameters

N

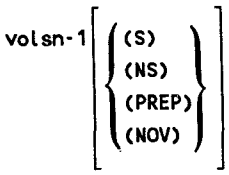
Suppress block number checking on input tape volumes or block number writing on output tape volumes.

If N is not used, tape block numbering depends upon the SYSGEN specification and the BKNO parameter of the DTFMT macro. The N parameter is effective only during initialized processing.

Mcc

Mode characteristics of the tape volumes specified in positional parameters 2 through *n*. The allowable values for *cc* are given in the *Consolidated Data Management Programming Guide* (UP-9978). If *Mcc* is not specified, the mode settings specified at system generation time are used.

NMcc
Combines both the *N* and *Mcc* options.



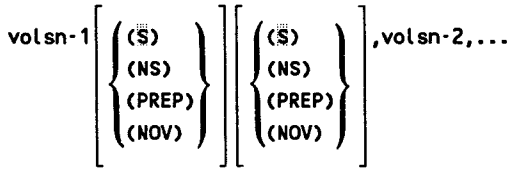
One to six alphanumeric characters specifying the volume serial number of the first volume of the file. If less than six characters are used, the number is left-justified and padded with blanks. The volume may be shareable (S) or nonshareable (NS), and volume checking may be suppressed (NOV). For tape files, a request to write volume labels may be made (PREP). Any information that is currently on the volume is effectively erased if PREP is specified.

SCRATCH
Used to specify a multivolume file without explicitly listing the serial numbers for each volume. May appear only once in a VOL statement and is always the last parameter specified.

Notes:

1. *OS/3 assumes that all volume serial numbers are unique. The loading of two volumes with identical volume serial numbers at the same time will yield unpredictable results.*

2. *Extreme care should be taken when using the PREP option on a file to be processed by the librarian. When PREP is specified, the tape is prepped every time it is opened as output. The librarian closes output tape files whenever they are to be used as input and then reopens them as input. If this tape file is to be reused as an output file within the same job, the librarian will close it as input and reopen it as output. This reopening will cause the file to be prepped (if PREP was specified), thereby effectively erasing all the information previously produced. The PREP option, therefore, should only be used if the file is going to be output only or output, then input. If this is not the case, use the TPREP utility routine to prep the file. The PREP option cannot be suppressed. You must redefine the tape file without specifying the PREP option on the VOL statement.*



Continuation of a variable amount of volume serial numbers; must appear in the same sequence required for mounting.

Notes:

1. *To suppress checking of volume serial numbers for multivolume files, you must specify* NOV *in the VOL statement for the last volume of the file.*

2. *For multivolume files, if* PREP *is specified for any of the volumes, all volumes in the file are prepped.*

3. *SCRATCH lets you mount additional tape volumes (unlimited processing); however, these additional volumes are not prepped if* PREP *is specified. If they must be initialized, use the TPREP utility routine.*

4. *When referencing multivolume files on the VOL statement, you must represent any undeclared volume serial numbers with commas. Additionally, if neither* Mcc, N, *or* NMcc *are specified for the first positional parameter, you must supply a comma.*

/$

Function

Start of data. All statements following the /$ statement up to and including the associated end-of-data (/*) statement are filed in YRUN.

The /$ must be the only statement on the line; however, comments may follow the /$ if preceded by one or more blanks.

Format

```
/$
```

No positional parameters are required.

Note: *For System 80, model 8 users: If using a Unisys 0716 card reader subsystem with the 96-column card feature supported, the data file can use all 96 columns. With data-set-label diskette, the data file can use up to 128 characters. Job control statements, however, can only use the first 72 columns (characters).*

/*

Function

End of data; used in conjunction with the start-of-data (/$) statement. When the /* statement is encountered by job control, the data file is closed and control stream processing is resumed. The /* statement is filed with the data.

The /* must be the only statement on the line; however, comments may follow the /* if preceded by one or more blanks.

Format

```
/*
```

No positional parameters are required.

/&

Function

End of a control stream.

The /& statement must be the only statement on the line; however, comments may follow the /& if they are preceded by one or more blanks.

Format

```
/&
```

No positional parameters are required.

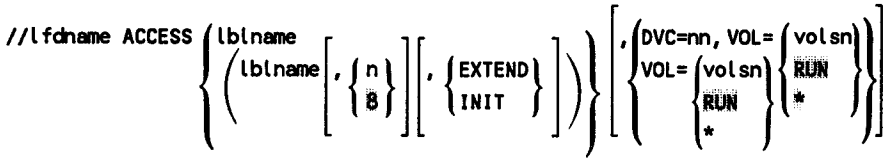
Section 3
Job Control Procs

ACCESS

Function

Generates the job control statements required to assign a device to a job so that the file on that device can be accessed at job execution time. The procedure call statement is included in the control stream at the point where the device assignment set is required. This statement can be used for tape files and previously allocated disk files.

Format



Label

lfdname
: Name of the file to be accessed. This corresponds to the 8-character name in the LFD job control statement.

Parameters

lblname
: File identifier under which the file was created. This corresponds to the tape or disk file identifier in the LBL job control statement.

n
: Number of extents in the file. Reserves extent table storage for use by the data access methods. The maximum value is 16. Space acquired by using this parameter increases the total storage requirement for the job and must be taken into consideration when allocating main storage.

EXTEND
Adds information to the present end of a tape file.

INIT

Initializes the file, starting with the first record each time the file is opened. Previous control information in the format labels is ignored at file open time and is overwritten by specifications contained in this DVC-LFD sequence at file close time.

DVC=nn
Device type, where nn is the logical unit number of the device.

If this keyword parameter is omitted, the file is assumed to be on the device containing the job's YRUN file.

VOL= { volsn / RUN / * }
Identifies the volume where the file resides. The options are:

VOL=volsn
Volume serial number.

VOL=*
Volume is identified in the file catalog.

If this keyword parameter is omitted, the file is assumed to be in the job's YRUN file.

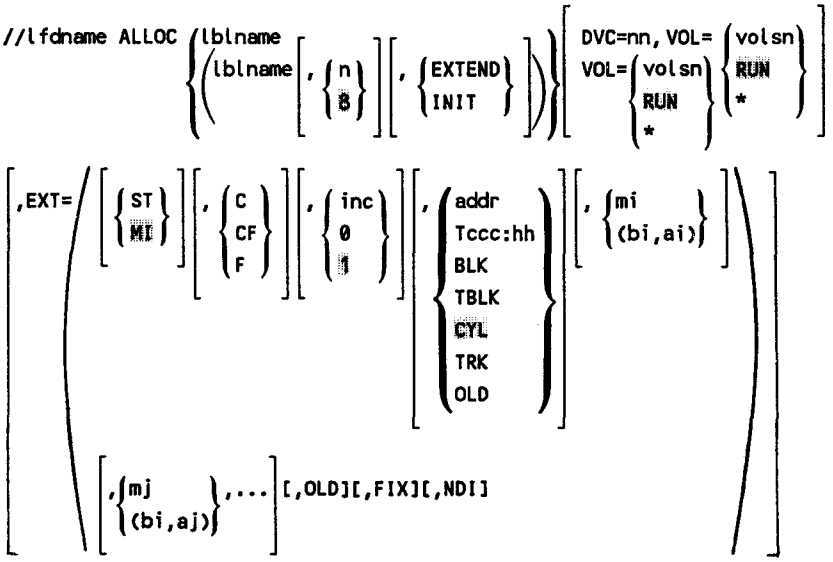
ALLOC

Function

Generates the job control statements required to assign a disk or diskette to a job step and to allocate space on that disk or diskette for the file. This statement is placed in the control stream where the device assignment set is required. Use it to create new files or extend current files.

See Appendix B for further information regarding extent specification.

Format



Label

lfdname
: Identifies the file; corresponds to the 8-character name in the LFD control statement.

Parameters

lblname
: File identifier used when the file was created; corresponds to the disk file identifier in the LBL job control statement.

n
: Number of extents in the file. Reserves extent table storage for use by the data access methods. The maximum value is 16. Space acquired by using this parameter increases the total storage requirement for the job and must be taken into consideration when allocating main storage.

EXTEND
Adds information to the end of a sequential file.

INIT
Initializes the file, starting with the first record each time the file is opened. Previous control information in the format labels is ignored at file open time and is overwritten by specifications contained in this DVC-LFD sequence at file close time.

DVC=nn
Device type, where nn is the logical unit number of the device.

If this keyword parameter is omitted, the file is assumed to be on the device containing the job's YRUN file.

VOL= { volsn | RUN | * }
Identifies the volume where the file resides. The options are:

VOL=volsn
Volume serial number.

VOL=*
Volume is identified in the file catalog.

If this keyword parameter is omitted, the file is assumed to be in the job's YRUN file.

EXT=
Supplies the extent specifications to be used when reserving system resources for a particular file. The information is coded within parentheses.

If this keyword parameter is completely omitted, job control allocates one cylinder of extent space for a MIRAM *(MI)* file and assumes one cylinder for dynamic extension.

ST
System access technique (SAT) file.

MI
MIRAM file; only valid specification for data-set-label diskette.

Notes:

1. *Data files must be allocated as MIRAM files.*

2. *All parameters that apply to disk (except* FIX*) apply to format-label diskette.*

`C`

Allocate contiguous space; must be specified for data-set-label diskette files.

`F`

Reformat the file at allocation time. Subparameter 4 must be *BLK*.

`CF`

Both *C* and *F*.

`inc`

Secondary increment (in cylinders or blocks) for automatic extension. An increment of zero indicates that there can be no dynamic extension of the file. If there is no *EXT* keyword parameter for this file, the value of the most recently specified secondary increment is used.

`0`

File cannot be dynamically extended. Must be specified for data-set-label diskette.

`addr`

Absolute beginning cylinder address, in terms of cylinders.

`Tccc:hh`

Absolute track address (hexadecimal) in cylinder/head format where the file is to begin. Allocation is in terms of tracks. Not to be used for allocating a disk file or format-label diskette when creating MIRAM files with IRAM characteristics.

`BLK`

Allocation in terms of blocks; actual allocation is in terms of cylinders; only valid specification for data-set-label diskette files.

`TBLK`

Allocates space in blocks; actual allocation is in terms of tracks. Not to be used for allocating a disk file or format-label diskette when creating MIRAM files with IRAM characteristics.

CYL
: Allocation in terms of cylinders.

TRK
: Allocates space in tracks. Not to be used for allocating a disk file or format-label diskette when creating MIRAM files with IRAM characteristics.

OLD
: Secondary allocation increment change (subparameter 3) for an existing file; cannot be followed by any other subparameters if specified; cannot be specified for data-set-label diskette.

mi
: Number of cylinders or tracks to allocate for this file; subparameter 4 must be *CYL*, *addr*, *TRK*, or *Tccc:hh*.

(bi,ai)
: Used when allocation is in terms of blocks (by cylinder or track) for disk and diskette files; subparameter 4 must be *BLK* or *TBLK*.

bi
: Is the average block length.

ai
: Is the number of blocks to allocate.

Subparameter 6

Same as subparameter 5 but used to describe additional extents; never specified for data-set-label diskette files.

OLD
: Indicates that the file's previously allocated extent is to be increased by the allocation amount *(mi, (bi,ai)*, etc.) specified.

  If OLD is omitted, the request is for a new extent; it may not be specified for data-set-label diskette.

NDI
: NDI (nondata interchange) must be specified to allocate space for all data-set-label diskette files that do not use the basic data exchange (BDE).

FIX
: Indicates you're allocating this extent in the fixed-head area of an 8417 disk.

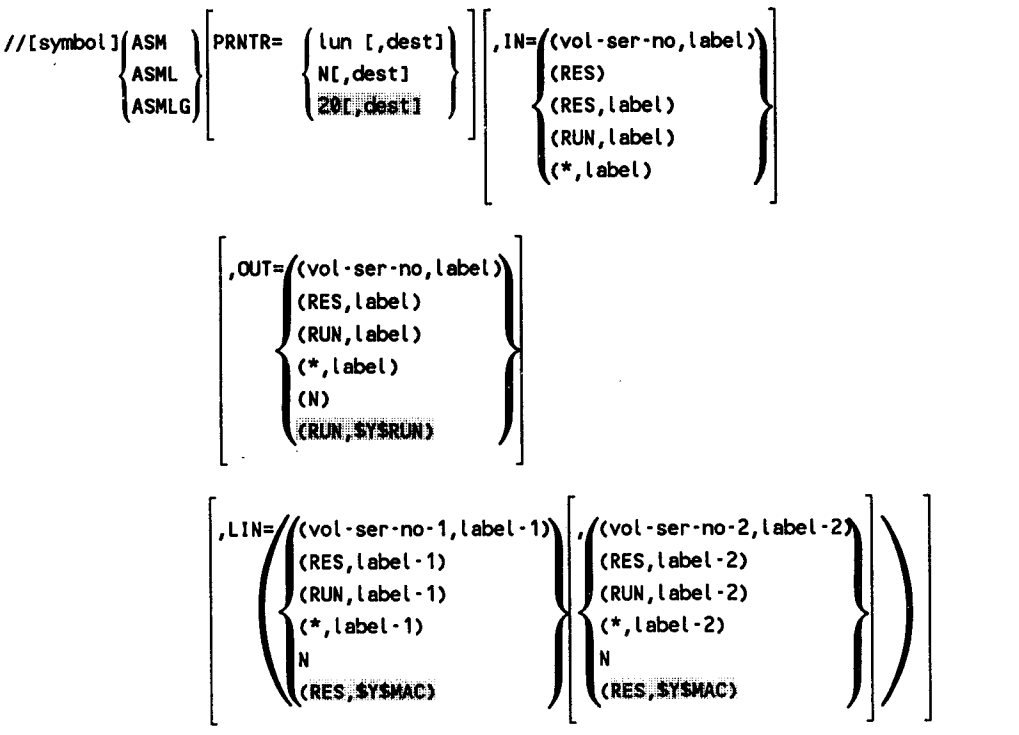
ASM

Function

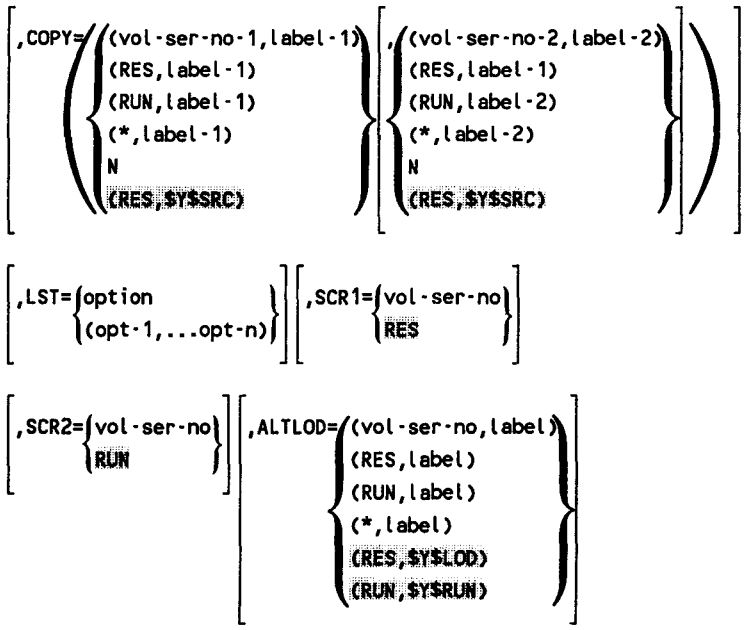
Generates the necessary job control statements to run an assembly language processor. Optionally, it can generate the job control statements that specify the following:

- Input source library
- Output object library
- Procedure, copy, and alternate load libraries
- PARAM job control statements to define the format of the assembler listing
- OPTION job control statements to automatically execute the linkage editor and the generated load module

Format



continued



Note: *The ASM procedure call is used to assemble a program. When the ASML procedure call is used, an OPTION LINK job control statement is generated, which allows for the automatic execution of the linkage editor. When the ASMLG procedure call is used, an OPTION LINK,GO job control statement is generated, which allows for the automatic execution of the linkage editor, followed by the automatic execution of the program.*

When either the ASML or ASMLG procedure calls are used, the OUT parameter to define a specific output library cannot be used.

