JCL

Chapter b5
Determinig the effective JCL

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

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

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

Chapter b5

Determining the effective JCL

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

Unit introduction.

This unit focuses on identifying and correcting common JCL errors that can occur when a procedure is used.

This unit discusses how to identify an effective JCL (the actual JCL resulting from the use of a procedure) in a message log.

This unit also explains how to look at system messages, error messages and the effective JCL to isolate and correct common JCL errors.

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

Course objectives.

Be able to:

- **Identify the JCL in effect at job execution time by examining a job log.**
- **Specify the parts of a job log that can help you analyze the effective JCL.**
- **Identify and correct common JCL errors that can occur when a procedure is used.**

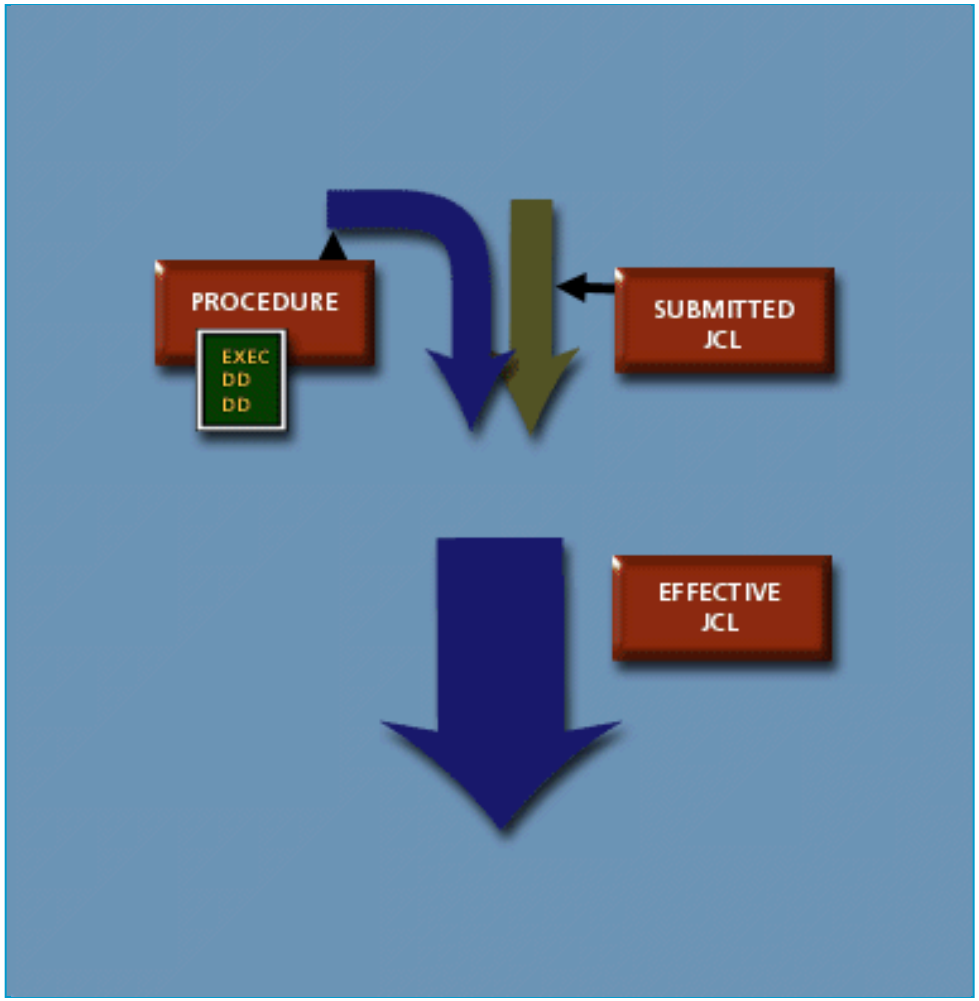
Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Effective JCL.

What is an effective JCL?

When a procedure is invoked, the system combines both the submitted JCL and the JCL that is stored with the procedure. The resulting JCL is called the effective JCL.

