



JCL

Chapter b6

Symbolic parameters

Job Control Language

Chapter a1. Introduction to JCL

Chapter a2. Coding JOB statements

Chapter a3. Coding EXEC statements

Chapter a4. Coding DD statements

Chapter a5. Analyzing job output

Chapter a6. Conditional processing

Job Control Language

Chapter b1. Using special DD statements

Chapter b2. Introducing procedures

Chapter b3. Modifying EXEC parameters

Chapter b4. Modifying DD parameters

Chapter b5. Determining the effective JCL

Chapter b6. Symbolic parameters

Job Control Language

Chapter c1. Nested procedures

Chapter c2. Cataloging procedures

Chapter c3. Using utility programs

Chapter c4. Sample utility application

Symbolic parameters.

Chapter b6

Symbolic parameters

Symbolic parameters.

Unit introduction.

In earlier units you have seen that procedures can be modified when they are executed using override and addition statements. However, these modifications are subject to strict sequencing rules, and they are not always convenient to implement.

This unit describes two techniques that can provide an alternative, and often easier, means of modifying procedures for execution:

- Using the DDNAME parameter on procedure DD statements.**
- Using symbolic parameters on procedure EXEC and DD statements.**

If these parameters are included in the procedure definition when it is coded, values can be easily assigned at the time of executing a procedure.

Symbolic parameters.

Course objectives.

Be able to:

- **Understand the purpose and form of the DDNAME parameter in a procedure definition.**
- **Resolve the DDNAME parameter when executing a procedure by identifying the actual data set names and attributes represented by the DDNAME parameter.**
- **Understand the purpose and form of symbolic parameters in a procedure definition.**
- **Assign values to symbolic parameters at the time of invoking a procedure.**

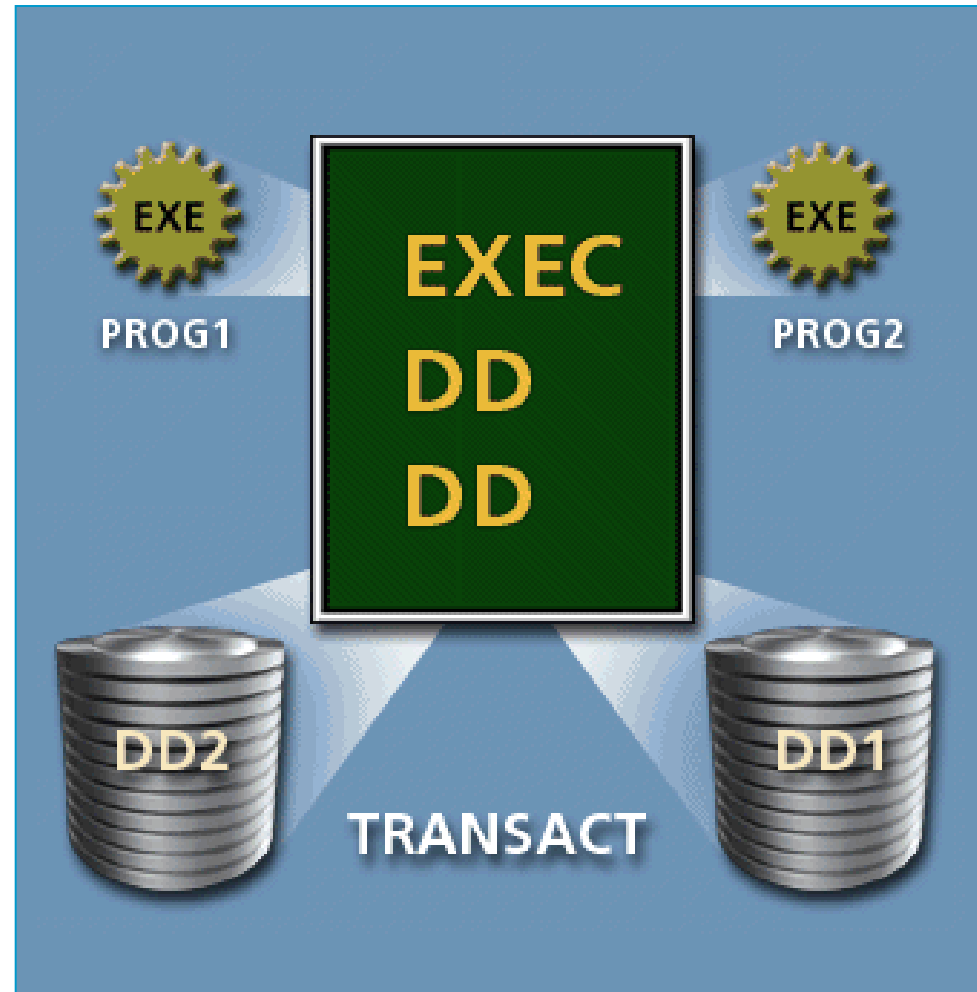
DDNAME and symbolic parameters.

Parameter variation in procedure.

Variations in procedure statements can be quite common between different instances of a procedure's use.

For example, in a procedure like TRANSACT, one or more items vary from one instance of the procedure's use to the next, such as:

- Name, location, or other attributes of one or more data sets identified in the procedure.
- Names of one or more programs to be executed.



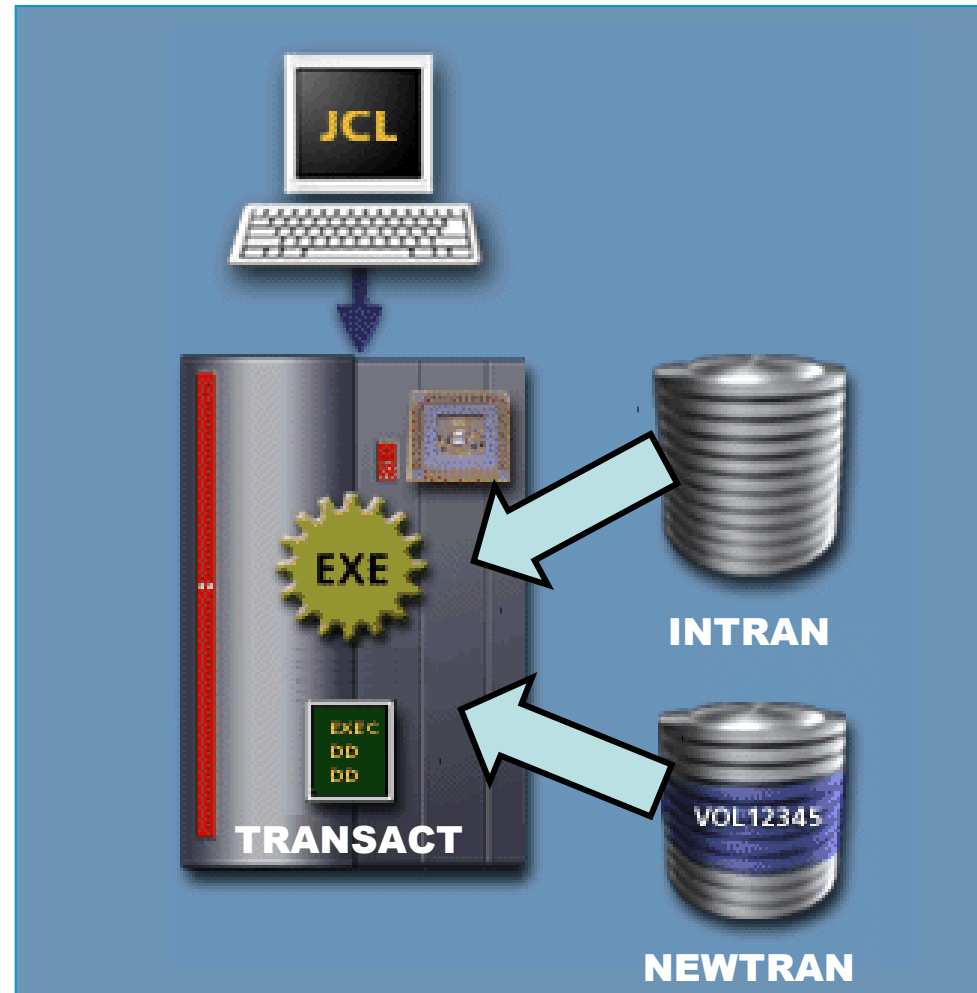
DDNAME and symbolic parameters.

Parameter variation in procedure.

These variations can be accommodated by temporarily modifying the procedure at the time the procedure is executed.

This is accomplished by adding or overriding parameters on the procedure EXEC and DD statements.

These modifications are subject to strict formatting and sequencing rules.



DDNAME and symbolic parameters.

Techniques to change parameters.

If variations in procedure are common, override and addition statements can be avoided.

Alternatives to using override and addition statements are:

- Use the DDNAME parameter on one or more DD statements in the procedure definition.
- Use symbolic parameters to represent EXEC or DD statement parameters in the procedure definition.



DDNAME and symbolic parameters.

TRANSACT procedure.

The TRANSACT procedure, on the right, meets two of the guidelines:

- It performs the exact functions required by the sample application.
- It does not require the procedure user to specify any execution-time alterations for typical procedure use.

But does it simplify as much as possible the JCL that the user has to code at procedure execution?

```
//PSTEP1 EXEC PGM=PROG1
//DD1 DD DSN=INTRAN,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```

DDNAME and symbolic parameters.

Are we on track?

A _____ parameter allows to postpone defining data set and its attributes.

DDNAME and symbolic parameters.

DDNAME parameters.

If the attributes of the data sets identified in TRANSACT vary from one instance of the procedure's use to the next, the user will need to code DD override and addition statements at the time of executing the procedure.

However, generalizing the TRANSACT procedure to process data input from a variety of sources, you will avoid the need for overrides and additions. One way of accomplishing this is the DDNAME parameter.



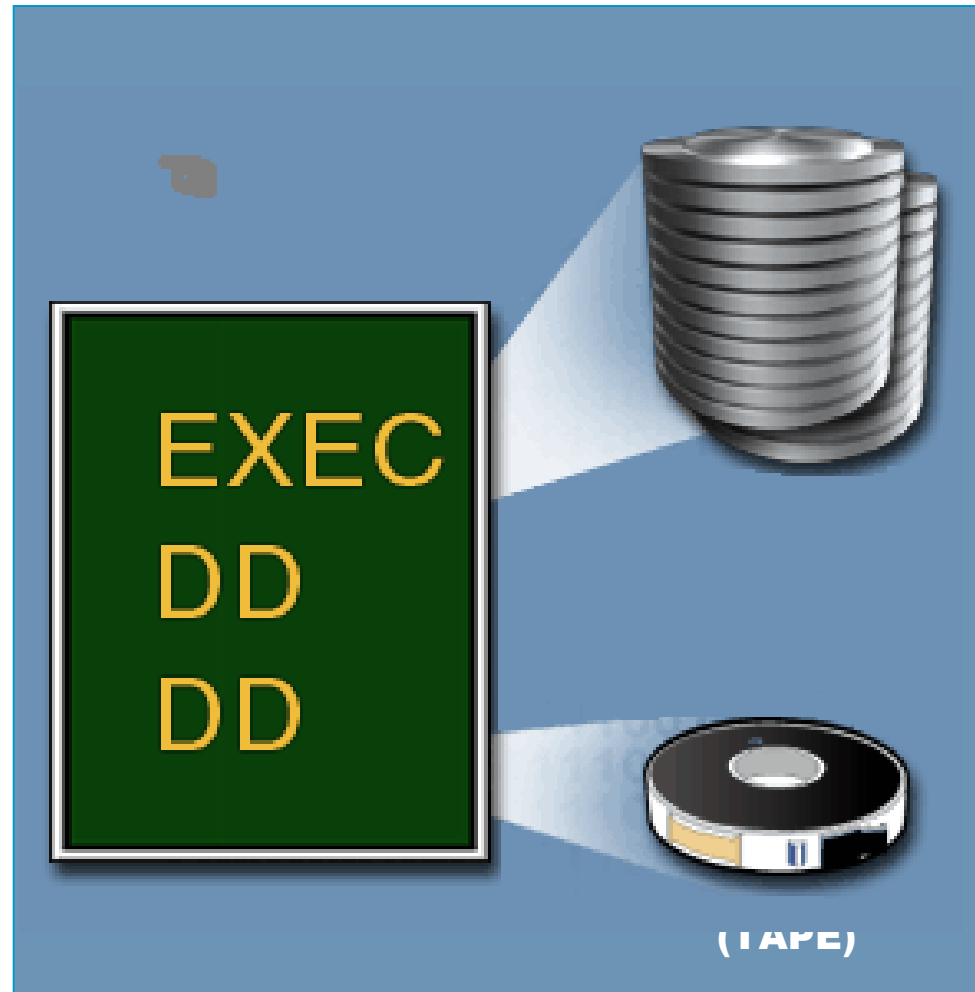
The DDNAME parameter.

Input data streams.

The DDNAME parameter is used in a procedure definition to represent an input data set.

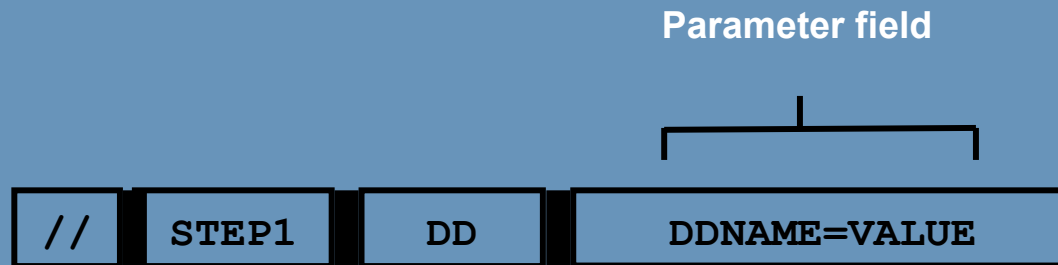
It permits the procedure user to define data set attributes at the time the procedure is invoked.

The example on the right shows that at various times the input data for a procedure could reside on a direct access volume or a tape.



The DDNAME parameter.

Coding a DDNAME.



You should code a DDNAME on the appropriate procedure step, generally to represent an input data set. This postpones definition of the data set attributes until the procedure is invoked.

The DDNAME parameter consists only of the operand name {DDNAME} followed by an equal sign and a Value (VALUE, in this example).

The DDNAME parameter.

Coding a DDNAME - an example.

The example on the right illustrates the TRANSACT procedure definition with a DDNAME parameter for the input data set.

Note that the value of the DDNAME parameter is not the same as the ddname of the statement (DD1 in this example).