Label

symbol
: Source module name of one to eight alphanumeric characters; only needed when the *IN* parameter is used.

Parameters

```
PRNTR= { lun[,dest] }
       { N[,dest]   }
       { 20[,dest]  }
```

Specifies a printer. The options are:

`PRNTR=lun[,dest]`
: Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric character identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

`PRNTR=N[,dest]`
: Indicates that you are supplying your own device assignment set for the printer; a DVC-LFD sequence is not to be generated. This option allows use of the // LCB and // VFB statements. A 1- to 6-alphanumeric character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

  If omitted, a local printer with a logical unit number of 20 is assumed.

```
IN={ (vol-ser-no,label) }
   { (RES)              }
   { (RES,label)        }
   { (RUN,label)        }
   { (*,label)          }
```

Input file definition; also replaces the PARAM IN control statement. The options are:

`IN=(vol-ser-no,label)`
: Indicates the volume serial number and file identifier.

`IN=(RES)`
: Input is on SYSRES, in YSRC.

`IN=(RES,label)`
: Input is on SYSRES, but in a file other than YSRC.

`IN=(RUN,label)`
: Input is on the volume containing the job's YRUN file, with the file identifier specified by *label*.

IN=(*,label)
Input volume is identified in the file catalog with a file identifier specified by *label.*

If omitted, source input is in the form of data cards.

Note: *The IN parameter must be used if the source correction feature of the assembler (i.e., SKI, REC, SEQ type cards) is to be used instead of strictly source cards.*

OUT={(vol-ser-no,label) | (RES,label) | (RUN,label) | (*,label) | (N) | (RUN,YRUN)}
Output file definition; also replaces the PARAM OUT control statement. The options are:

OUT=(vol-ser-no,label)
Indicates the volume serial number and file identifier.

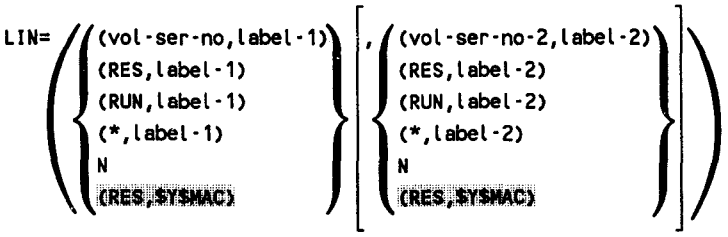
OUT=(RES,label)
Output is on SYSRES, with a file identifier specified by *label.*

OUT=(RUN,label)
Output is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

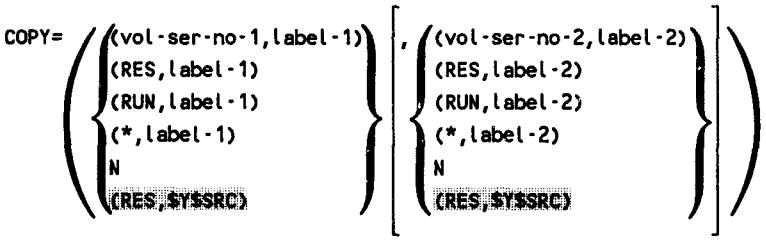
OUT=(*,label)
Output volume is identified in the file catalog with a file identifier specified by *label.*

OUT=(N)
No object code output.

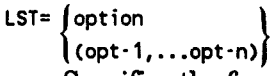
If omitted, output is in the job's YRUN file.



Indicates the location of the private procedure and macro libraries. RES indicates a library on SYSRES; RUN indicates a library on the volume containing the job's YRUN file; the * indicates the volume is identified in the file catalog. The file identifier is specified by *label.* N indicates no libraries. If omitted, the library is on SYSRES in the file YMAC.



Indicates the location of the copy libraries. RES indicates a library on SYSRES; RUN indicates a library on the volume containing the job's YRUN file; the * indicates the volume is identified in the file catalog. The file identifier is specified by *label.* N indicates no libraries. If omitted, the library is on SYSRES in the file YSRC.



Specifies the format of the assembler listing. The options are:

N

Specifies that no assembly listing is produced.

NC

Specifies that no cross-reference listing is produced.

ND

Specifies that no diagnostic listing is produced.

NR

Specifies that the cross-reference listing is to contain only symbols that have at least one reference each. The NC option, if specified with NR, always overrides it.

DBG

Specifies a proc or macro debug mode feature within the OS/3 assembler.

(NC,ND)
Specifies that neither a cross-reference nor diagnostic listing is produced.

(NR,ND)
Specifies that no diagnostic listing is produced and the cross-reference listing contains only symbols with references.

SCR1={vol-ser-no / RES}
Specifies the volume serial number of the first work file.

SCR2={vol-ser-no / RUN}
Specifies the volume serial number of the second work file.

ALTLOD={(vol-ser-no,label) / (RES,label) / (RUN,label) / (*,label) / (RES,YLOD) / (RUN,YRUN)}
Specifies the alternate load library that contains the assembler language processor. The options are:

ALTLOD=(vol-ser-no,label)
Indicates the volume serial number and file identifier.

ALTLOD=(RES,label)
Alternate load library is on SYSRES, with a file identifier specified by *label.*

ALTLOD=(RUN,label)
Alternate load library is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

ALTLOD=(*,label)
Alternate load library is on the volume identified in the file catalog, with a file identifier specified by *label.*

If omitted, the alternate load library is in YLOD when ASM and ASML are used and YRUN when ASMLG is used.

AUTO

Function

Generates the necessary job control statements to run RPG II auto report. The AUTO jproc has four forms:

AUTO
: Processes auto report source programs.

AUTRPG
: Processes auto report source programs and then compiles the RPG II generated source program.

AUTRPGL
: Processes auto report source programs, compiles the RPG II-generated source program, and link-edits the generated object module.

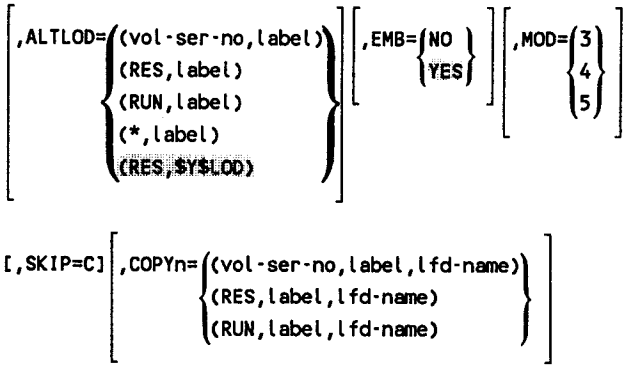
AUTRPGLG
: Processes auto report source programs, compiles the RPG II-generated source program, link-edits the generated object module, and executes the generated load module.

Format

```
//[symbol] { AUTO    } [ PRNTR= { lun[,dest] } ] [ ,IN= { (vol-ser-no,label) } ]
           { AUTRPG  }         { N[,dest]   }          { (RES)             }
           { AUTRPGL }         { 20[,dest]  }          { (RES,label)       }
           { AUTRPGLG}                                 { (RUN,label)       }
                                                       { (*,label)         }

[ ,OUT= { (vol-ser-no,label) } ]
        { (RES,label)        }
        { (RUN,label)        }
        { (*,label)          }
        { (N)                }
        { (RUN,$YSRUN)       }

[ ,OUTSRC= { (vol-ser-no,label,lfd-name,module-name) } ]
           { (RES,label,lfd-name,module-name)        }

[ ,LST= { K } ] [ ,SCR1= { vol-ser-no } ] [ ,SCR2= { vol-ser-no } ]
        { M }            { RES        }           { RUN        }
        { N }
        { S }
```

continued

(cont.)



Label

symbol
: Source module name of one to six alphanumeric characters; needed only when the IN parameter is used.

Parameters

PRNTR=lun[,dest]
N[,dest]
20[,dest]

Specifies a printer. The options are:

PRNTR=lun[,dest]
: Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

PRNTR=N[,dest]
: Indicates that you are supplying your own device assignment set for the printer; a DVC-LFD sequence is not to be generated. This option allows use of the // LCB and // VFB statements. A 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

  If omitted, a local printer with a logical unit number of 20 is assumed.

```
IN=(vol-ser-no,label)
   (RES)
   (RES,label)
   (RUN,label)
   (*,label)
```

Input file definition; it also replaces the PARAM IN control statement. The options are:

`IN=(vol-ser-no,label)`
Indicates the volume serial number and file identifier.

`IN=(RES)`
Input is on SYSRES in YSRC.

`IN=(RES,label)`
Input is on SYSRES, but in a file other than YSRC.

`IN=RUN,label)`
Input is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

`IN=(*,label)`
Input is on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, source input is in the form of data cards.

```
OUT=(vol-ser-no,label)
    (RES,label)
    (RUN,label)
    (N)
    (RUN $Y$RUN)
```

Output file definition; also replaces the PARAM IN control statement. The options are:

`OUT=(vol-ser-no,label)`
Indicates the volume serial number and file identifier.

`OUT=(RES,label)`
Output is on SYSRES, with a file identifier specified by *label.*

`OUT=(RUN,label)`
Output is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

`OUT=(*,label)`
Output is on the volume identified in the file catalog with a file identifier specified by *label.*

OUT=(N)
No object code output.

If omitted, output is in the job's YRUN file.

OUTSRC= {(vol-ser-no,label,lfd-name,module-name) / (RES,label,lfd-name,module-name)}
Output source file definition for the cataloged source file option on the auto report U specification. The options are:

OUTSRC=(vol-ser-no,label,lfd-name,module-name)
Indicates the volume serial number, file identifier *(label)*, logical file name *(lfd-name)*, and module-name of the output source file.

OUTSRC=(RES,label,lfd-name,module-name)
Indicates an output source file on SYSRES with a file identifier specified by *label*, a logical file name specified by *lfd-name*, and a module-name for the file.

If omitted, there is no output file definition for the cataloged source file option.

LST=(K / M / N / S)
Specifies the format of the compiler listing.

K
Inhibits error flags for sequence errors.

M
Inhibits source and diagnostic listings.

N
Inhibits all listable output.

S
Inhibits main storage map listing.

If omitted, the complete compiler listing is printed.

SCR1={vol-ser-no / RES}
Specifies the volume serial number of the first scratch work file.

SCR2={vol-ser-no / RUN}
Specifies the volume serial number of the second scratch work file.

```
ALTLOD=(vol-ser-no,label)
       (RES,label)
       (RUN,label)
       (*,label)
       (RES,SY$LOD)
```

Specifies the alternate load library containing the auto report product (AUTO#). The options are:

`ALTLOD=(vol-ser-no,label)`
Indicates the volume serial number and file identifier (label).

`ALTLOD=(RES,label)`
Alternate load library is on SYSRES; a file identifier is specified by *label.*

`ALTLOD=(RUN,label)`
Alternate load library is on the volume containing the job's YRUN file; a file identifier is specified by *label.*

`ALTLOD=(*,label)`
Alternate load library is on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, the AUTO# load module is located on SYSRES in YLOD.

```
EMB=NO
    YES
```

Indicates whether embedded linkage editor control statements are to be generated.

```
MOD=3
    4
    5
```

`MOD=3`
Specifies an RPG II compilation in IBM System/34 mode. When this mode is specified, MIRAM will be used to process all disk files; the logical file definition (LFD) for printer files is changed to PRNTR, PRNTR1...PRNTRn; and the control reader (CTLRDR) is used for card input even though the data management reader (READER) is specified.

`MOD=4`
Same as MOD=3 except that printer files are generated with the same names as used in the program and reader files use the data management reader (READER).

MOD=5
The program is to be compiled in the IBM®System/34 mode with native mode data management accessing the disk files.

SKIP=C
Causes auto report to generate SKIP to channel for the printer output file. If omitted, auto report generates SKIP to line numbers for the printer output file.

COPYn={(vol-ser-no,label,lfd-name) / (RES,label,lfd-name) / (RUN,label,lfd-name)}
Input source file definition for COPY statements within auto report source input. The number of copies corresponds to the number of unique files used to copy.

COPYn=(vol-ser-no,label,lfd-name)
Indicates the volume serial number, the file identifier *(label),* and the logical file name *(lfd-name).*

COPYn=(RES,label,lfd-name)
Indicates an input source file on SYSRES with a file identifier specified by *label* and a logical file name specified by *lfd-name.*

COPYn=(RUN,label,lfd-name)
Indicates an input source library in the job's YRUN file with a file identifier specified by *label* and a logical file name specified by *lfd-name.*

If omitted, there is no input source file definition for COPY statements.

ERRFIL=(vol-ser-no,label,module-name)
Specifies an error log file. Error diagnostic messages are written to this file by the error log processor. The parameter *vol-ser-no* specifies the volume serial number, *label* specifies the file identifier, *module-name* specifies the 1- to 8-alphanumeric-character name of the error file element being created. (Does not need to match or be related to the RPG II source object module name.)

IBM is a registered trademark of International Business Machines Corporation.

COBL74

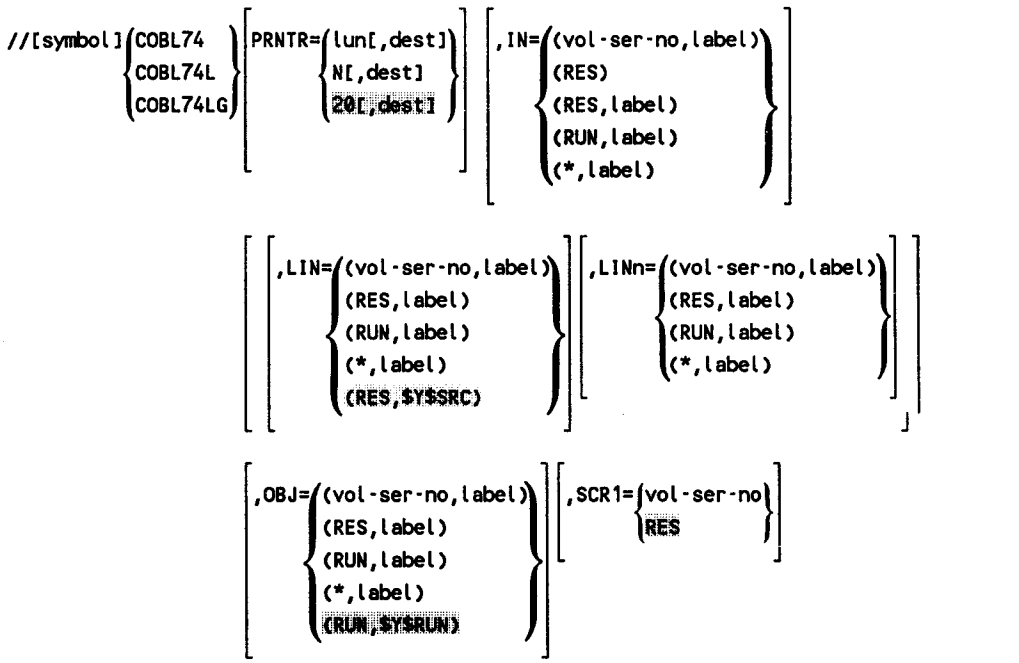
Function

Generates the job control statements to execute the ANSI 1974 COBOL language processor. The COBL74 jproc call has three forms:

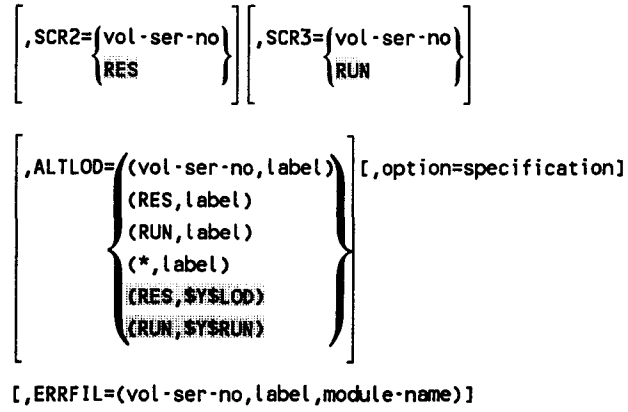
- COBL74: Compiles an ANSI 1974 COBOL source program.
- COBL74L: Compiles an ANSI 1974 COBOL source program and link-edits object modules.
- COBL74LG: Compiles an ANSI 1974 COBOL source program, link-edits object modules, and executes the load module.

