

IBM Job Control Language Conventions at CNS

CNS Document ID: D0013

Last Updated: 07/26/2006

This document provides a brief explanation of the format for job control language (JCL) statements used in CNS documentation. It is not intended to be a complete reference for JCL, but rather a guide.

For detailed documentation on JCL, please refer to the following:

The most recent version of the JCL text by Gary DeWard Brown

[[http://www.amazon.com/s/104-5512841-5030364?ie=UTF8&index=books&rank=-relevance%2C%](http://www.amazon.com/s/104-5512841-5030364?ie=UTF8&index=books&rank=-relevance%2C%2Fdate-sort-order)

CNS's Batch I/O, Tapes, and Datasets documents [<http://docweb/cgi-bin/docweb>]

UF Computing & Networking Services

112 Bryant Space Sciences Bldg, University of Florida

P.O. Box 112050

Gainesville Florida 32611-2050

(352) 392.2061

<editor@cns.ufl.edu>

Table of Contents

Job Control Language Conventions	3
JCL Conventions in CNS Documentation	3
JCL Examples	4
Job Control Statements	4
Job Control Statement Rules	5
Invalid Control Statements	6
The /*INCLUDE Statement In NERSP	6
The JOB Statement	7
JOB Statement Examples	9
The JOBPARM Statement	9
Syntax	10
Examples	11
The ROUTE Statement	12
ROUTE Statement Examples	13
Network Jobs	13
The SETUP Statement	13
Examples	14
The JCLLIB Statement	14
JCLLIB Statement Example	15
The EXEC Statement	15
Specifying a Cataloged Procedure	15
Specifying a Program Name	15
Specifying Options on the EXEC Statement	15
The DD Statement	15
SYSOUT : Output Stream Data Sets	16
More JCL Notes	17
Batch Input	17
Batch Output	17
Tape and Disk Storage	18

Job Control Language Conventions

This document provides a brief explanation of the format for job control language (JCL) statements used in CNS documentation.

This information is not intended to be a reference for JCL; it is simply a guide to how JCL is used in CNS documentation.

For complete documentation on JCL, please refer to the following:

- *IBM z/OS MVS JCL User's Guide* (available from the CNS BOOKSRV [<http://booksrv.cns.ufl.edu/>] service).
- *IBM z/OS MVS JCL Reference* (available from the CNS BOOKSRV [<http://booksrv.cns.ufl.edu/>] service).
- The most recent version of the JCL text by Gary DeWard Brown [<http://www.amazon.com/s/104-5512841-5030364?ie=UTF8&index=books&rank=-relevance%2C%2B>]
- CNS's Batch I/O, Tapes, and Datasets documents [<http://docweb/cgi-bin/docweb>]

JCL Conventions in CNS Documentation

Although specific JCL formats may change slightly from one document to another, several general conventions remain standard throughout all CNS documentation. Each reference will follow the general conventions listed below:

- Unless otherwise specified, the first character of the statement or set of statements begins in column 1.
- If a number appears above or below the beginning of an element of a statement, this indicates that the particular element must begin in the specified column. If no column number is shown, assume that the element in question does not have to appear in a specific column (unless it is the first character of a statement), but only in the relative position shown.
- Parameters shown in uppercase letters must be entered EXACTLY as shown.
- Parameters shown in lowercase letters represent entries to be supplied by you and simply show where these entries should go in relation to the other elements of the statement. All JCL must be in uppercase: when you replace the parameters shown in the JCL examples, you must do so in uppercase.
- Unless otherwise specified, any punctuation marks or special characters appearing as data must also be entered EXACTLY as they appear in the example.
- Blank spaces should not be inserted where they are not specifically shown, nor should they be removed if they are shown.

JCL Examples

Example 1. /*SETUP TAPE x,n

Using the conventions presented above, you must enter the indicated JCL as follows:

- The first slash (/) must be in column 1 since it is the first character of the statement.
- The /* are special characters and the comma is punctuation; therefore, they must be entered exactly as they appear.
- The words **SETUP** and **TAPE**, which appear in uppercase letters, must be entered exactly as they appear.
- The absence of a column number over the word **TAPE** indicates that it does not have to appear in a specific column, although it must be separated from the word **SETUP** by at least one blank space.
- The elements " x,n " indicate information to be supplied by you. In this example, you would type the word TAPE followed immediately by the type of tape drive needed (9 or C), followed by a comma, followed by the number of tape drives needed. No blank spaces are allowed except to separate the words SETUP and TAPE.