Effective JCL is the JCL in effect at the time of job execution.



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

Examining effective JCL.

An effective JCL is examined for the following reasons:

- **To ensure that the effective JCL satisfies all the processing requirements, particularly if it contains a number of overrides and additions.**

- **To find the source of the problem if a job is not successfully executed.**

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

Job log.

An effective JCL is checked by examining the job log.

In addition to the listing of effective JCL used during job execution, a job log can also contain the following:

- System messages.
- Detailed error messages for specific statements of effective JCL.
- Resource-allocation/job-step termination-status messages.

System Messages:

```
JES2 JOB LOG--SYSTEM EPP1 - NODE SPC

09.11.58 JOB0355 TEFC452I - JOB NOT RUN-
JCL ERROR
-----JES2 JOB STATISTICS-----
```

Detailed Error Message:

```
STMT NO. MESSAGE
3        IEFC630I UNIDENTIFIED KEYWORD
         PATM
```

Resource Allocation Messages:

```
IEF2371 4BB ALLOCATED TO DD1
```

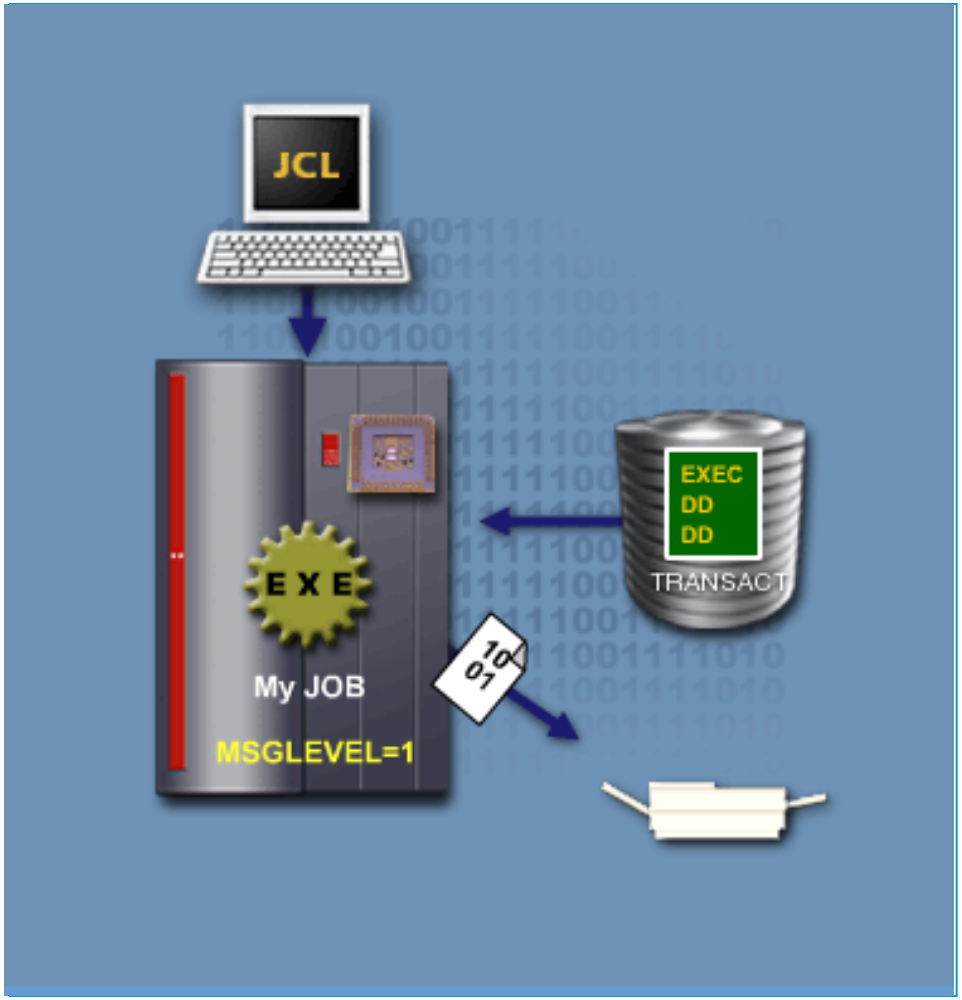
Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Requesting a JCL listing.