Note: *COBL74, COBL74L, and COBL74LG are the only COBOL jprocs supported in System 80.*

Format



continued



Label

symbol
: Source module name of one to six alphanumeric characters. Required only when the IN parameter is specified.

Parameters

PRNTR={lun[,dest] / N[,dest] / 20[,dest]}
: Specifies a printer. The options are:

  PRNTR=lun[,dest]
  : Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric-characer identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

  PRNTR=N[,dest]
  : Indicates that you are supplying your own device assignment set for the printer; a DVC-LFD sequence is not to be generated. This option allows use of the // LCB and // VFB statements. A 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

  If omitted, a local printer with a logical unit number of 20 is assumed.

```
IN=(vol-ser-no,label)
   (RES)
   (RES,label)
   (RUN,label)
   (*,label)
```

Input file definition; also generates a PARAM IN job control statement. The options are:

`IN=(vol-ser-no,label)`
Specifies the file identifier (*label*) and the volume serial number for the input file.

`IN=(RES)`
Input is on SYSRES in YSRC.

`IN=(RES,label)`
Input is on SYSRES, with the file identifier specified by *label.*

`IN=(RUN,label)`
Input is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

`IN=(*,label)`
Input volume is identified in the file catalog with a file identifier specified by *label.*

Note: *If this parameter is omitted, source input is in the form of data cards.*

```
LIN=(vol-ser-no,label)
    (RES,label)
    (RUN,label)
    (*,label)
    (RES,$Y$SRC)
```

Specifies the location of the copy modules. The lfdname is COPY$. The options are:

`LIN=(vol-ser-no,label)`
Specifies the file identifier (*label*) and the volume serial number.

`LIN=(RES,label)`
Modules are on SYSRES; the file identifier is specified by *label.*

`LIN=(RUN,label)`
Modules are on the volume containing the job's YRUN file; a file identifier is specified by *label.*

`LIN=(*,label)`

Modules are on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, modules are on SYSRES in the file YSRC.

`LINn=` { `(vol-ser-no,label)` | `(RES,label)` | `(RUN,label)` | `(*,label)` }

Specifies the location of additional copy libraries using LINn, where n equals 1 through 9 (i.e., LIN1=, LIN2=,...,LIN9=). The compiler searches the libraries in the order LIN, LIN1, LIN2, up to LIN9.

`OBJ=` { `(vol-ser-no,label)` | `(RES,label)` | `(RUN,label)` | `(*,label)` | `(RUN,$Y$RUN)` }

Specifies the output file definition and generates a PARAM OBJ job control statement. The options are:

`OBJ=(vol-ser-no,label)`

Specifies the location of the object module.

`OBJ=(RES,label)`

Specifies that the object module is located on the SYSRES device, with the file identifier specified by *label.*

`OBJ=(RUN,label)`

Specifies that the object module is located on the job's YRUN file, with the file identifier specified by *label.*

`OBJ=(*,label)`

The object module is on a volume identified in the file catalog with the file identifier specified by *label.*

`OBJ=(RUN,$Y$RUN)`

The object module is located on the job's YRUN file. This is the default value if the OBJ parameter is omitted.

Note: *The OBJ keyword parameter cannot be used with the COBL74L or COBL74LG forms of the jproc call.*

`SCR1=` { `vol-ser-no` | `RES` }

Specifies the volume serial number of the work file with an identifier of $SCR1.

SCR2={vol-ser-no / RES}

Specifies the volume serial number of the work file with an identifier of $SCR2.

SCR3={vol-ser-no / RUN}

Specifies the volume serial number of the work file with an identifier of $SCR3.

ALTLOD={(vol-ser-no,label) / (RES,label) / (RUN,label) / (*,label) / (RES,YLOD) / (RUN,YRUN)}

Specifies the alternate load library that contains the ANSI 1974 COBOL language processor. The options are:

ALTLOD=(vol-ser-no,label)
: Indicates the volume serial number and file identifier (*label*).

ALTLOD=(RES,label)
: Alternate load library is on YRES; a file identifier is specified by *label.*

ALTLOD=(RUN,label)
: Alternate load library is on the volume containing the job's YRUN file; a file identifier is specified by *label.*

ALTLOD=(*,label)
: Alternate load library is on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, the alternate load library is in YLOD when COBOL74 and COBOL74L are used, and YRUN when COBOL74LG is used.

option=specification
: Specifies the various compiler options.

ERRFIL=(vol-ser-no,label,module-name)
: Specifies an error log file. Error diagnostic messages are written to this file by the error log processor. The *vol-ser-no* parameter specifies the volume serial number, *label* specifies the file identifier, *module-name* specifies the 1- to 8-alphanumeric-character name of the error file element being created.

COBOL

Function

Generates the job control statements to execute the COBOL language processor. This form of the COBOL jprocs is only supported on the System 80, Model 8. (See COBL74.) Optionally, it can generate the job control statements that specify the following:

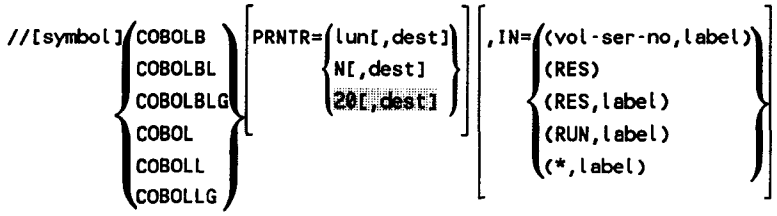
- Input-source library
- Output-object library
- Copy library
- PARAM job control statements to define the format of the compiler listing
- OPTION job control statements to automatically execute the linkage editor and the generated load module

The COBOL procedure call can be used for either the basic or extended version of COBOL. For the basic version, use the COBOLB, COBOLBL, or COBOLBLG procedure call. For the extended version, use the COBOL, COBOLL, or COBOLLG procedure call. When L is added to either the COBOL or COBOLB procedure call, an OPTION LINK control statement is generated, and the linkage editor is automatically executed. When LG is added to either the COBOL or COBOLB procedure call, an OPTION LINK,GO control statement is generated and the linkage editor and the generated load module are automatically executed.

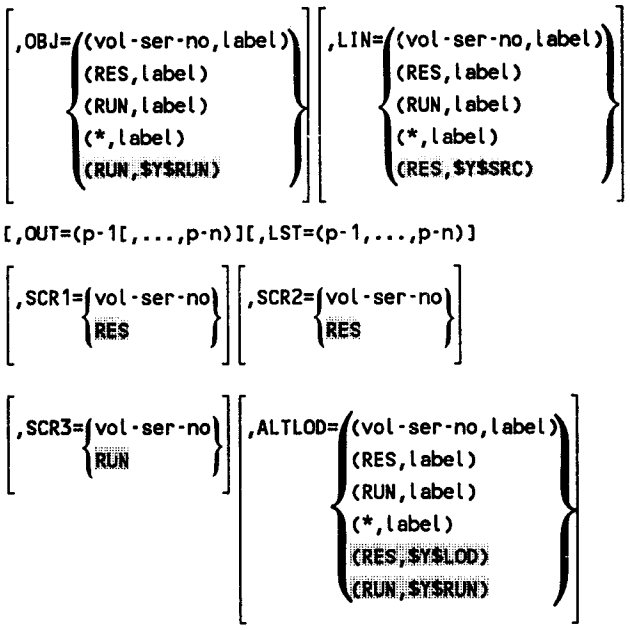
When L or LG is used, the OBJ parameter cannot be used to define a specific output library.

When the basic version is used, the specifications for the second and third scratch work files are ignored, as the basic version only needs one scratch work file.

Format



continued



Label

symbol
: Source module name of one to six alphanumeric characters; only needed when the IN parameter is used.

Parameters

PRNTR={lun[,dest]
N[,dest]
20[,dest]}
: Specifies a printer. The options are:

PRNTR=lun[,dest]
: Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

PRNTR=N[,dest]
: Indicates that you are supplying your own device assignment set for the printer; a DVC-LFD sequence is not to be generated. This option allows use of the // LCB and // VFB statements. A 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

If omitted, a local printer with a logical unit number of 20 is assumed.

```
IN=(vol-ser-no,label)
   (RES)
   (RES,label)
   (RUN,label)
   (*,label)
```

Input file definition; also replaces the PARAM IN control statement. The options are:

`IN=(vol-ser-no,label)`
Indicates the volume serial number and the file identifier.

`IN=(RES)`
Input is on SYSRES in YSRC.

`IN=(RES,label)`
Input is on SYSRES, but in a file other than YSRC.

`IN=(RUN,label)`
Input is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

`IN=(*,label)`
Input volume is identified in the file catalog with a file identifier specified by *label.*

If omitted, source input is in the form of data cards.

```
OBJ=(vol-ser-no,label)
    (RES,label)
    (RUN,label)
    (*,label)
    (RUN,SY$RUN)
```

Output file definition; also replaces the PARAM OBJ control statement. The options are:

`OBJ=(vol-ser-no,label)`
Indicates the volume serial number and file identifier.

`OBJ=(RES,label)`
Output is on SYSRES, with a file identifier specified by *label.*

`OBJ=(RUN,label)`
Output is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

`OBJ=(*,label)`
Output volume is identified in the file catalog with a file identifier specified by *label.*

LIN={(vol-ser-no-label) | (RES,label) | (RUN,label) | (*,label) | (RES,YSRC)}

Indicates the location of the copy module. The options are:

LIN=(vol-ser-no,label)
Specifies the file identifier (label) and the volume serial number.

LIN=(RES,label)
Modules are on SYSRES; the file identifier is specified by *label.*

LIN=(RUN,label)
Modules are on the volume containing the job's YRUN file; a file identifier is specified by *label.*

LIN=(*,label)
Modules are on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, modules are on SYSRES in the YSRC file.

OUT=(p-1,...p-n)
Parameter definitions for the COBOL compiler. See the COBOL supplementary reference manuals for your system for an explanation of these options.

LST=(p-1,...,p-n)
Specifies the format of the compiler listing. See the COBOL supplementary reference manuals for your system for an explanation of these options.

SCR1={vol-ser-no | RES}
Specifies the volume serial number of the first scratch work file.

SCR2={vol-ser-no | RES}
Specifies the volume serial number of the second work file. This is ignored in using the basic COBOL compiler.

SCR3={vol-ser-no | RUN}
Specifies the volume serial number of the third work file. This is ignored in using the basic COBOL compiler.

```
ALTLOD=(vol-ser-no,label)
       {(RES,label)      }
       {(RUN,label)      }
       {(*,label)        }
       {(RES,$Y$LOD)     }
       {(RUN,$Y$RUN)     }
```

Specifies the alternate load library that contains the COBOL compiler. The options are:

`ALTLOD=(vol-ser-no,label)`
: Indicates the volume serial number and file identifier (label).

`ALTLOD=(RES,label)`
: Alternate load library is on SYSRES; a file identifier is specified by *label.*

`ALTLOD=(RUN,label)`
: Alternate load library is on the volume containing the job's YRUN file; a file identifier is specified by *label.*

`ALTLOD=(*,label)`
: Alternate load library is on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, the alternate load library is YLOD unless you use LG (then the load library is in YRUN).

DVCDKT

Function

Assigns the same logical unit number to a diskette volume having different files used in a job. The logical unit numbers 130-133 (for diskette) are assigned by OS/3 to the different volumes as they are encountered in the control stream. Once a logical unit number is assigned to a volume, this logical unit number is associated with the volume for the length of the job. Any other specifications of a different logical unit number for this volume within the job is ignored.

Format

```
//[symbol] DVCDKT vol-ser-no[,lun][,NOVOL={Y}]
                                        {N}
```

Parameters

`vol-ser-no`
Volume serial number of the volume.

`lun`
Logical unit number to be used if different from the default logical unit number assigned by OS/3.

`NOVOL={Y}`
`      {N}`
Suppresses volume serial number checking (NOVOL=Y).

DVCVOL

Function

Assigns the same logical unit number to a disk or volume having different files used in a job. Up to 10 different volumes can be defined in a single job.

By default, the logical unit numbers 50-59 (for disk) are assigned by OS/3 to the different volumes as they are encountered in the control stream. Once a logical unit number is assigned to a volume, this logical unit number is associated with the volume for the length of the job. Any other specification of a different logical unit for this volume within the job is ignored.

Format

```
//[symbol] DVCVOL {vol-ser-no}[,lun][,NOVOL={Y}]
                  {RES      }       {N}
                  {RUN      }
```

Parameters

`vol-ser-no`
Volume serial number of the volume.

`RES`
Indicates the SYSRES volume is to be used.

`RUN`
Indicates the volume containing the job's YRUN file is to be used.

`lun`
Logical unit number to be used if different from the default logical unit numbers 50-59, which are assigned by OS/3.

`NOVOL={Y}`
`      {N}`
Suppresses volume serial number checking (NOVOL=Y).

Notes:

1. *The DVCVOL procedure call must be the last statement on a job control card if multistatement coding is used.*

2. *The LBL and LFD control statements must be present in the control stream after the DVCVOL procedure call. An EXT control statement must also be used when you are allocating the file.*

DVCVTP

Function

Assigns the same logical unit number to a tape volume having different files used in a job. Up to 10 different volumes can be defined in a single job.

By default, the logical unit numbers 90-99 are assigned by OS/3 to the different volumes as they are encountered in the control stream. Once a logical unit number is assigned to a volume, this logical unit number is associated with the volume for the length of the job. Any other specification of a different logical unit number for this number within the job is ignored.

Format

```
//[symbol] DVCVTP vol-ser-no[,lun][,PREP={Y|N}][,NOVOL={Y|N}]
```

Parameters

`vol-ser-no`
Volume serial number of the volume.

`lun`
Logical unit number to be used if different from the default logical numbers 90-99, which are assigned by OS/3.

`PREP={Y|N}`
If specified (PREP=Y), any information currently on the volume is effectively erased.

`NOVOL={Y|N}`
Suppresses volume serial number checking (NOVOL=Y).