Example 2. // EXEC SAS

Using the conventions described above, this JCL statement would be entered as follows:

- The absence of column numbers indicates that the elements do not have to appear in specific columns, except that the slashes must begin in column 1.
- The two slashes, as special characters, must be entered exactly as they appear. At least one blank space separates the slashes from EXEC.
- The EXEC and SAS elements, which appear in uppercase, must also be coded exactly as they appear with at least one blank space separating them.

Job Control Statements

A batch job includes instructions that tell the system what you want it to do. An MVS batch job requires that the instructions be in the form of IBM Job Control Language (JCL) and Job Entry Subsystem 2 (JES2) control statements. These control statements tell the operating system or JES2 how to process your job. Some of these statements are required in every job; others can be used to alter the default flow of processing. The most common of these

statements and local modifications of them are described here.

To help reduce the amount of JCL you need to write, CNS provides cataloged procedures (procs) for most of the higher-level languages and for most major programs and packages. We strongly encourage you to use these procedures because as versions of software change, CNS keeps these procedures current.

Job Control Statement Rules

The following rules apply to all job control statements:

- All statements must be card images, that is, 80-byte records.
- All text must be in UPPERCASE.
- The basic syntax of JCL statements is as follows:

Example 3. //name operator parameters

The "operator" is either **JOB**, **EXEC**, **DD**, **JCLLIB**, or **OUTPUT**. A space separates each part. No other spaces are allowed within the text.

- The basic syntax for a JES2 statement is as follows:

Example 4. /*name parameters

A space separates the name from the first parameter. No other spaces are allowed. Parameters may be either single keywords (such as **PRINT**) or a keyword and a value.

- All JCL statements begin with a double slash (//) in columns 1 and 2; JES2 statements begin with slash-asterisk (/*) in columns 1 and 2. The text begins in column 3.
- Text must appear in columns 3-71. Columns 73-80 can be used for comments or numbers, or be left blank. When JCL text is too lengthy to fit in the columns designated for it (columns 3-71 of an 80-byte line), statements must be continued on the next line(s). Statements may be interrupted only AFTER a complete logical parameter or subparameter AND the comma immediately following it. Do not add a comma to the statement to make a continuation; interrupt the statement only after an existing comma. Although the statement text cannot go past column 71, the continuation comma can appear in column 72 or earlier. Do not interrupt a statement following a comma that is part of a parameter enclosed by parentheses. Begin the continuation line of a JCL statement with // in columns 1 and 2. The text of the continuation statement can begin anywhere between columns 4-16 (i.e., there must be at least one blank between the // and the continuation text).

For more information about continuing statements, see one of the reference books listed in

Chapter 1 of this document.

Invalid Control Statements

If your job has an unrecognizable control statement, the job will be rejected as soon as the invalid statement is detected. Common errors that cause a JCL or JES2 statement to be invalid are misspelled keywords and syntax errors.

You should note, however, that you could supply incorrect information on a syntactically correct control statement (e.g., an incorrect data set name on a **DD** statement) and it will NOT be rejected. These types of errors usually are not detected until the program begins executing.

Invalid JES2 Statements

Invalid **/*** (JES2) statements in any MVS batch job will cause the job to be rejected as soon as the improper JES2 statement is detected. This ensures that jobs with invalid **/*** statements will not execute. It is designed to save you from the unnecessary cost of jobs that are likely to produce results other than what you had intended. For instance, if you code an invalid **/*ROUTE** statement, the job will be terminated as soon as that statement is encountered rather than running to completion and then routing the output to an invalid/undefined site.

An invalid JES2 statement will generate the following error message in the job output:

```
$HASP118 INVALID /* CONTROL STATEMENT
```

The error message will appear underneath the line that contains the invalid statement.