A programmer can request that procedure statements be listed in a job log by coding the value 1 as the first MSGLEVEL subparameter of the JOB statement.

```
//   JOB    ...MSGLEVEL=1
```

The installation may include procedure statements on a job log by default. If so, using the MSGLEVEL parameter to specify that procedure statements be listed is not required.



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

Are we on track?

Enter the subparameter that you would code on a job statement to obtain a listing of procedure statements, if they are not included by default:

//MYJOB JOB 123,D.GREEN_____

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

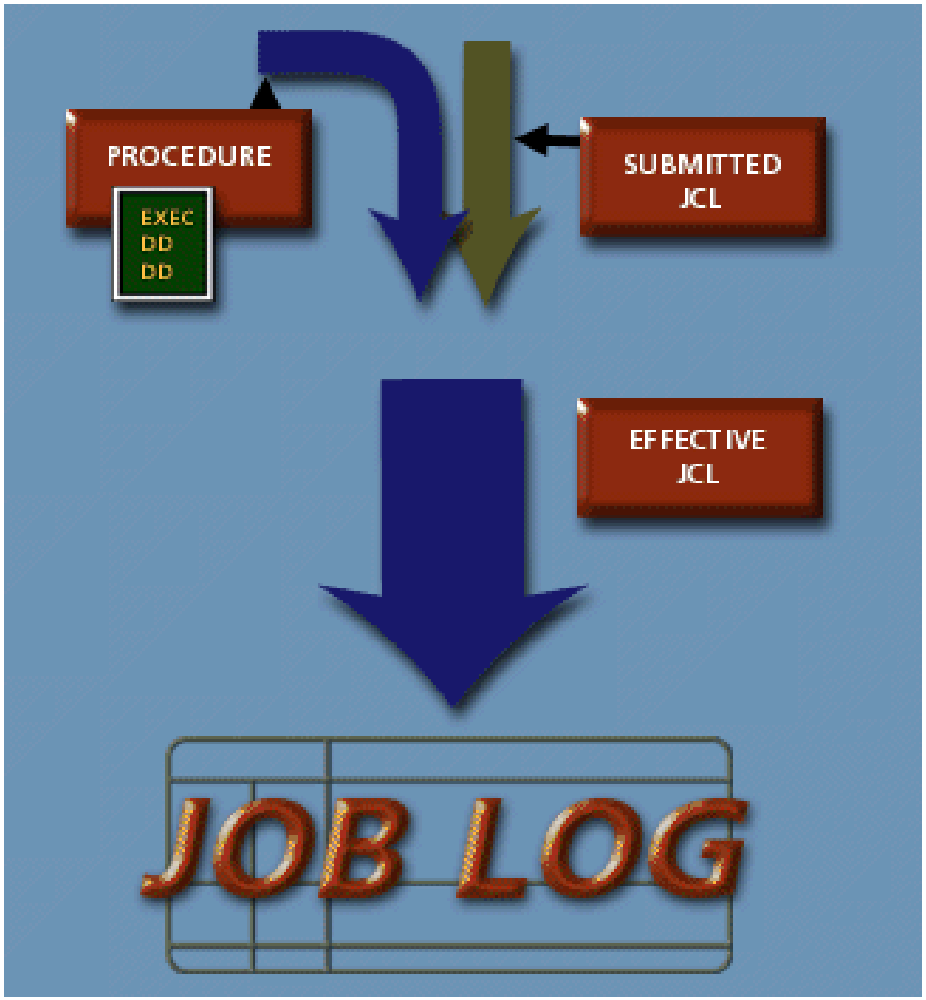
Job log listing.

The listing of a job log can be used to verify if the resulting JCL for the job is what is required.