Notes:

1. *The DVCVTP procedure call must be the last statement on a job control card if multistatement coding is used.*

2. *The LBL and LFD control statements must be present in the control stream after the DVCVTP procedure call.*

FORT

Function

Generates the job control statements to execute the FORTRAN language processor. Optionally, it can generate the job control statements that specify the following

- Input source library
- Output object library
- PARAM job control statement to define the format of the compiler listing
- OPTION job control statements to automatically execute the linkage editor and the generated load module

The FORTRAN procedure call can be used for basic, extended, or FORTRAN IV versions of FORTRAN.

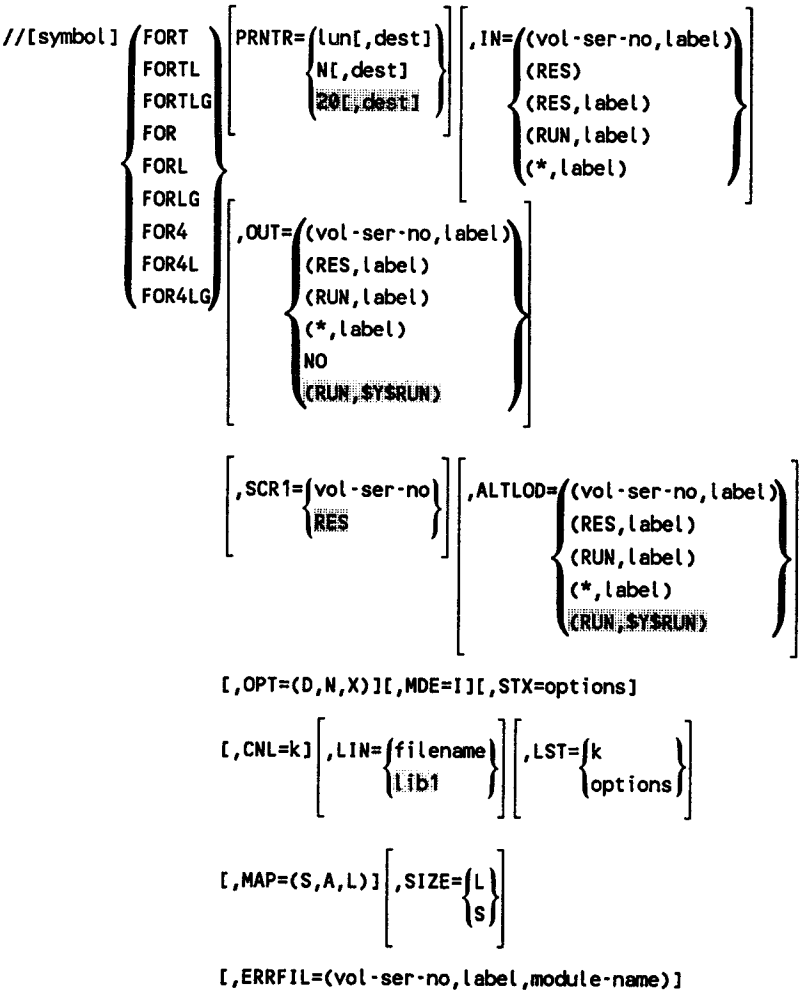
- Basic - Use the FORT, FORTL, or FORTLG procedure call.
- Extended - Use the FOR, FORL, or FORLG procedure call.
- FORTRAN IV - Use the FOR4, FOR4L, or FOR4LG procedure call.

When L is added to the FOR, FORT, or FOR4 procedure call, an OPTION LINK control statement is generated, and the linkage editor is automatically executed. When LG is added to the FOR, FORT, or FOR4 procedure call, an OPTION LINK,GO control statement is generated, and the linkage editor and the generated load module are automatically executed.

Note: *FOR4, FOR4L, and FOR4LG are the only jprocs supported for System 80, models 3, 4, 5, and 6. The System 80, model 8 supports all the FORTRAN jprocs.*

When L or LG is used, the OBJ parameter cannot be used to define a specific output library.

Format



Label

symbol

Source module name of one to six alphanumeric characters; only needed when the IN parameter is used.

Parameters

```
PRNTR={lun[,dest]}
      {N[,dest]  }
      {20[,dest] }
```

Specifies a printer. The options are:

`PRNTR=lun[,dest]`
: Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

`PRNTR=N[,dest]`
: Indicates that you are supplying your own device assignment set for the printer; a DVC-LFD sequence is not to be generated. This option allows the use of the // LCB and // VFB statements. A 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

If omitted, a local printer with a logical unit number of 20 is assumed.

```
IN={(vol-ser-no,label)}
   {(RES)             }
   {(RES,label)       }
   {(RUN,label)       }
   {(*,label)         }
```

Input file definition; also replaces the PARAM IN control statement. The options are:

`IN=(vol-ser-no,label)`
: Indicates the volume serial number and file identifier.

`IN=(RES)`
: Input is on SYSRES, in YSRC.

`IN=(RES,label)`
: Input is on SYSRES, but in a file other than YSRC.

`IN=(RUN,label)`
: Input is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

`IN=(*,label)`
: Input volume is identified in the file catalog with a file identifier specified by *label.*

If omitted, source input is in the form of data cards.

```
OUT={(vol-ser-no,label)}
    {(RES,label)       }
    {(RUN,label)       }
    {(*,label)         }
    {NO                }
    {(RUN,$Y$RUN)      }
```

Output file definition; also replaces the PARAM OUT control statement. The options are

`OUT=(vol-ser-no,label)`
Indicates the volume serial number and file identifier.

`OUT=(RES,label)`
Output is on SYSRES, with a file identifier specified by *label.*

`OUT=(RUN,label)`
Output is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

`OUT=(*,label)`
Output volume is identified in the file catalog with a file identifier specified by *label.*

`OUT=NO`
No object code output.

```
SCR1={vol-ser-no}
     {RES       }
```

Specifies the volume serial number of the scratch work file.

```
ALTLOD={(vol-ser-no,label)}
       {(RES,label)       }
       {(RUN,label)       }
       {(*,label)         }
       {(RUN,$Y$RUN)      }
```

Specifies the alternate load library that contains the FORTRAN language processor. The options are:

`ALTLOD=(vol-ser-no,label)`
Indicates the volume serial number and file identifier (label).

`ALTLOD=(RES,label)`
Alternate load library is on SYSRES; a file identifier is specified by *label.*

`ALTLOD=(RUN,label)`
Alternate load library is on the volume containing the job's YRUN file; a file identifier is specified by *label.*

ALTLOD=(*,label)
Alternate load library is on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, the alternate load library is in the job's YRUN file.

OPT=(D,N,X)
Specifies one or all of the following compilation options:

D
Specifies double spacing of the compiler listing.

N
Specifies that no object program is to be generated. The program units are merely compiled and cannot be executed.

X
Specifies compilation of all cards with the character X in column 1. If this option is not specified, these cards will be treated as comments.

This parameter is only supported for the extended and FORTRAN IV compilers.

MDE=I
Specifies that the compiler is to evaluate expressions in a strict left-to-right order when there is a choice, and that storage is to be allocated for variables and arrays in the sequence in which they were encountered.

STX=options
Used with the extended compiler to execute two STXIT program status words (PSWs).

CNL=k
Specifies compiler termination if a diagnostic with a severity level is generated. The values for *k* are:

2
Indicates academic messages, e.g., a truncated constant.

4
Indicates warning diagnostics, e.g., an extraneous comma in a list.

6
Indicates serious diagnostics, e.g., an array reference without a preceding array declarator.

8
Indicates fatal errors, e.g., insufficient storage to complete the compilation.

If the CNL parameter is not specified, the compiler processes all program units in the control stream, regardless of the errors encountered. When specified, the CNL parameter remains in force until redefined.

This parameter is only supported for the extended compiler.

LIN={filename / lib1}

Specifies the name of the default file name in which the source modules reside.

filename

Specifies a 1- to 8-alphanumeric-character file name.

This parameter is used with the IN parameter and is only supported for the extended and FORTRAN IV compilers.

LIB1 is the default and is not supported for the FORTRAN IV compiler.

LST={k / options}

Specifies the format of the compiler listings.

LST=k

Specifies the options for the basic compiler only. The values for *k* are

1

Indicates source code listing.

2

Indicates diagnostic listing.

4

Indicates storage allocation map.

7

Is the default value.

8

Indicates object code listing.

Multiple options may be specified by adding the values for k; i.e., to produce both the source code and diagnostic listings (options 1 and 2), specify a 3 for the value of *k*.

LST=options

Specifies the options for the extended and FORTRAN IV compilers only. The options are:

N

Indicates an abbreviated listing consisting of only the compiler identification, parameters, error counts, and termination conditions.

S

Indicates, in addition to the N listing, the source code listing with any serious diagnostics.

M

Indicates, in addition to the S listing, a storage map showing the addresses assigned to variables and arrays. (For FORTRAN IV, M can be superseded by the MAP parameter.)

W

Indicates, in addition to the M listing, academic and warning messages (extended compiler only).

O

Indicates, in addition to the W listing, an object code showing the instructions generated for the executable statements (extended compiler only).

The default is M for the extended compiler and S for the FORTRAN IV compiler.

MAP=(S,A,L)

Specifies the type of maps produced by the compiler. One or all of the following options may be chosen:

S

Specifies object summary information, including module size and external subroutines called.

A

Specifies an alphabetical listing of the addresses assigned to variables, arrays, and statement labels.

L

Specifies a listing of the variables, arrays, and statement labels in order by the storage locations assigned.

When a MAP argument is specified, it supersedes the maps selected by the LST parameter. Also, when a MAP argument is specified, it is not necessary to specify LST=M.

This parameter is only supported for the FORTRAN IV compiler.

SIZE={L / S}

Specifies the size of the FORTRAN IV compiler. The options are:

L

Specifies large version.

S

Specifies small version.

ERRFIL=(vol-ser-no,label,module-name)

Applies to FORTRAN IV only; specifies an error log file. Error diagnostic messages are written to this file by the error log processor. The *vol-ser-no* parameter specifies the volume serial number, *label* specifies the file identifier, *module-name* specifies the name of the error log file element being created.

LINK

Function

Generates the job and linkage editor control statements needed to execute the linkage editor for the purpose of creating a load module. The object modules to be specifically included in the load module are assumed to be in the job's YRUN file; any object code that may have to be automatically included in the load module is assumed to be in YOBJ, and the load module produced is assigned the default name LNKLOD and is stored in the job's YRUN file. All these default options may be altered by specific parameters. The generated load module can also be automatically executed by specifying the LINKG procedure call statement.

Format

```
//[symbol] { LINK  } [input-module-name-1,...,input-module-name-10]
           { LINKG }

    [ ,PRNTR={ lun[,dest]  } ]
             { N[,dest]    }
             { 20[,dest]   }

    [ ,IN={ (vol-ser-no,label) } ][ ,OUT={ (vol-ser-no,label) } ]
          { (RES)              }        { (RES,label)        }
          { (RES,label)        }        { (RUN,label)        }
          { (RUN,label)        }        { (*,label)          }
          { (*,label)          }        { (N)                }
          { (RUN,$Y$RUN)       }        { (RUN,$Y$RUN)       }

    [ ,RLIB={ (vol-ser-no,label) } ][ ,ALIB={ (vol-ser-no,label) } ]
            { (RES,label)        }         { (RES,label)        }
            { (RUN,label)        }         { (RUN,label)        }
            { (*,label)          }         { (*,label)          }

    [ ,SCR1={ vol-ser-no } ][ ,STD={ YES } ]
            { RES        }        { NO  }

    [ ,ALTLOD={ (vol-ser-no,label) } ]
              { (RES,label)        }
              { (RUN,label)        }
              { (*,label)          }
              { (RUN,$Y$RUN)       }
```

continued

```
[,OPT='options'][,CLIB=[(vol-ser-no,label,modname)
                       {(RES,label,modname)       }
                       {(RUN,label,modname)       }
                       {(*,label,modname)         }]
[,CMT='comments'][,ENTER=expression]
```

Notes:

1. *The LINK procedure call is used to produce a load module from object modules. When the LINKG procedure call is used, an OPTION GO job control statement is generated, which automatically executes the generated load module.*

2. *When the LINKG procedure call is used, the OUT parameter to define a specific output library cannot be used.*

Label

`symbol`

Output load module name of one to six alphanumeric characters; this name is used for LOADM linkage editor control statements. If *symbol* is omitted, the value for the first input-module-name is used for the load module name. If the *input-module-name* parameter is also omitted, the load module name is LNKLOD.

Parameters

`input-module-name`

Names up to 10 input modules for the generated INCLUDE linkage editor control statements, each name being from 1 to 8 alphanumeric characters in length. If COBOL-68 (extended or basic) is used, you cannot take the 6-character value of *symbol*; you must use all eight positions of the first *input-module-name,* zero-filled to the right if necessary. (You can use the *input-module-name* parameter by itself; the first *input-module-name* truncates to six characters for *symbol.)*

If the *input-module-name* parameter is omitted, the value in the symbol field is assumed to be the name of the object module to be link-edited. If the symbol field is blank, all object modules residing in the job's YRUN file are included.

```
PRNTR={lun[,dest]}
      {N[,dest]  }
      {20[,dest] }
```

Specifies a printer. The options are:

`PRNTR=lun[,dest]`
: Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

`PRNTR=N ,dest`
: Indicates that you are supplying your own device assignment set for the printer - a DVC-LFD sequence is not to be generated. This option allows use of the // LCB and // VFB statements. A 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

If omitted, a local printer with a logical unit number of 20 is assumed.

```
IN={(vol-ser-no,label)}
   {(RES)             }
   {(RES,label)       }
   {(RUN,label)       }
   {(*,label)         }
   {(RUN,$Y$RUN)      }
```

Input file definition; also adds a file name to the generated INCLUDE linkage editor control statement. The options are:

`IN=(vol-ser-no,label)`
: Indicates the volume serial number and file identifier.

`IN=(RES)`
: Input is on SYSRES, in YOBJ.

`IN=(RES,label)`
: Input is on SYSRES, but in a file other than YOBJ.

`IN=(RUN,label)`
: Input is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

`IN=(*,label)`
: Input volume is identified in the file catalog with a file identifier specified by *label.*

If omitted, the job's YRUN file is searched for object modules to include. The file defined in the RLIB parameter is searched next (if the RLIB parameter was specified), and finally YOBJ is searched.

```
OUT={(vol-ser-no,label)}
    {(RES,label)       }
    {(RUN,label)       }
    {(*,label)         }
    {(N)               }
    {(RUN,$Y$RUN)      }
```

Output file definition for the output load module; also replaces the PARAM OUT control statement. The options are:

`OUT=(vol-ser-no,label)`
Indicates the volume and serial number and file identifier.

`OUT=(RES,label)`
Output load module is on SYSRES; a file identifier is specified by *label.*

`OUT=(RUN,label)`
Output load module is on the volume containing the job's YRUN file; a file identifier is specified by *label.*

`OUT=(*,label)`
Output volume is identified in the file catalog with a file identifier specified by *label.*

`OUT=(N)`
Output load module is not written.

```
RLIB={(vol-ser-no,label)}
     {(RES,label)       }
     {(RUN,label)       }
     {(*,label)         }
```

Renames the file to be searched in place of YOBJ during automatic inclusion. This library is searched when no private-library (ALIB) is specified, or when the module being sought is not found in a private library. The options are:

`RLIB=(vol-ser-no,label)`
Indicates the volume serial number and file identifier.

`RLIB=(RES,label)`
Library is on SYSRES; a file identifier is specified by *label.*

`RLIB=(RUN,label)`
Library is on the volume containing the job's YRUN file; a file identifier is specified by *label.*

RLIB=(*,label)
Library is on the volume identified in the file catalog with a file identifier specified by *label.*

ALIB={(vol-ser-no,label) | (RES,label) | (RUN,label) | (*,label)}
Names an additional library to be used in the automatic inclusion. This library is always searched first during automatic inclusion in an attempt to locate the needed modules. If they are not found, YOBJ, or the library named by the RLIB parameter, is searched. The options are:

ALIB=(vol-ser-no-label)
Indicates the volume serial number and the file identifier.