The **/*INCLUDE** Statement In NERSP

Under NERSP [<http://nersp.cns.ufl.edu/>], the **SUBMIT** command allows you to use a **/*INCLUDE** statement in a NERSP file when you submit a batch job to the OS/390 (MVS) system. The **/*INCLUDE** statement allows the contents of one file to be included as part of the job stream when the job is submitted.

The format of the NERSP **/*INCLUDE** statement is:

```
/*INCLUDE fileidentifier
```

The file being included can contain another **/*INCLUDE** statement, which can in turn include another file; files can be included in this way up to 99 levels of nesting.

Note

You cannot use **/*INCLUDE** with the **SCHED** command.

Incorrect coding of the **/*INCLUDE** statement could result in a JCL error or in having the wrong file included.

Enter **man submit** from NERSP for details on the use of the **SUBMIT** command.

The JOB Statement

The **JOB** statement is the first statement in your batch job. It is required for every batch job submitted to the OS/390 system.

The **JOB** statement can be continued using standard JCL continuation procedures.

Figure 1. The JOB Statement

```
//jobname JOB , 'your name' , CLASS=class , TIME=(mm,ss) , LINES=lines ,  
//          REGION=region , TYPRUN=type , MSGCLASS=sysoutclass  
//          USER=userid , PASSWORD=password
```

<i>jobname</i>	is up to eight alphabetic or numeric characters specifying the name of the job. The first character must be alphabetic. The name cannot contain blanks or non-alphanumeric characters (you can include the national characters #, @, and \$), and it must begin in column three following two slashes.
JOB	is a keyword that identifies the JOB statement. It is preceded by at least one blank and followed by at least one blank.
TIME=(mm , ss)	specifies the maximum time, in CPU minutes and CPU seconds, that you want the job to run. If the job runs more than this time limit, it will not run to completion. This number must be four digits or fewer for minutes and two digits or fewer for seconds. A value of 0 means 1 second. The default is 6000 seconds (100 minutes). If you do not specify time inside parentheses, then it is assumed that are referring to minutes. For example, a TIME=5 means that you want your code to run for a maximum of 5 minutes. If you want to use seconds only, then use the following code: (,ss). You may use a maximum time of 357912 minutes and 59 seconds (357912,59).
LINES=	specifies the maximum number of lines, in thousands, the job can generate for one copy of the job. If the job attempts to spool more lines, it will be canceled. The maximum that can be specified is 999999 (999,999,000 lines); the default is 99,000 lines. A value of 0 allows up to 500 lines to print. The number of lines applies to any additional copies requested; that number of lines will be printed for each copy.
'your name'	is your name or some other identifying character string enclosed in single quotes. You can use up to 20 characters. If the string only uses alphanumeric characters and contains no blanks, you can omit the single quotes. (Note: a "." is allowed without quotes, but no other special characters are.)

CLASS=	is a single-character job class. Refer to CNS document D0058, <i>OS/390: Understanding CNS OS/390 Job Classes</i> [http://docweb.cns.ufl.edu/docs/d0058/d0058.html], or to the <i>CNS Charging Algorithm</i> [http://docweb.cns.ufl.edu/docs/d0001/d0001.html] document for a description of job classes. The default is Class A (normal priority).
REGION=	is a keyword to specify the maximum amount of main memory the job will need for any job step. The REGION parameter on the JOB statement overrides region specifications on EXEC statements. You can specify the amount as kilobytes (e.g., 2048K) or in megabytes (e.g., 2M). The default region size is 2MB. The maximum region for regular batch classes is 45MB. The maximum region for research job classes is 175MB. If you need more than 175MB, you must make special arrangements. Jobs that require large regions (more than 16MB) can be delayed in executing until the requested amount of memory becomes available.
TYPRUN=	can be used to request special processing. If not used, the job's JCL will be scanned when the job is read in and, if there are no syntax errors in the JCL, the job will be queued for execution. This process can be altered by using one of the operands below:
TYPRUN=SCAN	The JCL will be scanned only; the job will not be executed.
TYPRUN=COPY	A copy of the input will be produced. The JCL will not be scanned and the job will not be executed.
TYPRUN=HOLD	The job will be placed in HOLD after the JCL has been scanned. It will not be queued for execution until specifically released from HOLD by you from IOF or with the RELEASE utility.
TYPRUN=JCLHOLD	The job will be placed in HOLD immediately. The JCL will not be scanned until the job is released from HOLD.
MSGCLASS=	specifies the job scheduler message output class. The valid output classes are described in Chapter 9 under "SYSOUT: Output Stream data Sets."
USER=	specifies the userid under which the job is to run. This option is not needed if the job is to run using the same userid from which it was submitted.
PASSWORD=	specifies the password for the userid indicated on the USER= keyword. NEVER save passwords in your files. Leave this off the JOB statement except when submitting a batch job to a different userid than you are signed on to. If someone sees