The job log is also useful to verify the JCL and to distinguish between:

- JCL statements submitted.
- Statements stored in the procedure.
- JCL in effect.

In a job log listing, special notations in columns 1, 2, and 3 distinguish between different categories of statements as shown on the next slide.



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

Notations in a job log listing.

Notations in Columns 1, 2, and 3		Statement identified
Cataloged Procedure	In-stream Procedure	
//	//	Statements you submit with the job, including in-stream procedure definitions (if applicable) and any alterations to DD statements.
XX	++	A statement in a procedure definition that is used during a job execution.
X/	+/	A DD statement in a procedure definition that you have overridden.
XX*	++*	A statement in a procedure definition, other than a comment statement, that the system considers to be a comment.
***	***	A comment or job entry subsystem control statement.

The table above illustrates the notations used on a job log to identify JCL statements. Special notation in columns 1,2 and 3 of the JCL listing distinguish the different categories of JCL statements.

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

Identifying the JCL in effect.

Are we on track?

Match the job log notations below with the kinds of statements they identify.

1. +/	**A.**	**JCL statements you submit.**
2. XX	**B.**	**A DD statement in an in-stream procedure that is overridden.**
3. //	**C.**	**A comment or control statement.**
4. ***	**D.**	**A statement from a cataloged procedure that is used.**

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

JCL in effect – In-stream procedure.

The JCL on the right illustrates how to identify the JCL in effect at job execution time when using an in-stream procedure.

In the example, the JCL is for a job LA$TEST2 that invokes an in-stream procedure TRANSACT.

The JCL for LA$TEST2 comprises the following:

- The JOBLIB statement identifies the library storing the programs to be executed.
- The TRANSACT procedure definition appears between the PROC and PEND statements.
- The final statement executes the procedure.

```
//LA$TEST2 JOB 31SPC090156,ROSE,CLASS=B
//JOBLIB   DD DSN=TSOCHIS.TESTJCL.LOAD,
//            DISP=SHR
//TRANSACT PROC
//PSTEP1   EXEC PGM=PROG1
//DD1      DD  DSN=TSOCHIS.INTRAN,
//             DISP=SHR
//DD2      DD  DSN=TSOCHIS.MASTER,
//             DISP=SHR
//DD3      DD  SYSOUT=A
//DD4      DD  DSN=&&VALID,UNIT=SYSDA,
//             DISP=(NEW,PASS),
//             SPACE=(TRK,(1,1))
//PSTEP2   EXEC PGM=PROG2
//DD5      DD  DSN=&&VALID,
//             DISP=(OLD,DELETE)
//DD6      DD  SYSOUT=A
//         PEND
//JSTEP    EXEC TRANSACT
```

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

Identifying the JCL in effect.

JCL statements in effect.

```
1.  //LA$TEST2     JOB      (31SPCO3090156W),ROSE,CLASS=B
2.  //JOBLIB       DD       DSN=TSOCHIS.TESTJCL.LOAD,DISP=SHR
    //TRANSACT     PROC
    //PSTEP1       EXEC     PGM=PROG1
    //DD1          DD       DSN=TSOCHIS.INTRAN,DISP=SHR
    //DD2          DD       DSN=TSOCHIS.MASTER,DISP=SHR
    //DD3          DD       SYSOUT=A
    //DD4          DD       DSN=&&VALID,UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(1,1))
    //PSTEP2       EXEC     PGM=PROG2
    //DD5          DD       DSN=&&VALID,DISP=(OLD,DELETE)
    //DD6          DD       SYSOUT=A
    //            PEND
3.  //JSTEP        EXEC     TRANSACT
4.  ++TRANSACT     PROC
5.  ++PSTEP1       EXEC     PGM=PROG1
6.  ++DD1          DD       DSN=TSOCHIS.INTRAN,DISP=SHR
7.  ++DD2          DD       DSN=TSOCHIS.MASTER,DISP=SHR
8.  ++DD3          DD       SYSOUT=A
9.  ++DD4          DD       DSN=&&VALID,UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(1,1))
10. ++PSTEP2       EXEC     PGM=PROG2
11. ++DD5          DD       DSN=&&VALID,DISP=(OLD,DELETE)
12. ++DD6          DD       SYSOUT=A
```

The JCL portion of the job log for LA$TEST2 is shown above.

