



JCL

Chapter b4

Modifying DD 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

Modifying DD parameters.

Chapter b4

Modifying DD parameters

Modifying DD parameters.

Course objectives.

Be able to:

- **Override or nullify parameters on procedure step DD statements.**
- **Add DD statements to a procedure step if it does not already contain all the DD statements required for execution.**
- **Correctly sequence multiple temporary alterations to procedure step DD statements.**

Changing DD parameters.

Changing the DD statement parameters.

The procedure step DD statements specify the data sets and their characteristics for that procedure step.

If the specifications are not what are needed for a job, changes to the DD statement parameters can be made at the time of procedure execution.

Some of the changes to the DD statement parameters that can be accomplished at the time of execution include:

- Change a data set name or its storage location.
- Nullifying some of the specified parameters that are not applicable to the data set to be used.
- Add needed parameters that are not specified.

The general form for changing DD statement parameters is as follows:

```
//procstepname.ddname DD parameter=value
```



Changing DD parameters.

Important considerations while changing DD statement parameters.

```
//INVOKE          EXEC procname  
//procstepname.ddname DD parameter=value
```

**Procedure step where
the parameter occurs**

**Name of the DD statement
in the procedure step**

The DD statement has a two-part name. It consists of the name of the procedure step in which the DD statement to be changed occurs, followed by a period and the name of the DD statement in the procedure step.

The parameter to be changed, added or nullified is specified after the keyword DD, followed by an equal sign and the value of the parameter.

Changing DD parameters.

Changing DD parameters.

Why make the change?

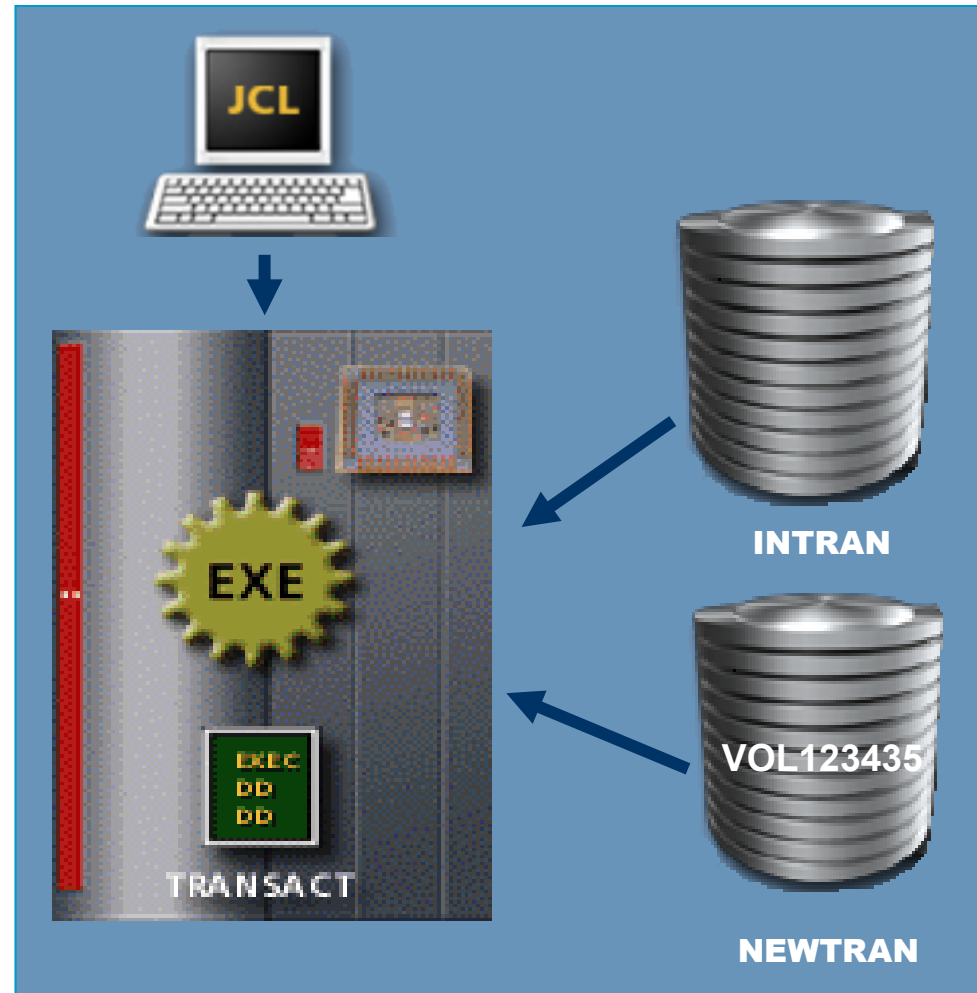
For example, you may want to execute a procedure using a different storage location for the output data set.

How to make the change?

Code an override DD statement immediately following the EXEC statement that executes the procedure.

The following override statement executes the TRANSACT procedure using NEWTRAN rather than INTRAN:

```
//PSTEP1.DD1 DD DSN=NEWTRAN,  
//                UNIT=3390,  
//                VOL=SER=123435
```



Changing DD parameters.

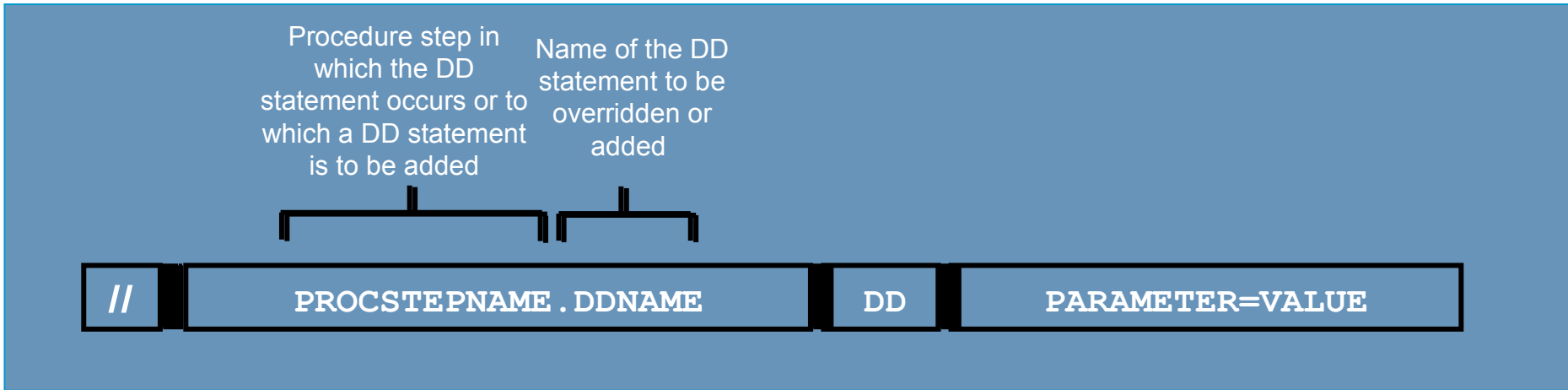
Rules for coding changes to DD parameter.

The rules for coding changes to DD statement parameters are as follows:

- **Code a special DD statement immediately following the EXEC statement to invoke the procedure.**
- **The DD statement has a two-part name:**
 - ✓ **The first part consists of the name of the procedure step where the DD parameter to be changed occurs or to which the DD statement is to be added, followed by a period.**
 - ✓ **The second part of the name is the DD statement to be overridden in the procedure step or the DDNAME defined for a data set to be added.**
- **Specify the parameter to be changed, added, or nullified, followed by an equal sign and the value of the parameter.**

Changing DD parameters.

General form for DD changing parameters.



How to code a stepname?

You can code the stepname in front of the DDNAME to override or to add. Notice that this sequence is the opposite of that used when coding EXEC statement overrides or additions.

The general form for DD additions, overrides, and nullifications is shown above.

Changing DD parameters.

Significance of DDname.

Why is DDname important?

The DDname on the submitted statement determines whether the system treats the statement as an addition or override.

If the DDname on the submitted statement matches a DDname within the procedure, the parameters on the procedure statement are overridden, as in the example shown on the right. If not, the statement is treated as an addition statement.

If a parameter in the procedure DD statement is matched by one in a submitted DD override statement, it is overridden. Otherwise it is retained.

TRANSACTION Procedure Definition

```
//PSTEP1      EXEC  PGM=PROG1,  
//              TIME=(1,30),  
//              PARM='01/29/99'  
//DD1         DD    DSN=INTRAN,  
//              DISP=SHR
```

Override Statement:

```
//PSTEP1.DD1  DD    DSN=NEWTRAN,  
//              UNIT=3390,  
//              VOL=SER=123435
```



Changing DD parameters.

Are we on track?

When invoking a procedure, you code changes to a procedure's data specifications on a DD statement that follows the EXEC statement. Which of the following items are included in the two-part name of that DD statement?

- A. The procedure step in which the parameter to be changed occurs.**
- B. The DDname of the statement in which the parameter to be changed occurs.**
- C. A period.**
- D. A parameter to be changed.**

Changing DD parameters.

Are we on track?

Following are the items you code in a DD statement to change procedure data specifications. Put them in the correct sequence.

- A. DD**
- B. Parameter to be changed.**
- C. DDname**
- D. Procedure step name followed by a period.**
- E. Value of the parameter to be changed.**
- F. //**
- G. An equal sign.**

Changing DD parameters.

Glossary.

DD statement parameters

Parameters such as DSN, RECFM, and DISP that are coded on a DD statement to give information necessary to read/write the data set.

Override DD statements.

Overriding DD statement parameters.

When the DD statement parameters that already exist in the procedure definition are not what you want for a particular job execution, you can code an override DD statement.

To override the DD statement parameters, the existing DD statement parameters in the procedure step need to be explicitly changed or nullified or a mutually exclusive parameter must be coded.

The general form for overriding DD statement parameters is:

```
//PROCSTEPNAME.DDNAME DD PARAMETER=VALUE
```


Override DD statements.

Coding an override DD statement.

When coding an override DD statement, the following considerations must be kept in mind:

- **Code the override DD statement immediately following the EXEC statement used to invoke the procedure.**
- **Supply a name for the override DD statement consisting of the name of the procedure step in which the DD statement to be overridden occurs, followed by a period and the name of the DD statement in the procedure step.**
- **Specify on the DD statement (in any sequence) the keyword parameters whose values are to be changed or nullified, separated by commas.**

Override DD statements.

Override DD statement – an example.

In the TRANSACT procedure, the input transactions are specified in a data set named INTRAN.

A particular week's input transactions reside in a data set named NEWTRAN.

NEWTRAN is an uncataloged data set that resides on a 3390 direct access volume whose volume identifier is 12345.