your password, they could gain access to your account.

You can change your password by entering the following:

PASSWORD=(*password*,*new-password*)

JOB Statement Examples

Figure 2. JOB Statement Example.

```
//MYJOB , 'JOHN O USER' ,CLASS=A,REGION=3M,TIME=( , 30 ) ,LINES=60
```

The jobname is MYJOB. The job will be charged to the userid from which the job was submitted since no USER= was used. It requests a maximum of 30 seconds of CPU time and 60,000 lines of printed output. The user's name is JOHN O USER. The job will be run at normal priority and requests a maximum of 3MB of memory for each job step.

Figure 3. JOB Statement Example.

```
//TEST JOB , 'MORGAN' ,CLASS=P, TYPRUN=HOLD, TIME=( , 20 ) ,LINES=200
```

This job is named TEST. It requests up to 20 seconds of CPU time and 200,000 lines of output. It will be run in the PRODUCTION class. The JCL will be scanned for syntax errors but the job will not be executed until it is released.

The JOBPARM Statement

Note

The **JOBPARM** statement may be phased out. You should use the keywords on the **JOB** statement whenever possible.

The **JOBPARM** statement specifies job-related parameters for JES2 and can be used to

- override values for parameters that are coded in the parentheses of the **JOB** statement
- include information not specified on the **JOB** statement.

There can be multiple **JOBPARM** statements in a job. Parameters coded on the last **JOBPARM** statement override any previously specified or implied values for those parameters. If there are conflicting values for any particular parameter specified, the last value takes precedence, even over the **JOB** statement. The **JOBPARM** statement cannot be continued and cannot be coded

past column 71. However, you can use multiple **JOBPARM** statements with different parameters.

The **JOBPARM** statement should immediately follow your **JOB** statement.

Syntax

The **JOBPARM** statement uses a keyword-type format. The syntax of the **JOBPARM** statement is as follows:

```
/*JOBPARM parameter,...,parameter
```

The **/*** must begin in column one; there is no space between the ***** and the word **JOBPARM**; and there must be one blank between the word **JOBPARM** and the list of parameters. The parameters are separated by commas. The following parameters can be specified:

LINES=nnnnnn	is the estimated line count (for one copy of the job) in thousands, in which "nnnnnn" is less than or equal to 999999. Can be abbreviated as L=nnnnnn .
CARDS=nnnnnnnn	indicates the number of cards to be punched. Can be abbreviated as C=nnnnnnnn . Default is 100. The maximum that can be specified in the JOBPARM statement is 99999999.
FORMS=dddd	indicates the special forms number to be used for the output. All SYSOUT datasets for the job will be printed using this forms number unless otherwise indicated in the SYSOUT parameter of the DD statement. Can be abbreviated as F=dddd . See CNS document D0077, " <i>OS/390: Special Forms Output</i> " [http://docweb.cns.ufl.edu/docs/d0077/d0077.html] for more information about form codes.
COPIES=nn	indicates the total number of copies of the entire job to be printed. Can be abbreviated as N=nn . To get a different number of printed or punched output or additional copies of specific datasets, the COPIES= parameter would have to be specified in the DD statement.
LINECT=kkk	specifies the number of lines to be printed per page. Must be a number in the range 0-254. Can be abbreviated as K=kkk .
NOLOG	can be specified to suppress printing of the JES2 job log. Has no operands. Can be abbreviated as J .
TIME=tttt	is a numeric field specifying the maximum time, in CPU seconds, that the job can execute. If the job exceeds this limit, it will not run to completion. The time specification on the /*JOBPARM statement overrides any time specification

expressed in the form of positional parameters; e.g., it will override (*, , T, L*) in the positional parameters of the **JOB** statement. This parameter can be abbreviated as **T=ttttt**.