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

Are we on track?

Review the effective JCL for job LA$TEST2, on the previous page. The // notation identifies the JCL statements that are submitted with the job that invokes the TRANSACT procedure. Which of the following are included?

A. JOB statement.

B. JOBLIB DD statement.

C. DD override statement.

D. TRANSACT procedure definition.

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

JCL in effect – Cataloged procedure.

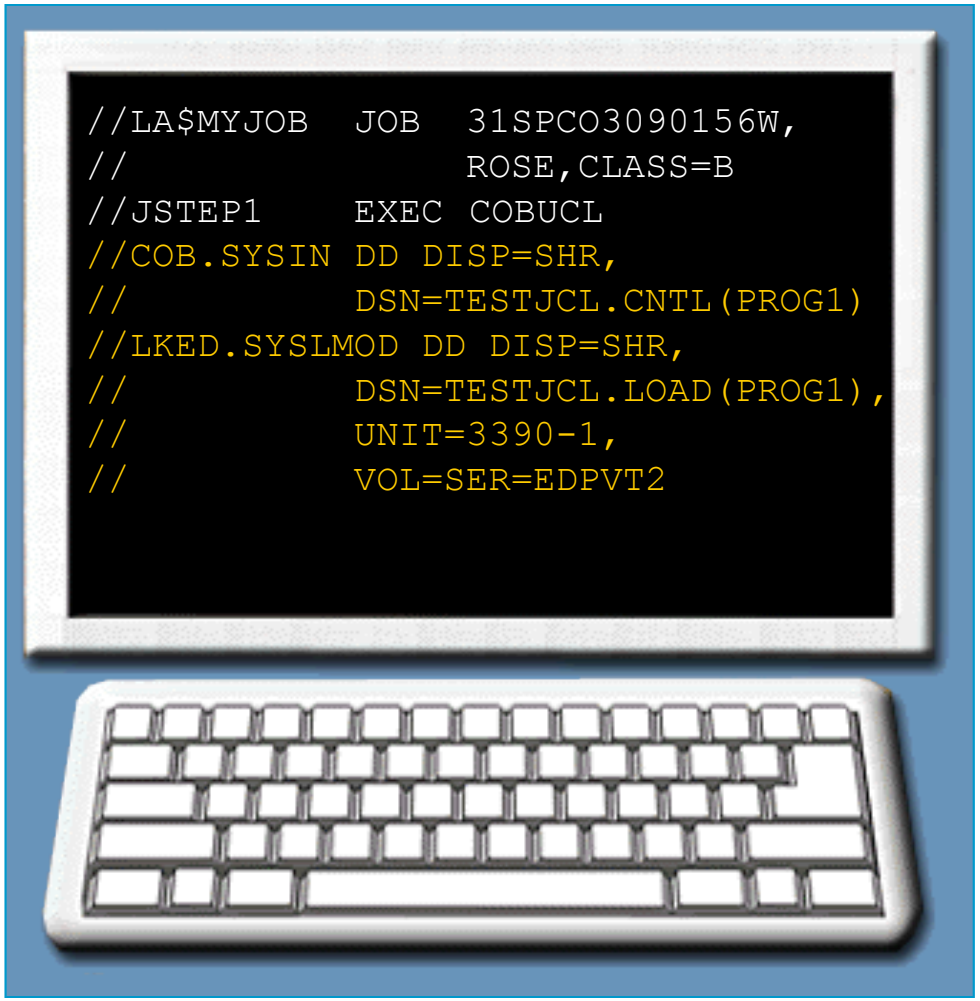
An example of JCL that invokes a cataloged procedure named COBUCL is shown on the right. COB step of COBUCL compiles a COBOL

program and LKED link-edits the resulting COBOL object program.