```
//PSTEP1 EXEC PGM=PROG1
//DD1      DD      DDNAME=INPUT
//DD2      DD
DSN=MASTER, DISP=SHR
//DD3      DD      SYSOUT=A
//DD4      DD      DSN=&&VALID,
//          DISP= (NEW, PASS) ,
//          UNIT=SYSDA,
//          SPACE= (TRK, (1, 1) )
//PSTEP2 EXEC PGM=PROG2
//DD5      DD      DSN=&&VALID,
//          DISP= (OLD, DELETE)
//DD6      DD      SYSOUT=A
```



The DDNAME parameter.

Resolving the DDNAME parameter.

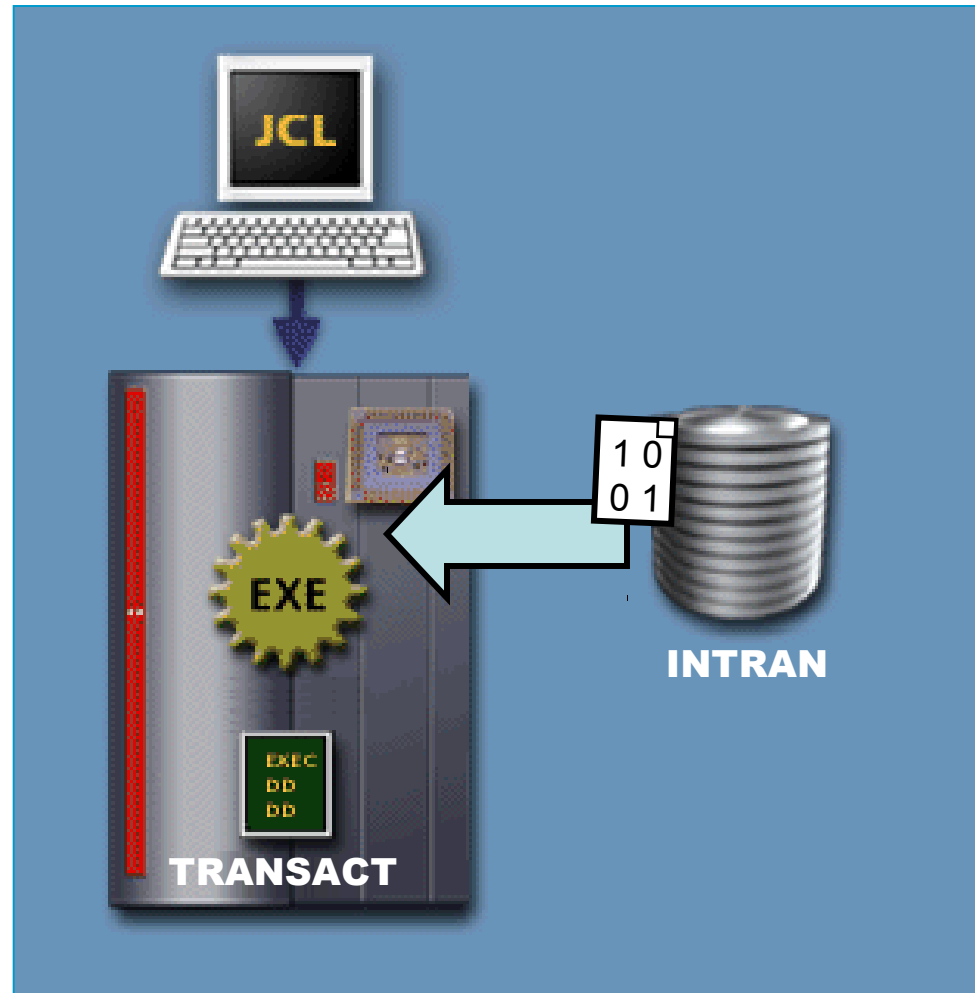
At the time of execution to invoke the procedure, you would code:

- An EXEC statement that executes the procedure.
- A DD statement to identify the attributes of the data set represented by DDNAME=INPUT.

This is called resolving the DDNAME parameter.

For example, the DDNAME is resolved with a data set INTRAN as shown in the code:

```
//STEP          EXEC TRANSACT  
//PSTEP1.INPUT DD DSN=INTRAN, ..
```

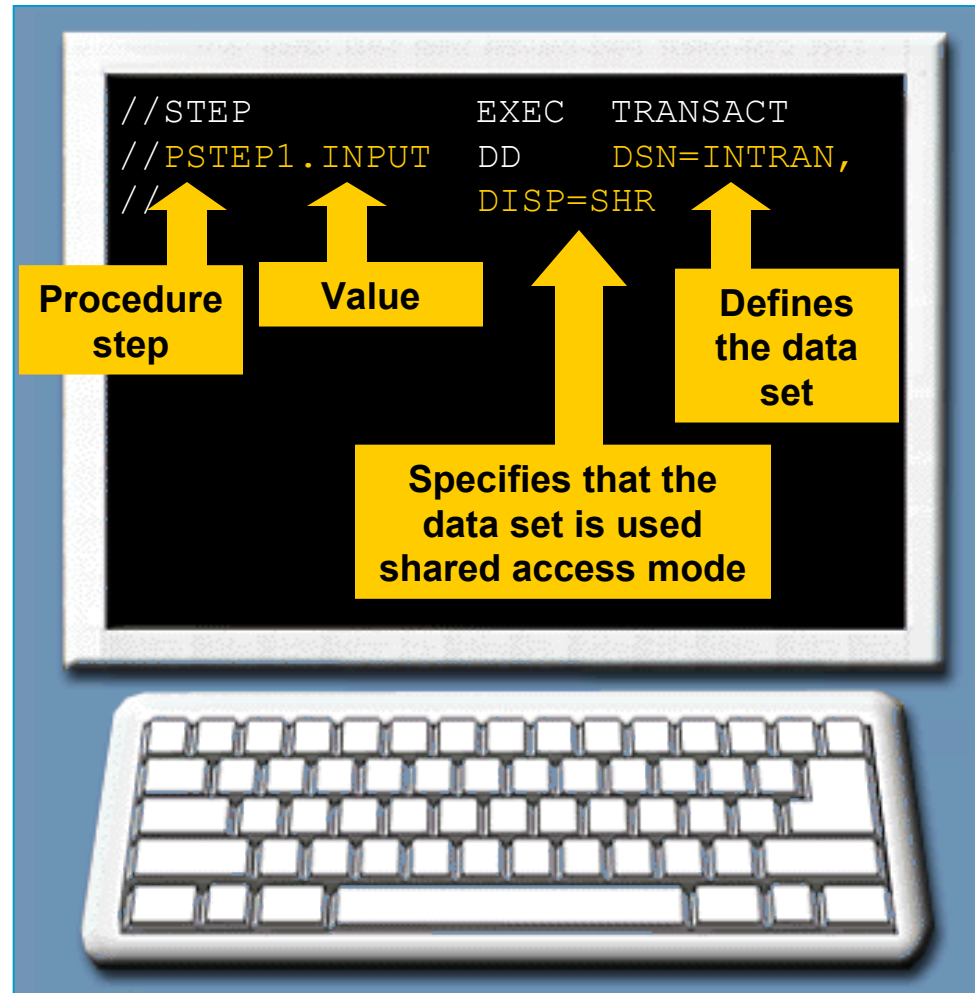


The DDNAME parameter.

Resolving the DDNAME parameter.

The DD statement that resolves the DDNAME parameter has two components:

- The name of the procedure step in which the DD statement containing the DDNAME parameter is used, followed by a period.
- The value as specified in the DDNAME parameter.



The DDNAME parameter.

Are we on track?

Which of the following is an advantage of using the DDNAME parameter in a procedure?

- A. You can postpone specifying data set attributes until procedures execution.**
- B. You enable the user to execute the procedure using different programs.**
- C. You can relieve the procedure user of the need to specify data set attributes.**

The DDNAME parameter.

Are we on track?

The following is a procedure called MYPROC. The procedure uses the DDNAME parameter to represent the data set in PSTEP1.

```
//PSTEP1 EXEC PGM=PGM1  
//DD1 DD DDNAME=INPUT
```

Code a statement to invoke the procedure.

//JSTEP EXEC _____

The DDNAME parameter.

Are we on track?

Code a DDNAME parameter to define the temporary data set for PROG1 that refers to the DD name DD4. Define the actual data set characteristics using the name TSTDATA.

TRANSACT Procedure:

```
//PSTEP1      EXEC  PGM=PROG1
//DD1         DD    DSN=...
//DD2         DD    DSN=MASTER, DISP=SHR
//DD3         DD    SYSOUT=A
//DD4         DD    _____
```

The DDNAME parameter.

Are we on track?

The following is a procedure called MYPROC. The procedure uses the DDNAME parameter to represent the data set in PSTEP1.

```
//PSTEP1 EXEC PGM=PGM1  
//DD1      DD  DDNAME=INPUT
```

Code a statement to resolve the DDNAME parameter with a data set named MYDATA.

```
//JSTEP EXEC MYPROC
```

The DDNAME parameter.

Are we on track?

Complete the statement required at procedure invocation time to resolve DDNAME as an existing data set named MYDATA.

JCL to invoke TRANSACT:

```
//SS1          EXEC TRANSACT  
//PSTEP1.TSTDATA DD DSN=MYDATA,DISP_____
```

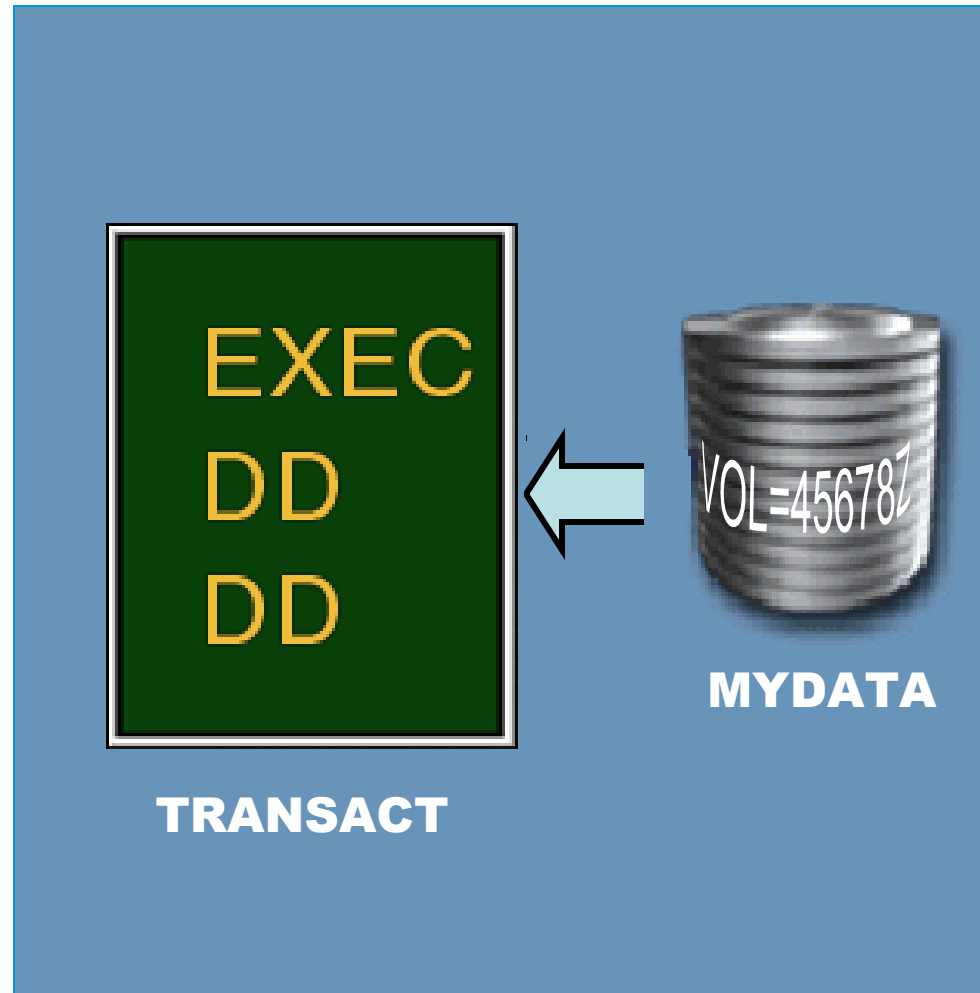
The DDNAME parameter.

Resolving the DDNAME parameter – example 1.

To execute the TRANSACT procedure, the user executes the procedure using a data set MYDATA in shared access mode.

MYDATA resides on a 3390 direct access volume with a volume identifier of 45678Z. In this case, the DDNAME parameter is resolved with the data set name of MYDATA.

```
//STEP          EXEC TRANSACT
//PSTEP1.INPUT DD DSN=MYDATA,
//              UNIT=3390,
//              VOL=SER=45678Z
```



The DDNAME parameter.

Resolving the DDNAME parameter – example 2.

The DDNAME parameter in the same procedure is resolved for using different input data.

The user invokes the TRANSACT procedure and directs that input data be taken from card images supplied in the job stream.



The DDNAME parameter.

Avoiding an input data set.

If an input data set is not needed for the execution of the procedure then the DD statement to identify the data set should not be coded.

The unresolved DDNAME is automatically treated as a dummy.

The DDNAME parameter.

Rules for resolving DDNAME parameters.

The DDNAME parameter can be used in a procedure definition in as many as five DD statements within each procedure step.

The sequencing rules to be followed when resolving DDNAME parameters are:

- **They must be coded in procedure step sequence.**
- **DD statements to resolve DDNAME parameters are considered additions and should be sequenced as such (that is, in procedure step sequence).**
- **If multiple DDNAME parameters are specified in a single step of the procedure definition, the DD statements to resolve them can be coded in any sequence.**

The DDNAME parameter.

Are we on track?

The following is a procedure called ANYPROC:

```
//PSTEP1 EXEC PGM=CBL5
//IN      DD      DDNAME=WEEKNO
//OUT     DD      DSN=&&TEMP, DISP= (NEW, PASS) ,
//                UNIT=SYSDA, SPACE= (TRK, (1, 1))
//PSTEP2 EXEC PGM=CBL6
//PSD     DD      DSN=&&TEMP, DISP= (OLD, DELETE)
//RPT     DD      DDNAME=OUTPUT
```

Code the statements required to resolve the procedure's WEEKNO DDNAME parameters to specify WEEKNO as an existing data set named WEEK2.

```
//JSTEP EXEC ANYPROC
//_____
```

The DDNAME parameter.

Are we on track?

The following is a procedure called ANYPROC:

```
//PSTEP1 EXEC PGM=CBL5
//IN      DD      DDNAME=WEEKNO
//OUT     DD      DSN=&&TEMP, DISP= (NEW, PASS) ,
//                UNIT=SYSDA, SPACE= (TRK, (1, 1))
//PSTEP2 EXEC PGM=CBL6
//PSD     DD      DSN=&&TEMP, DISP= (OLD, DELETE)
//RPT     DD      DDNAME=OUTPUT
```

Code the statements required to resolve the procedure's OUTPUT DDNAME parameters to specify OUTPUT to be a data set named SAVE.

```
//JSTEP EXEC ANYPROC
//PSTEP1.WEEKNO DD DSN=WEEK2
//_____
```

The DDNAME parameter.

Glossary.

DUMMY

A parameter coded in the DD statement that instructs the system to ignore all other parameters, with the exception of DCB.

DDNAME Parameter

A parameter coded in a procedure definition that represents an actual data set and its attributes.

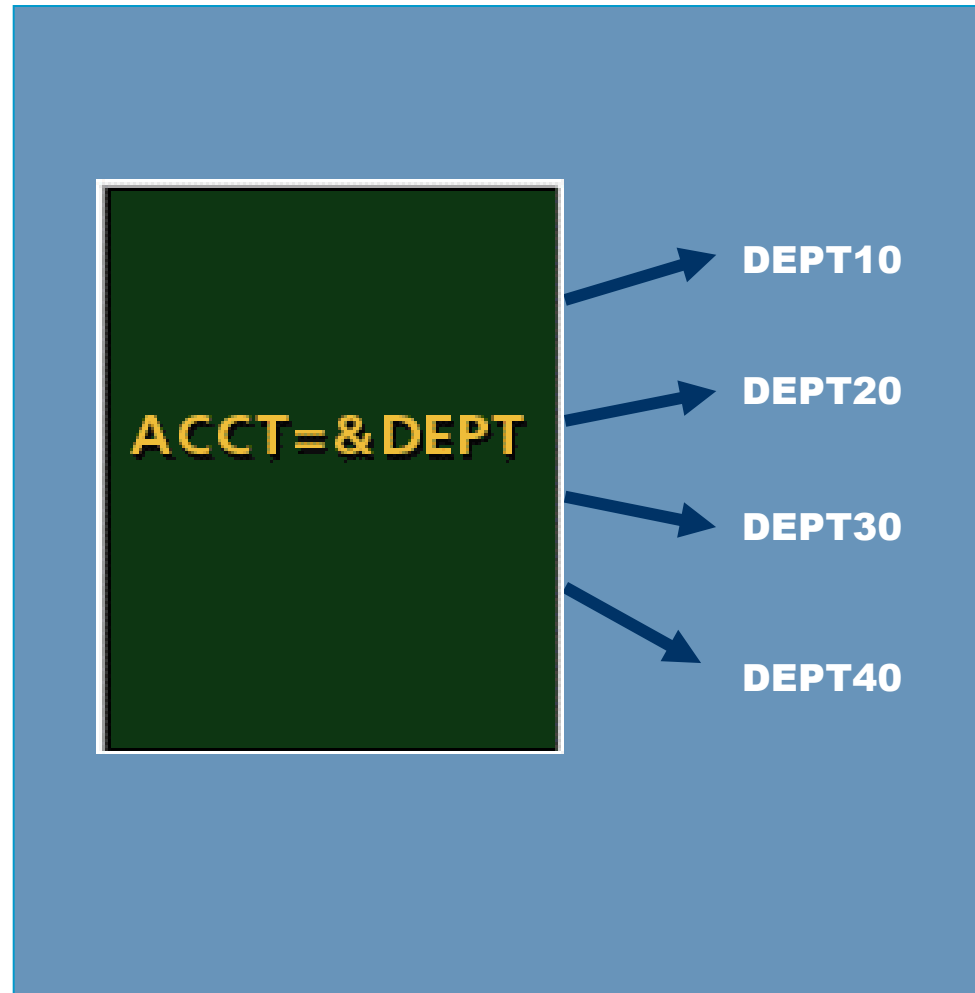
Symbolic parameters.

Symbolic parameters.

What are symbolic parameters?

Symbolic parameters are a technique to build flexibility into procedures. A symbolic parameter represents a procedure statement parameter or subparameter.

The procedure statement operands whose values are most likely to change from one procedure invocation to another, are suitable candidates to be coded as symbolic parameters. The example on the right illustrates such a procedure statement operand, ACCT.



Symbolic parameters.

Significance of symbolic parameters.

How do symbolic parameters help?

Coding symbolic parameters simplifies the procedure user's task in a number of ways:

- **The user can assign the appropriate value when the procedure is invoked.**
- **The user can simply accept a default value if one is assigned.**
- **The user can assign a value to the symbolic parameter only once to change all recurring like operands in statements occurring through the entire procedure. This is useful if the same value is to be used for like operands in statements occurring throughout the procedure.**

Symbolic parameters.

Standards for symbolic parameter names.

Rules for coding symbolic parameter names are as follows:

- Are preceded by an ampersand (&).
- Consist of up to seven alphanumeric (A-Z, 0-9) or national (#,@,\$) characters, beginning with an alphabetic or national character.
- Can represent EXEC or DD statement parameters.
- It cannot conflict with an EXEC statement keyword. That is, keywords such as PGM or COND cannot be used.
- It cannot conflict with the name of another symbolic parameter in the same procedure. If more than one value is assigned to a symbolic parameter, only the first can be used.

Examples of Symbolic Parameters