Note

This parameter does NOT override time expressed on the Job Card in the form of keyword parameters, e.g.,

TIME=(mm,ss) keyword.

Note

The **/*JOBPARM** parameter may be phased out in the future.

HEADER=cccccccc	causes the character string "cccccccc" to be printed on batch output (in block letters just under the jobname) in place of the JES2 job number. Can be 1-8 alphabetic or numeric characters plus the characters @, #, and \$. The first character of the name must be alphabetic. NOTE: Be aware of the effect this parameter could have on the filing of your output. All output is filed according to the last three digits of the JES2 job number. If you plan to use the HEADER option, contact the operator at that site to arrange for pickup of the output.
QUEUE=FETCH	causes the job output to be routed to the TSO FETCH queue. Can be abbreviated as Q=F . See the CNS Document D0037, <i>TSO Introduction</i> [http://docweb.cns.ufl.edu/docs/d0037/d0037.html], for more information on the FETCH queue.
INFORM	causes a message to be printed at your terminal when the job has finished execution. Can also be specified as I .

Examples

The following JOBPARM statement specifies special forms number 3001, and no JES2 job log:

Example 5.

```
/*JOBPARM FORMS=3001,NOLOG
```

The following JOBPARM statement specifies a maximum of 5000 lines of printed output and two copies of all output:

Example 6.

```
/*JOBPARM COPIES=2,LINES=5
```

The following statement specifies that output is to be sent to the TSO FETCH queue and to send an information message when the job finishes executing.

Example 7.

```
/*JOBPARM Q=F,I
```

The following statement will notify CNS Operations that you would like to have your output kept inside (not filed in the lobby).

Example 8.

```
/*JOBPARM HEADER=INSIDE
```

The ROUTE Statement

Output is normally returned to the site of submission. The ROUTE statement can be used to direct output to other locations. From NERSP the ROUTE statement is required.

The designation of a remote location can be obtained from the operations staff at that site.

Figure 4 shows the format of the ROUTE statement. Note that the statement begins with a /* and, therefore, is a JES2 statement.

Figure 4. The ROUTE Statement

```
/*ROUTE PRINT node.location
```

PRINT	is a keyword that refers to printed output, whether it be on paper or routed back to your session.
<i>node</i>	is NER for CNS's MVS system, NERSP for CNS's AIX/6000, or the name of one of the other nodes in the State University System (SUS) network.
<i>location</i>	is the remote site or userid at the selected node, depending on the type of operating system at that node.

For output routed to MVS nodes (NER), you can specify locations such as

R0 or **LOCAL** to indicate that the output should be sent to the CNS central site, which is Bryant Space Sciences Research Building (SSRB) on the UF campus.

Rn to indicate the remote site 'n', where 'n' is a 1- to 3-digit number specifying the particular site.

For output routed to NERSP, you can specify the location as

NERSP.userid The output arrives as an e-mail in the user's in-box.

ROUTE Statement Examples

Example 9. ROUTE Statement Example

This example requests that all printed output be routed to the CNS central site (the SSRB).

```
/*ROUTE PRINT NER.R0
```

Example 10. ROUTE Statement Example

This example requests that all printed output be sent as an e-mail message to the userid STRTREK on the NERSP.

```
/*ROUTE PRINT NERSP.STRTREK
```

Network Jobs

The **/*ROUTE** statement can also be used to request that a job execute or generate output at another regional data center in the State University System computer network.

The SETUP Statement

Any job that uses tapes must include a JES2 **/*SETUP** statement. It is placed after the **JOB** statement and before any **EXEC** statements.

JES2 uses the **SETUP** statement to coordinate the mounting of tapes. It avoids initiating jobs that require more tape drives than are currently available. Only one **SETUP** statement can be used for each type of tape drive (**TAPE9** and **TAPEC**) in a job. For **TAPE9** or **TAPEC**, you can request more than one unit.