There are addition and override statements for JSTEP1 as follows:

- COB.SYSIN is an addition DD statement that identifies the source module to be compiled.

- LKED.SYSLMOD is an override DD statement that specifies the data set and member name.



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

Identifying the JCL in effect.

Effective JCL in a job log.

```
1.  //LA$MYJOB     JOB      3ISPCO3090156W,ROSE,MSGCLASS=T,CLASS=T,MSGLEVEL(1,1)
2.  //JSTEP1       EXEC     COBUCL
3.  XXCOBUCL       PROC
4.  XXCOB          EXEC     PGM=IKFCBL00
5.  XXSYSPRINT     DD       SYSOUT=*
6.  XXSYSUT1       DD       UNIT=SYSDA,SPACE=(CYL,(1,1))
7.  XXSYSUT2       DD       UNIT=SYSDA,SPACE=(CYL,(1,1))
8.  XXSYSUT3       DD       UNIT=SYSDA,SPACE=(CYL,(1,1))
9.  XXSYSUT4       DD       UNIT=SYSDA,SPACE=(CYL,(1,1))
10. XXSYSLIN       DD       DSN=**LOADSET,UNIT=SYSDA,
    XX                      DISP=(MOD,PASS),SPACE=9TRK,93,300,DCB=BLKSIZE=800
11. //COB.SYSIN    DD       DSN=TESTJCL.CNTL(PROG1),DISP=SHR
12. XXLKED         EXEC     PGM=IEW,PARM='LIST,MAP',COND=(5,LT,COB),
. . .
15. //LKED.SYSLMODDD       DSN=TESTJCL.LOAD(PROG1),UNIT=SYDA,DISP=SHR
    X/SYSLMOD      DD       DSN=&&GOSET,DISP=(,PASS),NIT=SYSDA,SPACE=(CYL,(1,1,1,)0
```

A portion of the effective JCL for LA$MYJOB is shown above. The JCL statements that are submitted are identified by the // notation. The statements from the procedure definition are identified by the XX notation. The addition DD statement for procedure step COB is statement 11. The override statement for procedure step LKED (statement 15) appears at the point where the DD statement to be overridden appears. The statement to be overridden is preceded by the X/ notation.

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

Identifying the JCL in effect.

Are we on track?

Review the effective JCL in job log for LA$MYJOB, on the previous slide. Notice that the system merges the cataloged procedure statements into the job stream at the appropriate places. Order the following statements to reflect the sequence of effective JCL in the example.

A. Statements of procedure step COB.

B. EXEC statement to invoke the procedure.

C. Addition statement for COB.

D. JOB statement.

E. DD statement that is overridden.

F. Statement of procedure step LKED.

G. Override statement for procedure step LKED.

Copyright © 2016 CA. All rights reserved. All marks used herein are the property of their respective owners.

JCL statements causing error messages.

Information provided in the job log can also be used to isolate JCL statements causing error messages. The errors can be corrected in the EXEC and DD statements that are submitted when you invoke the procedure.

To isolate and correct JCL errors, you may have to examine the following portions of the job log, illustrated in the next slide:

- The system messages.
- The listing of effective JCL.
- The detailed error messages for specific statements.
- The resource-allocation messages for specific statements.

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

JCL error – system messages.

```
JES2 JOB LOG - SYSTEM EPP1 - NODE SPC

09.11.58 JOB0355  TEFC452I - JOB NOT RUN - JCL ERROR
----JES2  JOB  STATISTICS ----
        17 CARDS         READ
        42 SYSOUT        PUNCH    RECORDS
         0 SYSOUT        PUNCH    KBYTES
                         0.00 MINUTES             EXECUTION        TIME
TIME
```

An example of a system message is shown above. System messages appear at the beginning of a job log. They give information such as time of the job, the job number assigned internally, and job statistics.

If the job is not run because of a JCL error, a system message indicating the error will appear in this part of the job log.