```
ACCT=&DEPT
```

```
PGM=&PROG1
```

```
//STEP1 EXEC PGM=PROG1,ACCT=&DEPT
```

```
//DD1 DD DSN=&WEEK
```

Symbolic parameters.

Are we on track?

Which of the following statements apply to the use of symbolic parameters?

- A. They can be specified for a DD or EXEC statement parameter.**
- B. They represent a procedure statement parameter or subparameter.**
- C. They are identified in a procedure by a preceding # sign.**
- D. They are coded in a procedure definition.**

Symbolic parameters.

Are we on track?

Which of the following items contains the symbolic parameter:

A. ACCT=&GRP

B. SYSOUT=A

C. DSN=&&SYN

D. VOL=SER=&SERNO

Symbolic parameters.

Are we on track?

Which of the following are advantages of symbolic parameters?

- A. They enable the user to simply accept default values, if appropriate.**
- B. They can be given a name that is meaningful, such as &DEPT.**
- C. They enable the user to assign values when the procedure is executed.**
- D. They can be assigned different values within a job.**
- E. They can represent a data set and all its attributes.**

Symbolic parameters.

Are we on track?

Which are the symbolic parameters in the following JCL from a procedure named ANYPROC?

```
//S2          EXEC      PGM=&A, ACCT=&GRP  
//DD5        DD        DSN=&&SYM, DISP={OLD, DELETE}  
//DD6        DD        SYSOUT=A
```

A. PGM=&A,ACCT=&GRP,DSN=&&SYM

B. DSN=&&SYM,ACCT=&GRP

C. PGM=&A,ACCT=&GRP

EXEC statement parameters.

Recognizing symbolic parameter – an example.

An example of a symbolic parameter to represent an EXEC statement parameter is shown on the right.

Here the TRANSACT procedure is defined with a symbolic ACCT parameter (&DEPT) for PROG1 (in PSTEP1) and PROG2 (in PSTEP2).

ACCT=&DEPT

```
//PSTEP1 EXEC PGM=PROG1,ACCT=&DEPT
//DD1 DD DSN=INTRAN,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2,ACCT=&DEPT
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```



EXEC statement parameters.

Assign or nullify values to EXEC statement parameters.

The procedure user can assign or nullify a value to the symbolic parameter on the EXEC statement.

How to assign a value to a symbolic parameter?

The user should specify the symbolic parameter without the preceding ampersand, followed by an equal sign and the value.

How to nullify a symbolic parameter?

The user should specify the symbolic parameter without the preceding ampersand, followed by only an equal sign.

To assign a value to a symbolic parameter:

```
//JSTEP EXEC TRANSACT,DEPT=GRP50
```

To nullify the symbolic parameter:

```
//JSTEP EXEC TRANSACT,DEPT=
```



EXEC statement parameters.

Rules for assigning values to nullifying EXEC statement parameters.

The rules for assigning values to or nullifying EXEC statement parameters are:

- **Separate the value assignments or nullifications from operands and from each other using commas.**
- **Specify the symbolic parameter without the preceding ampersand, followed by an equal sign and the value.**
- **To nullify a symbolic parameter specify the symbolic parameter without the preceding ampersand, followed by an equal sign. No value is specified for the symbolic parameter while nullifying it.**
- **Symbolic parameters can be nullified or assigned values in any sequence.**

EXEC statement parameters.

Are we on track?

The effect of not coding a value for a symbolic parameter is to _____ the symbolic parameter.

EXEC statement parameters.

Are we on track?

The following is the TRANSACT procedure definition:

```
//PSTEP1 EXEC PGM=PROG1,ACCT=&DEPT
//DD1      DD   DSN=INTRAN,DISP=SHR
//DD2      DD   DSN=MASTER,DISP=SHR
//DD3      DD   SYSOUT=A
//DD4      DD   DSN=&&VALID,DISP=(NEW,PASS),
//          DD   UNIT=SYSDA,SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2,ACCT=&DEPT
//DD5      DD   DSN=&&VALID,DISP=(OLD,DELETE)
//DD6      DD   SYSOUT=A
```

Code an EXEC statements to execute the procedure and assign a value of GRP25 to the &DEPT symbolic parameter.

//JSTEP EXEC _____

Assigning different values.

Conditions.

Different values cannot be assigned to the same symbolic parameter when invoking the procedure.

For example, it is not possible to assign GRP50 to &DEPT in PSTEP1 and GRP100 to &DEPT in PSTEP2 unless one of the following conditions exists:

- A different symbolic parameter represents each occurrence of the parameter.
- An EXEC or DD statement override or addition statement is used when invoking the procedure.

```
//PSTEP1 EXEC PGM=PROG1,ACCT=&DEPT
//DD1      DD   DSN=INTRAN,DISP=SHR
//DD2      DD   DSN=MASTER,DISP=SHR
//DD3      DD   SYSOUT=A
//DD4      DD   DSN=&&VALID,
//          DD   DISP=(NEW,PASS),
//          DD   UNIT=SYSDA,
//          DD   SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2,ACCT=&DEPT
//DD5      DD   DSN=&&VALID,
//          DD   DISP=(OLD,DELETE)
//DD6      DD   SYSOUT=A
```



Assigning different values.

Assigning different values - example 1.

To illustrate the first condition, assume the TRANSACT procedure has been created to allow for two different values for the EXEC statement ACCT parameters, as follows:

```
//PSTEP1 EXEC PGM=PROG1 ,ACCT=&DEPT1  
...  
//PSTEP2 EXEC PGM=PROG2 ,ACCT=&DEPT2
```

In the example on the right, the TRANSACT procedure has been invoked with the values GRP50 and GRP100 for PSTEP1 and PSTEP2 respectively.



Assigning different values.

Assigning different values – example 2.

Here the procedure has been created without symbolic parameters:

```
//PSTEP1 EXEC PGM=PROG1  
...  
//PSTEP2 EXEC PGM=PROG2
```

At the time of invoking the procedure, an addition statement as shown on the right can be coded to assign an ACCT value of GRP50 for PSTEP1 and GRP100 for PSTEP2.



```
//JSTEP EXEC TRANSACT,  
// ACCT.PSTEP1=GRP50,  
// ACCT.PSTEP2=GRP100
```

Assigning different values.

Are we on track?

As a procedure user, when can you assign different values to the same symbolic parameter?

A. When a symbolic parameter represents the parameter.

B. When a different symbolic parameter represents each occurrence of the parameter.

C. When you code an addition or an override statement when invoking a procedure.

D. Never.

Assigning different values.

Assigning values to the PGM parameter.

The PGM parameter is the only EXEC statement parameter that cannot be modified with an override statement when a procedure is executed.

If a symbolic parameter is specified in the procedure definition, an appropriate value can be assigned when invoking the procedure.

Assigning different values.

The PGM parameter – an example.

In this example, different versions of the programs executed by the TRANSACT procedure are being tested to see which versions best suit departmental requirements.

Symbolic parameters are assigned to the PGM parameter in the TRANSACT procedure EXEC statement, so the procedure user can specify which program versions are to be executed.

Assigning different values.

Symbolic PGM parameters.

The TRANSACT procedure definition with symbolic PGM parameters for the PSTEP1 and PSTEP2 EXEC statements is shown on the right.

```
//PSTEP1 EXEC PGM=&PROG1
//DD1      DD      DSN=INTRAN,DISP=SHR
//DD2      DD      DSN=MASTER,DISP=SHR
//DD3      DD      SYSOUT=A
//DD4      DD      DSN=&&VALID,
//          DD      DISP=(NEW,PASS),
//          DD      UNIT=SYSDA,
//          DD      SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=&PROG2
//DD5      DD      DSN=&&VALID,
//          DD      DISP=(OLD,DELETE)
//DD6      DD      SYSOUT=A
```



Assigning different values.

Symbolic PGM parameters.

After specifying symbolic parameter for PGM in the procedure definition, you can assign the appropriate value (or values) to the symbolic parameter when invoking the procedure, as shown in the Statement to invoke TRANSACT on the right.

```
//JSTEP EXEC TRANSACT,  
//          PROG1=TEST1,  
//          PROG2=TEST2
```



Assigning different values.

Are we on track?

The following is the TRANSACT procedure definition:

```
//PSTEP1 EXEC PGM=&PROG1
//DD1      DD      DSN=INTRAN,DISP=SHR
//DD2      DD      DSN=MASTER,DISP=SHR
//DD3      DD      SYSOUT=A
//DD4      DD      DSN=&&VALID,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=&PROG2
//DD5      DD      DSN=&&VALID,DISP=(OLD,DELETE)
//DD6      DD      SYSOUT=A
```

Code an EXEC statement to execute

a.) TEST4 for PSTEP1 and

b.) TEST5 for PSTEP2.

//JSTEP EXEC _____

Assigning different values.

Are we on track?

Complete the S1 EXEC statement for a procedure named ANYPROC with the following symbolic parameters.

Code a symbolic parameter for PGM and name it &PROG:

```
//S1          EXEC  PGM=_____
```

Code a symbolic parameter for ACCT and name it &DEPT:

```
//          ACCT=_____
//DD1      DD    DSN=&INPUT,DISP=SHR
```

Assigning different values.

Are we on track?

Complete the JCL statement required to invoke procedure ANYPROC from the previous exercise, and specify the following:

- 1. Nullify the &DEPT symbol.**
- 2. Assign a value of MYPROG to &PROG.**

//JSTEP EXEC ANYPROC,_____

Assigning different values.

Assigning values.

When executing a procedure with symbolic parameters, you cannot assign different values to the same symbolic parameter within one job.

However, if you anticipate the need to assign different values, you can define a different symbolic parameter to represent each occurrence of the operand.

For example, when different departments want to execute TRANSACT in the same job you can define a different symbolic parameter for each occurrence of the ACCT operand.

TRANSACT definition:

```
//PSTEP1 EXEC PGM=PROG1,  
//          ACCT=&DEPT1  
//PSTEP2 EXEC PGM=PROG2,  
//          ACCT=&DEPT2
```

Assigning values to &DEPT:

```
//JSTEP EXEC TRANSACT,  
//          DEPT1=GRP50,  
//          DEPT2=GRP100
```



Using symbolic parameters.

Assigning values using SET statement.

How to code a SET statement?

Rather than assigning values on the EXEC statement used to invoke the procedure, you code a SET statement as follows:

```
//      SET parameter=value,  
//      parameter=value
```

The example on the right shows how to assign values to &DEPT using a SET statement.

TRANSACT definition:

```
//PSTEP1 EXEC PGM=PROG1,  
//          ACCT=&DEPT1  
//PSTEP2 EXEC PGM=PROG2,  
//          ACCT=&DEPT2
```

Assigning values to &DEPT1:

```
//          SET   DEPT1=CRP100  
//JSTEP   EXEC  TRANSACT
```



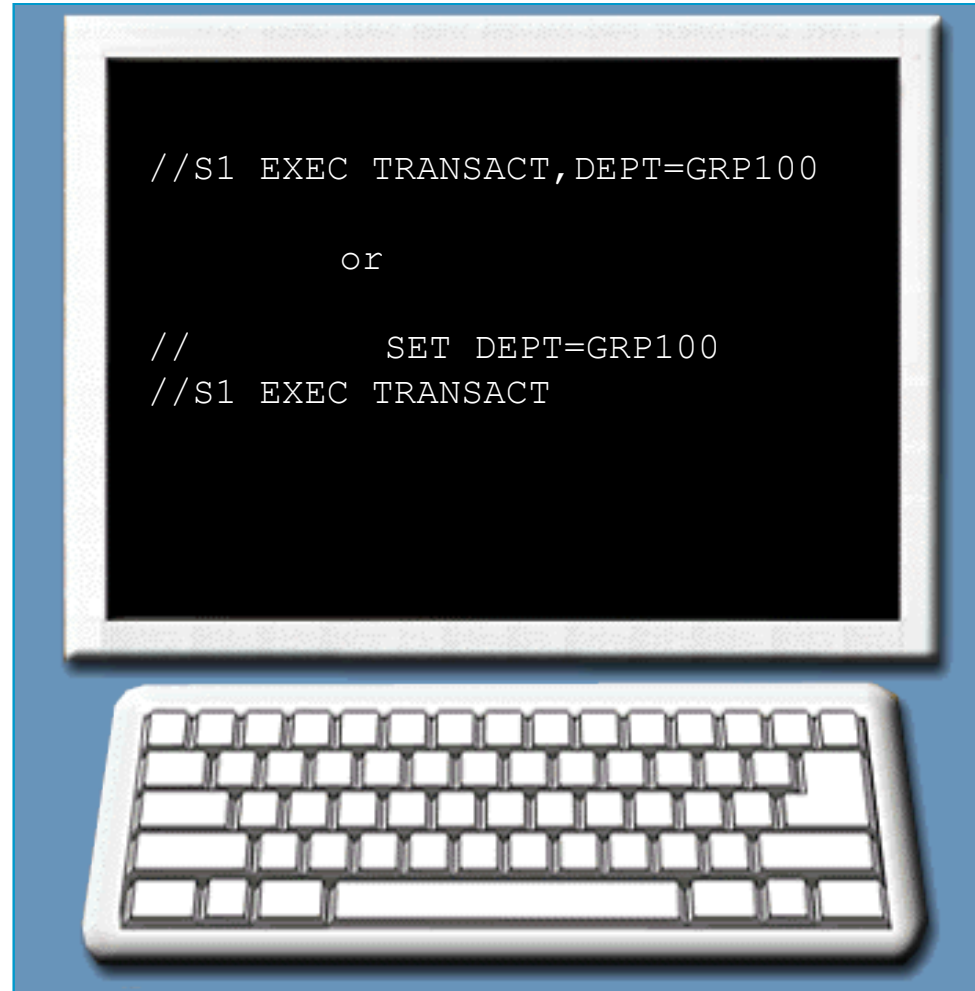
Using symbolic parameters.

Assigning values using SET statement.

SET statements can be placed anywhere in a job stream following the JOB statement. However, where they are placed will affect where they take effect.

The system assigns values to symbolic parameters as it encounters them. Therefore, a new SET statement will change the value of a previous SET statement. Similarly, values that are assigned on EXEC or PROC statements will change the values assigned on previous SET statements.

In the examples on the right, GRP100 would be the value set by the system, since it is the last value the system encounters.



Using symbolic parameters.

Are we on track?

When you use a SET statement to assign a value to a symbolic parameter, when does the new value apply?

- A. Only if the symbolic parameter did not have a previous value.**
- B. For the next use of the symbolic parameter only.**
- C. To all subsequent uses of the symbolic parameter, unless changed again.**

DD statement parameters.

Symbolic parameters for DD statement operands.

Like DDNAME, symbolic parameters can also be specified for DD statement operands, if the values for those operands are likely to change.

When do you use DDNAME and when do you specify symbolic parameters?