If your job requests a tape mount on a unit you have not specified in a **SETUP** statement, or if the tape type does not match the unit type specified (i.e., the tape is a cartridge and the drive is a reel drive), the job will be canceled.

For more information on the use of tapes, see *Using Magnetic Tapes at CNS* (D0017) [<http://docweb.cns.ufl.edu/docs/d0017/d0017.html>].

Figure 7 shows the format of the **SETUP** statement.

Example 11. The **SETUP** Statement

```
/*SETUP unit,n
```

unit is the type of unit required, which can be 9-track or 3480 cartridge tape drive (**TAPE9** or **TAPEC**). CNS has several 9-track tape drives and several cartridge drives.

n is the maximum number of tape drives to be used simultaneously during the job.

Examples

The following **/*SETUP** statement allocates two 3480 cartridge tape drives:

```
/*SETUP TAPEC,2
```

The following statement requests one 9-track tape drive and one 3480 cartridge tape drive.

```
/*SETUP TAPE9,1 /*SETUP TAPEC,1
```

The **JCLLIB** Statement

The **JCLLIB** statement allows you to use a cataloged procedure in your own procedure library. This statement must precede the **EXEC** statement. If your job does not have a **JCLLIB** statement, the system procedure libraries will be searched for the procedure specified in your job's **EXEC** statement. The **JCLLIB** statement will indicate that your procedure library is to be searched first.

Figure 8 shows the format of the **JCLLIB** statement.

Example 12. The **JCLLIB** Statement

Replace "*proc.lib.name*" with the name of your procedure library.

```
// JCLLIB ORDER=proc.lib.name
```

JCLLIB Statement Example

Example 13. Example using the JCLLIB Statement

This example accesses a procedure called PHOTON that is in a procedure library called U.STRTREK.PROCLIB.

```
// JCLLIB ORDER=U.STRTREK.PROCLIB  
// EXEC PHOTON
```

The EXEC Statement

The **EXEC** statement tells the operating system which cataloged procedure or program you want to use. The **EXEC** statement has two forms, depending on whether you specify the name of a cataloged procedure or the name of a program.

Specifying a Cataloged Procedure

Code the **EXEC** statement as follows to use a cataloged procedure:

```
// EXEC procname
```

You would replace "procname" with the name of the cataloged procedure. For example, to use SAS (the Statistical Analysis System), the statement would be:

```
// EXEC SAS
```

Specifying a Program Name

Code the **EXEC** statement as follows to execute a specific program:

```
// EXEC PGM=program
```

You would replace "program" with the name of the program you want to execute. For example, to execute a program named **IEBGENER**, you would use:

```
// EXEC PGM=IEBGENER
```

Specifying Options on the EXEC Statement

Most programs and cataloged procedures allow you to pass options to them on the **EXEC** statement. When you specify a program, you pass options with the **PARM** parameter. However, if you are using a cataloged procedure, you will need to refer to the documentation for the particular procedure you are using to determine the keyword for specifying parameters.

The DD Statement

The **DD** statement can be very complex, depending on the format and location of your data. A complete description of the **DD** statement, therefore, is beyond the scope of this document.

DD statements for magnetic tape or disk data sets are covered in detail in other CNS documents. See especially the *CNS Using Magnetic Tapes at CNS* (D0017) [???] document and the *Disk Data Sets at CNS* (D0045) [<http://docweb.cns.ufl.edu/docs/d0045/d0045.html>] document. Also, documentation for specific programs and utilities contains the **DD** statements and other JCL needed to use those programs at CNS.

Another excellent source of information on JCL and **DD** statements is the latest version of the JCL text by Gary DeWard Brown, or *OS/390 MVS JCL Reference*.

A typical **DD** statement would resemble the following:

```
//ddname DD positional-parameters and keyword-parameters
```

The **DD** statement consists of the characters `//` in columns 1 and 2 and four fields: name, operation (**DD**), parameter, and comments.

A **DD** statement is usually required for each data set.