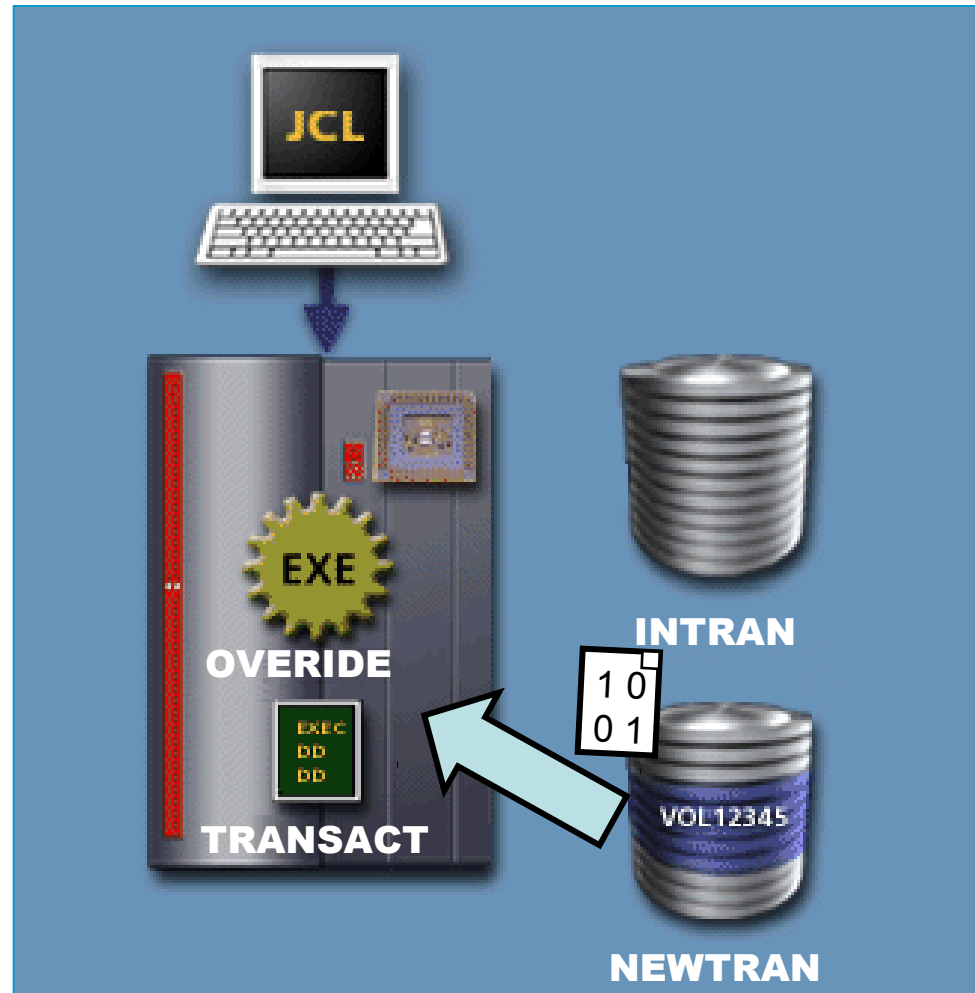
Override DD statements.

Override DD statement – an example.

In order to use the existing TRANSACT procedure to process that week's transactions a DD override statement that identifies NEWTRAN, rather than INTRAN, as the name of the transaction data set must be coded.

The override DD statement is as follows:

```
//PSTEP1.DD1 DD DSN=NEWTRAN,  
//          UNIT=3390,  
//          VOL=SER=12345
```



Override DD statements.

Override DD statements – procedure definition.

The code alongside shows the TRANSACT procedure definition.

The DD statement parameter that is to be overridden is highlighted.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//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, TIME=5
//DD5      DD  DSN=&&VALID,
//          DISP= (OLD, DELETE)
```

Override DD statements.

Override DD statements – EXEC statement.

The code alongside shows the EXEC statement used to invoke the TRANSACT procedure.

The override DD statement is coded immediately after the EXEC statement, specifying the procedure step containing the DD statement and name of the DD statement in the procedure, the new data set's name, and its location.

```
//STEP          EXEC TRANSACT
//PSTEP1.DD1 DD  DSN=NEWTRAN,
//              UNIT=3390,
//              VOL=SER=12345
```



Override DD statements.

Override DD statement – resulting JCL.

The code on the right shows the resulting JCL statement.

The data set to be used during execution has been changed to NEWTRAN.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//DD1      DD  DSN=NEWTRAN, DISP=SHR,
//          UNIT=3390,
//          VOL=SER=12345
//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, TIME=5
//DD5      DD  DSN=&&VALID,
//          DISP=(OLD, DELETE)
```



Override DD statements.

Are we on track?

Under which of the following conditions would you code an override DD statement?

- A. The Procedure step has no DD statements.**
- B. There is no DD statement with the ddname you need.**
- C. The data set attributes in a DD statement are not appropriate.**

Override DD statements.

Are we on track?

The following is a procedure called MYPROC

```
//STEP1 EXEC PGM=PROGA
//DD1 DD DSN=A, DISP=SHR
//DD2 DD DSN=OUTA, DISP=(NEW, CATLG) ,
// UNIT=SYSDA, SPACE=(TRK, (1, 1))
//STEP2 EXEC PGM=PROGB
//DD3 DD DSN=B, DISP=SHR
//DD4 DD DSN=RPT, DISP=(NEW, CATLG) ,
// UNIT=SYSDA, SPACE=(TRK, (1, 1))
```

Complete the following JCL statements so that when MYPROC is invoked, PROGB creates a SYSOUT class A output rather than write the output to the data set named RPT.

```
//JSTEP EXEC MYPROC
```


Nullification DD statements.

Nullifying a DD statement parameter.

To return the value of a DD statement parameter to the system default, the parameter must be nullified.

To nullify a DD statement parameter, you:

- Give the name of the parameter, followed by the equal sign.**
- Do not code a value for the parameter.**

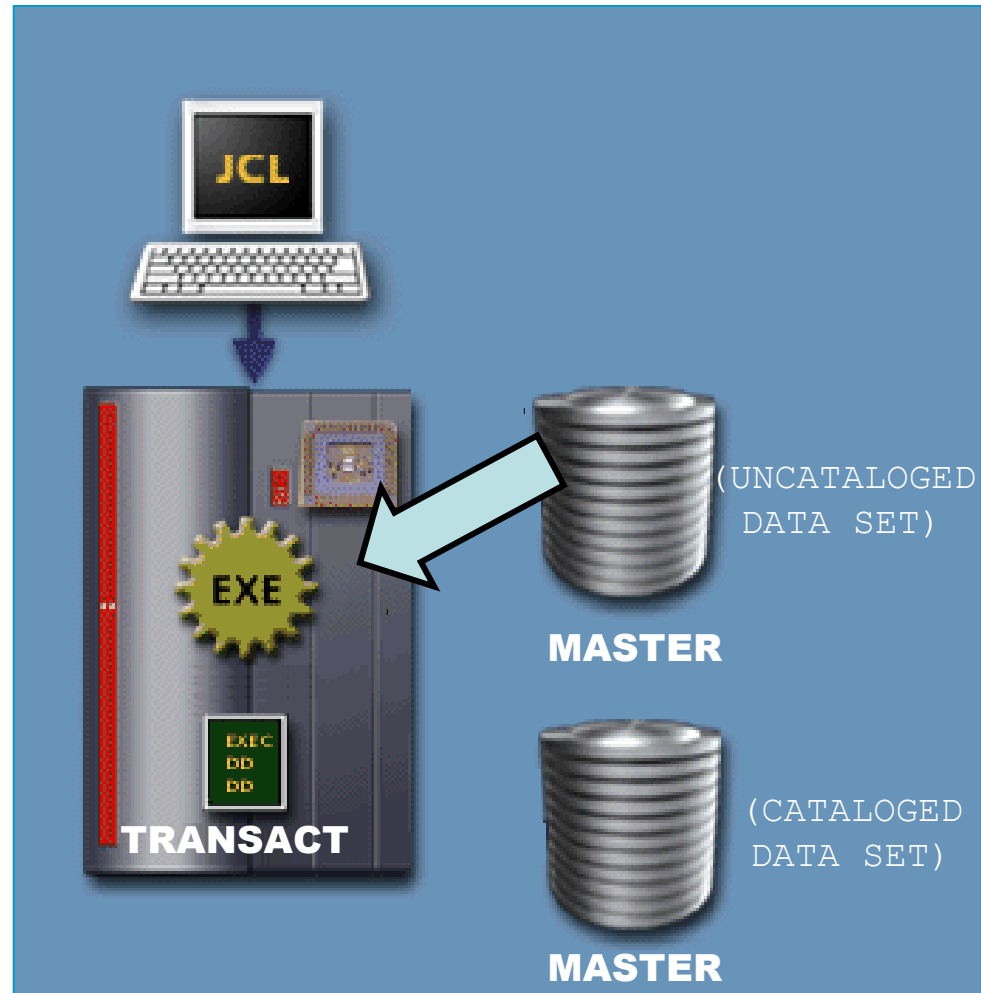
The general form for nullifying a DD statement parameter is:

//procstepname.ddname DD parameter=

Nullification DD statements.

Nullifying a DD statement parameter – an example.

In this example assume the TRANSACT procedure definition identifies data set MASTER as an uncataloged data set residing on a 3480 tape volume with a volume identifier of 987762.



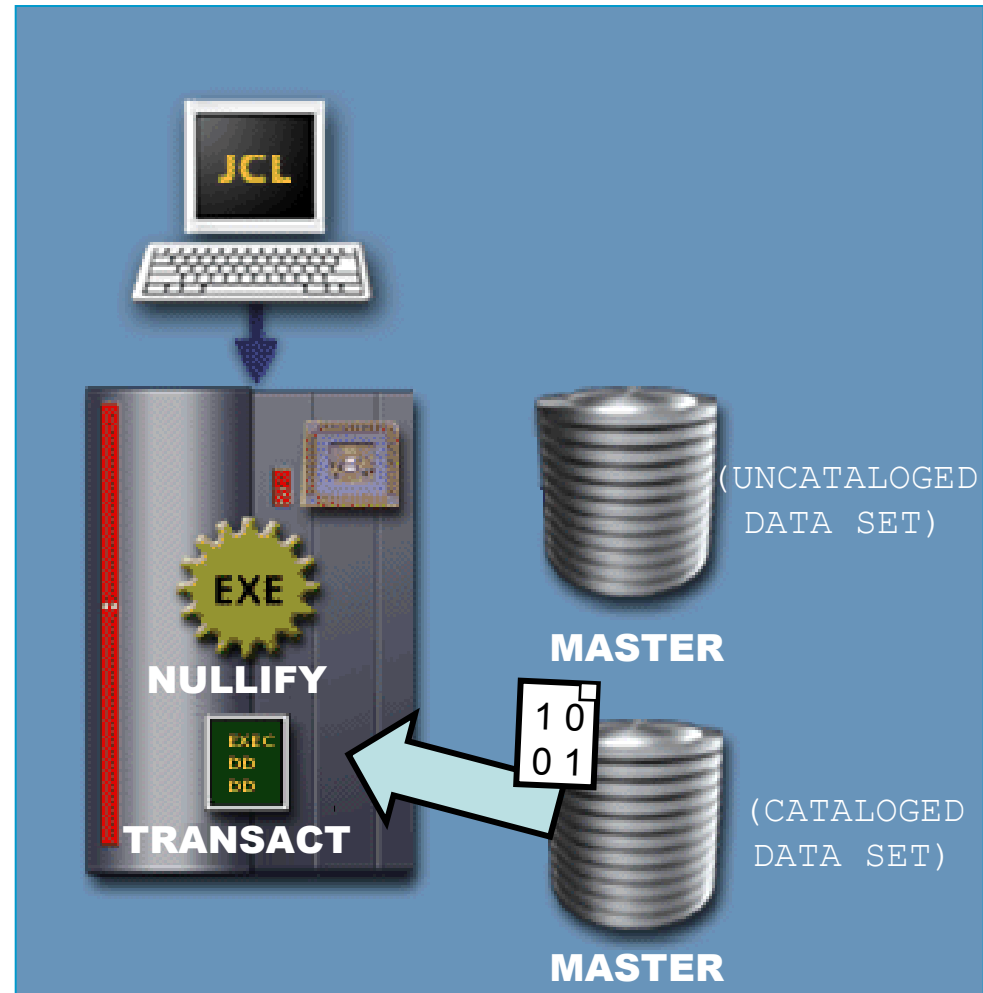
Nullification DD statements.