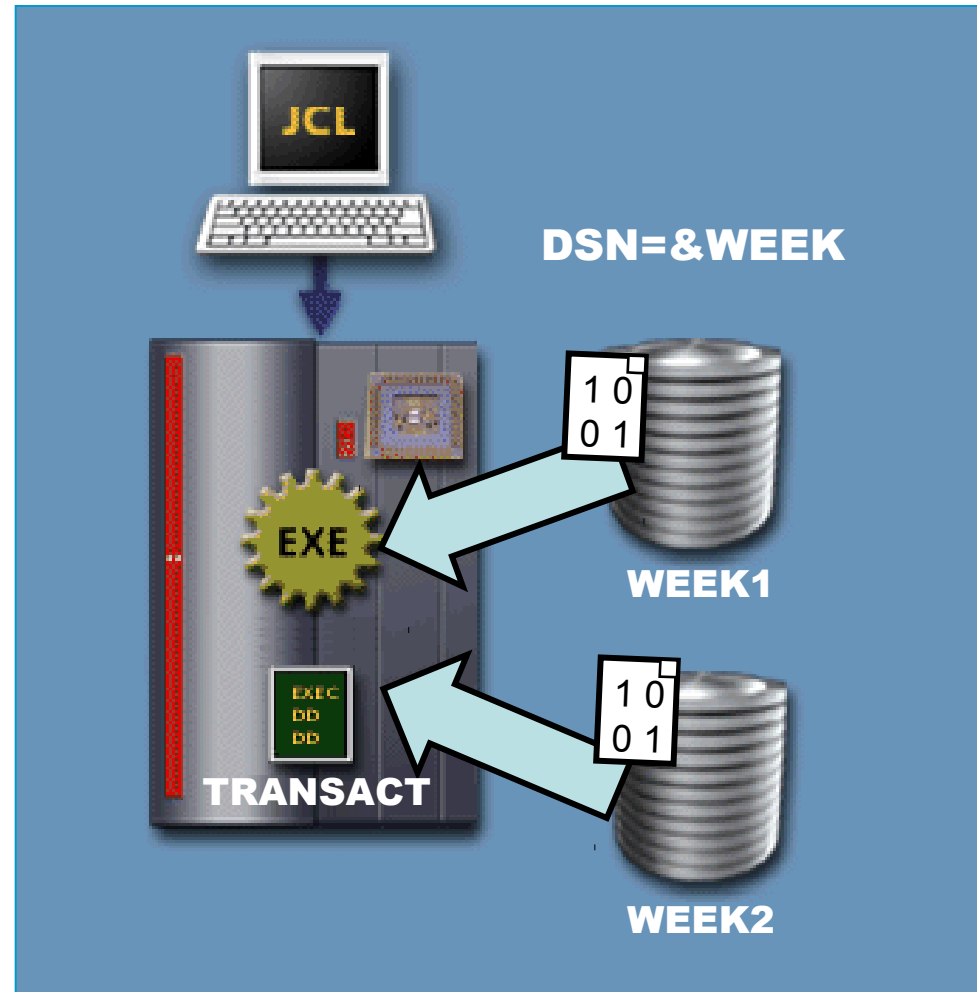
Specify symbolic parameters in a procedure if only a limited number of DD statement operands are likely to vary.

If, however, completely different data sets with different attributes are likely to be used, consider using the DDNAME operand instead. In that way, the users will not have to assign multiple values to multiple symbolic parameters.

DD statement parameters.

Changing DD statement parameters.

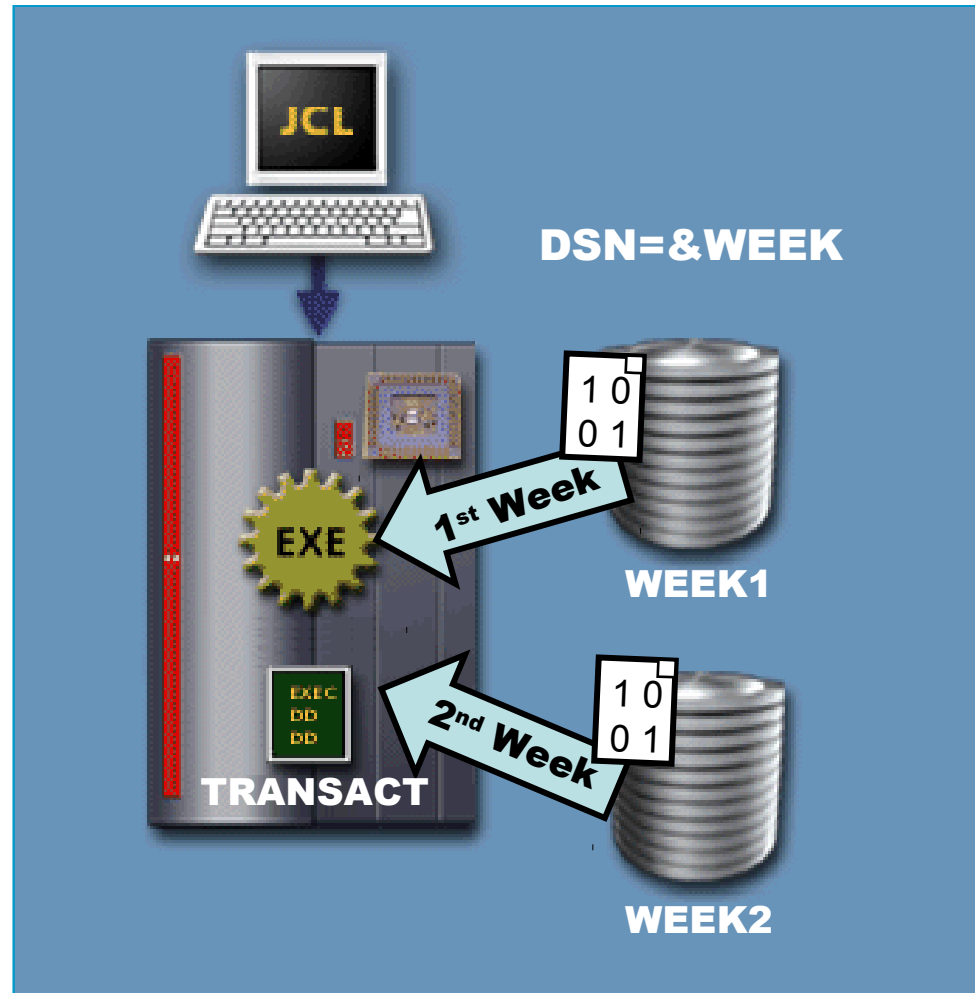
Symbolic parameters are often specified for DD statement parameters, if the values for those parameters are likely to change.



DD statement parameters.

DD statement parameters – an example.

For example, suppose the identification of the data set containing customer transactions for TRANSACT varies from week to week.



DD statement parameters.

DD statement parameters – an example.

The TRANSACT procedure definition with a symbolic DSN parameter value for the DD1 DD statement is shown on the right.

Assume that the identification of the data set containing customer transaction for TRANSACT varies from week to week.

```
//PSTEP1 EXEC PGM=PROG1,ACCT=&DEPT
//DD1      DD      DSN=&WEEK,DISP=SHR
//DD2      DD      DSN=MASTER,DISP=SHR
//DD3      DD      SYSOUT=A
//DD4      DD      DSN=&&VALID,
//          DISP=(NEW,PASS),
//          UNIT=SYSDA,
//          SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2,ACCT=&DEPT
//DD5      DD      DSN=&&VALID,
//          DISP=(OLD,DELETE)
//DD6      DD      SYSOUT=A
```



Using symbolic parameters.

Assigning values.

How to assign values with the EXEC statement?

When invoking the procedure, you simply code the appropriate value for &WEEK on EXEC statement, as shown on the right.

How to assign values with the SET statement?

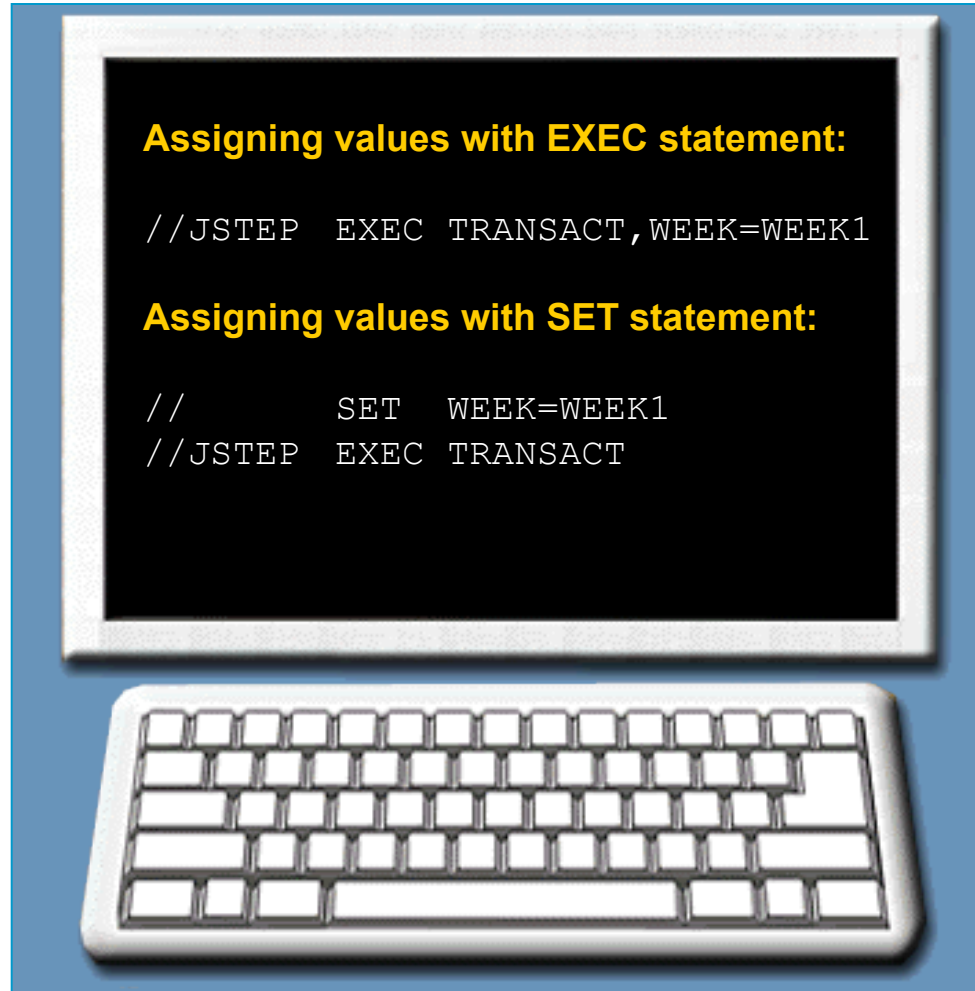
Alternatively, you can assign a value with a SET statement coded at an appropriate place in the job stream, as shown on the right.

Assigning values with EXEC statement:

```
//JSTEP EXEC TRANSACT,WEEK=WEEK1
```

Assigning values with SET statement:

```
//          SET WEEK=WEEK1  
//JSTEP EXEC TRANSACT
```



Using symbolic parameters.

Are we on track?

Review the JCL statements from a procedure named ANYPROC. Code symbolic parameters to meet the following specifications:

Code a symbolic parameter for the data set name for the input data set in statement DD1. Name it &INPUT.

```
//S1          EXEC  PGM=PROG1,ACCT=&GRP  
//DD1        DD    DSN=_____,DISP=SHR
```

Code a symbolic parameter for the VOL=SER= parameter for the data set in statement DD2. Name it &SERNO. Specify shared access mode {SHR}.

```
//DD2        DD    DSN=CKDATA,UNIT=3390,  
//          VOL=SER=_____
```

Using symbolic parameters.

Are we on track?

Code a statement to invoke procedure ANYPROC from the previous example, and assign values to the symbolic parameters as follows:

1. Assign a value of MYDATA to the &INPUT symbol.
2. Assign a value of 692912 to the &SERNO symbol.

```
//JSTEP EXEC ANYPROC,_____
```


DD statement parameters.

DD statement parameters – an example.

The following EXEC statement is used to invoke the procedure using WEEK1 as the input data set:

```
//JSTEP EXEC TRANSACT,WEEK=WEEK1
```

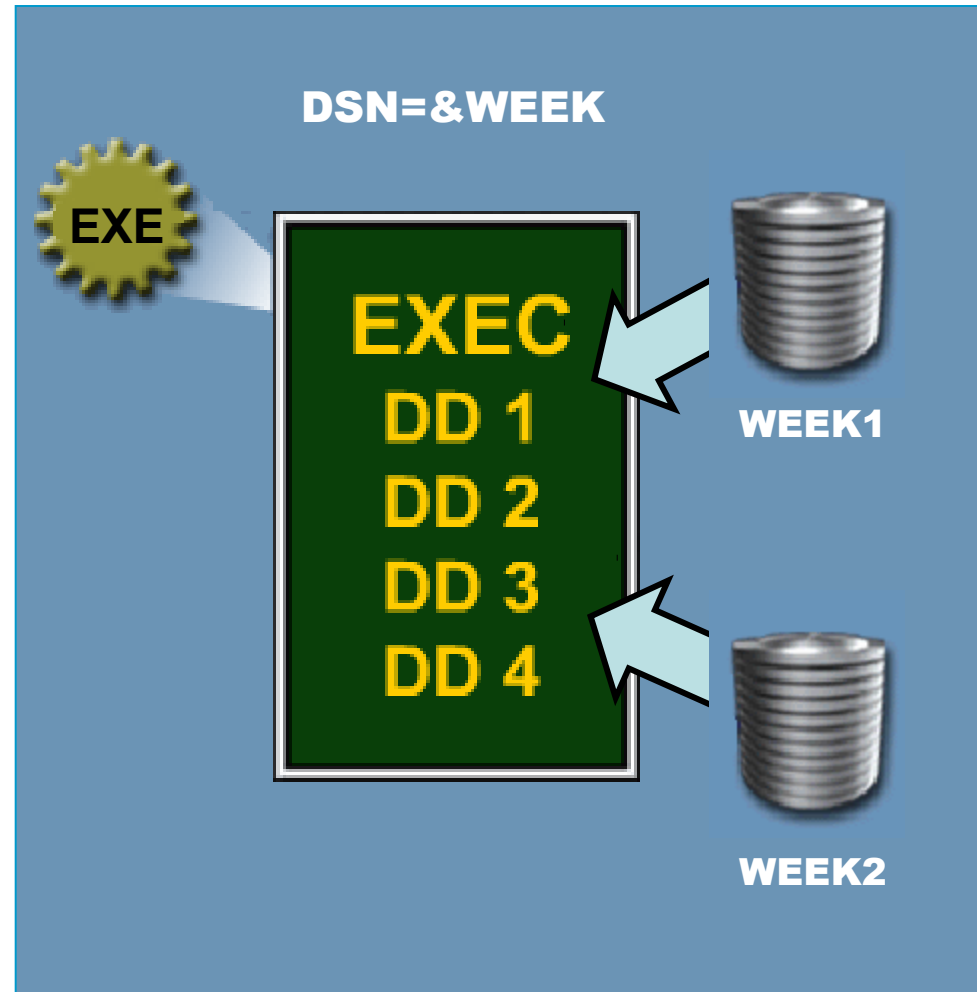
By simply assigning the appropriate data set name to &WEEK when the procedure is invoked, the data sets to be used for the other weeks can be specified.

DD statement parameters.

Assigning values.

If the same symbolic parameter is used to represent an parameter (such as the UNIT parameter) in several DD statements, then the value needs to be assigned to the symbolic parameter only once.

The value is applied to all occurrences of the parameter. The need to code several override DD statements can be eliminated.



DD statement parameters.

Rules for assigning values to DD statement parameters.

The rules for assigning values to or nullifying DD statement parameters are:

- **Separate the value assignments or nullifications from parameters and from each other using comas.**
- **Specify the symbolic parameter without the preceding ampersand, followed by an equal sign and value.**
- **To nullify a symbolic parameter specify the symbolic parameter without the preceding ampersand, followed by an equal sign. The value for the symbolic parameter should not be coded.**
- **Symbolic parameters can be nullified or assigned values in any sequence.**

DD statement parameters.

Are we on track?