ALIB=(RES,label)
Library is on SYSRES; a file identifier is specified by *label.*

ALIB=(RUN,label)
Library is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

ALIB=(*,label)
Library is on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, the library specified by the RLIB parameter or YOBJ is searched, in that order.

SCR1={vol-ser-no | RES}
Specifies the volume serial number of the scratch work file.

STD={YES | NO}
Indicates whether the LOADM and INCLUDE linkage editor control statement should be generated automatically or placed physically in the control stream as data. If YES is used, the proper LOADM and INCLUDE linkage editor control statements are generated between the /$ and /* job control statements. If NO is used, no linkage editor control statements are generated; they must be placed physically in the control stream following the procedure call.

```
ALTLOD=(vol-ser-no,label)
       (RES,label)
       (RUN,label)
       (*,label)
       (RUN,$Y$RUN)
```

Specifies the alternate load library containing the linkage editor. The options are:

`ALTLOD=(vol-ser-no,label)`
Indicates the volume serial number and file identifier (label).

`ALTLOD=(RES,label)`
Alternate load library is on SYSRES; a file identifier is specified by *label.*

`ALTLOD=(RUN,label)`
Alternate load library is on the volume containing the job's YRUN file; a file identifier is specified by *label.*

`ALTLOD=(*,label)`
Alternate load library is on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, the alternate load library is in the job's YRUN file.

`OPT='options'`
Specifies the linkage editor options to be in effect.

```
CLIB=(vol-ser-no,label,modname)
     (RES,label,modname)
     (RUN,label,modname)
     (*,label,modname)
```

Specifies the location of the source module containing the linkage editor control statements to be processed. RES indicates a library on SYSRES; RUN indicates a library on the volume containing the job's YRUN file; the * indicates the volume is identified in the file catalog. The file identifier is specified by *label.* The name of the source module is specified by *modname.*

`CMT='comment'`
Indicates the character string that will be inserted in the comment field of each phase header record produced for the generated load module. The comment must be enclosed in apostrophes and may be up to 30 characters in length. It may contain blanks, commas, and other special symbols, excluding apostrophes.

ENTER=expression

Specifies the transfer address. The expression may be a decimal number from one to eight digits, a hexadecimal number from one to six digits in the form X'nnnnnn', a previously defined symbol, or a previously defined symbol plus or minus a decimal or hexadecimal number as just described. If this parameter is not specified, control is transferred to the address obtained from the first object module in the phase that had a valid transfer address. If no valid transfer address is found, control is transferred to the first CSECT specifically involved in the phase.

Note: *If the ASM, COBOL, FORT, or RPG procedure call is used previously in the job control stream, the same disk unit assigned in that procedure call must be assigned in this procedure call.*

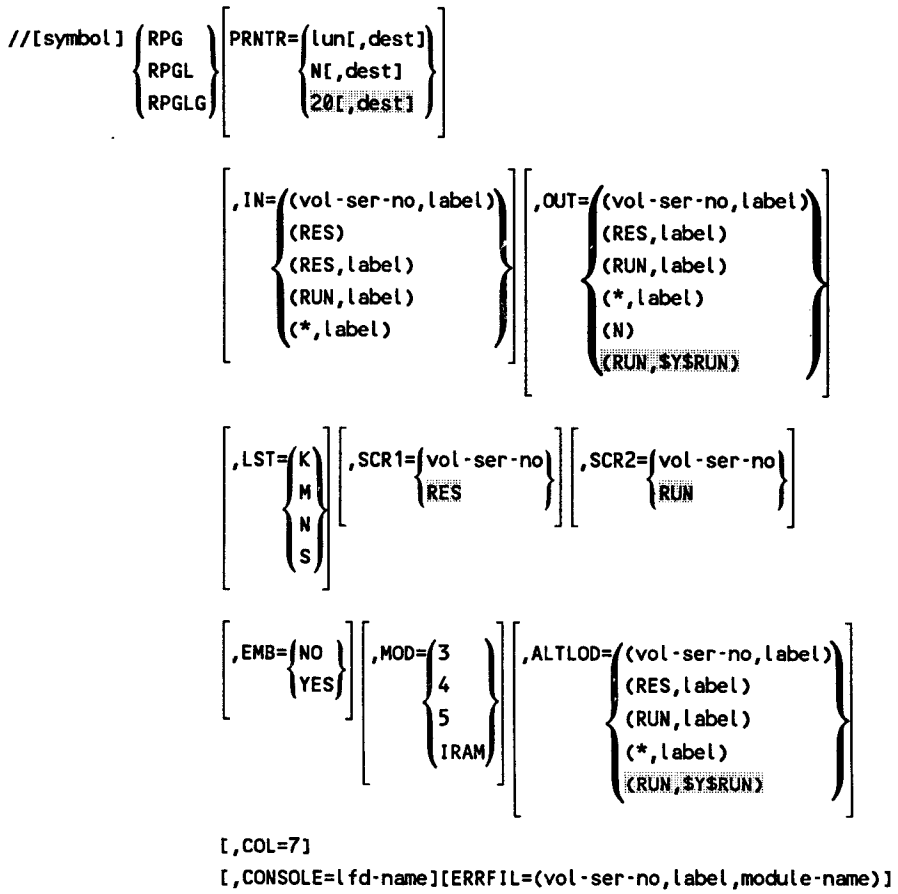
RPG II

Function

Generates the necessary job control statements to run the RPG II language processor. Optionally, it can generate the job control statements that specify the following:

- Input source library
- Output object library
- PARAM job control statements to define the format of the compiler listing
- OPTION job control statements to automatically execute the linkage editor and the generated load module

Format



Note: *The RPG procedure call is used to compile a program. When the RPGL procedure call is used, an OPTION LINK job control statement is generated,*

which allows for the automatic execution of the linkage editor. The RPGLG procedure call generates an OPTION LINK,GO job control statement, which automatically executes the linkage editor, followed by the automatic execution of the program.

When either the RPGL or RPGLG procedure call is used, the OUT parameter to define a specific output library cannot be used.

Label

symbol
: Source module name of one to six alphanumeric characters; only needed when the IN parameter is used.

Parameters

PRNTR=$\left\{\begin{array}{l}\text{lun[,dest]}\\ \text{N[,dest]}\\ \text{20[,dest]}\end{array}\right\}$
: Specifies a printer. The options are:

PRNTR=lun[,dest]
: Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

PRNTR=N ,dest
: Indicates that you are supplying your own device assignment set for the printer - a DVC-LFD sequence is not to be generated. This option allows use of the // LCB and // VFB statements. A 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

If omitted, a local printer with a logical unit number of 20 is assumed.

```
IN=(vol-ser-no,label)
   (RES)
   (RES,label)
   (RUN,label)
   (*,label)
```

Input file definition; also replaces the PARAM IN control statement. The options are:

`IN=(vol-ser-no,label)`
Indicates the volume serial number and file identifier.

`IN=(RES)`
Input is on SYSRES, in YSRC.

`IN=(RES,label)`
Input is on SYSRES, but in a file other than YSRC.

`IN=(RUN,label)`
Input is on the volume containing the job's YRUN file; a file identifier is specified by *label.*

`IN=(*,label)`
The input volume is identified in the file catalog with a file identifier specified by label.

If omitted, source input is in the form of data cards.

```
OUT=(vol-ser-no,label)
    {(RES,label)       }
    {(RUN,label)       }
    {(*,label)         }
    {(N)               }
    {(RUN,$Y$RUN)      }
```

Output file definition; also replaces the PARAM IN control statement. The options are:

`OUT=(vol-ser-no,label)`
: Indicates the volume serial number and file identifier.

`OUT=(RES,label)`
: Output is on SYSRES, with a file identifier specified by *label.*

`OUT=(RUN,label)`
: Output is on the volume containing the job's YRUN file, with a file identifier specified by *label.*

`OUT=(*,label)`
: The output volume is identified in the file catalog with a file identifier specified by label.

`OUT=(N)`
: No object code output.

LST=(K / M / N / S)

Specifies the format of the compiler listing:

K
: Inhibits error flags for sequence errors.

M
: Inhibits source and diagnostic listings.

N
: Inhibits all listable output.

S
: Inhibits main storage map listing.

SCR1=(vol-ser-no / RES)

Specifies the volume serial number of the first scratch work file.

SCR2=(vol-ser-no / RUN)

Specifies the volume serial number of the second scratch work file.

EMB=(NO / YES)

editor control statements are to be generated.

MOD=(3 / 4 / 5)

MOD=3
: Specifies an RPG II compilation in IBM System/3 and System/34 mode. When this mode is specified, MIRAM will be used to process all disk files; the logical file definition (LFD) for printer files is changed to PRNTR,PRNTR1,...PRNTRn; and the control reader is used for card input even though the data management reader (READER) is specified.

MOD=4
: Same as MOD=3 except that printer files are generated with the same names as used in the program and reader files use the data management reader (READER).

MOD=5
: The program is to be compiled in the IBM System 3 and System 34 mode with native mode data management accessing.

ALTLOD=(vol-ser-no,label)
(RES,label)
(RUN,label)
(*,label)
(RUN,SY$RUN)

Specifies the alternate load library that contains the RPG II compiler. The options are:

ALTLOD=(vol-ser-no,label)
Indicates the volume serial number and file identifier (label).

ALTLOD=(RES,label)
Alternate load library is on SYSRES; a file identifier is specified by *label.*

ALTLOD=(RUN,label)
Alternate load library is on the volume containing the job's YRUN file; a file identifier is specified by *label.*

ALTLOD=(*,label)
Alternate load library is on the volume identified in the file catalog with a file identifier specified by *label.*

If omitted, the alternate load library is in the job's YRUN file.

COL=7
Indicates that RPG II code is to begin in column 7 of the source code (columns 1 through 5 being used for a sequence field and column 6 being left blank).

If omitted, code will begin in column 1.

CONSOLE
Specifies the lfd-name of a CONSOLE (interactive data entry) file - not a system console file.

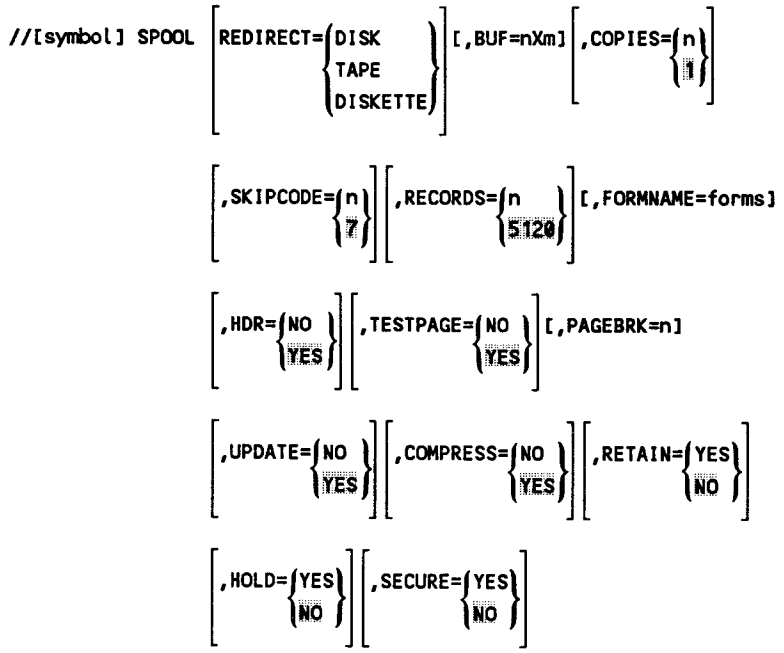
ERRFIL=(vol-ser-no,label,module-name)
Specifies an error log file. Error diagnostic messages are written to this file by the error log processor. The parameter *vol-ser-no* specifies the volume serial number, *label* specifies the file identifier, *module-name* specifies the 1- to 8-alphanumeric-character name of the error file element being created. (Does not need to match or be related to the RPG II source object module name.)

SPOOL

Function

Controls output spooling. When used, SPOOL must occur within the DVC-LFD sequence for the file to be spooled. Ignored in a nonspooling environment.

Format



Note: *Only the* BUF, RETAIN, UPDATE, COMPRESS, *and* HOLD *keyword parameters can be used for spooled data-set-label diskette output files.*

Parameters

REDIRECT={DISK / TAPE / DISKETTE}

Redirects spooled output (for temporary storage) to a disk, tape, or format-label diskette; output is printed or punched later.

REDIRECT=DISK

Redirects spooled output to another disk volume.

REDIRECT=TAPE
Redirects spooled output to a tape volume.

REDIRECT=DISKETTE
Redirects spooled output to a format-label diskette.

Note: *A DEV operator command must not be in effect (for this copy of the output writer) if you specify* REDIRECT=TAPE.

BUF=nXm
Establishes buffers for use by the spool subfile where *n* is the number of buffers, *X* is a constant, and *m* is the size of each buffer in 256-byte increments. No advantage is gained by making *n* greater then 2 unless system symbionts are being used. If this keyword parameter is omitted, the file shares the buffer pool with the job log and other spooled files not having their own buffers.

COPIES={n / 1}
Number of times the spooled file is to be printed or punched (output), in a range of 0 through 255. Zero indicates no output. If this keyword parameter is not specified, the file is output once and then deleted from the spool file.

SKIPCODE={n / 7}
Used when a filed vertical format buffer having more than seven skip codes is requested (via // VFB) or when the system default buffer has more than seven skip codes. Indicates the number of lines in the buffer that contain skip codes, plus three (one for forms overflow and two for home paper position). Zero indicates no skip codes.

RECORDS={n / 5120}
Maximum number of records (lines, including spaces and skipped lines from print files, cards for punch files) that can be placed in the spool file before the operator is queried as to whether the job should be continued or terminated. The largest number you can specify is 262144. If this parameter is omitted, 5120 is assumed.

FORMNAME=forms
A 1- to 8-alphanumeric-character name identifying a special printer form or card type to the operator.

HDR={NO / YES}
Suppresses the printing of page headers in burst mode. If omitted, a page header is automatically printed.

TESTPAGE=$\left\{\begin{matrix}\text{NO}\\\text{YES}\end{matrix}\right\}$

Suppresses the query to the operator asking if a sample test pattern page is to be printed when a formname (*FORMNAME=forms*) is detected. If omitted, a query is automatically directed to the operator.

PAGEBRK=n

Specifies the number of pages or cards to be spooled out before a file is breakpointed and printed or punched. The largest number you can specify is 32000.

UPDATE=$\left\{\begin{matrix}\text{NO}\\\text{YES}\end{matrix}\right\}$

Indicates that the spool file subdirectory is to be updated only when a file is closed; if the program cancels, any output generated before cancellation is lost. If omitted, the spooler updates the subdirectory entry each time it crosses a logical track in the program file; if the program cancels, output generated before cancellation can be printed.

COMPRESS=$\left\{\begin{matrix}\text{NO}\\\text{YES}\end{matrix}\right\}$

Prevents the system from attempting to compress data that's directed to the output spool file. Normally, it should not be specified if data in the file contains a large number of embedded blanks or if the file has a block size larger than 120. Specifying COMPRESS=NO when the block size is 121 or greater results in an output spool file containing only one line per sector and requires that nXm be at least 2X4.