Nullifying a DD statement parameter – an example.

For a particular week, a previous version of MASTER (whose location is recorded in the system catalog) is required during the execution of TRANSACT.

An override DD statement to nullify the UNIT and VOL=SER parameters and referring to the uncataloged MASTER data set would be coded as follows:

```
//procstepname.ddanme DD UNIT=,  
//                               VOL=SER=
```



Nullification DD statements.

Nullifying a parameter – procedure definition.

The TRANSACT procedure definition is shown on the right.

The DD statement parameter that is to be overridden is highlighted.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//DD1      DD  DSN=NEWTRAN, DISP=SHR
//DD2      DD  DSN=MASTER, DISP=SHR,
//          UNIT=3480,
//          VOL=SER=987762
//DD3      DD  SYSOUT=A
//DD4      DD  DSN=&&VALID,
//          DISP= (NEW, PASS) ,
//          UNIT=SYSDA,
//          SPACE= (TRK, (1, 1) )
//PSTEP2 EXEC PGM=PROG2, TIME=5
//DD5      DD  DSN=&&VALID,
//          DISP=(OLD, DELETE)
```



Nullification DD statements.

Nullifying a parameter – EXEC statement.

The EXEC statement used to invoke the TRANSACT procedure is shown on the right.

The override DD statement is coded immediately after the EXEC statement, specifying the procedure step containing the DD statement and name of the DD statement in the procedure.

The override DD statement nullifies the UNIT and VOL parameters.



```
//STEP          EXEC TRANSACT  
//PSTEP1.DD2 DD  UNIT=,VOL=SER=
```

Nullification DD statements.

Nullifying a parameter – resulting JCL.

The resulting JCL statement is shown on the right.

The data set to be used during execution has been changed to default value i.e. the cataloged MASTER data set.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//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, TIME=5
//DD5 DD DSN=&&VALID,
// DISP=(OLD, DELETE)
```

Nullification DD statements.

Are we on track?

The following is a procedure called MYPROC

```
//STEP1 EXEC PGM=PROGA
//DD1 DD DSN=A, DISP=SHR
//DD2 DD DSN=OUTA, DISP=(NEW, CATLG) ,
// UNIT=SYSDA, SPACE=(TRK, (1, 1))
//STEP2 EXEC PGM=PROGB
//DD3 DD DSN=B, DISP=SHR
//DD4 DD DSN=RPT, DISP=(NEW, CATLG) ,
// UNIT=SYSDA, SPACE=(TRK, (1, 1))
```

Code statements to override the UNIT parameter to 3480 and nullify the SPACE parameter in STEP2.

```
//JSTEP EXEC MYPROC _____
```

Nullification DD statements.

DUMMY parameter.

A DUMMY parameter is used to bypass a data set, without resulting in a program ABEND.

A DUMMY parameter may be specified in a procedure step DD statement. If you require to execute the procedure with real data, the DUMMY parameter must be overridden with the name of the data set to be used.

Nullification DD statements.

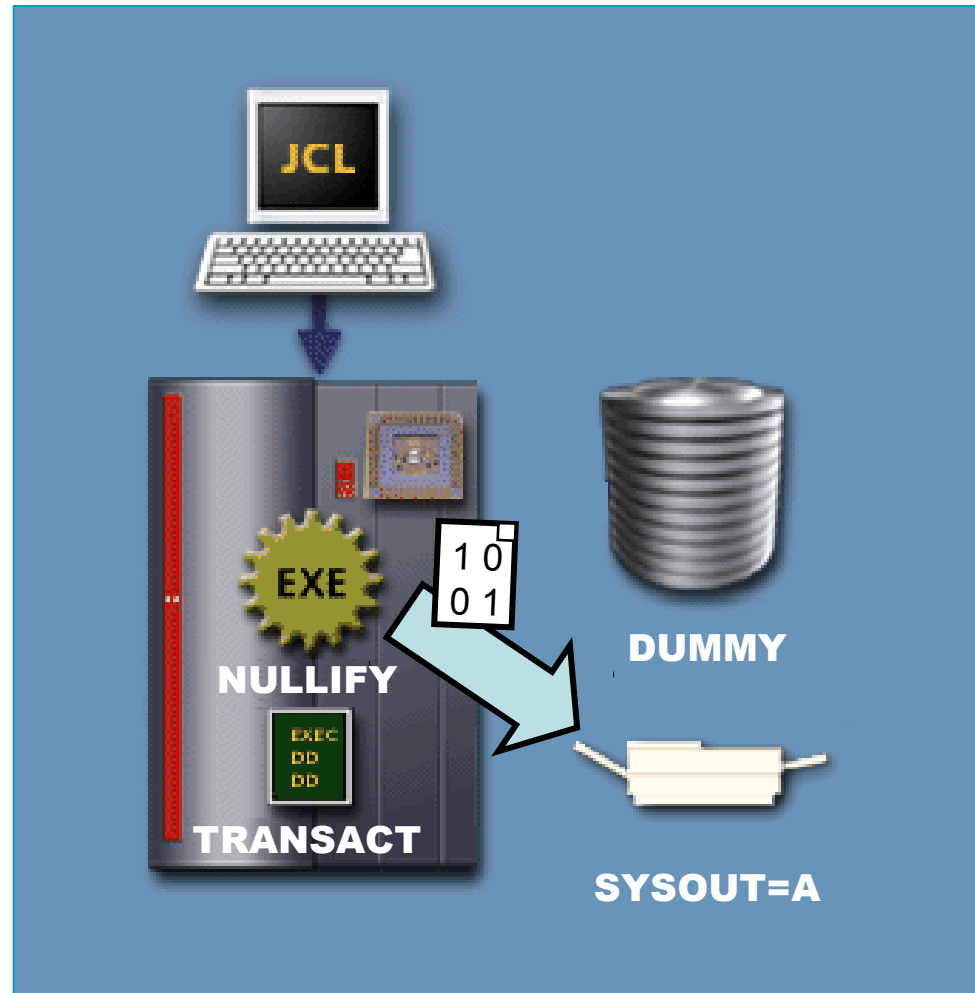
Nullifying the DUMMY parameter – an example.

In this example, a procedure step named PSTEP1 contains the following DD statement to identify the output data set:

```
//DDOUT DD DUMMY
```

If a SYSOUT class A output has to be created for this invocation then the following DD statement needs to be coded:

```
//PSTEP1.DDOUT DD SYSOUT=A
```



Nullification DD statements.

Nullifying the DSN parameter.

If a procedure is to be executed without using a data set specified on a procedure step DD statement, a dummy status can be assigned to the data set.

The override DD statement can either specify DUMMY or DSN=NULLFILE.

```
//procstepname.ddname DD DUMMY
```

or

```
//procstepname.ddname DD DSN=NULLFILE
```

When DUMMY or DSN=NULLFILE is coded on an override DD statement, it nullifies all parameters specified on the corresponding DD statement in the procedure definition except the DCB parameter.

Nullification DD statements.

Nullifying the DSN parameter – procedure definition.

The TRANSACT procedure definition is shown on the right.

The DD statement parameter that is to be overridden is highlighted.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//DD1      DD  DSN=INTRAN, DISP=SHR
//DD2      DD  DSN=MASTER, DISP=SHR,
//          UNIT=3480,
//          VOL=SER=98762
//DD3      DD  SYSOUT=A
//DD4      DD  DSN=&&VALID,
//          DISP=(NEW, PASS),
//          UNIT=SYSDA,
//          SPACE=(TRK, (1, 1))
//PSTEP2 EXEC PGM=PROG2, TIME=5
//DD5      DD  DSN=&&VALID,
//          DISP=(OLD, DELETE)
```



Nullification DD statements.

Nullifying the DSN parameter – EXEC statement.

The EXEC statement used to invoke the TRANSACT procedure is shown on the right.

The override DD statement nullifies the DSN parameter and any other parameters associated with that data set.



```
//STEP EXEC TRANSACT
//PSTEP1.DD2 DD DUMMY
```

Nullification DD statements.

Nullifying the DSN parameter – resulting JCL.

The resulting JCL statement is shown on the right.

During execution of the procedure, the system will not consider the data set that was designated by DD2.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//DD1      DD  DSN=INTRAN, DISP=SHR
//DD2      DD  DUMMY
//DD3      DD  SYSOUT=A
//DD4      DD  DSN=&&VALID,
//          DISP=(NEW, PASS) ,
//          UNIT=SYSDA,
//          SPACE=(TRK, (1, 1) )
//PSTEP2 EXEC PGM=PROG2, TIME=5
//DD5      DD  DSN=&&VALID,
//          DISP=(OLD, DELETE)
```

Nullification DD statements.

Are we on track?

Which one of the following is the result of coding DUMMY or DSN=NULLFILE on an override DD statement?

- A. The procedure is executed with a dummy data set.**
- B. All parameters on the corresponding DD statement except DCB are nullified.**
- C. The procedure is executed without the data set.**
- D. All parameters on the corresponding procedure DD statement are nullified.**

Nullification DD statements.

DCB subparameters.

The data control block (DCB) unlike other parameters, is not overridden in its entirety, but by subparameter.

DCB subparameters must be nullified explicitly. You code the DD override statement with the parameters to be overridden in parentheses. The others are retained, as shown on the right. To nullify the entire DCB parameter, you must explicitly nullify each subparameter as shown below:

```
//      DCB= (BUFNO= ,BLKSIZE= ,  
//      RECFM= ,BUFL=)
```

Original Parameters:

```
//DD1 DD DSN=MYDSET,DISP=SHR,  
//      DCB= (BUFNO=1,BLKSIZE=800,  
//      RECFM=FB,BUFL=800)
```

DCB Override Statement:

```
//PSTEP1.DD1 DD DCB= (BLKSIZE=320,  
//      BUFL=320)
```

Resulting Parameters:

```
//DD1 DD DSN..  
//      DCB= (BUFNO=1,BLKSIZE=320,  
//      RECFM=FB,BUFL=320)
```



Nullification DD statements.

Consideration while nullifying DCB parameters.

Some special rules apply when overriding or nullifying the DCB parameter:

- **Code only those DCB keyword subparameters whose values need to be changed. DCB keyword subparameters which are not coded remain unchanged.**
- **Code any required DCB positional subparameters regardless of whether or not they are specified on the DD statement in the procedure definition.**
- **To nullify an existing DCB positional subparameter simply omit it from the DCB parameter given in the override DD statement.**
- **To nullify the DCB parameter completely, omit all existing positional subparameters and explicitly nullify each existing keyword subparameter.**
- **Positional subparameters (such as data set name) are essential to the system. These must be coded, even if they are specified in the procedure definition.**
- **Keyword subparameters (such as BLKSIZE) supply additional information to the operating system. If they are specified in the procedure definition, then code only those that are to be changed.**



Nullification DD statements.

Nullifying the DCB parameter – example 1.

This is a DD statement in a procedure step called PSTEP1:

```
//DD1 DD DSN=MYDSET,DISP=SHR,  
//          DCB=(BUFNO=1,BLKSIZE=800,RECFM=FB,BUFL=800)
```

If the block size and buffer length needs to be changed to 320 at the time of execution, then the following override DD statement needs to be coded:

```
//PSTEP1.DD1 DD DCB=(BLKSIZE=320,BUFL=320)
```

Nullification DD statements.