SYSDOUT: Output Stream Data Sets

The **SYSDOUT** parameter of the **DD** statement routes output to a printer. For printed output, code the **DD** statement as:

```
//ddname DD SYSDOUT=class
```

The CNS-defined output classes are:

A	Normal print
B	Punch
C	Special forms
D	Dummy (writes to spool, then discards the output)
H	HELD
K	Punch
L	Reserved for operating system
M	Reserved for mail
N	Reserved for mail notes
T	"throw-away" class (if there is a JCL error, the output prints; otherwise there is no output)

The remaining output classes are reserved for other special system output processing.

More JCL Notes

Examples of the most commonly used control statements are briefly described below. The words typed in upper case and all punctuation must be specified exactly as given. You need to supply the items in lowercase.

The **JOB** statement defines the beginning of the job and identifies the job to the system. This statement is required for all jobs.

```
//jobname JOB , 'programmer' , CLASS=class ,  
// TIME=(mm,ss) , LINES=lines , COPIES=copies
```

Use the **ROUTE** statement to route output to a site other than the one from which the job was submitted. For example, "*node.location*" may be replaced with **NER.R0** to route output to the local (SSRB) site.

```
/*ROUTE PRINT node.location
```

Use the **SETUP** statement to request the use of tape drives. This example requests one cartridge tape drive.

```
/*SETUP TAPEC,1
```

Use the **EXEC** statement to tell the system what program or cataloged procedure to use when executing the job. This example requests the Statistical Analysis System (SAS) cataloged procedure. You must have a space between "/" and "**EXEC**" for the job to run.

```
// EXEC SAS
```

Use the **DD** (data definition) statement to define to the system what data are to be used for the job and where they are located. The example shown is incomplete. The **DD** statement can be very complex, depending on the format and location of your data. You need to supply your own data definition parameters here.

```
//ddname DD . . .
```

The optional **/*** statement may be used to tell the system that the job's data input stream has ended. It is not required.

```
/*
```

Batch Input

Card-image batch jobs may be submitted from time-sharing systems such as MVS/TSO. Input of programs and data may be from the following:

- interactive workstations
- magnetic reel or cartridge tapes
- direct access storage devices (disk)

Batch Output

Output may be of the following types:

- files routed back to your interactive session
- printed pages
- magnetic tape files
- disk data sets
- plots
- indexed CD-ROM

For printed output directed to CNS local printers at the SSRB, you may request a variety of paper types (special forms), including gummed mailing labels, IBM 3130 page printer (letter quality), HP 5000 PS large-format color printer, upper- or lower-case print, and six or eight lines-per-inch. Printed output may be picked up at the location to which it was routed. Special forms output may be picked up in the SSRB lobby. indexed CD-ROM and output for batch jobs submitted with an INSIDE header may be picked up from the Operations shift supervisor or the tape librarian.

Tape and Disk Storage

The two primary forms of data storage available at CNS are magnetic disk and magnetic cartridge and reel tape. We do not charge for tape storage, but you must purchase your own tapes and must pay a setup charge each time a tape is used. CNS recommends that you use cartridge tapes. Tapes must be logged into CNS's Tape Management System (TMS) before they can be used. Detailed information about using tapes at CNS is in *Using Magnetic Tapes at CNS* (D0017) [<http://docweb.cns.ufl.edu/docs/d0017/d0017.html>]. You may call the tape librarian at (352) 392-2291.

The charge for storing data files or programs on CNS disk volumes is based on the amount of space you allocate. The rates for disk storage are listed in the *CNS Charging Algorithm* [<http://docweb.cns.ufl.edu/docs/d0001/d0001.html>]. Detailed information on using disk data sets at CNS is in *OS/390 Disk Data Sets at CNS*. You are responsible for creating, correctly naming, cataloging, managing, backing up, uncataloging, and deleting your own data sets.

Your Comments are Welcome

We welcome your comments and suggestions on this and all CNS documentation. Please send your comments to:

UF Computing & Networking Services

112 Bryant Space Sciences Bldg, University of Florida

P.O. Box 112050

Gainesville Florida 32611-2050

(352) 392.2061

<editor@cns.ufl.edu>