The following JCL statements are from a procedure named ANYPROC.

```
//S1 EXEC PGM=PROG1,ACCT=&GRP  
//DD1 DD DSN=&INPUT,DISP=SHR  
//...  
//S2 EXEC PGM=&A,ACCT=&GRP
```

Which of the following statements are correct to invoke the procedure and specify the following nullify the &GRP symbol, assign a value of MYDATA to &INPUT, assign a value of TSTPRG to &A.

- A. **//JSTEP EXEC ANYPROG,GRP=,INPUT=MYDATA**
- B. **//JSTEP EXEC ANYPROC,INPUT=MYDATA,A=TSTPRG,GRP=**
- C. **//JSTEP EXEC ANYPROC,A=TSTPRG,GRP=,INPUT=MYDATA**
- D. **//JSTEP EXEC ANYPROC,A=TSTPRG,INPUT=MYDATA**

DD statement parameters.

Assigning values to DD statement parameters.

While assigning values to symbolic parameters:

- **An appropriate value that represents a data set name must be assigned to the symbolic parameter.**
- **If the symbolic parameter is not assigned a value, the system considers the symbolic parameter to be the name of a temporary data set because it begins with the ampersand character.**

DD statement parameters.

Are we on track?

If no value is assigned to a symbolic parameter in a procedure DD statement at the time of invoking a procedure, it is treated as a _____ data set.

Default values.

Recognizing default values.

At the time symbolic parameters are created, creator of the procedure can also assign default values to the parameters.

What are the characteristics of default values?

The characteristics of default values are:

- **They are specified in the parameter field of the PROC statement.**
- **They do not have a preceding ampersand.**
- **In the case of multiple symbolic parameters, they are separated by a comma.**
- **They can be assigned in any sequence.**

Default default values.

General rules and format.

At the time you create symbolic parameters, you can also assign default values. You do so on the PROC statement of the procedure. If the user determines that the values are appropriate at the time the procedure is executed, the user does not have to assign values.

The rules for specifying default values for symbolic parameters are as follows:

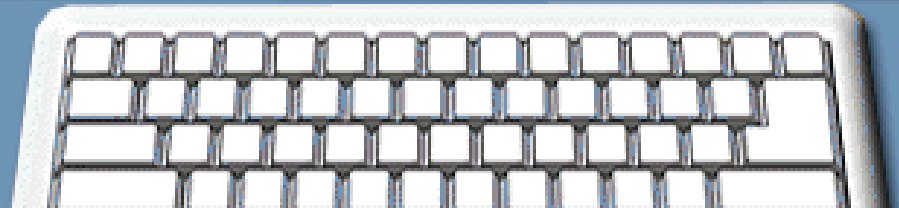
- **Specify default values for symbolic parameters in the operand field of the PROC statement. (A PROC statement must always precede an in-stream procedure definition and is required in a cataloged procedure only if you are assigning default values to symbolic parameters.)**
- **Specify each symbolic parameter without the preceding ampersand, followed by an equal sign and the default value.**
- **Separate the default value specifications for multiple symbolic parameters by a comma.**
- **You can assign default values for symbolic parameters in any sequence. You are not required to follow the sequence in which the symbolic parameters occur in the procedure definition.**

Default values.

Default values – an example.

In the TRANSACT procedure definition shown on the right, the PROC statement assigns the default value G300 to &DEPT and the default value PROG2 to &PROG.

```
//TRANSACT PROC DEPT=G300,PROG=PROG2
//PSTEP1 EXEC PGM=PROG1,
// ACCT=&DEPT
//DD1 DD DSN=&WEEK,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=&PROG,ACCT=&DEPT
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```



DD statement parameters.

Are we on track?

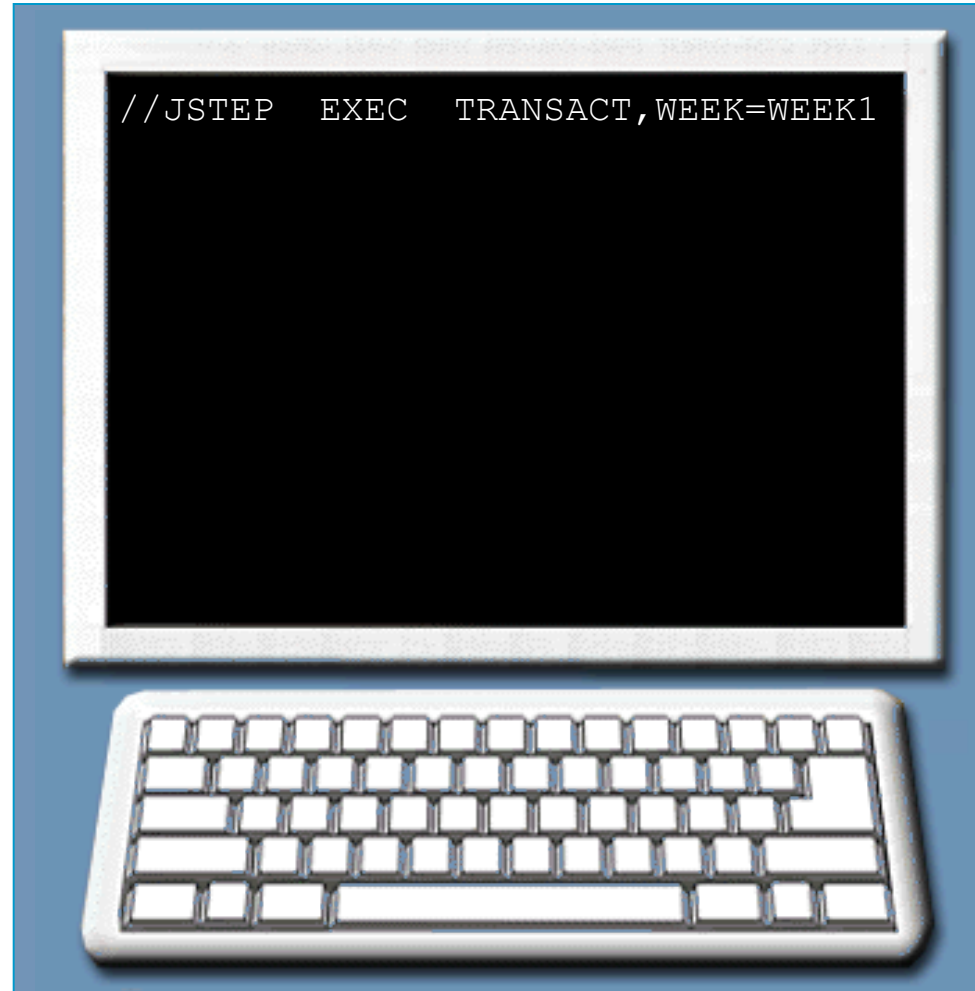
Which of the following statements are true for default values?

- A. They are assigned on the PROC statement of a procedure.**
- B. They are preceded by an ampersand.**
- C. They must be assigned in a specific sequence.**
- D. They can be overridden when the procedure is executed.**

Default values.

Accepting default values.

The EXEC statement required to accept the default values and specify the data set named WEEK1 is shown on the right.



Default values.

Are we on track?

Code an EXEC statements to invoke the TRANSACT procedure and accept default values.

```
// EXEC _____
```

Default values.

Nullifying default values.

When should you assign a value?

A value for a symbolic parameter should be assigned (or nullified) only if:

- **The default values specified in the PROC statement is not appropriate for that particular execution of the procedure.**
- **No default value is specified for the symbolic parameter on the PROC statement in the procedure.**

Default values.

Nullifying default values.

The rules for overriding or nullifying the default values are:

- **To override a default value, specify the symbolic parameter without the preceding ampersand, followed by an equal sign and the appropriate value.**
- **To nullify a default value, specify the symbolic parameter without the preceding ampersand, followed by only an equal sign.**
- **The overrides and nullifications for default values can be specified in any sequence.**

Default values.

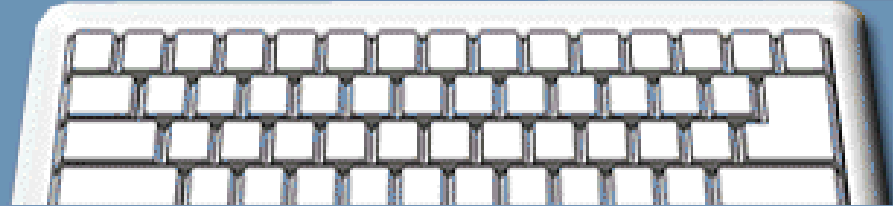
Nullifying default values – procedure definition.

In this example, the TRANSACT procedure is to be invoked with some input test data (DD1 DD statement) that resides in a data set named TSTDATA and a test program named TESTPRG for step PSTEP2.

Because this execution is a test, a department number need not be specified.

Thus, the default values G300 (specified for &DEPT) and PROG2 (specified for &PROG) are not needed.

```
//TRANSACT PROC DEPT=G300,PROG=PROG2
//PSTEP1 EXEC PGM=PROG1,
// ACCT=&DEPT
//DD1 DD DSN=&WEEK,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=&PROG,ACCT=&DEPT
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```



Default values.

Nullifying default values – EXEC statement.

The EXEC statement for the invocation is shown on the right. The EXEC statement does the following:

- Nullifies the default values specified for &DEPT.
- Overrides the default value specified for &PROG and assigns the value TESTPRG instead.
- Assigns the value TSTDATA to &WEEK.

```
//JSTEP EXEC TRANSACT,  
//          DEPT=,  
//          PROG=TESTPRG,  
//          WEEK=TSTDATA
```



Default values.

Are we on track?

Which of the following statements are true of assigning default values for symbolic parameters?

- A. They are assigned on the EXEC statement that invokes the procedure.**
- B. You can assign default values for as many symbolic parameters as needed.**
- C. They can be nullified by the procedure user.**
- D. They are preceded by an ampersand {&}.**
- E. They are assigned on the PROC statement.**

Summary example.

Are we on track?

Which of the following should you include in the procedure to meet the needs of the processing environment? Choose all that apply.

- A. A symbolic parameter for PGM=in the first procedure step.**
- B. A symbolic parameter for PGM=in the second procedure step.**
- C. A SET statement.**
- D. A symbolic parameter for ACCT=in the EXEC statement.**
- E. Specific values for all data set names and attributes.**

Summary example.

Are we on track?

Given that the data set names and attributes of the input data set are likely to vary widely, which of the following would best achieve the execution-time objectives?

- A. Use the DDNAME parameter to represent the input data set.**
- B. Code data specifications needed for typical use for the input data set.**
- C. Assign symbolic parameters for the input data set name and attributes.**

Summary example.

Are we on track?

Complete the JCL statements for the sample procedure as per the following instructions:

In line 1, specify a symbolic parameter for ACCT in PSTEP1. Name it &DEPT.

In line 2, use the DDNAME parameter for the input data set. Code the value as INPUT.

In line 6, specify a symbolic parameter for PGM in PSTEP2. Name it &PROG2.

```
1. //PSTEP1 EXEC PGM=PROG1, _____
2. //DD1      DD      _____
3. //DD2      DD      DSN=MASTER, DISP=SHR
4. //DD3      DD      SYSOUT=A
5. //DD4      DD      DSN=&&VALID, DISP=(NEW, PASS), UNIT=SYSDA,
                      SPACE=(TRK, (1, 1))
6. //PSTEP2 EXEC _____
7. //DD5      DD      DSN=&&VALID, DISP=(OLD, DELTE)
8. //DD6      DD      SYSOUT=A
```

Summary example.

Are we on track?

Code the JCL used to invoke the sample procedure and provide the following execution-time information:

In line 1, code a department number of G300 and the program named TESTPRG for execution in PSTEP2

In line 2, code input transactions from cards in the job stream.

```
1. //JSTEP EXEC          TRANSPORT, _____
2. //PSTEP1. _____ *
   (input data records)
   /*
```

Symbolic parameters.

Unit summary.

Now that you have completed this unit, you should be able to:

- **Understand the purpose and form of the DDNAME parameter in a procedure definition.**
- **Resolve the DDNAME parameter when executing a procedure by identifying the actual data set names and attributes represented by the DDNAME parameter.**
- **Understand the purpose and form of symbolic parameters in a procedure definition.**
- **Assign values to symbolic parameters at the time of invoking a procedure.**

JCL

Chapter b6 Symbolic parameters

Job Control Language

Chapter a1. Introduction to JCL

Chapter a2. Coding JOB statements

Chapter a3. Coding EXEC statements

Chapter a4. Coding DD statements

Chapter a5. Analyzing job output

Chapter a6. Conditional processing

Job Control Language

Chapter b1. Using special DD statements

Chapter b2. Introducing procedures

Chapter b3. Modifying EXEC parameters

Chapter b4. Modifying DD parameters

Chapter b5. Determining the effective JCL

Chapter b6. Symbolic parameters

Job Control Language

Chapter c1. Nested procedures

Chapter c2. Cataloging procedures

Chapter c3. Using utility programs

Chapter c4. Sample utility application

Symbolic parameters.

Chapter b6

Symbolic parameters

5

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Symbolic parameters.

Unit introduction.

In earlier units you have seen that procedures can be modified when they are executed using override and addition statements. However, these modifications are subject to strict sequencing rules, and they are not always convenient to implement.

This unit describes two techniques that can provide an alternative, and often easier, means of modifying procedures for execution:

- Using the **DDNAME** parameter on procedure **DD** statements.
- Using symbolic parameters on procedure **EXEC** and **DD** statements.

If these parameters are included in the procedure definition when it is coded, values can be easily assigned at the time of executing a procedure.

Symbolic parameters.

Course objectives.

Be able to:

- **Understand the purpose and form of the DDNAME parameter in a procedure definition.**
- **Resolve the DDNAME parameter when executing a procedure by identifying the actual data set names and attributes represented by the DDNAME parameter.**
- **Understand the purpose and form of symbolic parameters in a procedure definition.**
- **Assign values to symbolic parameters at the time of invoking a procedure.**

DDNAME and symbolic parameters.

Parameter variation in procedure.

Variations in procedure statements can be quite common between different instances of a procedure's use.

For example, in a procedure like TRANSACT, one or more items vary from one instance of the procedure's use to the next, such as:

- Name, location, or other attributes of one or more data sets identified in the procedure.
- Names of one or more programs to be executed.



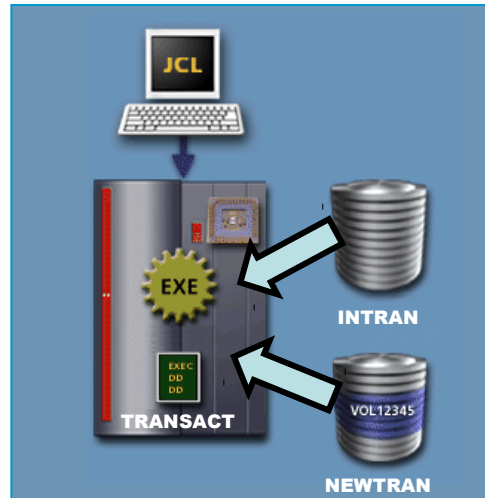
DDNAME and symbolic parameters.

Parameter variation in procedure.

These variations can be accommodated by temporarily modifying the procedure at the time the procedure is executed.

This is accomplished by adding or overriding parameters on the procedure EXEC and DD statements.

These modifications are subject to strict formatting and sequencing rules.



Also the parameters that can be overridden, the step names within the procedure, the DDnames of the statements overridden, and so on, must be known.

DDNAME and symbolic parameters.

Techniques to change parameters.

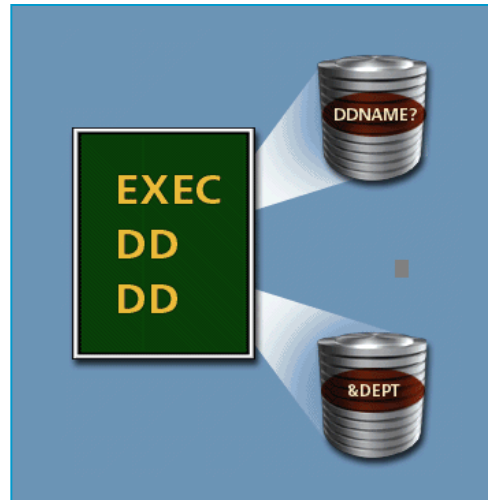
If variations in procedure are common, override and addition statements can be avoided.

Alternatives to using override and addition statements are:

- Use the DDNAME parameter on one or more DD statements in the procedure definition.
- Use symbolic parameters to represent EXEC or DD statement parameters in the procedure definition.