Nullifying the DCB parameter – example 2.

This is a DD statement in a procedure step called PSTEP1:

```
//DD2 DD DSN=MYDSET,DISP=SHR,DCB=(DS1,BLKSIZE=80)
```

If the block size needs to be changed to 400 and the remaining DCB information has to be copied from a cataloged data set (DS1) at the time of execution, then the following override DD statement needs to be coded:

```
//PSTEP1.DD2 DD DCB=(DS1,BLKSIZE=400)
```

Nullification DD statements.

Nullifying the DCB parameter – example 3.

This is a DD statement in a procedure called PSTEP1:

```
//DD3 DD DSN=MYDSET,DISP=SHR,  
//      DCB=(RECFM=FB,BLKSIZE=160,LRECL=80)
```

In order to nullify the entire DCB parameter of the DD statement, each DCB keyword subparameter specified in the procedure definition must be nullified as shown below:

```
//PSTEP1.DD3 DD DCB=(RECFM=,BLKSIZE=,LRECL=)
```

Nullification DD statements.

Are we on track?

Which of the following rules apply to coding DD override statements for the DCB parameter?

- A. Code only those keyword subparameters you want to change.**
- B. To nullify, omit positional parameters and nullify all keyword parameters.**
- C. Specify any positional subparameters you want to retain.**
- D. To nullify, code DCB=NULLFILE.**

Nullification DD statements.

Are we on track?

A DD statement named DD1 in a procedure step called PSTEP2 has the following DCB parameter

```
//      DCB=(DS1, BLKSIZE=400, RECFM=FB, BUFL=400)
```

Complete the following JCL statements to change the block size and buffer length to 320 and copy other DCB information from data set DS2.

```
//PSTEP2.DD1 DD DCB=_____
```

Nullification DD statements.

Glossary.

DCB subparameters

A variable placed within parenthesis within the DCB parameter, to specify information such as block size or record format.

Keyword subparameters

This is a list of variable information contained within parentheses, and followed by an equal sign and value.

Positional subparameters

This is a list of variable information coded in parentheses.

Addition DD statements.

Addition DD statements.

If a required data set is not found in the procedure definition then an additional DD statement can be coded along with the invocation of the procedure.

The general form is same as an override DD statement:

```
//prostepname.ddname DD parameter=value
```

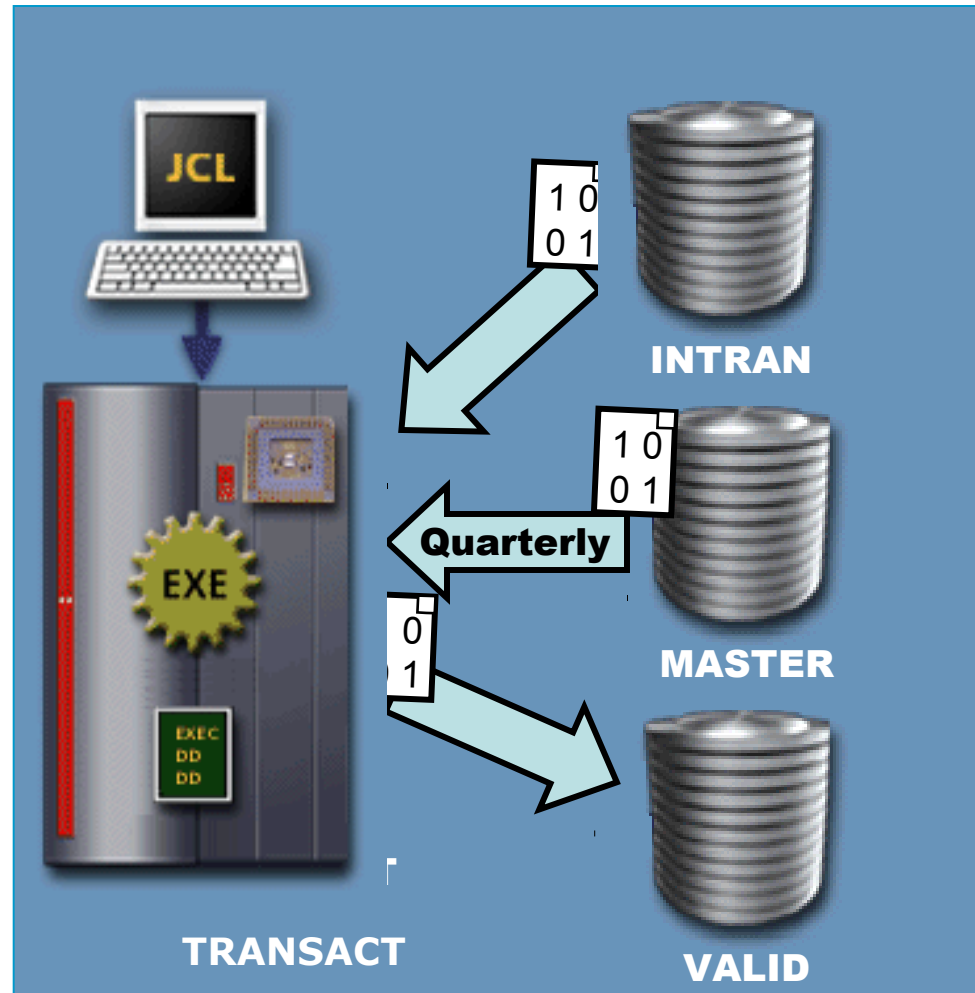
Addition DD statements.

Adding DD statements – an example.

In this example, the TRANSACT procedure is used to process customer orders.

Every quarter the transaction input file (data set INTRAN) has to be checked against the master customer list (data set MASTER) to ensure that each transaction has valid customer.

During the routine weekly execution the check is not made, and data set MASTER is not used.



Addition DD statements.

Adding DD statements – procedure definition.

The TRANSACT procedure definition is shown on the right.

In this procedure:

- The PARM value of NOCHECK is specified in the PSTEP1 EXEC statement.
- PSTEP1 does not include a reference to data set MASTER.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30),  
//          PARM=NOCHECK  
//DD1      DD DSN=INTRAN, DISP=SHR  
//DD3      DD SYSOUT=A  
//DD4      DD DSN=&&VALID,  
//          DISP= (NEW, PASS),  
//          UNIT=SYSDA,  
//          SPACE= (TRK, (1, 1))  
//PSTEP2 EXEC PGM=PROG2, TIME=5  
//DD5      DD DSN=&&VALID,  
//          DISP= (OLD, DELETE)  
//DD6      DD SYSOUT=A
```



Addition DD statements.

Adding DD statements – EXEC statement.

To execute the procedure routinely every week without making the check, you would code the following EXEC statement to invoke the procedure:

```
//JSTEP EXEC TRANSACT
```

The code on the right represents the effective JCL for this particular job.

```
//PSTEP1 EXEC PGM=PROG1,TIME(1,30),  
//          PARM=NOCHECK  
//DD1      DD DSN=INTRAN,DISP=SHR  
//DD3      DD SYSOUT=A  
//DD4      DD DSN=&&VALID,  
//          DISP=(NEW,PASS),  
//          UNIT=SYSDA,  
//          SPACE=(TRK,(1,1))  
//PSTEP2 EXEC PGM=PROG2,TIME=5  
//DD5      DD DSN=&&VALID,  
//          DISP=(OLD,DELETE)  
//DD6      DD SYSOUT=A
```



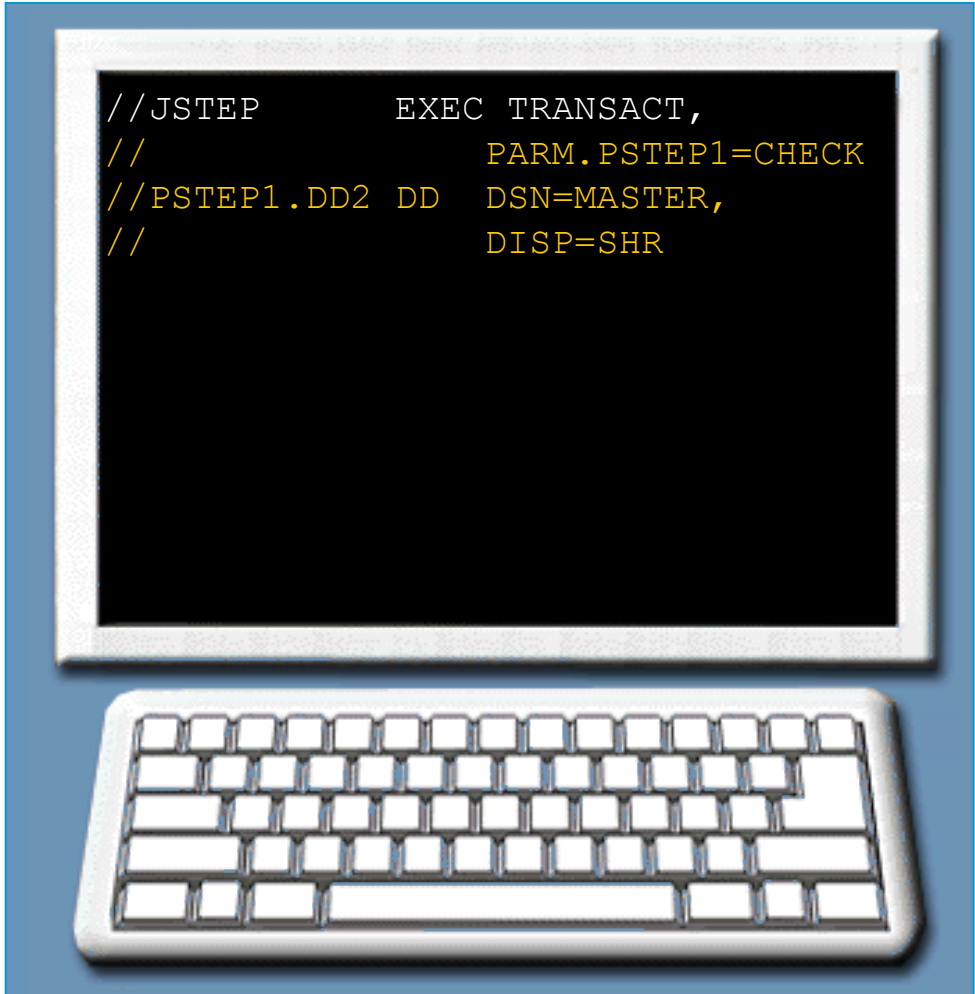
Addition DD statements.

Addition DD statements – EXEC statement.

For the quarterly execution the transaction input file has to be checked against the master customer list.

To perform the required quarterly check, two statements are required to invoke the procedure:

- An EXEC statement override that includes a PARM parameter value of CHECK to override the value of NOCHECK.
- An addition DD statement to identify data set MASTER that PROG1 needs to access to perform the check.



```
//JSTEP      EXEC TRANSACT,  
//          PARM.PSTEP1=CHECK  
//PSTEP1.DD2 DD  DSN=MASTER,  
//          DISP=SHR
```

Addition DD statements.

Addition DD statements – effective JCL.

The resulting JCL statement is shown on the right.

The PARM value has been overridden to CHECK and an addition statement to identify data set MASTER has been specified.