RETAIN=$\left\{\begin{matrix}\text{YES}\\\text{NO}\end{matrix}\right\}$

The output file is retained in the spool file after it is printed, punched, or placed on data-set-label diskette; the file can be printed, punched, or placed on data-set-label diskette again at a later time. If *RETAIN=YES* is specified with *REDIRECT*, the output file is redirected but is also retained in the spool file for later processing.

HOLD=$\left\{\begin{matrix}\text{YES}\\\text{NO}\end{matrix}\right\}$

Spooled output file (print, punch, or data-set-label diskette) is to be held for later processing. Files on hold can be released by a BEGIN SPL command or by a CC job control statement specifying the BEGIN SPL command. If *HOLD=YES* is specified with *RETAIN*, *REDIRECT*, or both, the file is put on hold and *RETAIN* and *REDIRECT* are acted upon when the file is released.

SECURE=$\left\{\begin{matrix}\text{YES}\\\text{NO}\end{matrix}\right\}$

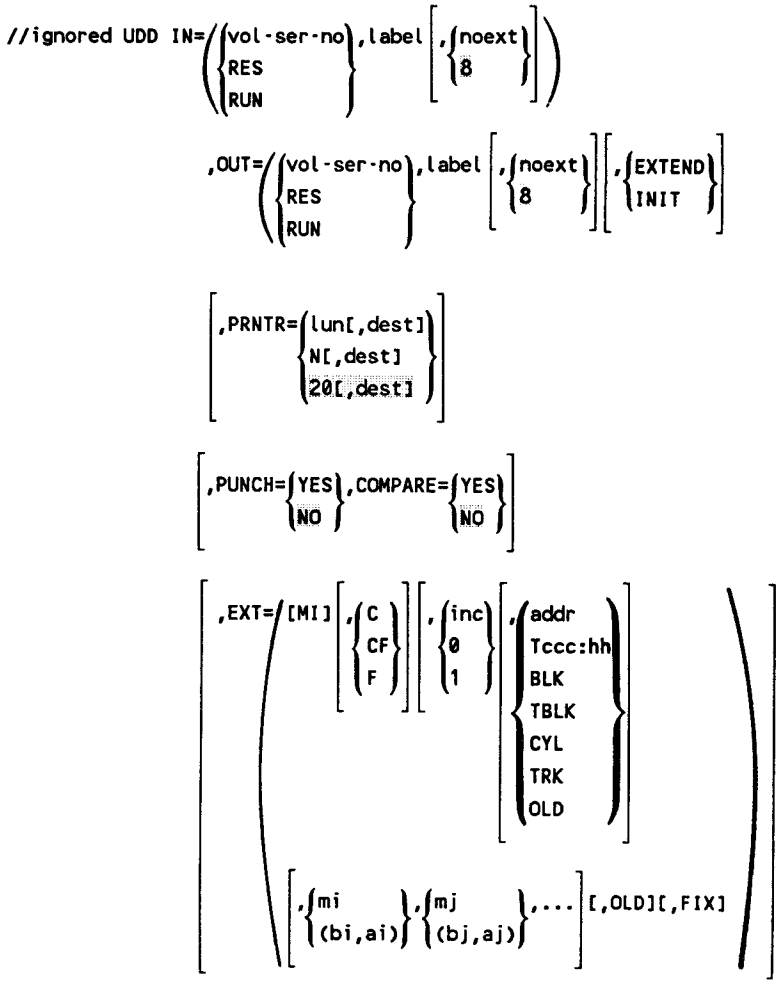
Suppresses the printing of the output file if the workstation to which the auxiliary printer is connected is not logged on. If not specified, the file is printed at the auxiliary printer whether or not the workstation is logged on.

UDD

Function

Generates the job control statements for the device assignment sets needed by the data utility routine to copy or compare one disk, data-set-label diskette, or format-label diskette to another disk, data-set-label diskette, or format-label diskette.

Format



Label

The label will be ignored.

Parameters

IN=

Supplies the information needed to define the input file. The information is coded within parentheses.

vol-ser-no
: Volume serial number.

RES
: Input is on SYSRES.

RUN
: Input is on the volume containing the job's YRUN file.

label
: File identifier for the input file.

noext
: Specifies the number of extents in the file. Reserves extent table storage for use by the data access method.

OUT=

Supplies the information needed to define the output file for a copy operation or the secondary input for a compare operation.

vol-ser-no
: Volume serial number.

RES
: Output (or secondary input) is on SYSRES.

RUN
: Output (or secondary input) is on the volume containing the job's YRUN file.

label
: File identifier for the output (or secondary input) file.

noext
: Number of extents in the file. Reserves extent table storage for use by the data access method.

EXTEND
: Adds information to the end of a sequential file.

INIT

Initializes the file, starting with the first record each time the file is opened. Previous control information in the format labels is ignored at file open time and is overwritten by specifications contained in this DVC-LFD sequence at file close time.

PRNTR={lun[,dest] / N[,dest] / 20[,dest]}

Specifies a printer. The options are:

PRNTR=lun[,dest]

Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

PRNTR=N[,dest]

Indicates that you are supplying your own device assignment set for the printer - a DVC-LFD sequence is not to be generated. This option allows use of the // LCB and // VFB statements. A 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

If omitted, a local printer with a logical unit number of 20 is assumed.

PUNCH={NO / YES}

Generates the device assignment set for the punch. This is needed when the punch dual output feature (DC) is selected on the data utility control card.

COMPARE={NO / YES}

Indicates that this is a compare operation; will generate the appropriate file name for the secondary input.

EXT=

Supplies the extent specifications to be used when reserving system resources for a particular file. The information is coded within parentheses. All the parameters (except FIX) that apply to disk also apply to format-label-diskette.

MI

MIRAM file; only valid specification for data-set-label diskette.

C
: Allocate contiguous space; only valid specification for a data-set-label diskette file.

F
: Reformat the file at allocation time. Subparameter 4 must be **BLK.**

CF
: Both of the preceding.

inc
: Secondary increment (in cylinders) for automatic extension. If there is no *EXT* keyword parameter for this file, the value of the most recently specified secondary increment is used.

0
: File cannot be dynamically extended; must be specified for data-set-label diskette file.

addr
: Absolute beginning cylinder address, in terms of cylinders.

Tccc:hh
: Absolute track address (hexadecimal) in cylinder/head format where the file is to begin. Allocation is in terms of tracks.

BLK
: Allocation in terms of blocks; only valid specification for data-set-label diskette files.

TBLK
: Allocates space in blocks; actual allocation is in terms of tracks.

CYL
: Allocation in terms of cylinders.

TRK
: Allocates space in tracks.

OLD
: Secondary allocation increment change (subparameter 3) for an existing file; cannot be followed by any other subparameters if specified.

mi
: Numbers of cylinders or tracks to allocate for this file; subparameter 4 must be *CYL, addr, TRK,* or *Tccc:hh.*

(bi,ai)
Allocation in terms of blocks (by cylinder or track); subparameter 4 must be *BLK* or *TBLK*.

bi
Is the average block length.

ai
Is the number of blocks to allocate.

Subparameters 6 through n

Same as subparameter 5 and used to describe additional extents.

Subparameter n+1

OLD
Indicates the file's previously allocated extent is to be increased by the allocation amount (*mi,(bi,ai),* etc) specified.

If OLD is omitted, the request is for a new extent.

FIX
Indicates this extent is in the fixed-head area of an 8417 disk.

UDT

Function

Generates the job control statements for the device assignment sets needed by the data utility routine to copy or compare a disk file to a tape file.

Format

```
//ignored UDT IN=((vol-ser-no|RES|RUN),label[,(noext|8)])
            ,OUT=(vol-ser-no,label)[,PRNTR=(lun[,dest]|N[,dest]|20[,dest])]
            [,PUNCH=(NO|YES)][,COMPARE=(NO|YES)]
```

Label

The label will be ignored.

Parameters

IN=

Supplies the information needed to define the input file. The information is coded within parentheses.

vol-ser-no

Volume serial number.

RES

Input is on SYSRES.

RUN

Input is on the volume containing the job's YRUN file.

label
File identifier for the input file.

noext
Number of extents in the file. Reserves extent table storage for use by the data access method.

OUT=
Specifies the information needed to define the output file for a copy operation or the secondary input for a compare operation.

vol-ser-no
Volume serial number.

label
File identifier for the output (or secondary input) file.

PRNTR={lun[,dest] / N[,dest] / [redacted]}
Specifies a printer. The options are:

PRNTR=lun[,dest]
Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

PRNTR=N[,dest]
Indicates that you are supplying your own device assignment set for the printer - a DVC-LFD sequence is not to be generated. This option allows use of the // LCB and // VFB statements. A 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

If omitted, a local printer with a logical unit number of 20 is assumed.

PUNCH={[redacted] / YES}
Generates the device assignment set for the punch. This is needed when the punch dual output feature (DC) is selected on the data utility control card.

COMPARE={[redacted] / YES}
Indicates this is a compare operation; will generate the appropriate file name for the secondary input.

UPLCNV

Function

Generates the job control statements for the device assignment set needed by the UTS 400 up-line conversion routine.

Format

```
//ignored UPLCNV [PRNTR={n  }],FILn={vsn,label,filename }
                 [     {20 }]      {RES,label,filename }
                                     {RUN,label,filename }
```

Label

The label will be ignored.

Parameters

`PRNTR={n }`
`      {20}`

Specifies the logical unit number for the printer.

`FILn={vsn,label,filename }`
`     {RES,label,filename }`
`     {RUN,label,filename }`

Generates the device assignment set for the disk pack.

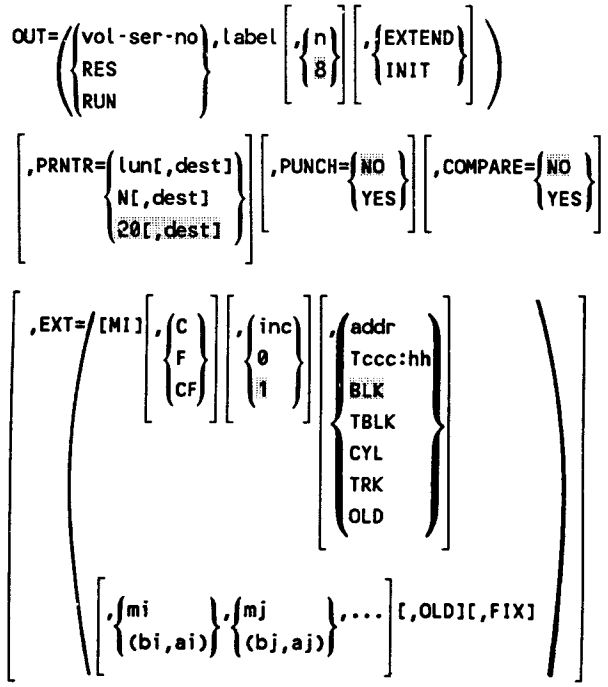
UTD

Function

Generates the job control statements for the device assignment sets needed by the data utility routine to copy or compare a tape file to a disk format-label diskette, or data-set-label diskette file.

Format

```
//ignored UTD IN=(vol-ser-no,label),
```



Label

The label will be ignored.

Parameters

`IN=(vol-ser-no,label)`

Supplies the information needed to define the input file. The information is coded within parentheses.

`vol-ser-no`

Volume serial number.

label
File identifier for the input file.

OUT=
Supplies the information needed to define the output file for a copy operation or the secondary input for a compare operation.

vol-ser-no
Volume serial number.

RES
Output (or secondary input) is on SYSRES.

RUN
Output (or secondary input) is on the job's YRUN file.

label
File identifier for the output (or secondary input) file.

n
Number of extents in the file. Reserves extent table storage for use by the data access method.

EXTEND
Adds information to the end of a sequential file.

INIT
Initializes the file, starting with the first record each time the file is opened. Previous control information in the format labels is ignored at file open time and is overwritten by specifications contained in this DVC-LFD sequence at file close time.

PRNTR={lun[,dest] / N[,dest] / 20[,dest]}
Specifies a printer. The options are:

PRNTR=lun[,dest]
Indicates a logical unit number for a printer *(lun)* and optionally a 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote destination for spooled printer output (when dealing with remote batch processing).

PRNTR=N[,dest]

Indicates that you are supplying your own device assignment set for the printer - a DVC-LFD sequence is not to be generated. This option allows use of the // LCB and // VFB statements. A 1- to 6-alphanumeric-character identifier *(dest)* indicating a remote device for spooled printer output may also be specified when dealing with remote batch processing.

If omitted, a local printer with a logical unit number of 20 is assumed.

PUNCH={[redacted] / YES}

Generates the device assignment set for the punch. This is needed when the punch dual output feature (DC) is selected on the data utility control card.

COMPARE={[redacted] / YES}

Indicates this is a compare operation; will generate the appropriate file name for the secondary input.

EXT=

Supplies extent specification to be used when reserving system resources for a particular file. This information is coded within parentheses. All the parameters (except FIX) that apply to disk also apply to format-label diskette.

MI

MIRAM file; only valid specification for System 80 data-set-label diskette.

C

Allocate contiguous space; only valid specification for data-set-label diskette file.

F

Reformat the file at allocation time. Subparameter 4 must be **BLK.**

CF

Both of the preceding.

inc

Secondary increment (in cylinders) for automatic extension. An increment of zero indicates that there can be no dynamic extension of the file. If there is no *EXT* keyword parameter for this file, the value of the most recently specified secondary increment is used.

0

File cannot be dynamically extended; must be specified for data-set-label diskette files.

addr

Absolute beginning cylinder address, in terms of cylinders.

Tccc:hh

Absolute track address (hexadecimal) in cylinder/head format where the file is to begin. Allocation is in terms of tracks.

BLK

Allocation in terms of blocks; only valid specification for data-set-label diskette file.

TBLK

Allocates space in blocks; actual allocation is in terms of tracks.

CYL

Allocation in terms of cylinders.

TRK

Allocates space in tracks.

OLD

Secondary allocation increment change (subparameter 3) for an existing file; cannot be followed by any other subparameters if specified.

mi

Number of cylinders or tracks to allocate for this file; subparameter 4 must be *CYL, addr, TRK,* or *Tccc:hh.*

(bi,ai)

Allocation in terms of blocks (by cylinder or track); subparameter 4 must be *BLK* or *TBLK.*

bi

Is the average block length.

ai

Is the number of blocks to allocate.

Subparameters 6 through n

Same as subparameter 5 and used to describe additional extents.

Subparameter n+1

OLD

Indicates the file's previously allocated extent is to be increased by the allocation amount specified (*mi*, (*bi,ai*), etc).

If this subparameter is omitted, the request is for a new extent.

FIX

Indicates this extent is in the fixed-head area of an 8417 disk.

WORK/TEMP

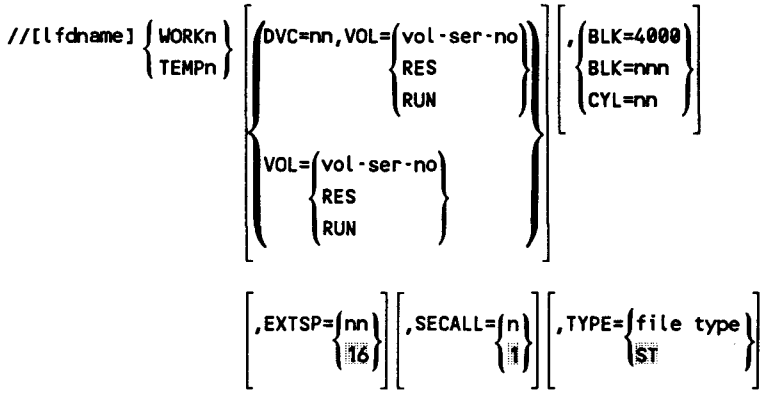
Function

The WORK and TEMP jproc calls generate any device assignment sets needed for temporary work files. The differences between the two jproc calls are:

- WORK sets up temporary files for one job step and deletes them at the end of the job step.
- TEMP sets up temporary files for the duration of a job and deletes them at the end of the job.
- WORK generates default file identifiers beginning with $SCRn.
- TEMP generates default file identifiers beginning with $JOB.