10

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.



The DDNAME parameter is used in place of an actual data set name and its attributes. The effect of coding the DDNAME parameter is to postpone the definition of the data set until procedure execution.

The symbolic parameter is used in place of an actual parameter value, and is assigned a value at procedure execution.

DDNAME and symbolic parameters.

TRANSACT procedure.

The TRANSACT procedure, on the right, meets two of the guidelines:

- It performs the exact functions required by the sample application.
- It does not require the procedure user to specify any execution-time alterations for typical procedure use.

But does it simplify as much as possible the JCL that the user has to code at procedure execution?

```
//PSTEP1 EXEC PGM=PROG1
//DD1 DD DSN=INTRAN, DISP=SHR
//DD2 DD DSN=MASTER, DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID
// DISP=(NEW, PASS),
// UNIT=SYSDA,
// SPACE=(TRK, (1, 1))
//PSTEP2 EXEC PGM=PROG2
//DD5 DD DSN=&&VALID,
// DISP=(OLD, DELETE)
//DD6 DD SYSOUT=A
```

DDNAME and symbolic parameters.

Are we on track?

A _____ parameter allows to postpone defining data set and its attributes.

12

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is DDNAME.

DDNAME and symbolic parameters.

DDNAME parameters.

If the attributes of the data sets identified in TRANSACT vary from one instance of the procedure's use to the next, the user will need to code DD override and addition statements at the time of executing the procedure.

However, generalizing the TRANSACT procedure to process data input from a variety of sources, you will avoid the need for overrides and additions. One way of accomplishing this is the DDNAME parameter.



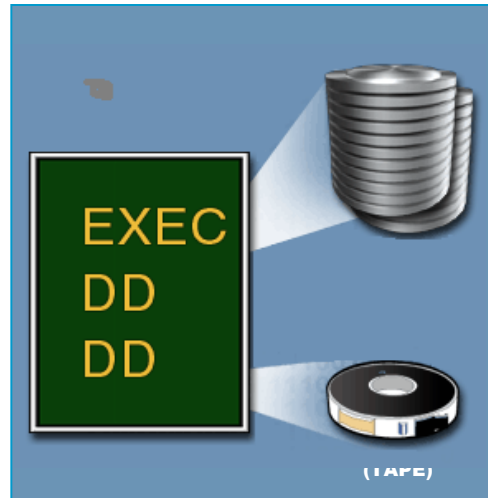
The DDNAME parameter.

Input data streams.

The DDNAME parameter is used in a procedure definition to represent an input data set.

It permits the procedure user to define data set attributes at the time the procedure is invoked.

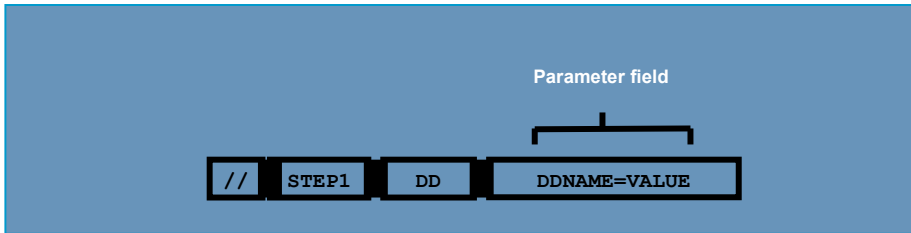
The example on the right shows that at various times the input data for a procedure could reside on a direct access volume or a tape.



Defining data set attributes at the time a procedure is invoked is useful when a procedure may process input transactions from any one of several different sources.

The DDNAME parameter.

Coding a DDNAME.



You should code a DDNAME on the appropriate procedure step, generally to represent an input data set. This postpones definition of the data set attributes until the procedure is invoked.

The DDNAME parameter consists only of the operand name {DDNAME} followed by an equal sign and a Value (VALUE, in this example).

The DDNAME parameter.

Coding a DDNAME - an example.

The example on the right illustrates the TRANSACT procedure definition with a DDNAME parameter for the input data set.

Note that the value of the DDNAME parameter is not the same as the ddname of the statement (DD1 in this example).

```
//PSTEP1 EXEC PGM=PROG1
//DD1 DD DDNAME=INPUT
//DD2 DD
DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
```

The value of the DDNAME parameter is not the same as the ddname of the statement.

The DDNAME parameter.

Resolving the DDNAME parameter.

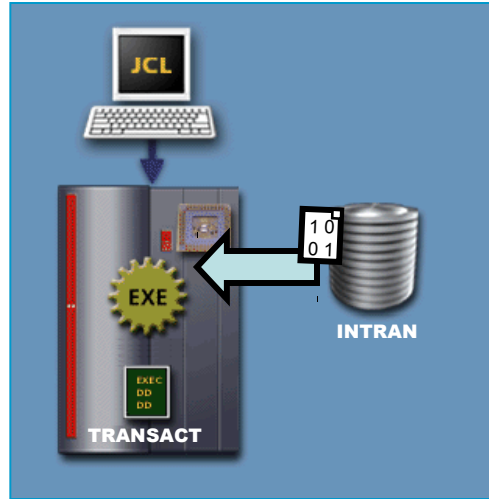
At the time of execution to invoke the procedure, you would code:

- An EXEC statement that executes the procedure.
- A DD statement to identify the attributes of the data set represented by DDNAME=INPUT.

This is called resolving the DDNAME parameter.

For example, the DDNAME is resolved with a data set INTRAN as shown in the code:

```
//STEP          EXEC TRANSACT  
//PSTEP1.INPUT DD DSN=INTRAN, . .  
17
```



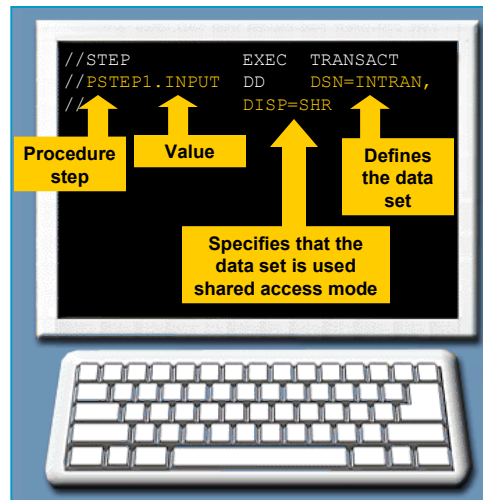
Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The DDNAME parameter.

Resolving the DDNAME parameter.

The DD statement that resolves the DDNAME parameter has two components:

- The name of the procedure step in which the DD statement containing the DDNAME parameter is used, followed by a period.
- The value as specified in the DDNAME parameter.



The DDNAME parameter.

Are we on track?

Which of the following is an advantage of using the DDNAME parameter in a procedure?

- A. You can postpone specifying data set attributes until procedures execution.**
- B. You enable the user to execute the procedure using different programs.**
- C. You can relieve the procedure user of the need to specify data set attributes.**

The correct answer is A.

The DDNAME parameter.

Are we on track?

The following is a procedure called MYPROC. The procedure uses the DDNAME parameter to represent the data set in PSTEP1.

```
//PSTEP1 EXEC PGM=PGM1  
//DD1 DD DDNAME=INPUT
```

Code a statement to invoke the procedure.

```
//JSTEP EXEC _____
```

20

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is MYPROC

But this only invokes the procedure. You should also specify the DDNAME parameter resolution:

```
//PSTEP1.INPUT DD DSN=...
```

See next slides...

The DDNAME parameter.

Are we on track?

Code a DDNAME parameter to define the temporary data set for PROG1 that refers to the DD name DD4. Define the actual data set characteristics using the name TSTDATA.

TRANSACT Procedure:

```
//PSTEP1 EXEC PGM=PROG1
//DD1 DD DSN=...
//DD2 DD DSN=MASTER, DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD _____
```

21

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is: DDNAME=TSTDATA

The DDNAME parameter.

Are we on track?

The following is a procedure called MYPROC. The procedure uses the DDNAME parameter to represent the data set in PSTEP1.

```
//PSTEP1 EXEC PGM=PGM1  
//DD1 DD DDNAME=INPUT
```

Code a statement to resolve the DDNAME parameter with a data set named MYDATA.

```
//JSTEP EXEC MYPROC
```

22

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is //PSTEP1.INPUT DD DSN=MYDATA

The DDNAME parameter.

Are we on track?

Complete the statement required at procedure invocation time to resolve DDNAME as an existing data set named MYDATA.

JCL to invoke TRANSACT:

```
//SS1          EXEC TRANSACT  
//PSTEP1.TSTDATA DD DSN=MYDATA,DISP_____
```

23

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is: =OLD

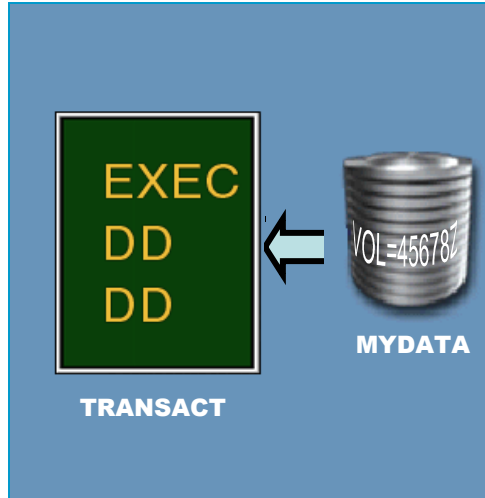
The DDNAME parameter.

Resolving the DDNAME parameter – example 1.

To execute the TRANSACT procedure, the user executes the procedure using a data set MYDATA in shared access mode.

MYDATA resides on a 3390 direct access volume with a volume identifier of 45678Z. In this case, the DDNAME parameter is resolved with the data set name of MYDATA.

```
//STEP          EXEC TRANSACT
//PSTEP1.INPUT DD DSN=MYDATA,
//              UNIT=3390,
//              VOL=SER=45678Z
//
```



Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The DDNAME parameter.

Resolving the DDNAME parameter – example 2.

The DDNAME parameter in the same procedure is resolved for using different input data.

The user invokes the TRANSACT procedure and directs that input data be taken from card images supplied in the job stream.



Since job stream data is not allowed to be coded within a procedure, the DDNAME parameter is a good way to define data sets that will normally be on cards.

The DDNAME parameter.

Avoiding an input data set.

If an input data set is not needed for the execution of the procedure then the DD statement to identify the data set should not be coded.

The unresolved DDNAME is automatically treated as a dummy.

The DDNAME parameter.

Rules for resolving DDNAME parameters.

The DDNAME parameter can be used in a procedure definition in as many as five DD statements within each procedure step.

The sequencing rules to be followed when resolving DDNAME parameters are:

- **They must be coded in procedure step sequence.**
- **DD statements to resolve DDNAME parameters are considered additions and should be sequenced as such (that is, in procedure step sequence).**
- **If multiple DDNAME parameters are specified in a single step of the procedure definition, the DD statements to resolve them can be coded in any sequence.**

27

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

See DDNAMEUsage.ppt example for more information about the DDNAME parameter.

The DDNAME parameter.

Are we on track?

The following is a procedure called ANYPROC:

```
//PSTEP1 EXEC PGM=CBL5
//IN      DD      DDNAME=WEEKNO
//OUT     DD      DSN=&&TEMP, DISP=(NEW,PASS),
//          UNIT=SYSDA, SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=CBL6
//PSD     DD      DSN=&&TEMP, DISP=(OLD,DELETE)
//RPT     DD      DDNAME=OUTPUT
```

Code the statements required to resolve the procedure's WEEKNO DDNAME parameters to specify WEEKNO as an existing data set named WEEK2.

```
//JSTEP EXEC ANYPROC
//_____
```

28

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is: PSTEP1.WEEKNO DD DSN=WEEK2, DISP=OLD

The DDNAME parameter.

Are we on track?

The following is a procedure called ANYPROC:

```
//PSTEP1 EXEC PGM=CBL5
//IN      DD      DDNAME=WEEKNO
//OUT     DD      DSN=&&TEMP, DISP=(NEW,PASS),
//          UNIT=SYSDA, SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=CBL6
//PSD     DD      DSN=&&TEMP, DISP=(OLD,DELETE)
//RPT     DD      DDNAME=OUTPUT
```

Code the statements required to resolve the procedure's OUTPUT DDNAME parameters to specify OUTPUT to be a data set named SAVE.

```
//JSTEP EXEC ANYPROC
//PSTEP1.WEEKNO DD DSN=WEEK2
//_____
```

29

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is: PSTEP2.OUTPUT DD DSN=SAVE

The DDNAME parameter.

Glossary.

DUMMY

A parameter coded in the DD statement that instructs the system to ignore all other parameters, with the exception of DCB.

DDNAME Parameter

A parameter coded in a procedure definition that represents an actual data set and its attributes.

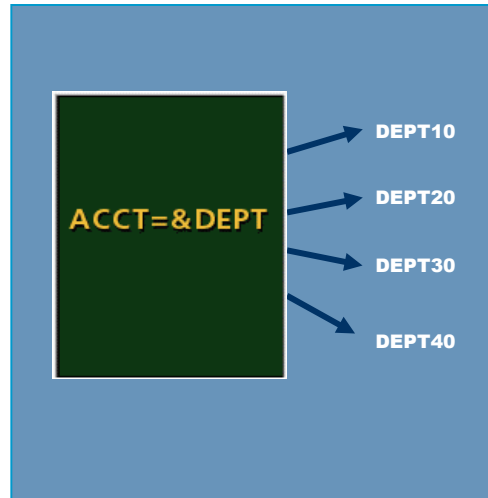
Symbolic parameters.

Symbolic parameters.

What are symbolic parameters?

Symbolic parameters are a technique to build flexibility into procedures. A symbolic parameter represents a procedure statement parameter or subparameter.

The procedure statement operands whose values are most likely to change from one procedure invocation to another, are suitable candidates to be coded as symbolic parameters. The example on the right illustrates such a procedure statement operand, ACCT.



31

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Symbolic parameters are often assigned default values. When the procedure is invoked the default values can be accepted or appropriate values can be assigned to them on the EXEC statement.

Symbolic parameters.

Significance of symbolic parameters.

How do symbolic parameters help?

Coding symbolic parameters simplifies the procedure user's task in a number of ways:

- **The user can assign the appropriate value when the procedure is invoked.**
- **The user can simply accept a default value if one is assigned.**
- **The user can assign a value to the symbolic parameter only once to change all recurring like operands in statements occurring through the entire procedure. This is useful if the same value is to be used for like operands in statements occurring throughout the procedure.**

Recurring – opakující se.

Symbolic parameters.

Standards for symbolic parameter names.

Rules for coding symbolic parameter names are as follows:

- Are preceded by an ampersand (&).
- Consist of up to seven alphanumeric (A-Z, 0-9) or national (#,@,\$) characters, beginning with an alphabetic or national character.
- Can represent EXEC or DD statement parameters.
- It cannot conflict with an EXEC statement keyword. That is, keywords such as PGM or COND cannot be used.
- It cannot conflict with the name of another symbolic parameter in the same procedure. If more than one value is assigned to a symbolic parameter, only the first can be used.

Examples of Symbolic Parameters

```
ACCT=&DEPT
```

```
PGM=&PROG1
```

```
//STEP1 EXEC PGM=PROG1,ACCT=&DEPT
```

```
//DD1 DD DSN=&WEEK
```

Symbolic parameters are preceded by a single ampersand, whereas temporary data set names are preceded by two ampersands.

The name should be meaningful (such as &DEPT to represent a department number, &NUMBER to represent a volume identifier, or &DEVICE to represent an auxiliary storage unit).

Symbolic parameters.

Are we on track?

Which of the following statements apply to the use of symbolic parameters?

- A. They can be specified for a DD or EXEC statement parameter.**
- B. They represent a procedure statement parameter or subparameter.**
- C. They are identified in a procedure by a preceding # sign.**
- D. They are coded in a procedure definition.**

34

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is A., B., and D.

Symbolic parameters.

Are we on track?

Which of the following items contains the symbolic parameter:

- A. ACCT=&GRP**
- B. SYSOUT=A**
- C. DSN=&&SYN**
- D. VOL=SER=&SERNO**

35

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is A. and D.

Symbolic parameters.

Are we on track?

Which of the following are advantages of symbolic parameters?