```
/PSTEP1 EXEC PGM=PROG1, TIME=(1, 30),  
/  
          PARM=CHECK  
/DD1     DD  DSN=INTRAN, DISP=SHR  
/DD3     DD  SYSOUT=A  
/DD4     DD  DSN=&&VALID,  
/  
          DISP=(NEW, PASS),  
/  
          UNIT=SYSDA,  
/  
          SPACE=(TRK, (1, 1))  
/ /DD2     DD  DSN=MASTER, DISP=SHR  
/PSTEP2 EXEC PGM=PROG2, TIME=5  
/DD5     DD  DSN=&&VALID,  
/  
          DISP=(OLD, DELETE)
```



Addition DD statements.

Are we on track?

Under which one of the following conditions would you code an addition DD statement?

A. The ddname you need is not specified.

B. The ddname is specified, but the attributes are not appropriate.

C. The procedure step has a DD statement with the DUMMY parameter.

Addition DD statements.

Are we on track?

The following is a procedure called ANYPROC

```
//PSTEPA EXEC PGM=PROG1  
//DD1      DD  DSN=DATFIL,DISP=SHR  
//DD2      DD  SYSOUT=A  
//PSTEPB EXEC PGM=PROG2  
//DD3      DD  SYSOUT=A
```

Code an addition statement for ANYPROC. Specify, for PROG2, a data set named INDATA. PROG2 refers to the data set by the ddname TEST.

```
//JSTEP EXEC ANYPROC
```

Sequencing multiple parameters.

Coding multiple parameters.

Any number of override and addition DD statements can be coded to invoke a procedure.

The rules for coding multiple parameters are:

- **Code the EXEC statement to invoke the procedure.**
- **Code the override statements (if any) for a step in the same DDname sequence as in the procedure definition.**
- **Code addition DD statements (if any) for that step, following the override statements.**
- **Within each procedure step, the override statements must be coded in DDname sequence.**

Sequencing multiple parameters.

Sequencing multiple parameters – an example.

The TRANSACT procedure definition is shown on the right. The following changes need to be made:

- PROG1 obtains its input transactions from a cataloged data set named MYDATA, instead of from INTRAN.
- PROG1 checks the contents of MYDATA against the contents of a second cataloged data set named CKDATA. The PARM value of NOCHECK needs to be overridden with the value CHECK, and a DD statement named DD2 to reference CKDATA has to be added.
- PROG2 writes its output (using ddname DD6) to a data set named INVOICE on a 3390 volume with volume identifier 6929L. The data set requires 10 tracks of space and should be cataloged.

```
//PSTEP1 EXEC PGM=PROG1,TIME(1,30),  
//          PARM=NOCHECK  
//DD1      DD DSN=INTRAN,DISP=SHR  
//DD3      DD SYSOUT=A  
//DD4      DD DSN=&&VALID,  
//          DISP=(NEW,PASS),  
//          UNIT=SYSDA,  
//          SPACE=(TRK,(1,1))  
//PSTEP2 EXEC PGM=PROG2,TIME=5  
//DD5      DD DSN=&&VALID,  
//          DISP=(OLD,DELETE)  
//DD6      DD SYSOUT=A
```



Sequencing multiple parameters.

Are we on track?

Refer to the previous example. Which of the following JCL statements would you need when you invoke TRANSACT?

- A. An EXEC statement with a PARM parameter override.**
- B. An override DD statement for PSTEP2.DD6.**
- C. An addition DD statement for PSTEP1.**
- D. An override DD statement for PSTEP1.DD1.**

Sequencing multiple parameters.

Sequencing rules – an example.

The JCL used to invoke TRANSACT is shown on the right.

- The PARM parameter value has been overridden to CHECK.
- The override for DD1 in step PSTEP1 is specified.
- The addition statement DD2 for step PSTEP1 is specified.
- The override statement for DD6 in step PSTEP2 is specified.

```
//JSTEP      EXEC TRANSACT,  
//          PARM.PSTEP1=CHECK  
//PSTEP1.DD1 DD  DSN=MYDATA  
//PSTEP1.DD2 DD  DSN=CKDATA,  
//          DISP=SHR  
//PSTEP2.DD6 DD  DSN=INVOICE,  
//          DISP=(NEW,CATLG),  
//          UNIT=3390,  
//          VOL=SER=6929L,  
//          SPACE=(TRK,10)
```

Sequencing multiple parameters.

Shorthand form for multiple override statements.

Shorthand form of coding multiple override and addition DD statements can be used in the procedure step sequence.

To code the shorthand form:

- Follow the general form for the first override or addition statement.
- For subsequent override or addition statements in that procedure step use

//ddname DD . . .

```
//JSTEP      EXEC TRANSACT,  
//          PARM.PSTEP1=CHECK  
//PSTEP1.DD1 DD  DSN=MYDATA  
//DD2       DD  DSN=CKDATA,  
//          DISP=SHR  
//PSTEP2.DD6 DD  DSN=INVOICE,  
//          DISP=(NEW,CATLG),  
//          UNIT=3390,  
//          VOL=SER=6929L,  
//          SPACE=(TRK,10)
```



Sequencing multiple parameters.

Are we on track?

The procedure named WEEKLY is shown below.

```
//S1 EXEC PGM=PROCESS,TIME=(1,30)
//DD1 DD DSN=MYDATA,DISP=SHR
      ...
//S2 EXEC PGM=BILL
//DD3 DD DSN=&&TEMP,DISP=(NEW,DELETE)
      ...
```

Code the EXEC statement required to pass:

- a. PARM value of YES to the program named PROCESS.**
- b. Impose a CPU time limit of 3 minutes for execution of the program named BILL.**
- c. Use the installation-define time default for the program named PROCESS.**

//JSTEP EXEC _____

Sequencing multiple parameters.

Are we on track?

The procedure named WEEKLY is shown below.

```
//S1 EXEC PGM=PROCESS, TIME=(1, 30)
//DD1 DD DSN=MYDATA, DISP=SHR
      ...
//S2 EXEC PGM=BILL
//DD3 DD DSN=&&TEMP, DISP=(NEW, DELETE)
      ...
```

Complete the DD override statement to specify that PROCESS obtain its input from a cataloged data set named NEWDATA.

_____ DD DSN=NEWDATA

Sequencing multiple parameters.

Are we on track?

Review the following DD statements from STEPC of a sample procedure:

```
. . .  
//DD4 DD DSN=A.B.C...  
//DD5 DD DSN=STRP...  
//DD6 DD DSN=TYPE3...  
//DD7 DD DSN=A.B.D...
```

In this exercise, code the following override DD statements using the shorthand form:

Code an override statement to specify a data set named TEST1 instead of A.B.C

//STEPC.DD4 DD _____

Code an override statement to specify a data set named TESTDATA rather than STRP

//DD5 DD _____

Code an override statement to specify a data set named TEST2 instead of A.B.D

//DD7 DD _____

Sequencing multiple parameters.

Are we on track?

Place the following override and addition DD statements in the order in which they must be specified. (Assume procedure DD statements are in alphanumeric order.)

- A. An override DD statement for DD6 in PSTEP3**
- B. An override DD statement for DD2 in PSTEP1**
- C. An addition DD statement for DD3 in PSTEP1**
- D. An addition DD statement for DD5 in PSTEP3**
- E. An override DD statement for DD4 in PSTEP1**

Modifying DD parameters.

Unit summary.

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

- **Override or nullify parameters on procedure step DD statements.**
- **Add DD statements to a procedure step if it does not already contain all the DD statements required for execution.**
- **Correctly sequence multiple temporary alterations to procedure step DD statements.**

JCL

Chapter b4 Modifying DD 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

Modifying DD parameters.

Chapter b4

Modifying DD parameters

5

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

Modifying DD parameters.

Course objectives.

Be able to:

- **Override or nullify parameters on procedure step DD statements.**
- **Add DD statements to a procedure step if it does not already contain all the DD statements required for execution.**
- **Correctly sequence multiple temporary alterations to procedure step DD statements.**

Changing DD parameters.

Changing the DD statement parameters.

The procedure step DD statements specify the data sets and their characteristics for that procedure step.

If the specifications are not what are needed for a job, changes to the DD statement parameters can be made at the time of procedure execution.

Some of the changes to the DD statement parameters that can be accomplished at the time of execution include:

- Change a data set name or its storage location.
- Nullifying some of the specified parameters that are not applicable to the data set to be used.
- Add needed parameters that are not specified.

The general form for changing DD statement parameters is as follows:

```
7//procstepname.ddname DD parameter=value
```

To change the DD statement parameter when invoking a procedure, the DD statement must be coded immediately following the EXEC statement.

Changing DD parameters.

Important considerations while changing DD statement parameters.

```
//INVOKE EXEC procname  
//procstepname.ddname DD parameter=value
```

The diagram shows two yellow boxes with black text. The first box, labeled 'Procedure step where the parameter occurs', has a yellow arrow pointing to the 'procstepname' part of the DD statement. The second box, labeled 'Name of the DD statement in the procedure step', has a yellow arrow pointing to the 'ddname' part of the DD statement.

The DD statement has a two-part name. It consists of the name of the procedure step in which the DD statement to be changed occurs, followed by a period and the name of the DD statement in the procedure step.

The parameter to be changed, added or nullified is specified after the keyword DD, followed by an equal sign and the value of the parameter.

For nullifying a parameter the value is not coded.

Changing DD parameters.

Changing DD parameters.

Why make the change?

For example, you may want to execute a procedure using a different storage location for the output data set.

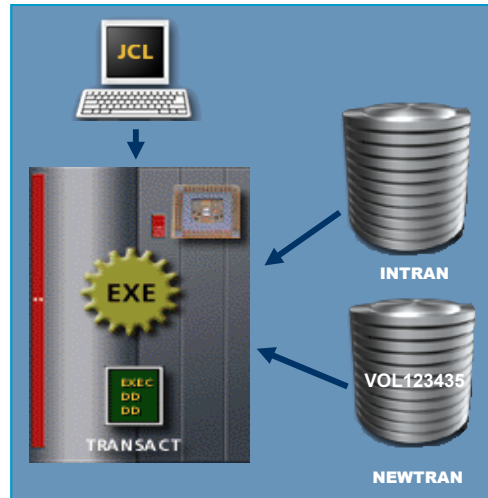
How to make the change?

Code an override DD statement immediately following the EXEC statement that executes the procedure.

The following override statement executes the TRANSACT procedure using NEWTRAN rather than INTRAN:

```
//PSTEP1.DD1 DD DSN=NEWTRAN,  
//                UNIT=3390,  
9/                VOL=SER=123435
```

Copyright © 2006 CA. All trademarks herein belong to their respective companies.



Notice that the coding sequence (STEPname.DDname) is opposite of that used when coding EXEC statement (PARMname.STEPname).

Changing DD parameters.

Rules for coding changes to DD parameter.

The rules for coding changes to DD statement parameters are as follows:

- **Code a special DD statement immediately following the EXEC statement to invoke the procedure.**
- **The DD statement has a two-part name:**
 - ✓ **The first part consists of the name of the procedure step where the DD parameter to be changed occurs or to which the DD statement is to be added, followed by a period.**
 - ✓ **The second part of the name is the DD statement to be overridden in the procedure step or the DDNAME defined for a data set to be added.**
- **Specify the parameter to be changed, added, or nullified, followed by an equal sign and the value of the parameter.**

10

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

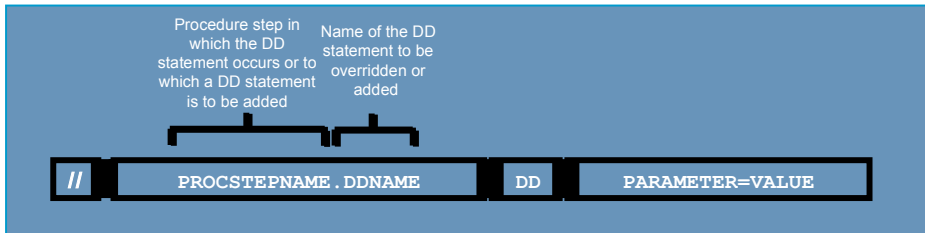
The name of an addition DD statement consists of the name of the procedure step to which the DD statement is to be added and the DDname defined for the data set in the program.

The name of an override DD statement consists of the name of the procedure step in which the DD parameter to be changed occurs and the DD statement to be overridden.

If you are nullifying a parameter, do not code a value.

Changing DD parameters.

General form for DD changing parameters.



How to code a stepname?

You can code the stepname in front of the DDNAME to override or to add. Notice that this sequence is the opposite of that used when coding EXEC statement overrides or additions.

The general form for DD additions, overrides, and nullifications is shown above.

Changing DD parameters.

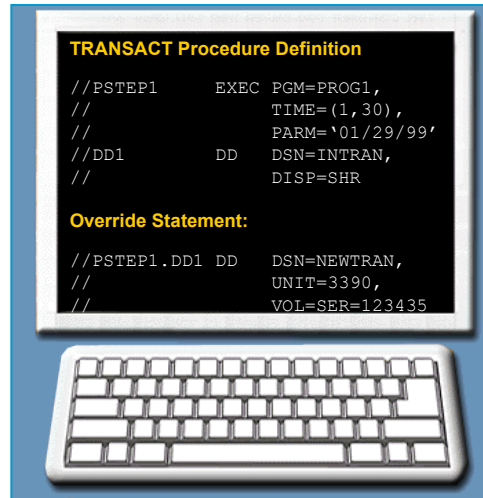
Significance of DDname.

Why is DDname important?

The DDname on the submitted statement determines whether the system treats the statement as an addition or override.

If the DDname on the submitted statement matches a DDname within the procedure, the parameters on the procedure statement are overridden, as in the example shown on the right. If not, the statement is treated as an addition statement.

If a parameter in the procedure DD statement is matched by one in a submitted DD override statement, it is overridden. Otherwise it is retained.



12

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

DD statements are added in their entirety. Overridden statements are modified parameter by parameter.

Changing DD parameters.

Are we on track?

When invoking a procedure, you code changes to a procedure's data specifications on a DD statement that follows the EXEC statement. Which of the following items are included in the two-part name of that DD statement?

- A. The procedure step in which the parameter to be changed occurs.**
- B. The DDname of the statement in which the parameter to be changed occurs.**
- C. A period.**
- D. A parameter to be changed.**

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

Changing DD parameters.

Are we on track?

Following are the items you code in a DD statement to change procedure data specifications. Put them in the correct sequence.

- A. DD**
- B. Parameter to be changed.**
- C. DDname**
- D. Procedure step name followed by a period.**
- E. Value of the parameter to be changed.**
- F. //**
- G. An equal sign.**

14

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

The correct order is F., D., C., A., B, G, E.

Changing DD parameters.

Glossary.

DD statement parameters

Parameters such as DSN, RECFM, and DISP that are coded on a DD statement to give information necessary to read/write the data set.

Override DD statements.

Overriding DD statement parameters.

When the DD statement parameters that already exist in the procedure definition are not what you want for a particular job execution, you can code an override DD statement.

To override the DD statement parameters, the existing DD statement parameters in the procedure step need to be explicitly changed or nullified or a mutually exclusive parameter must be coded.

The general form for overriding DD statement parameters is:

```
//PROCSTEPNAME.DDNAME DD PARAMETER=VALUE
```


Override DD statements.

Coding an override DD statement.

When coding an override DD statement, the following considerations must be kept in mind:

- **Code the override DD statement immediately following the EXEC statement used to invoke the procedure.**
- **Supply a name for the override DD statement consisting of the name of the procedure step in which the DD statement to be overridden occurs, followed by a period and the name of the DD statement in the procedure step.**
- **Specify on the DD statement (in any sequence) the keyword parameters whose values are to be changed or nullified, separated by commas.**

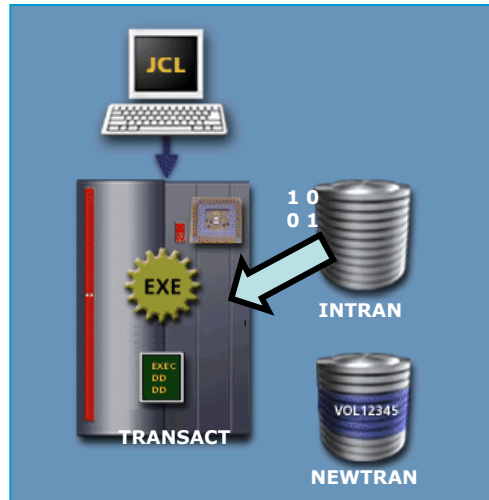
Override DD statements.

Override DD statement – an example.

In the TRANSACT procedure, the input transactions are specified in a data set named INTRAN.

A particular week's input transactions reside in a data set named NEWTRAN.

NEWTRAN is an uncataloged data set that resides on a 3390 direct access volume whose volume identifier is 12345.



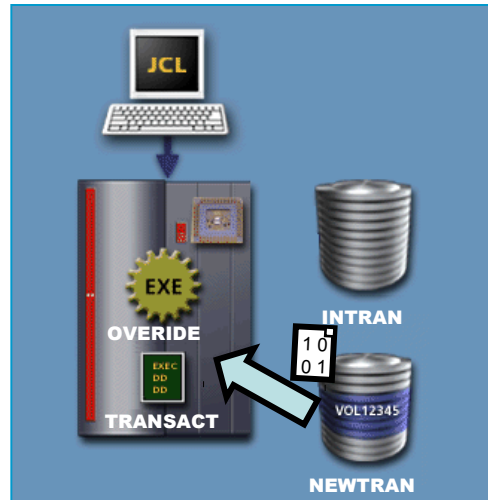
Override DD statements.

Override DD statement – an example.

In order to use the existing TRANSACT procedure to process that week's transactions a DD override statement that identifies NEWTRAN, rather than INTRAN, as the name of the transaction data set must be coded.

The override DD statement is as follows:

```
//PSTEP1.DD1 DD DSN=NEWTRAN,  
//                UNIT=3390,  
//                VOL=SER=12345
```



Override DD statements.

Override DD statements – procedure definition.

The code alongside shows the TRANSACT procedure definition.

The DD statement parameter that is to overridden is highlighted.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//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, TIME=5
//DD5 DD DSN=&&VALID,
// DISP=(OLD, DELETE)
```

Override DD statements.

Override DD statements – EXEC statement.

The code alongside shows the EXEC statement used to invoke the TRANSACT procedure.

The override DD statement is coded immediately after the EXEC statement, specifying the procedure step containing the DD statement and name of the DD statement in the procedure, the new data set's name, and its location.



Override DD statements.

Override DD statement – resulting JCL.

The code on the right shows the resulting JCL statement.

The data set to be used during execution has been changed to NEWTRAN.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//DD1 DD DSN=NEWTRAN, DISP=SHR,
// UNIT=3390,
// VOL=SER=12345
//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, TIME=5
//DD5 DD DSN=&&VALID,
```

In this example, you will see that the parameter DISP=SHR, specified for DD1 in the procedure definition, is retained.

The existing DD statement keyword parameters that are not overridden are retained.

Override DD statements.

Are we on track?

Under which of the following conditions would you code an override DD statement?

- A. The Procedure step has no DD statements.**
- B. There is no DD statement with the ddname you need.**
- C. The data set attributes in a DD statement are not appropriate.**

23

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

The correct answer is C.

Override DD statements.

Are we on track?

The following is a procedure called MYPROC

```
//STEP1 EXEC PGM=PROGA
//DD1 DD DSN=A, DISP=SHR
//DD2 DD DSN=OUTA, DISP=(NEW, CATLG),
// UNIT=SYSDA, SPACE=(TRK, (1, 1))
//STEP2 EXEC PGM=PROGB
//DD3 DD DSN=B, DISP=SHR
//DD4 DD DSN=RPT, DISP=(NEW, CATLG),
// UNIT=SYSDA, SPACE=(TRK, (1, 1))
```

Complete the following JCL statements so that when MYPROC is invoked, PROGB creates a SYSOUT class A output rather than write the output to the data set named RPT.

```
//JSTEP EXEC MYPROC
```

24

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

The correct answer is

```
//STEP2.DD4 DD SYSOUT=A
```

The most precise specification would be:

```
//STEP2.DD4 DD SYSOUT=A,DSN=,UNIT=,SPACE=
```

You should nullify rest of parameters originally coded in the procedure, except the DISP parameter which is „mutually exclusive with keyword SYSOUT“.

Nullification DD statements.

Nullifying a DD statement parameter.

To return the value of a DD statement parameter to the system default, the parameter must be nullified.

To nullify a DD statement parameter, you:

- **Give the name of the parameter, followed by the equal sign.**
- **Do not code a value for the parameter.**

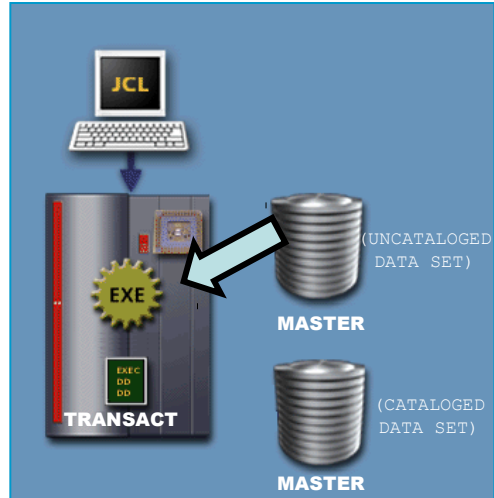
The general form for nullifying a DD statement parameter is:

//procstepname.ddname DD parameter=

Nullification DD statements.

Nullifying a DD statement parameter – an example.

In this example assume the TRANSACT procedure definition identifies data set MASTER as an uncataloged data set residing on a 3480 tape volume with a volume identifier of 987762.



Nullification DD statements.

Nullifying a parameter – procedure definition.

The TRANSACT procedure definition is shown on the right.

The DD statement parameter that is to be overridden is highlighted.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//DD1 DD DSN=NEWTRAN, DISP=SHR
//DD2 DD DSN=MASTER, DISP=SHR,
// UNIT=3480,
// VOL=SER=987762
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP= (NEW, PASS) ,
// UNIT=SYSDA,
// SPACE= (TRK, (1, 1))
//PSTEP2 EXEC PGM=PROG2, TIME=5
//DD5 DD DSN=&&VALID,
```

Nullification DD statements.

Nullifying a parameter – EXEC statement.

The EXEC statement used to invoke the TRANSACT procedure is shown on the right.

The override DD statement is coded immediately after the EXEC statement, specifying the procedure step containing the DD statement and name of the DD statement in the procedure.

The override DD statement nullifies the UNIT and VOL parameters.



Nullification DD statements.

Nullifying a parameter – resulting JCL.

The resulting JCL statement is shown on the right.

The data set to be used during execution has been changed to default value i.e. the cataloged MASTER data set.



```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//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, TIME=5
//DD5 DD DSN=&&VALID,
// DISP= (OLD, DELETE)
```

The parameter DISP=SHR, specified for DD2 in the procedure definition, is retained because the existing DD statement keyword parameters, that are not overridden, are retained.

Nullification DD statements.

Are we on track?