Note: *The file name in your program must match the file identifier generated by WORK or TEMP.*

Format



Note: *The n of WORKn/TEMPn is a number in the range 1 to 10.*

Label

lfdname

File name used when the file is defined by the generated procedure definition. This name corresponds to the name in the LFD control statement.

If omitted, the name generated for WORK1 is $SCR1; for WORK2, $SCR2; for WORK3, $SCR3, etc; the name generated for TEMP1 is $JOB1; for TEMP2, $JOB2, etc.

Parameters

DVC=nn,VOL={vol-ser-no / RES / RUN}

Specifies the work file device type.

nn

Is the logical unit number of the device.

If this keyword parameter is omitted, the file is assigned to the volume specified by the VOL parameter, or to the device containing the job's YRUN file for even-numbered files (WORK2, TEMP2, WORK4, TEMP4, etc) or the SYSRES volume for odd-numbered files (WORK3, TEMP3, WORK5, TEMP5, etc) if the VOL parameter is omitted. These may both be the same physical volume.

VOL={vol-ser-no / RES / RUN}

Specifies the volume serial number where the temporary file resides.

VOL=RES

Specifies the default if n in WORKn or TEMPn is odd.

VOL=RUN

Specifies the default if n in WORKn or TEMPn is even.

If this keyword parameter is omitted, SYSRES on the volume containing the YRUN file is used.

BLK=nnnn

Specifies the file size.

nnnn

Is the number of 256-byte blocks to be allocated.

Although specified in increments of 256 bytes, the file may be formatted to other block sizes at OPEN time. The size is rounded up to an integral number of cylinders.

If this keyword parameter is omitted and *CYL=nn* is also omitted, four thousand 256-byte blocks are allocated to the file.

CYL=nn

Specifies the file size.

nn

Is the number of cylinders to be allocated.

If this keyword parameter is omitted and *BLK=nnnn* is also omitted, four thousand 256-byte blocks are allocated to the file.

EXTSP={nn / 16}

Number of extents reserved in the prologue.

SECALL={n / 1}

Number of cylinders a file is extended if overflow occurs during a write operation.

TYPE={file type / ST}

Specifies the file type for the extent. It can be either ST (SAT) or MI (MIRAM). If this keyword parameter is omitted, the file type is ST(SAT).

WRTBIG/WRTSML

Function

WRTBIG and WRTSML print block letters on your printed output. They function the same, except that the characters produced by WRTBIG are larger than the characters produced by WRTSML. The following characters may be used

Letters A through Z

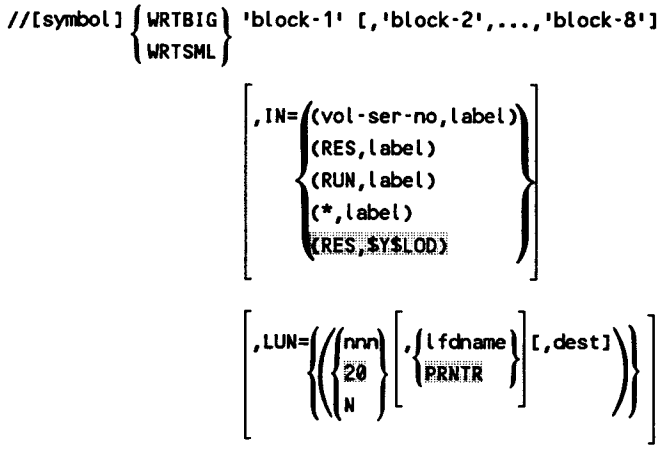
Digits 0 through 9

Special characters () + & * - / ? # = . $ [ ] " ' @ > < ! ! ; % , e

Embedded blanks

Note: *On 48-character printers, some special characters may not print.*

Format



Parameters

WRTBIG

This option allows block printing of up to 12 characters on each line and up to 4 lines on each page.

WRTSML

This option allows block printing of up to 20 characters on each line and up to 6 lines on each page.

Note: *It may be necessary to use the statement continuation feature of job control when coding your block parameters. Put a nonblank character in column 72 and begin the next line with / / n (n is a decimal number in the range 1-9 and is used to keep track of the number of continued lines).*

`block`

Information to be printed on a line. Can be up to 12 or 20 characters long and must be enclosed within apostrophes. Blanks may be used for visual presentation. If the first four characters are coded as 'JOB$', 'DAT$', 'TIM$', or 'VER$', the job name, system date, system time, or version number of the operating system in use is printed. Each may be specified alone, or with other options.

```
IN= { (vol-ser-no,label) }
    { (RES,label)        }
    { (RUN,label)        }
    { (*,label)          }
    { (RES,$Y$LOD)       }
```

Identifies the file that contains the load module called by the WRTBIG/WRTSML jproc. The options are:

`IN=(vol-ser-no,label)`

Indicates the volume serial number and file identifier.

`IN=(RES,label)`

Load module is on SYSRES, in the file identified by *label.*

`IN=(RUN,label)`

Load module is on the volume containing the job's YRUN file, in the file identified by *label.*

`IN=(*,label)`

Load module is on the volume identified in the file catalog with a file identifier specified by *label.*

```
LUN= { ( { nnn } [ , { lfdname } ] [,dest] ) }
     { ( { 20  } [   { PRINTR  } ]        ) }
     { ( { N   }                          ) }
```

Provides a logical unit number *(nnn)* and a file name *(lfd-name)* for the printer. A 1- to 6-alphanumeric-character identifier *(dest)* indicates a remote destination for spooled printer output (when dealing with remote batch processing).

If the job control proc is not to generate a printer device assignment set, code N as the value for this parameter and manually insert a device assignment set. This allows for the insertion of LCB and VFB control statements.

Section 4
Proc Definition Statements

END

Function

End of a procedure definition. The statements between the PROC and END proc definition statements are considered to be the body of the procedure definition, and, therefore, these statements are paired.

Format

LABEL	ΔOPERATIONΔ	OPERAND
[[//]symbol]	END	unused

Label

symbol
: Specifies a 1- to 8-character symbol used as the target address of a branch instruction.

If omitted, the END statement cannot be referenced.

No parameters required.

NAME

Function

Supplies the procedure definition name. It must immediately follow the PROC statement. You may use more than one NAME statement, but all must be in the beginning of the definition. Each statement provides a unique name for the same definition. This name provides a reference point within the body and a source code level.

Multiple NAME statements allow you to specify different parameters in the operand fields of each NAME statement; then you may select the desired parameters by calling the proc by that particular name.

Format

LABEL	ΔOPERATIONΔ	OPERAND
[//]symbol	NAME	parameter

Label

symbol
: Symbolic name by which the procedure definition may be called. May not be any valid job control statement names (DVC, QGBL, etc.).

Parameters

parameter
: Parameter or a parameter sublist to be used at job execution time.

PROC

Function

Procedure definition start. A valid symbol in the label field is used as a dummy label that represents the *procname* statement label. This dummy label attaches to the expanded procedure definition each time it enters the control stream. The *procname* statement label replaces the dummy label when the procedure definition is called. If the *procname* statement has no label, a null character string is used.

Format

LABEL	ΔOPERATIONΔ	OPERAND
[[//]symbol]	PROC	[pos,n][,k,...,k]

Label

symbol
: Specifies a 1- to 8-character entry point to this procedure definition when it is expanded and inserted into the control stream. There may be a corresponding label on the procname statement.

Parameters

pos
: Positional parameter reference symbol in the body of the procedure definition.

  Positional parameters specified in a jproc call are associated with positional parameters specified in job control statements in the body of the jproc definition. The PROC directive specifies the number of positional parameters allowed.

  If omitted, no positional parameter substitution is possible.

n
: Decimal number that represents the total number of positional parameters to be found in the procedure call statement.

  If omitted, zero is assumed.

Note: *If you omit the pos and n parameters, you must still code two commas before you can code any keyword name values.*

k

Represents the name or names used in referencing keyword parameters contained in the body of the procedure definition.

When a keyword parameter is given a value in the jproc definition, it takes that value if the keyword parameter is omitted in the jproc call. To preset a keyword value, the *k* parameter takes the form [,K=value-1,...,K=value-n].

If omitted, no keyword parameters are contained in the body of the procedure definition.

procname

Function

Calls a control stream procedure definition. This definition must be either defined and available in YJCS or embedded within the control stream and must precede any reference to it. The specified positional parameters are associated with certain individual positional parameters of a job control statement in the procedure definition body. The amount of positional parameters in this *procname* statement must be equal to the amount in the PROC statement.

If the value of an omitted positional or keyword parameter has been preset in the PROC statement, then the preset value is used in the called procedure definition. If the value has not been preset, then the omitted positional or keyword parameter is set to a null character string. Multiple procedure calls are not allowed on the same line.

Format

```
//[symbol] procname p1,p2,...,pn,ki=vi,kj,...,km=vm
```

Label

symbol
: Dummy label. When specified, this symbol must correspond to the symbol specified in the label field of the PROC statement.

  If omitted, the value of the keyword parameter in the label field is null.

Operation

procname
: Procedure definition name. This must be the same name as that specified in the label field of the NAME statement of the called procedure definition.

Parameters

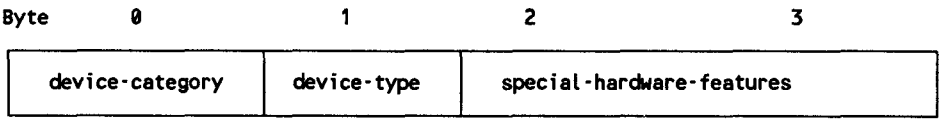
p1,p2,...,pn
: Represent positional parameters.

ki=vi,kj,...,km=vm
: Represent keyword parameters.

Appendix A
Logical Unit Number Assignment Information

A device may be designated by a logical unit number at system generation (SYSGEN) time. You may use a standard table or relate the logical unit numbers according to the conventions of the installation. The range of numbers should be as compact as possible to save main storage space. The logical unit numbers are set so that certain ranges of numbers refer to certain categories of devices. These ranges cannot be changed; only the specific entries within the ranges can be changed. Within each category may be several types of devices. For example, tape is a category; Unisys magnetic tape units are the types within that category. The association of the logical unit number to the category and the type of device is maintained as a 512-byte table on the system resident device (SYSRES). The logical unit number is used as an index into the table. This association may be altered at job execution time via the *EQU* statement.

Each entry in the table is four bytes long. The general format for an entry is as follows:



The following paragraphs summarize the logical unit number values for each device category.

Printers

The value range of logical unit numbers for printers is 14-29. The values 20 and 21 indicate that any printer may be used. If a specific type of printer is required, the proper logical unit number must be used in assigning the device. The standard logical unit number assignments are listed in Table A-1.

System 80 uses the following printers:

- Unisys 0789 printer subsystem
- Unisys 0776 printer subsystem (non-SDMA) (model 8 only)
- Unisys 0776 printer subsystem (all models)
- Unisys 0770 printer subsystem (model 8 only)
- Unisys 0798 printer subsystem
- Unisys 0791 printer subsystem

Card Reader Subsystems

The value range of logical unit numbers for card readers is 30-39. The values 30 and 31 indicate that any reader may be used. If a specific type of reader is required, the proper logical unit number must be used in assigning the device. The standard logical unit number assignments are listed in Table A-1.

System 80 uses the following card readers:

- Unisys 0719 card reader subsystem
- Unisys 0716 card reader subsystem (model 8 only)

Card Punch Subsystems

The value range of logical unit numbers for punch units is 40-49. The values 40 and 41 indicate that any punch unit may be used. If a specific type of punch unit is required, the proper logical unit number must be used in assigning the device. The standard logical unit number assignments are listed in Table A-1. System 80 uses the Unisys 0608 Card Punch Subsystem.

Disk Subsystems

The value range of logical unit numbers for disk units is 50-89 and 160-177. The values of 50 to 59 indicate that any disk unit may be used; the specific type of disk unit is not essential to the execution of the program. If a specific type of disk unit is required, the proper logical unit number must be used in assigning the device. The standard logical unit number assignments are listed in Table A-1.

System 80 uses the following disk subsystems:

- Unisys 8417 disk subsystem
- Unisys 8419 disk subsystem
- Unisys 8416 disk subsystem (model 8 only)
- Unisys 8418 disk subsystem (model 8 only)
- Unisys 8430 disk subsystem (model 8 only)
- Unisys 8433 disk subsystem (model 8 only)
- Unisys 8470 disk subsystem (models 4, 6, and 8 only)

Diskette Subsystems

The range value of logical unit numbers for diskettes is 130-153. The values 130-133 indicate that any diskette unit may be used. System 80 uses the Unisys 8420 and 8422 diskettes.

Magnetic Tape Subsystems

The value range of logical unit numbers for tape units is 90-127. The value of 90 indicates that any tape unit may be used; the specific type of tape unit is not essential to the execution of the program. If a specific type of tape unit is required, the proper logical unit number must be used in assigning the device. The standard logical unit number assignments are listed in Table A-1. System 80 uses the following magnetic tape subsystems:

- UNISERVO® 10 magnetic tape subsystem
- UNISERVO 10/14 magnetic tape subsystem (model 8 only)
- UNISERVO 12/16 magnetic tape subsystem (model 8 only)
- UNISERVO 20 magnetic tape subsystem (model 8 only)
- UNISERVO 22/24 magnetic tape subsystem (model 8 only)
- Streaming Tape magnetic tape subsystem

UNISERVO is a registered trademark of Unisys Corporation.

Workstations

The value range of logical unit numbers for workstations is 200-219. The values 200-215 indicate that any workstation may be used.

Standard Logical Unit Numbers

Table A-1 provides the standard logical unit numbers and their corresponding device types.

Table A-1. Standard Logical Unit Number Assignments

Logical Unit No.	Device Type and Features	Device Type Code
1-13	Spare	FFFFFFFF
14, 15	0791 Correspondence quality printer (CQP-1)	04040000
16,17	0798 printer, no features specified	04010000
18,19	0789 printer	04020000
20,21	Any printer, no feature specified	04FF0000
22,23	Spare	FFFFFFFF
24,25	0776 printer, no optional features specified	04100000
26,27	Spare	FFFFFFFF
28, 29	0770 printer, no optional features specified	04800000
30,31	Any card reader subsystem, no features specified	08FF0000
32,33	0717/0719 card reader, no feature specified	08200000
34, 35	0716 card reader, no features specified	08800000
36, 37	0711 card reader, no features specified	08400000
38, 39	Spare	08100000
40,41	Any card punch subsystem, no features specified	02FF0000
42-45	Spare	FFFFFFFF
46, 47	0608 card punch	02010000
48	Any remote printer, no features specified	04FF8000
49	Spare	FFFFFFFF
50-59	Any disk	20FF0000

continued

Table A-1. Standard Logical Unit Number Assignments (cont.)