- A. They enable the user to simply accept default values, if appropriate.**
- B. They can be given a name that is meaningful, such as &DEPT.**
- C. They enable the user to assign values when the procedure is executed.**
- D. They can be assigned different values within a job.**
- E. They can represent a data set and all its attributes.**

36

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is A., B., C., and E.

Symbolic parameters.

Are we on track?

Which are the symbolic parameters in the following JCL from a procedure named ANYPROC?

```
//S2      EXEC    PGM=&A, ACCT=&GRP  
//DD5     DD      DSN=&&SYM, DISP={ OLD, DELETE}  
//DD6     DD      SYSOUT=A
```

A. PGM=&A,ACCT=&GRP,DSN=&&SYM

B. DSN=&&SYM,ACCT=&GRP

C. PGM=&A,ACCT=&GRP

The correct answer is C.


EXEC statement parameters.

Recognizing symbolic parameter – an example.

An example of a symbolic parameter to represent an EXEC statement parameter is shown on the right.

Here the TRANSACT procedure is defined with a symbolic ACCT parameter (&DEPT) for PROG1 (in PSTEP1) and PROG2 (in PSTEP2).

```
ACCT=&DEPT
```



```
//PSTEP1 EXEC PGM=PROG1,ACCT=&DEPT
//DD1 DD DSN=INTRAN,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2,ACCT=&DEPT
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```

In this way, several different departments in the company can easily use the TRANSACT procedure. Each department will simply provide its department number, for accounting purposes, when the procedure is used.

EXEC statement parameters.

Assign or nullify values to EXEC statement parameters.

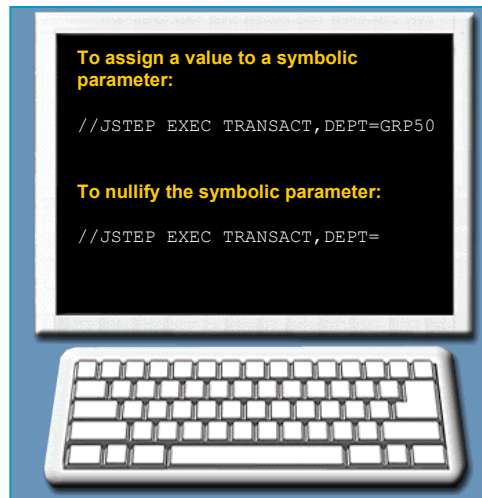
The procedure user can assign or nullify a value to the symbolic parameter on the EXEC statement.

How to assign a value to a symbolic parameter?

The user should specify the symbolic parameter without the preceding ampersand, followed by an equal sign and the value.

How to nullify a symbolic parameter?

The user should specify the symbolic parameter without the preceding ampersand, followed by only an equal sign.



If the value of the symbolic parameter is not coded while invoking the procedure, the symbolic parameter is nullified:

```
//JSTEP EXEC TRANSACT,DEPT=
```

Nullifying the &DEPT in this example has the same effect as omitting the symbolic parameter from the procedure definition.

The following statement assigns the value GRP50 to the &DEPT symbolic parameter:

```
//JSTEP EXEC TRANSACT,DEPT=GRP50
```

If several symbolic parameters are assigned, the user can assign values in any order.

EXEC statement parameters.

Rules for assigning values to nullifying EXEC statement parameters.

The rules for assigning values to or nullifying EXEC statement parameters are:

- **Separate the value assignments or nullifications from operands and from each other using commas.**
- **Specify the symbolic parameter without the preceding ampersand, followed by an equal sign and the value.**
- **To nullify a symbolic parameter specify the symbolic parameter without the preceding ampersand, followed by an equal sign. No value is specified for the symbolic parameter while nullifying it.**
- **Symbolic parameters can be nullified or assigned values in any sequence.**

EXEC statement parameters.

Are we on track?

**The effect of not coding a value for a symbolic parameter is to
 the symbolic parameter.**

41

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is nullify.

EXEC statement parameters.

Are we on track?

The following is the TRANSACT procedure definition:

```
//PSTEP1 EXEC PGM=PROG1,ACCT=&DEPT
//DD1 DD DSN=INTRAN,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2,ACCT=&DEPT
//DD5 DD DSN=&&VALID,DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```

Code an EXEC statements to execute the procedure and assign a value of GRP25 to the &DEPT symbolic parameter.

//JSTEP EXEC _____

The correct answer is: TRANSACT, DEPT=GRP25

Assigning different values.

Conditions.

Different values cannot be assigned to the same symbolic parameter when invoking the procedure.

For example, it is not possible to assign GRP50 to &DEPT in PSTEP1 and GRP100 to &DEPT in PSTEP2 unless one of the following conditions exists:

- A different symbolic parameter represents each occurrence of the parameter.
- An EXEC or DD statement override or addition statement is used when invoking the procedure.



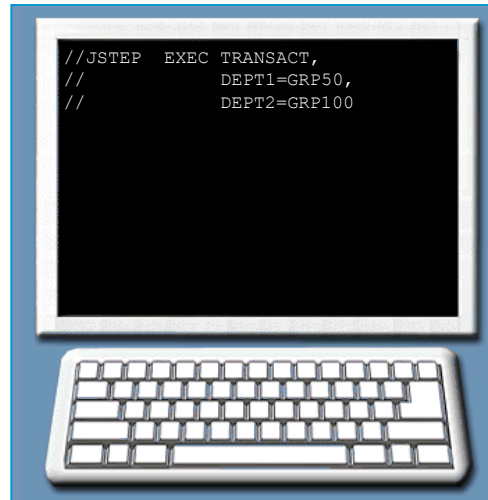
Assigning different values.

Assigning different values - example 1.

To illustrate the first condition, assume the TRANSACT procedure has been created to allow for two different values for the EXEC statement ACCT parameters, as follows:

```
//PSTEP1 EXEC PGM=PROG1,ACCT=&DEPT1  
...  
//PSTEP2 EXEC PGM=PROG2,ACCT=&DEPT2
```

In the example on the right, the TRANSACT procedure has been invoked with the values GRP50 and GRP100 for PSTEP1 and PSTEP2 respectively.



While invoking the procedure, the symbolic parameter &DEPT1 is assigned a value of GRP50, &DEPT2 a value of GRP100. Values can be assigned to the symbolic parameters in any order.

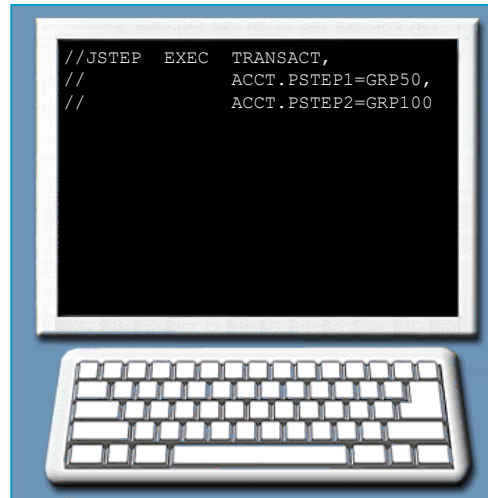
Assigning different values.

Assigning different values – example 2.

Here the procedure has been created without symbolic parameters:

```
//PSTEP1 EXEC PGM=PROG1  
...  
//PSTEP2 EXEC PGM=PROG2
```

At the time of invoking the procedure, an addition statement as shown on the right can be coded to assign an ACCT value of GRP50 for PSTEP1 and GRP100 for PSTEP2.



Assigning different values.

Are we on track?

As a procedure user, when can you assign different values to the same symbolic parameter?

- A. When a symbolic parameter represents the parameter.**
- B. When a different symbolic parameter represents each occurrence of the parameter.**
- C. When you code an addition or an override statement when invoking a procedure.**
- D. Never.**

46

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answers is D.

Assigning different values.

Assigning values to the PGM parameter.

The PGM parameter is the only EXEC statement parameter that cannot be modified with an override statement when a procedure is executed.

If a symbolic parameter is specified in the procedure definition, an appropriate value can be assigned when invoking the procedure.

Assigning different values.

The PGM parameter – an example.

In this example, different versions of the programs executed by the TRANSACT procedure are being tested to see which versions best suit departmental requirements.

Symbolic parameters are assigned to the PGM parameter in the TRANSACT procedure EXEC statement, so the procedure user can specify which program versions are to be executed.

Assigning different values.

Symbolic PGM parameters.

The TRANSACT procedure definition with symbolic PGM parameters for the PSTEP1 and PSTEP2 EXEC statements is shown on the right.

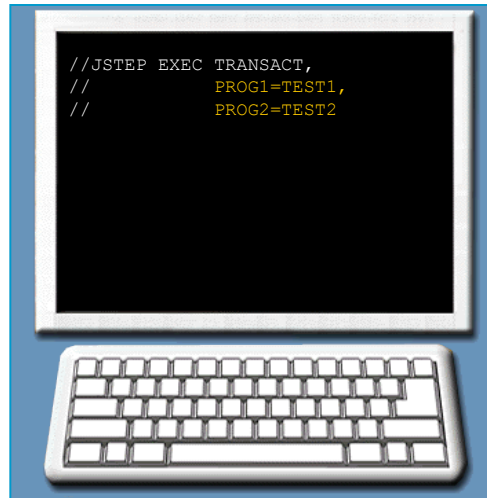
```
//PSTEP1 EXEC PGM=&PROG1
//DD1 DD DSN=INTRAN,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=&PROG2
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```



Assigning different values.

Symbolic PGM parameters.

After specifying symbolic parameter for PGM in the procedure definition, you can assign the appropriate value (or values) to the symbolic parameter when invoking the procedure, as shown in the Statement to invoke TRANSACT on the right.



Assigning different values.

Are we on track?

The following is the TRANSACT procedure definition:

```
//PSTEP1 EXEC PGM=&PROG1
//DD1 DD DSN=INTRAN,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=&PROG2
//DD5 DD DSN=&&VALID,DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```

Code an EXEC statement to execute

a.) TEST4 for PSTEP1 and

b.) TEST5 for PSTEP2.

//JSTEP EXEC _____

51

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is: TRANSACT,PROG1=TEST4,PROG2=TEST5

Assigning different values.

Are we on track?

Complete the S1 EXEC statement for a procedure named ANYPROC with the following symbolic parameters.

Code a symbolic parameter for PGM and name it &PROG:

```
//S1 EXEC PGM=_____
```

Code a symbolic parameter for ACCT and name it &DEPT:

```
// DD ACCT=_____  
//DD1 DD DSN=&INPUT,DISP=SHR
```

52

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is:

1. &PROG,
2. &DEPT

Assigning different values.

Are we on track?

Complete the JCL statement required to invoke procedure ANYPROC from the previous exercise, and specify the following:

- 1. Nullify the &DEPT symbol.**
- 2. Assign a value of MYPROC to &PROG.**

//JSTEP EXEC ANYPROC,_____

The correct answer is DEPT=,PROG=MYPROC or PROG=MYPROC,DEPT=

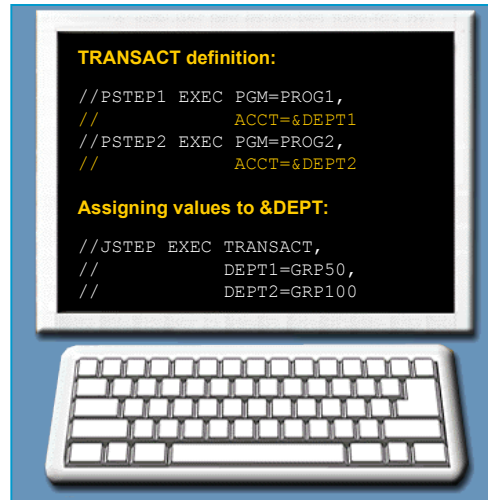
Assigning different values.

Assigning values.

When executing a procedure with symbolic parameters, you cannot assign different values to the same symbolic parameter within one job.

However, if you anticipate the need to assign different values, you can define a different symbolic parameter to represent each occurrence of the operand.

For example, when different departments want to execute TRANSACT in the same job you can define a different symbolic parameter for each occurrence of the ACCT operand.



54

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

To invoke the procedure, you should code an appropriate value for each &DEPT symbolic parameter.

Using symbolic parameters.

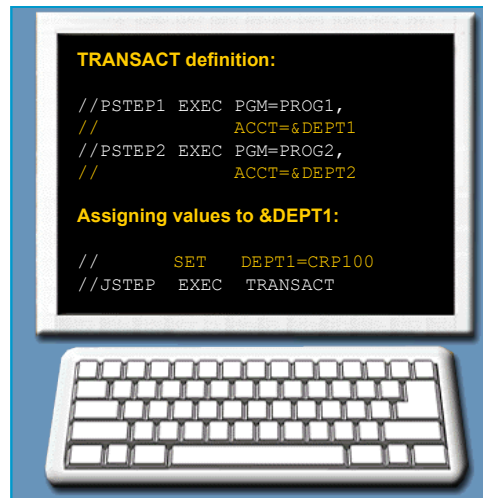
Assigning values using SET statement.

How to code a SET statement?

Rather than assigning values on the EXEC statement used to invoke the procedure, you code a SET statement as follows:

```
//      SET parameter=value,  
//      parameter=value
```

The example on the right shows how to assign values to &DEPT using a SET statement.



55

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

SET Statement

Purpose

Use the SET statement to:

- Define and assign initial values to symbolic parameters that are to be used when processing JCL statements.
- Change or nullify the values of defined symbolic parameters (those that are defined on previous SET statements) by assigning new values or nullifying current values.

Syntax

```
//[name] SET symbolic-parameter=value  
// [,symbolic-parameter=value]... [comments]
```

Using symbolic parameters.

Assigning values using SET statement.

SET statements can be placed anywhere in a job stream following the JOB statement. However, where they are placed will affect where they take effect.

The system assigns values to symbolic parameters as it encounters them. Therefore, a new SET statement will change the value of a previous SET statement. Similarly, values that are assigned on EXEC or PROC statements will change the values assigned on previous SET statements.

In the examples on the right, GRP100 would be the value set by the system, since it is the last value the system encounters.



Using symbolic parameters.

Are we on track?

When you use a SET statement to assign a value to a symbolic parameter, when does the new value apply?

- A. Only if the symbolic parameter did not have a previous value.**
- B. For the next use of the symbolic parameter only.**
- C. To all subsequent uses of the symbolic parameter, unless changed again.**

57

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is C.

DD statement parameters.

Symbolic parameters for DD statement operands.

Like DDNAME, symbolic parameters can also be specified for DD statement operands, if the values for those operands are likely to change.

When do you use DDNAME and when do you specify symbolic parameters?

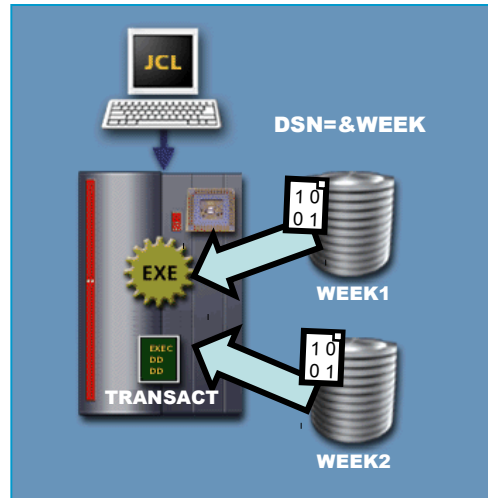
Specify symbolic parameters in a procedure if only a limited number of DD statement operands are likely to vary.

If, however, completely different data sets with different attributes are likely to be used, consider using the DDNAME operand instead. In that way, the users will not have to assign multiple values to multiple symbolic parameters.

DD statement parameters.

Changing DD statement parameters.

Symbolic parameters are often specified for DD statement parameters, if the values for those parameters are likely to change.

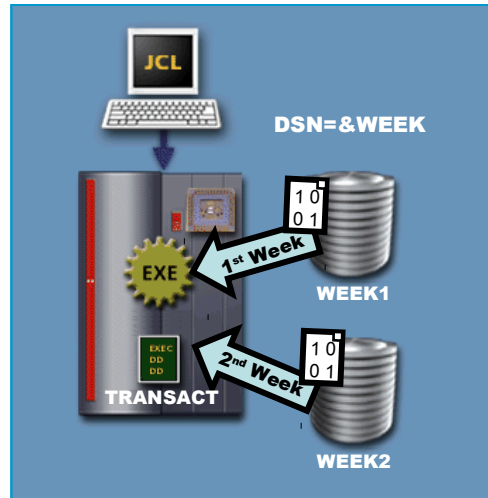


The procedure user does not have to code an override or addition DD statement when executing the procedure.