The following is a procedure called MYPROC

```
//STEP1 EXEC PGM=PROGA
//DD1 DD DSN=A,DISP=SHR
//DD2 DD DSN=OUTA,DISP=(NEW,CATLG),
// UNIT=SYSDA,SPACE=(TRK,(1,1))
//STEP2 EXEC PGM=PROGB
//DD3 DD DSN=B,DISP=SHR
//DD4 DD DSN=RPT,DISP=(NEW,CATLG),
// UNIT=SYSDA,SPACE=(TRK,(1,1))
```

Code statements to override the UNIT parameter to 3480 and nullify the SPACE parameter in STEP2.

```
//JSTEP EXEC MYPROC _____
```

31

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

The correct answer is

```
//STEP2.DD4 DD UNIT=3480,SPACE=
```

Nullification DD statements.

DUMMY parameter.

A DUMMY parameter is used to bypass a data set, without resulting in a program ABEND.

A DUMMY parameter may be specified in a procedure step DD statement. If you require to execute the procedure with real data, the DUMMY parameter must be overridden with the name of the data set to be used.

Nullification DD statements.

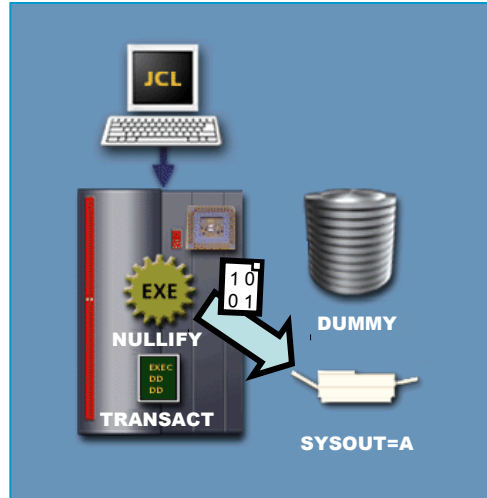
Nullifying the DUMMY parameter – an example.

In this example, a procedure step named PSTEP1 contains the following DD statement to identify the output data set:

```
//DDOUT DD DUMMY
```

If a SYSOUT class A output has to be created for this invocation then the following DD statement needs to be coded:

```
//PSTEP1.DDOUT DD SYSOUT=A
```



Nullification DD statements.

Nullifying the DSN parameter.

If a procedure is to be executed without using a data set specified on a procedure step DD statement, a dummy status can be assigned to the data set.

The override DD statement can either specify DUMMY or DSN=NULLFILE.

//procstepname.ddname DD DUMMY

or

//procstepname.ddname DD DSN=NULLFILE

When DUMMY or DSN=NULLFILE is coded on an override DD statement, it nullifies all parameters specified on the corresponding DD statement in the procedure definition except the DCB parameter.

Nullification DD statements.

Nullifying the DSN parameter – procedure definition.

The TRANSACT procedure definition is shown on the right.

The DD statement parameter that is to be overridden is highlighted.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//DD1 DD DSN=INTRAN, DISP=SHR
//DD2 DD DSN=MASTER, DISP=SHR,
// UNIT=3480,
// VOL=SER=98762
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW, PASS),
// UNIT=SYSDA,
// SPACE=(TRK, (1, 1))
//PSTEP2 EXEC PGM=PROG2, TIME=5
//DD5 DD DSN=&&VALID,
```



Nullification DD statements.

Nullifying the DSN parameter – EXEC statement.

The EXEC statement used to invoke the TRANSACT procedure is shown on the right.

The override DD statement nullifies the DSN parameter and any other parameters associated with that data set.



Nullification DD statements.

Nullifying the DSN parameter – resulting JCL.

The resulting JCL statement is shown on the right.

During execution of the procedure, the system will not consider the data set that was designated by DD2.

```
//PSTEP1 EXEC PGM=PROG1, TIME (1, 30)
//DD1 DD DSN=INTRAN, DISP=SHR
//DD2 DD DUMMY
//DD3 DD SYSOUT=A
//DD4 DD DSN=&&VALID,
// DISP=(NEW, PASS),
// UNIT=SYSDA,
// SPACE=(TRK, (1, 1))
//PSTEP2 EXEC PGM=PROG2, TIME=5
//DD5 DD DSN=&&VALID,
// DISP=(OLD, DELETE)
```

Nullification DD statements.

Are we on track?

Which one of the following is the result of coding DUMMY or DSN=NULLFILE on an override DD statement?

- A. The procedure is executed with a dummy data set.**
- B. All parameters on the corresponding DD statement except DCB are nullified.**
- C. The procedure is executed without the data set.**
- D. All parameters on the corresponding procedure DD statement are nullified.**

38

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

The correct answer is B.

(But I could accept even A and C - see previous slide 34.)

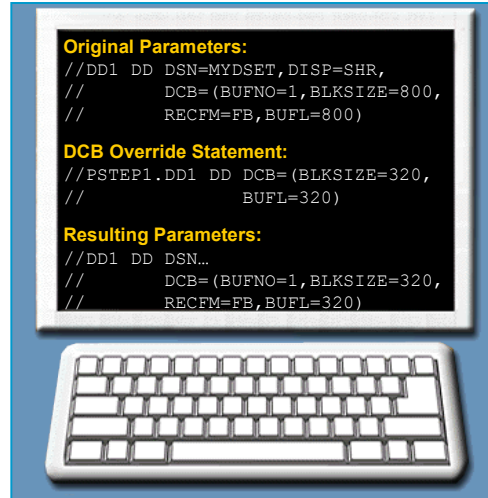
Nullification DD statements.

DCB subparameters.

The data control block (DCB) unlike other parameters, is not overridden in its entirety, but by subparameter.

DCB subparameters must be nullified explicitly. You code the DD override statement with the parameters to be overridden in parentheses. The others are retained, as shown on the right. To nullify the entire DCB parameter, you must explicitly nullify each subparameter as shown below:

```
//      DCB=(BUFNO=,BLKSIZE=,  
//      RECFM=,BUFL=)  
39
```



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

Nullification DD statements.

Consideration while nullifying DCB parameters.

Some special rules apply when overriding or nullifying the DCB parameter:

- Code only those DCB keyword subparameters whose values need to be changed. DCB keyword subparameters which are not coded remain unchanged.
- Code any required DCB positional subparameters regardless of whether or not they are specified on the DD statement in the procedure definition.
- To nullify an existing DCB positional subparameter simply omit it from the DCB parameter given in the override DD statement.
- To nullify the DCB parameter completely, omit all existing positional subparameters and explicitly nullify each existing keyword subparameter.
- Positional subparameters (such as data set name) are essential to the system. These must be coded, even if they are specified in the procedure definition.
- Keyword subparameters (such as BLKSIZE) supply additional information to the operating system. If they are specified in the procedure definition, then code only those that are to be changed.

Nullification DD statements.

Nullifying the DCB parameter – example 1.

This is a DD statement in a procedure step called PSTEP1:

```
//DD1 DD DSN=MYDSET,DISP=SHR,  
//      DCB=(BUFNO=1, BLKSIZE=800, RECFM=FB, BUFL=800)
```

If the block size and buffer length needs to be changed to 320 at the time of execution, then the following override DD statement needs to be coded:

```
//PSTEP1.DD1 DD DCB=(BLKSIZE=320, BUFL=320)
```

The values of the DSN and DISP parameters and the BUFNO and RECFM subparameters remain unchanged.

Nullification DD statements.

Nullifying the DCB parameter – example 2.

This is a DD statement in a procedure step called PSTEP1:

```
//DD2 DD DSN=MYDSET,DISP=SHR,DCB=(DS1,BLKSIZE=80)
```

If the block size needs to be changed to 400 and the remaining DCB information has to be copied from a cataloged data set (DS1) at the time of execution, then the following override DD statement needs to be coded:

```
//PSTEP1.DD2 DD DCB=(DS1,BLKSIZE=400)
```

42

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

If the DCB parameter is being modified and a DCB positional subparameter needs to be retained, the positional subparameter must be coded. This applies whether the positional subparameter exists on the DD statement in the procedure definition or not.

The value DS1 represents a positional subparameter, which must be explicitly specified in the override statement if it is to be retained.

Nullification DD statements.

Nullifying the DCB parameter – example 3.

This is a DD statement in a procedure called PSTEP1:

```
//DD3 DD DSN=MYDSET,DISP=SHR,  
//      DCB=(RECFM=FB,BLKSIZE=160,LRECL=80)
```

In order to nullify the entire DCB parameter of the DD statement, each DCB keyword subparameter specified in the procedure definition must be nullified as shown below:

```
//PSTEP1.DD3 DD DCB=(RECFM=,BLKSIZE=,LRECL=)
```

Nullification DD statements.

Are we on track?

Which of the following rules apply to coding DD override statements for the DCB parameter?

- A. Code only those keyword subparameters you want to change.**
- B. To nullify, omit positional parameters and nullify all keyword parameters.**
- C. Specify any positional subparameters you want to retain.**
- D. To nullify, code DCB=NULLFILE.**

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

Nullification DD statements.

Are we on track?

A DD statement named DD1 in a procedure step called PSTEP2 has the following DCB parameter

```
//      DCB=(DS1,BLKSIZE=400,RECFM=FB,BUFL=400)
```

Complete the following JCL statements to change the block size and buffer length to 320 and copy other DCB information from data set DS2.

```
//PSTEP2.DD1 DD DCB=_____
```

45

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

The correct answer is (DS2,BLKSIZE=320,BUFL=320)

Nullification DD statements.

Glossary.

DCB subparameters

A variable placed within parenthesis within the DCB parameter, to specify information such as block size or record format.

Keyword subparameters

This is a list of variable information contained within parentheses, and followed by an equal sign and value.

Positional subparameters

This is a list of variable information coded in parentheses.

Addition DD statements.

Addition DD statements.

If a required data set is not found in the procedure definition then an additional DD statement can be coded along with the invocation of the procedure.

The general form is same as an override DD statement:

```
//prostepname.ddname DD parameter=value
```

Explanation of the general form to add a DD statement:

A name for the addition DD statement has to be supplied. The name consists of the name of the procedure step to which the DD statement is to be added, followed by a period, and the ddname defined for the data set in the program.

The parameters required to identify the data set to be used must be included.

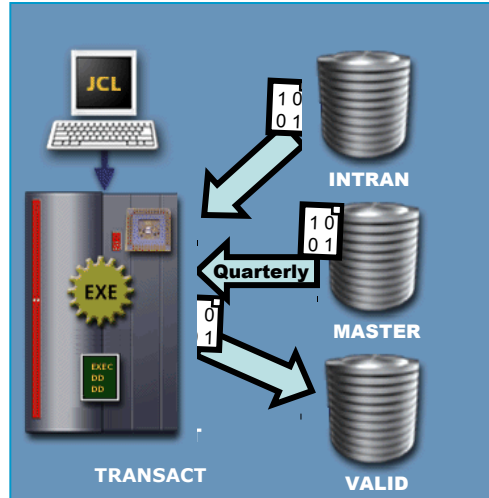
Addition DD statements.

Adding DD statements – an example.

In this example, the TRANSACT procedure is used to process customer orders.

Every quarter the transaction input file (data set INTRAN) has to be checked against the master customer list (data set MASTER) to ensure that each transaction has valid customer.

During the routine weekly execution the check is not made, and data set MASTER is not used.




Addition DD statements.

Adding DD statements – procedure definition.

The TRANSACT procedure definition is shown on the right.

In this procedure:

- The PARM value of NOCHECK is specified in the PSTEP1 EXEC statement.
- PSTEP1 does not include a reference to data set MASTER.