Logical Unit No.	Device Type and Features	Device Type Code
60-63	8419 disk subsystem	20108000
64-66	8416/8418 MOD1 disk subsystem (low density)	20028000
67-69	8418 MODII disk subsystem (high density)	20028004
70-74	8430 disk subsystem	20200000
75-79	8433 disk subsystem	20200004
80-85	8494 disk subsystem	2001C000
86-90	Spare	FFFFFFFF
90-99	Any tape, no features specified	10FF0000
100-102	Any tape, 9-track phase-encoded	10FF000A
103-105	Any tape, 9-track NRZI	10FF0006
106-109	Any tape, 7-track NRZI	10FF0005
110-112	Slow tape, 9-track phase-encoded (UNISERVO 10,12)	10C8000A
113-115	Slow 9-track NRZI (UNISERVO 10, 12)	10C80006
116-119	Slow tape, 7-track NRZI (UNISERVO 10, 12)	10C80005
120-122	Fast tape, 9-track phase-encoded (UNISERVO 14, 16, 20)	1034000A
123-125	Fast tape, 9-track NRZI (UNISERVO 14, 16, 20)	10340006
126-127	Fast tape, 7-track NRZI (UNISERVO 14, 16, 20)	10340005
128-129	Streaming tape	1002000A
130-133	Any diskette	40FF0000
136-137	8420/8422 diskette	40010000

continued

Table A-1. Standard Logical Unit Number Assignments (cont.)

Logical Unit No.	Device Type and Features	Device Type Code
138, 139	Any diskette, 128-byte	400F0001
140-141	Any diskette, 256-byte	400F0002
142-143	Any diskette, 512-byte	400F0004
144-145	Any diskette, 1024-byte	400F0008
146-147	Spare	FFFFFFFF
148-149	Double-density diskette (MFM recording)	40FF0020
150, 151	Any diskette, autoload	40FF0100
152, 153	Any diskette, double-sided	40FF0040
154-159	Spare	FFFFFFFF
160-163	Spare	FFFFFFFF
168, 169	Any fixed-head disk	20080020
170-174	8417 disk subsystem	2008C000
175-179	8470 disk subsystem	20040000
180-188	Spare	FFFFFFFF
190-194	8430/8433 disk subsystem (IDA-NUK)	20020008
195-199	8433/8433M disk subsystem (IDA-NUK)	2002000C
200-215	Any workstation	01FF0000
216-219	Any workstation with 24 x 80 screen	01FF0004
220-223	Any printer, class=1	04FF0001

continued

Table A-1. Standard Logical Unit Number Assignments (cont.)

Logical Unit No.	Device Type and Features	Device Type Code
224-227	Any printer, class=2	04FF0011
228-231	Any printer, class=3	04FF0111
232-254	Spare	FFFFFFFF
255, 256	Any workstation with printer attached	01FF0080

Appendix B
Extent Specification Information

Files are defined on direct access storage devices in terms of extents. An extent is space on the volume reserved for the file and is made up of contiguous cylinders. Files may be 1 to 16 extents for each volume in the file and may be allocated to any available area on the disk pack. Space on a disk may be allocated on a contiguous or noncontiguous basis, depending on your job requirements. You can request space on a disk in one of the following ways:

- Absolute address (cylinder or track)

  The starting address of your file specified as the absolute address of a cylinder or track. The number of cylinders and tracks required also must be specified.

- Number of tracks

  You can allocate files by specifying the number of tracks needed. This allows you to manage your file space more efficiently than if you allocate files by cylinder.

- Number of cylinders

  You can state the number of cylinders your file needs, rather than stating the exact starting address. Job control places your file in an area that is the best fit with the other files on the disk. Your starting address is placed in the volume table of contents and returned to the job in the file control block. However, you need only request the file by name; the software accesses it for you.

  The allocation is in terms of whole cylinders.

- Number of blocks (by cylinder)

  You can state the number of blocks in the file and the average length of the blocks; job control calculates the amount of space needed to fulfill your request. The amount of space assigned varies with the type of disk subsystem used. This method is the most flexible way to allocate extents on a disk since it is not device dependent.

- Number of blocks (by track)

  You can also allocate a file by specifying the number and average length of blocks it requires and have that specification converted to the number of tracks needed. The actual allocation, in this case, is by track.

Glossary

A

audit file
Optional file produced by the audit version of the dialog processor. It contains a record of your responses during a dialog session.

audit version of the dialog processor
The version of the dialog processor that produces an audit file.

C

COMREG
A 12-byte communications region in the *job preamble* available to the user for passing vital information from one *job step* to another. The last byte is the *UPSI* byte (user program switch indicator).

D

data set delimiters
The /$ and the /* control statements. They should follow the EXEC control statement in the control stream or, if used, any PARAM control statements. When job control detects the /$ statement, it reads without verifying and places in the job's YRUN file all subsequent statements up to and including the /* statement. The data set within the delimiters may be source code or other control statements.

data set labels
The data set labels are similar in function to a VTOC. They reside on head 0 and describe the types of records and their characteristics that make up the diskette file.

device assignment sets
The job control statements used to identify your devices and files to the system. A device assignment set can include these statements: DVC, VOL, EXT, LBL, UID, USE, LFD.

dialog processor
An OS/3 system program that manages interactive dialogs, including the *job control dialog.*

dummy data set
A dummy data set consists of only a /$ and /* statements. They are used with some language jprocs.

F

file cataloging
The method of maintaining, in a centralized catalog, all the information needed by a job to access a file that is common to other jobs as well. Also provides protection against unauthorized access or update by using read/write passwords.

file control block (FCB)
Contains information needed to control a specific file.

file identifier
The physical label of a file. It is specified in the LBL control statement and may be up to 17 characters for a tape file and 44 characters for a disk or format-label diskette file. It is used to locate the *volume table of contents (VTOC)* entry that contains control information for the file.

It is advisable to have the file identifier different from the *file name.*

file name
The name of the logical file specified on the LFD statement is considered to be an "internal" file name. It must be the same as the name specified in the label of the DTF macroinstruction when the file is defined for data management, or in the label field of the DTFPF macroinstruction when the physical IOCS function of the supervisor is used. It is logical file name that you use when you want to access the file.

The LFD file name is used to access the *file control block* associated with this particular file. Job control generates several control tables/blocks; each is made up of related information.

file processor
Reads control streams from the card reader and permanently stores them in YJCS. The control streams may consist of control statements, procedure definitions, or both, and each control stream must be given a unique name. No validity check is made at the time a control stream is stored in YJCS. The file processor is activated by the FILE operator command or the FILE workstation command.

I

interactive processing
A computing environment where you communicate with the operating system through a workstation.

interstep processor
Performs end-of-job functions and final housekeeping duties required at the end of a *job step*. These include.

- releasing devices not required by the remaining job steps in the *job;*
- releasing temporary and user-specified disk space;
- storing pertinent logging data.
- calling the *job step processor* (if this is not the last job step); and
- calling the *job termination* function (if this is the last job step).

J

JC$BLD
A system program that uses your responses to the job control dialog to build a job control stream or user jproc. The program is initiated by the SC JC$BLD workstation command.

job
A unit of work to be performed. Each job can be divided into *job steps* (programs) to be executed serially and must be given a unique name.

job control dialog
Unisys supplied dialog that guides you step-by-step through the process of building a job control stream or user jprocs.

job run library file
The default job temporary library for most system processors to use for input/output storage.

The *file identifier* for the temporary job run library file is YRUN.

job prologue
Area reserved at the beginning of the job region in main storage that includes the job preamble, any extent request tables, the task control blocks, open file table, and job accounting table.

job queue
Contains an entry for each *job* defined and stored in a YRUN file. The entry is filed according to one of three priorities: normal, high, or preemptive.

job scheduler
Schedules jobs for execution based on a 3-level priority queue.

job step

Unit of work associated with one processing program. A job step is an executable program consisting of one or more tasks that requires a specific amount of the hardware resources of the system.

job step processor

Prepares a *job step* for execution. This includes:

- allocating devices for the job step;
- locating and updating the *file control blocks* for this job step;
- posting the disk address of data in the job preamble;
- setting option indicators;
- requesting disk space allocation or file extension
- storing system logging information;
- performing any requested utility functions; and
- performing housekeeping and bookkeeping functions for the job step as required by the system.

The job step processor terminates when the user program is brought in for execution.

job termination

Either the normal or abnormal end of a job. All terminations result in the deallocation of all system resources (such as peripheral devices, main storage, and disk scratch area) previously allocated to the *job*. Any remaining data images or control statements in the control stream are bypassed.

- Normal Terminations

  These are initiated by the program, control stream, or system operator.

- Abnormal Terminations

  These are caused by program errors, control stream errors, or the expiration of the estimated processing time for the job.

L

logical unit number

A number preassigned to a unit by device type and characteristics at each installation. This number is used to access an entry in a control table contains the type of device being assigned, plus information regarding the use of that device.

M

main storage allocation/requirements
The assignment of blocks of main storage to jobs and system functions.

The minimum amount of space need be only that required for the longest load module in your program. Job control calculates how much additional space is required for the various control tables. This establishes the true minimum amount of main storage space that your program uses for its execution.

multijobbing
The concurrent scheduling, loading and execution of more than one job at a time. OS/3 job control can initiate up to 7 or 47 jobs concurrently, depending on your system configuration.

multiple indexed random access method (MIRAM)
Allows processing of logical records by relative record number in a random or a consecutive order or by multiple keys (indexed) in a random or a sequential order. Up to five keys can be searched.

multitasking
Executing asynchronous multiple tasks within a given job step.

P

phase name
The program name plus the 2-digit phase number appended by the linkage editor. The combined program name/phase number is placed in the phase header generated by the linkage editor.

physical unit block (PUB)
A control block in main storage containing the peripheral device information used by the physical I/O control system (PIOCS).

program name
The name of the program that is to be executed in the *job step*. This is the name given to the load module by the linkage editor.

R

run processor
Identifies, interprets, and analyzes the statements in the second stream. Based on this information, the run symbiont:

- builds the control blocks that describe the job requirements to the system:

(Continued)

(cont.)

- creates a temporary run library for the *job;*
- expands job control procedure calls;
- places an entry for the job in the *job queue* for scheduling; and
- passes control to the job scheduler.

S

scheduling priorities
The control streams for jobs submitted for execution are queued on the system resident device (SYSRES) by a scheduling priority. This priority is specified on either the JOB statement or the RUN statement and is normal, high, or preemptive. The priority applies to the entire *job* and is considered to be normal unless otherwise specified. High priority should be reserved for emergency scheduling situations and is, therefore, rarely used. Preemptive priority is higher and causes the currently executing job to be preempted. A priority specified on the RUN statement overrides a priority specified on the JOB statement.

spooling
The process of having the central processor reading or writing records from or to a high speed device, rather than directly from a slower device.

summary report
An optional printed report output by the dialog processor that contains a summary of a dialog session organized by sequentially numbered paragraphs.

switching priority
Determines the order in which central processor control is passed from task to task. The number of user switching priorities varies from 1 to 62. The number of priorities is established at system generation time. The minimum supervisor configuration requires only one priority, regardless of whether there is *multijobbing or multitasking* environment.

system access technique (SAT)
A specialized block level device handler that provides great efficiency in handling disk files.

system information block (SIB)
An area in main storage containing system control information.

system job control library file (YJCS)
Provides permanent storage for control streams. It can be used as the output file for the *file processor* and can be input to the *run processor* whether by specification or by default.

system library

A name, generally used to refer to the standard system files in an operating system. In OS/3, the system library includes five permanent system library files and a temporary job run library file for each job input to the system:

- System load library file (YLOD)
- System object library file (YOBJ)
- System source library file (YSRC)
- System macro, or procedure, library file (YMAC)
- System job control stream library file (YJCS)
- Temporary job run library file (YRUN jobname)

The preceding breakdown is defined by the operating system. OS/3 supports different system programs. Library files may be composed of both user and system programs. Such a mix is transparent to the librarian.

system load library file

Provides permanent storage for the executable programs (load modules) supplied as part of the operating system. Can also be used to provide permanent storage for user load modules. This file is used as the input library by the system loader.

The *file identifier* for the system load library file is YLOD.

system macro library file

Provides permanent storage for the standard system macro definitions. Can also provide permanent storage for user macro definitions. This file is used as an input library by the assembler macro facility.

The *file identifier* for the system load library file is YMAC.

system object library file

Provides permanent storage for the object modules applied as part of the operating system. Can also provide permanent storage for object modules of user programs. This file is used as an input library by the linkage editor.

The *file identifier* for the system object library file is YOBJ.

system source library file

Provides permanent storage for source modules that consist of source coding as processed by language processors.

The *file identifier* for the system source library file is YSRC.

T

task

A point in a program where central processor control is required for continued program execution. Every job step has at least one task, and there may be a maximum of 256 tasks. Also see *switching priority* and *multitasking.*

task control block (TCB)

An area in the job prologue containing information needed to control a specific task.

U

user program switch indicator (UPSI)

Last byte of the 12-byte communication region in the *job preamble.* The two most significant bits of the UPSI byte are used to communicate error conditions to the user.

V

volume serial number

One to six alphanumeric characters used to identify a physical tape or disk volume. All disk packs and magnetic tape units, including scratch volumes, should be assigned a volume serial number. Each installation may have its own system for numbering volumes.

volume table of contents (VTOC)

Contains the address of all information stored on a direct access volume.

The VTOC is made up of seven different format labels, defined as follows:

- Format 0 Label

The format 0 label represents available space within the extent of the VTOC itself. The format 0 label contains all binary 0's. When a format 1, 2, 3, 5, or 6 label is deleted from the VTOC, a format 0 label is written in its place.

- Format 1 label

The format 1 label identifies any file on a disk volume except for the VTOC file itself. There must be one format 1 label for each file or part of a file on each volume. Up to three contiguous or noncontiguous areas, or extents, occupied by the file can be identified in the format 1 label. One format 2 label or one format 3 label can be chained to a format 1 label.

- Format 2 Label

The format 2 label provides additional description for a format 1 label. If present, the format 2 label is chained to a format 1 label.

- Format 3 Label

The format 3 label is used if the file consists of more than three extents. If present, the format 3 label is chained to a format 1 label, format 2 label, or another format 3 label. A format 3 label can describe up to 13 extents.

- Format 4 label

The format 4 label describes the VTOC itself. It is always the first record in the VTOC

- Format 5 label

The format 5 label describes up to 26 contiguous or noncontiguous extents that are available for allocation on a volume. It is always the second record in the VTOC. Several format 5 labels may be chained together if more than 26 contiguous or noncontiguous extents are available on the volume.

- Format 6 Label

The format 6 label describes up to 26 split-cylinder extents. Format 6 labels may be chained together, with the first format 6 label being chained to the format 4 label.

All format labels are described in detail in the data management user guide.

W

workstation

A terminal device with a screen for display of system and user information (including dialog text) and a keyboard for user input (including responses to a dialog).

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

UNISYS

USER COMMENTS

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

_____

(Document Title)

_____ _____ _____

(Document No.) *(Revision No.)* *(Update Level)*

Comments:

From:

_____

(Name of User)

_____

(Business Address)

Fold on dotted lines, and mail. (No postage is necessary if mailed in the U.S.A.)
Thank you for your cooperation

CUT

FOLD

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
E/MSG Product Information Development
PO Box 500 — E5-114
Blue Bell, PA 19422-9990

FOLD

UNISYS

USER COMMENTS

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

_____

(Document Title)

_____ _____ _____

(Document No.) *(Revision No.)* *(Update Level)*

Comments:

From:

_____

(Name of User)

_____

(Business Address)

Fold on dotted lines, and mail. (No postage is necessary if mailed in the U.S.A.)
Thank you for your cooperation

CUT

FOLD

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
E/MSG Product Information Development
PO Box 500 — E5-114
Blue Bell, PA 19422-9990

FOLD