DD statement parameters.

DD statement parameters – an example.

For example, suppose the identification of the data set containing customer transactions for TRANSACT varies from week to week.



60

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

A data set named WEEK1 contains the transactions for the first week in the quarter, a data set named WEEK2 contains transactions for the second week in the quarter, and so on.

The creator of the procedure can accommodate this week-to-week change in the data set name by specifying the symbolic parameter DSN=&WEEK in the procedure definition.

DD statement parameters.

DD statement parameters – an example.

The TRANSACT procedure definition with a symbolic DSN parameter value for the DD1 DD statement is shown on the right.

Assume that the identification of the data set containing customer transaction for TRANSACT varies from week to week.

```
//PSTEP1 EXEC PGM=PROG1,ACCT=&DEPT
//DD1 DD DSN=&WEEK,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2,ACCT=&DEPT
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```



Using symbolic parameters.

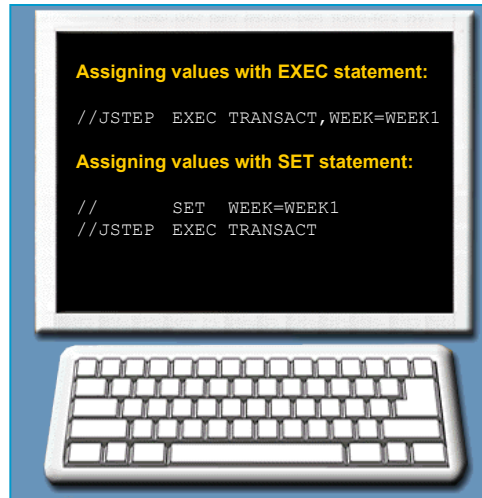
Assigning values.

How to assign values with the EXEC statement?

When invoking the procedure, you simply code the appropriate value for &WEEK on EXEC statement, as shown on the right.

How to assign values with the SET statement?

Alternatively, you can assign a value with a SET statement coded at an appropriate place in the job stream, as shown on the right.



Using symbolic parameters.

Are we on track?

Review the JCL statements from a procedure named ANYPROC. Code symbolic parameters to meet the following specifications:

Code a symbolic parameter for the data set name for the input data set in statement DD1. Name it &INPUT.

```
//S1      EXEC  PGM=PROG1,ACCT=&GRP  
//DD1     DD   DSN=_____,DISP=SHR
```

Code a symbolic parameter for the VOL=SER= parameter for the data set in statement DD2. Name it &SERNO. Specify shared access mode {SHR}.

```
//DD2     DD   DSN=CKDATA,UNIT=3390,  
//        VOL=SER=_____
```

63

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is:

1. &INPUT,
2. &SERNO, DISP=SHR

Using symbolic parameters.

Are we on track?

Code a statement to invoke procedure ANYPROC from the previous example, and assign values to the symbolic parameters as follows:

- 1. Assign a value of MYDATA to the &INPUT symbol.**
- 2. Assign a value of 692912 to the &SERNO symbol.**

```
//JSTEP EXEC ANYPROC,_____
```

The correct answer is INPUT=MYDATA,SERNO=692912

DD statement parameters.

DD statement parameters – an example.

The following EXEC statement is used to invoke the procedure using WEEK1 as the input data set:

```
//JSTEP EXEC TRANSACT,WEEK=WEEK1
```

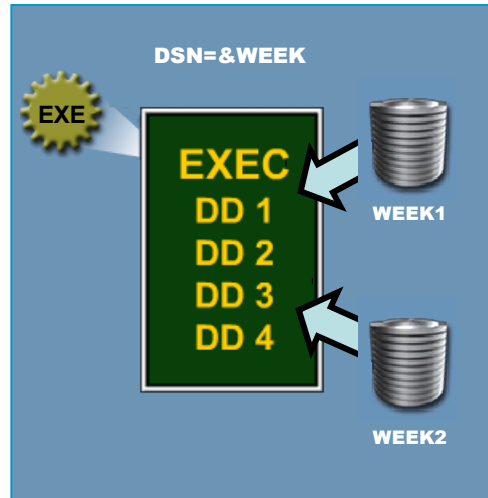
By simply assigning the appropriate data set name to &WEEK when the procedure is invoked, the data sets to be used for the other weeks can be specified.

DD statement parameters.

Assigning values.

If the same symbolic parameter is used to represent an parameter (such as the UNIT parameter) in several DD statements, then the value needs to be assigned to the symbolic parameter only once.

The value is applied to all occurrences of the parameter. The need to code several override DD statements can be eliminated.



DD statement parameters.

Rules for assigning values to DD statement parameters.

The rules for assigning values to or nullifying DD statement parameters are:

- **Separate the value assignments or nullifications from parameters and from each other using comas.**
- **Specify the symbolic parameter without the preceding ampersand, followed by an equal sign and value.**
- **To nullify a symbolic parameter specify the symbolic parameter without the preceding ampersand, followed by an equal sign. The value for the symbolic parameter should not be coded.**
- **Symbolic parameters can be nullified or assigned values in any sequence.**

67

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The rules that apply to assigning values to EXEC statement parameters also apply to DD statement parameters.

DD statements are assigned a value on the EXEC statement invoking the procedure. Moreover, if the same symbolic parameter is used to represent an parameter (such as UNIT) in several DD statements, the procedure user needs to assign a value only once. The value is applied to all occurrences of the parameter.

DD statement parameters.

Are we on track?

The following JCL statements are from a procedure named ANYPROC.

```
//S1 EXEC PGM=PROG1,ACCT=&GRP  
//DD1 DD DSN=&INPUT,DISP=SHR  
//...  
//S2 EXEC PGM=&A,ACCT=&GRP
```

Which of the following statements are correct to invoke the procedure and specify the following nullify the &GRP symbol, assign a value of MYDATA to &INPUT, assign a value of TSTPRG to &A.

- A. //JSTEP EXEC ANYPROG,GRP=,INPUT=MYDATA**
- B. //JSTEP EXEC ANYPROC,INPUT=MYDATA,A=TSTPRG,GRP=**
- C. //JSTEP EXEC ANYPROC,A=TSTPRG,GRP=,INPUT=MYDATA**
- D. //JSTEP EXEC ANYPROC,A=TSTPRG,INPUT=MYDATA**

The correct answer is B. and C.

DD statement parameters.

Assigning values to DD statement parameters.

While assigning values to symbolic parameters:

- **An appropriate value that represents a data set name must be assigned to the symbolic parameter.**
- **If the symbolic parameter is not assigned a value, the system considers the symbolic parameter to be the name of a temporary data set because it begins with the ampersand character.**

Assume the value to the symbolic parameter &WEEK in the previous example was not assigned.

When the procedure is invoked, the DD1 DD statement is assumed to address a temporary data set with the name &WEEK. This assumption will likely result in a JCL error message.

Submit the MCOE.EDU.JCL.JCL(TEMPDSN) to check that system really allocates the temporary data set beginning with SYS..., if you do not assign the temporary data set name.

DD statement parameters.

Are we on track?

If no value is assigned to a symbolic parameter in a procedure DD statement at the time of invoking a procedure, it is treated as a _____ data set.

70

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is Temporary.

Default values.

Recognizing default values.

At the time symbolic parameters are created, creator of the procedure can also assign default values to the parameters.

What are the characteristics of default values?

The characteristics of default values are:

- They are specified in the parameter field of the PROC statement.
- They do not have a preceding ampersand.
- In the case of multiple symbolic parameters, they are separated by a comma.
- They can be assigned in any sequence.

If the procedure user determine that the values are appropriate at the time the procedure is executed, then the values do not have to be assigned.

Default default values.

General rules and format.

At the time you create symbolic parameters, you can also assign default values. You do so on the PROC statement of the procedure. If the user determines that the values are appropriate at the time the procedure is executed, the user does not have to assign values.

The rules for specifying default values for symbolic parameters are as follows:

- **Specify default values for symbolic parameters in the operand field of the PROC statement. (A PROC statement must always precede an in-stream procedure definition and is required in a cataloged procedure only if you are assigning default values to symbolic parameters.)**
- **Specify each symbolic parameter without the preceding ampersand, followed by an equal sign and the default value.**
- **Separate the default value specifications for multiple symbolic parameters by a comma.**
- **You can assign default values for symbolic parameters in any sequence. You are not required to follow the sequence in which the symbolic parameters occur in the procedure definition.**

72

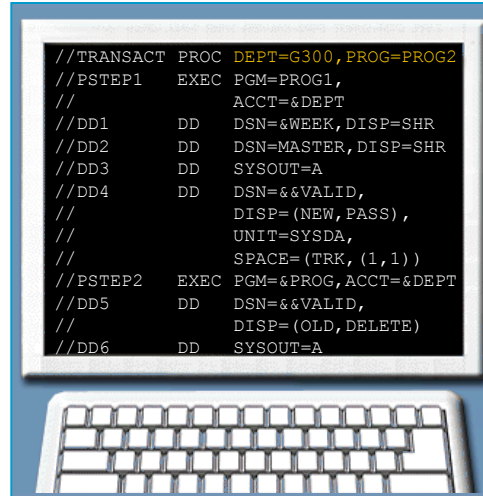
For example:

```
//TRANSACTION      PROC      DEPT=G300
//PSTEP1           EXEC      PGM=PROG1,ACCT=&DEPT
```

Default values.

Default values – an example.

In the TRANSACT procedure definition shown on the right, the PROC statement assigns the default value G300 to &DEPT and the default value PROG2 to &PROG.



```
//TRANSACT PROC DEPT=G300,PROG=PROG2
//PSTEP1 EXEC PGM=PROG1,
// ACCT=&DEPT
//DD1 DD DSN=&WEEK,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=&PROG,ACCT=&DEPT
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```

This would be useful if members of the same department (G300) were the primary users of the procedure and the program executed in step PSTEP2 was usually PROG2.

DD statement parameters.

Are we on track?

Which of the following statements are true for default values?

- A. They are assigned on the PROC statement of a procedure.**
- B. They are preceded by an ampersand.**
- C. They must be assigned in a specific sequence.**
- D. They can be overridden when the procedure is executed.**

74

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is A. and D.

Default values.

Accepting default values.

The EXEC statement required to accept the default values and specify the data set named WEEK1 is shown on the right.



75

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

You are not required to assign values to &DEPT and &PROG if their default values are appropriate for procedure invocation.

Since no default value is assigned to &WEEK, you must assign a value on the EXEC statement you code to invoke the procedure.

Default values.

Are we on track?

Code an EXEC statements to invoke the TRANSACT procedure and accept default values.

// EXEC _____

76

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is: TRANSACT

Default values.

Nullifying default values.

When should you assign a value?

A value for a symbolic parameter should be assigned (or nullified) only if:

- **The default values specified in the PROC statement is not appropriate for that particular execution of the procedure.**
- **No default value is specified for the symbolic parameter on the PROC statement in the procedure.**

Default values.

Nullifying default values.

The rules for overriding or nullifying the default values are:

- **To override a default value, specify the symbolic parameter without the preceding ampersand, followed by an equal sign and the appropriate value.**
- **To nullify a default value, specify the symbolic parameter without the preceding ampersand, followed by only an equal sign.**
- **The overrides and nullifications for default values can be specified in any sequence.**

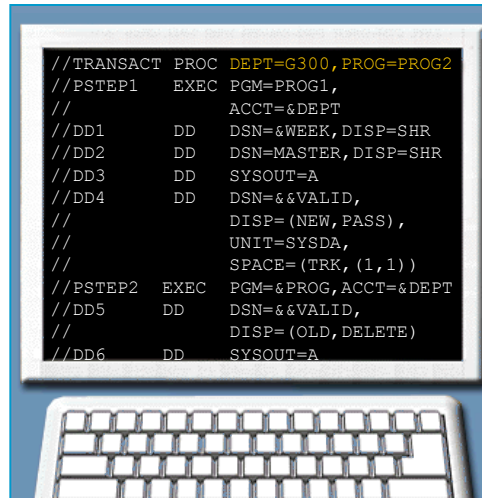
Default values.

Nullifying default values – procedure definition.

In this example, the TRANSACT procedure is to be invoked with some input test data (DD1 DD statement) that resides in a data set named TSTDATA and a test program named TESTPRG for step PSTEP2.

Because this execution is a test, a department number need not be specified.

Thus, the default values G300 (specified for &DEPT) and PROG2 (specified for &PROG) are not needed.



```
//TRANSACT PROC DEPT=G300,PROG=PROG2
//PSTEP1 EXEC PGM=PROG1,
// ACCT=&DEPT
//DD1 DD DSN=&WEEK,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW,PASS),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=&PROG,ACCT=&DEPT
//DD5 DD DSN=&&VALID,
// DISP=(OLD,DELETE)
//DD6 DD SYSOUT=A
```

Because this execution is a test, you can omit specifying a department number, which is only required for routine procedure use.

Thus, the default values G300 (specified for &DEPT) and PROG2 (specified for &PROG) are not appropriate for your needs.

Default values.

Nullifying default values – EXEC statement.

The EXEC statement for the invocation is shown on the right. The EXEC statement does the following:

- Nullifies the default values specified for &DEPT.
- Overrides the default value specified for &PROG and assigns the value TESTPRG instead.
- Assigns the value TSTDATA to &WEEK.



80

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The nullification, override and assignment of new values can be coded in any order on the EXEC statement used to invoke the procedure.

Default values.

Are we on track?

Which of the following statements are true of assigning default values for symbolic parameters?

- A. They are assigned on the EXEC statement that invokes the procedure.**
- B. You can assign default values for as many symbolic parameters as needed.**
- C. They can be nullified by the procedure user.**
- D. They are preceded by an ampersand {&}.**
- E. They are assigned on the PROC statement.**

81

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is B., C., and E.

Summary example.

Are we on track?

Which of the following should you include in the procedure to meet the needs of the processing environment? Choose all that apply.

- A. A symbolic parameter for PGM=in the first procedure step.**
- B. A symbolic parameter for PGM=in the second procedure step.**
- C. A SET statement.**
- D. A symbolic parameter for ACCT=in the EXEC statement.**
- E. Specific values for all data set names and attributes.**

82

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is B. and D.

Relates to the TRANSACT procedure on slide 79.

Summary example.

Are we on track?

Given that the data set names and attributes of the input data set are likely to vary widely, which of the following would best achieve the execution-time objectives?

- A. Use the DDNAME parameter to represent the input data set.**
- B. Code data specifications needed for typical use for the input data set.**
- C. Assign symbolic parameters for the input data set name and attributes.**

83

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is A.

Summary example.

Are we on track?

Complete the JCL statements for the sample procedure as per the following instructions:

In line 1, specify a symbolic parameter for ACCT in PSTEP1. Name it &DEPT.

In line 2, use the DDNAME parameter for the input data set. Code the value as INPUT.

In line 6, specify a symbolic parameter for PGM in PSTEP2. Name it &PROG2.

```
1. //PSTEP1 EXEC PGM=PROG1, _____
2. //DD1 DD _____
3. //DD2 DD DSN=MASTER, DISP=SHR
4. //DD3 DD SYSOUT=A
5. //DD4 DD DSN=&&VALID, DISP=(NEW, PASS), UNIT=SYSDA,
SPACE=(TRK, (1, 1))
6. //PSTEP2 EXEC _____
7. //DD5 DD DSN=&&VALID, DISP=(OLD, DELTE)
8. //DD6 DD SYSOUT=A
```

84

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is:

1. ACCT=&DEPT
2. DDNAME=INPUT
6. PGM=&PROG2

Summary example.

Are we on track?

Code the JCL used to invoke the sample procedure and provide the following execution-time information:

In line 1, code a department number of G300 and the program named TESTPRG for execution in PSTEP2

In line 2, code input transactions from cards in the job stream.

```
1. //JSTEP EXEC          TRANSACT, _____
2. //PSTEP1._____ *
   (input data records)
   /*
```

The correct answer is:

1. DEPT=G300,PROG2=TESTPRG
2. INPUT DD

Symbolic parameters.

Unit summary.

Now that you have completed this unit, you should be able to:

- Understand the purpose and form of the DDNAME parameter in a procedure definition.
- Resolve the DDNAME parameter when executing a procedure by identifying the actual data set names and attributes represented by the DDNAME parameter.
- Understand the purpose and form of symbolic parameters in a procedure definition.
- Assign values to symbolic parameters at the time of invoking a procedure.