```
//PSTEP1 EXEC PGM=PROG1,TIME(1,30),  
//          PARM=NOCHECK  
//DD1      DD DSN=INTRAN,DISP=SHR  
//DD3      DD SYSOUT=A  
//DD4      DD DSN=&&VALID,  
//          DISP=(NEW,PASS),  
//          UNIT=SYSDA,  
//          SPACE=(TRK,(1,1))  
//PSTEP2 EXEC PGM=PROG2,TIME=5  
//DD5      DD DSN=&&VALID,  
//          DISP=(OLD,DELETE)  
//DD6      DD SYSOUT=A
```

The PARM value of NOCHECK is specified in the PSTEP1 EXEC statement, because the check is usually not required.

PSTEP1 does not include a reference to data set MASTER, because the data set is also not usually required. The omission ensures that system resources (such as a tape drive) are not allocated for the data set when it is not used.

Addition DD statements.

Adding DD statements – EXEC statement.

To execute the procedure routinely every week without making the check, you would code the following EXEC statement to invoke the procedure:

```
//JSTEP EXEC TRANSACT
```

The code on the right represents the effective JCL for this particular job.

```
//PSTEP1 EXEC PGM=PROG1,TIME(1,30),  
// PARM=NOCHECK  
//DD1 DD DSN=INTRAN,DISP=SHR  
//DD3 DD SYSOUT=A  
//DD4 DD DSN=&&VALID,  
// DISP=(NEW,PASS),  
// UNIT=SYSDA,  
// SPACE=(TRK,(1,1))  
//PSTEP2 EXEC PGM=PROG2,TIME=5  
//DD5 DD DSN=&&VALID,  
// DISP=(OLD,DELETE)  
//DD6 DD SYSOUT=A
```



Addition DD statements.

Addition DD statements – EXEC statement.

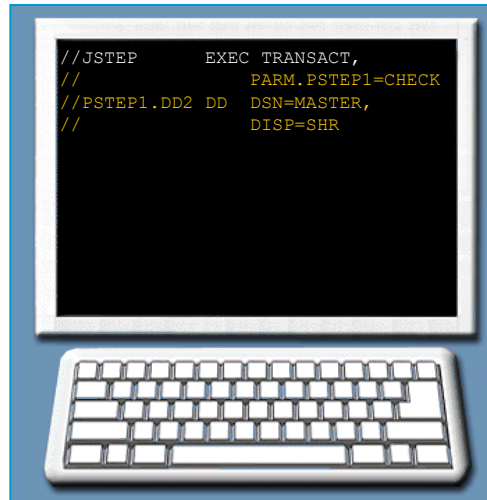
For the quarterly execution the transaction input file has to be checked against the master customer list.

To perform the required quarterly check, two statements are required to invoke the procedure:

- An EXEC statement override that includes a PARM parameter value of CHECK to override the value of NOCHECK.
- An addition DD statement to identify data set MASTER that PROG1 needs to access to perform the check.

51

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



The code on the right represents the EXEC statement used to invoke the procedure.

Addition DD statements.

Addition DD statements – effective JCL.

The resulting JCL statement is shown on the right.

The PARM value has been overridden to CHECK and an addition statement to identify data set MASTER has been specified.



```
/*PSTEP1 EXEC PGM=PROG1,TIME=(1,30),
/
/ PARM=CHECK
/DD1 DD DSN=INTRAN,DISP=SHR
/DD3 DD SYSOUT=A
/DD4 DD DSN=*&VALID,
/ DISP=(NEW,PASS),
/ UNIT=SYSDA,
/ SPACE=(TRK,(1,1))
/DD2 DD DSN=MASTER,DISP=SHR
/*PSTEP2 EXEC PGM=PROG2,TIME=5
/DD5 DD DSN=*&VALID,
/ DISP=(OLD,DELETE)
```

Addition DD statements.

Are we on track?

Under which one of the following conditions would you code an addition DD statement?

A. The ddname you need is not specified.

B. The ddname is specified, but the attributes are not appropriate.

C. The procedure step has a DD statement with the DUMMY parameter.

53

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

The correct answer is A.

Addition DD statements.

Are we on track?

The following is a procedure called ANYPROC

```
//PSTEPS EXEC PGM=PROG1
//DD1 DD DSN=DATFIL,DISP=SHR
//DD2 DD SYSOUT=A
//PSTEPB EXEC PGM=PROG2
//DD3 DD SYSOUT=A
```

Code an addition statement for ANYPROC. Specify, for PROG2, a data set named INDATA. PROG2 refers to the data set by the ddname TEST.

```
//JSTEP EXEC ANYPROC
```

54

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

The correct answer is

```
//PSTEPB.TEST DD DSN=INDATA
```

Sequencing multiple parameters.

Coding multiple parameters.

Any number of override and addition DD statements can be coded to invoke a procedure.

The rules for coding multiple parameters are:

- **Code the EXEC statement to invoke the procedure.**
- **Code the override statements (if any) for a step in the same DDname sequence as in the procedure definition.**
- **Code addition DD statements (if any) for that step, following the override statements.**
- **Within each procedure step, the override statements must be coded in DDname sequence.**

55

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

To facilitate override sequencing some installation require that DD statements in procedure steps be in alphabetical order by ddname.

You can add or override DD statements in the order in which they appear within the procedure (procedure step order).

[Within a procedure step, you must code overrides before additions.](#)

Sequencing multiple parameters.

Sequencing multiple parameters – an example.

The TRANSACT procedure definition is shown on the right. The following changes need to be made:

- PROG1 obtains its input transactions from a cataloged data set named MYDATA, instead of from INTRAN.
- PROG1 checks the contents of MYDATA against the contents of a second cataloged data set named CKDATA. The PARM value of NOCHECK needs to be overridden with the value CHECK, and a DD statement named DD2 to reference CKDATA has to be added.
- PROG2 writes its output (using ddname DD6) to a data set named INVOICE on a 3390 volume with volume identifier 6929L. The data set requires 10 tracks of space and should be cataloged.

```
//PSTEP1 EXEC PGM=PROG1,TIME(1,30),
//          PARM=NOCHECK
//DD1      DD DSN=INTRAN,DISP=SHR
//DD3      DD SYSOUT=A
//DD4      DD DSN=&&VALID,
//          DISP=(NEW,PASS),
//          UNIT=SYSDA,
//          SPACE=(TRK,(1,1))
//PSTEP2 EXEC PGM=PROG2,TIME=5
//DD5      DD DSN=&&VALID,
//          DISP=(OLD,DELETE)
//DD6      DD SYSOUT=A
```



Sequencing multiple parameters.

Are we on track?

Refer to the previous example. Which of the following JCL statements would you need when you invoke TRANSACT?

- A. An EXEC statement with a PARM parameter override.**
- B. An override DD statement for PSTEP2.DD6.**
- C. An addition DD statement for PSTEP1.**
- D. An override DD statement for PSTEP1.DD1.**

57

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 D.

Sequencing multiple parameters.

Sequencing rules – an example.

The JCL used to invoke TRANSACT is shown on the right.

- The PARM parameter value has been overridden to CHECK.
- The override for DD1 in step PSTEP1 is specified.
- The addition statement DD2 for step PSTEP1 is specified.
- The override statement for DD6 in step PSTEP2 is specified.



58

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

These statements must be specified in the order as shown.

Sequencing multiple parameters.

Shorthand form for multiple override statements.

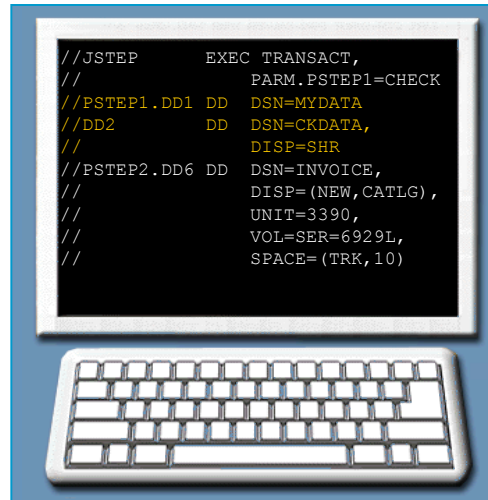
Shorthand form of coding multiple override and addition DD statements can be used in the procedure step sequence.

To code the shorthand form:

- Follow the general form for the first override or addition statement.
- For subsequent override or addition statements in that procedure step use

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

59

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

The TRANSACT invocation from the previous example, this time using shorthand form is shown on the right.

Generally, you code override and addition DD statements with the name specified as:

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

In my opinion – it is confusing. I would not recommend it.

Sequencing multiple parameters.

Are we on track?

The procedure named WEEKLY is shown below.

```
//S1 EXEC PGM=PROCESS,TIME=(1,30)
//DD1 DD DSN=MYDATA,DISP=SHR
      ...
//S2 EXEC PGM=BILL
//DD3 DD DSN=&&TEMP,DISP=(NEW,DELETE)
      ...
```

Code the EXEC statement required to pass:

- a. PARM value of YES to the program named PROCESS.**
- b. Impose a CPU time limit of 3 minutes for execution of the program named BILL.**
- c. Use the installation-define time default for the program named PROCESS.**

```
//JSTEP EXEC _____
60
```

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

The correct answer is

WEEKLY,PARM.S1=YES,TIME.S1=,TIME.S2=3

Sequencing multiple parameters.

Are we on track?

The procedure named WEEKLY is shown below.

```
//S1 EXEC PGM=PROCESS,TIME=(1,30)
//DD1 DD DSN=MYDATA,DISP=SHR
      . . .
//S2 EXEC PGM=BILL
//DD3 DD DSN=&&TEMP,DISP=(NEW,DELETE)
      . . .
```

Complete the DD override statement to specify that PROCESS obtain its input from a cataloged data set named NEWDATA.

_____ **DD DSN=NEWDATA**

61

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

The correct answer is:

//S1.DD1 DD DSN=NEWDATA

Sequencing multiple parameters.

Are we on track?

Review the following DD statements from STEPC of a sample procedure:

```
. . .  
//DD4 DD DSN=A.B.C...  
//DD5 DD DSN=STRP...  
//DD6 DD DSN=TYPE3...  
//DD7 DD DSN=A.B.D...
```

In this exercise, code the following override DD statements using the shorthand form:

Code an override statement to specify a data set named TEST1 instead of A.B.C

//STEPC.DD4 DD _____

Code an override statement to specify a data set named TESTDATA rather than STRP

//DD5 DD _____

Code an override statement to specify a data set named TEST2 instead of A.B.D

//DD7 DD _____

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

The correct answer is:

1. DSN=TEST1
2. DSN=TESTDATA
3. DSN=TEST2

Sequencing multiple parameters.

Are we on track?

Place the following override and addition DD statements in the order in which they must be specified. (Assume procedure DD statements are in alphanumeric order.)

- A. An override DD statement for DD6 in PSTEP3**
- B. An override DD statement for DD2 in PSTEP1**
- C. An addition DD statement for DD3 in PSTEP1**
- D. An addition DD statement for DD5 in PSTEP3**
- E. An override DD statement for DD4 in PSTEP1**

63

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

The correct order is B., E., C., A., D.

Modifying DD parameters.

Unit summary.

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

- **Override or nullify parameters on procedure step DD statements.**
- **Add DD statements to a procedure step if it does not already contain all the DD statements required for execution.**
- **Correctly sequence multiple temporary alterations to procedure step DD statements.**