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

JCL error messages.

```
STMT NO. MESSAGE

19  IEF6861 DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
35  IEF6861 DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
----------
----------
IEF2371 DMY ALLOCATED TO
```

The example above shows detailed error messages for LA$TEST. These types of messages appear at the end of the job log.

The two warning messages that appear here to draw the user’s attention to unresolved DDNAME keyword operands are JCL statements 19 and 35. In this case, the unresolved operands do not prevent the job from executing. A later resource allocation statement indicates that the data set was assigned dummy status.

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

Are we on track?

Match the items from a job log with the kind of information they can give you about a procedure:

1. Resource allocation messages.
2. Effective JCL.
3. Detailed Error Messages.
4. System Messages.

A. Whether or not a job has run.

B. A specific problem and the statement in which it may appear.

C. The JCL the system has executed.

D. How system resources are used.

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

JCL error – statement number.

```
Error Message:

19  IEF6861 DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVEI

Effective JCL:

18XX              DD  DDNAME=SYSIN
19//JSTEP2        EXEC COBUCL
```

Sometimes the statement number associated with an error message is not the actual statement to which the message applies. In this example, statement 19 is associated with the error message. However, the error actually occurs in statement 18 as shown in the job log.

As the system interpreted the JCL for this job, it expected statement 19 to be a DD statement to assign a value to the DDNAME operand in statement 18. When the system encountered an EXEC statement instead, it created a message for statement 19. However, the reason for the diagnostic actually appears in statement 18.

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

System message – an example.

```
                                                JES2  JOB LOG==SYSTEM EPPI—MODE
   SPC
16.31.28          JOBO3361          LASTEST9 STARTED—INIT 37—CLASST—SYS EPPI
16.31.28          JOBO3361          IEF4581  LASTEST9 - STARTED
16.31.29          JOBO3361          IEC1301  DD2      DD STATEMENT MISSING
16.31.29          JOBO3361          LASTEST9 ENDED
```

As another example of interpreting error messages, examine the system messages for a job named LASTEST9 as shown above to answer the question on the next slide.

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

Are we on track?

In the example on the previous slide, which one of the following problems is identified by the system messages:

A. The job did not execute.

B. DD statement DD2 is missing.

C. There was a JCL error.

D. An operand was unresolved.

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

JCL statements causing error messages.

Examine the JCL listing for LASTEST9 on the right. Note that the procedure definition does not include a DD statement named DD2.

However, a later addition statement (statement 9) refers to a data set named DDIN in PSTEP1.

```
1.//LASTEST9     JOB 12345,ROSE,...
2.//JOBLIB       DD  DSN=TEST.JCL,
  //                   DISP=SHR
  //TRANSACT     PROC
  //PSTEP1       EXEC PROG=PROG1
  //DD1          DD  DSN=INTRAN,
  //                   DISP=SHR
  //DD3          DD  SYSOUT=A
 . . .
9.//PSTEP1.DDIN DD  DSN=MASTER,
  //                   DISP=SHR
```

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

Are we on track?

Refer to the effective JCL for LASTEST9 on the previous slide. Based on this JCL, which of the following do you think is the likely cause of the error?

A. An incorrectly sequenced addition statement.

B. An invalid name for an addition statement (DDIN, not DD2).

C. An incorrectly sequenced override statement.

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

Glossary.

Unresloved
Not assigned a value at procedure execution.

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

Common JCL errors.

It can be helpful to know the way the system interprets JCL, while tracking the source of JCL errors. The actual cause of the error may differ from the cause identified in a detailed error message.

Common JCL errors made when invoking procedures are:

- **Specifying EXEC statement modifications in an incorrect sequence.**
- **Misspelling keyword parameters.**
- **Specifying override and addition DD statements in an incorrect sequence.**
- **Specifying an invalid name for an override or addition DD statements.**
- **Violating JCL syntax rules.**

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

EXEC statement modifications.

```
09.11.58 JOB0355  TEFC452I - JOB NOT RUN - JCL ERROR
----JES2  JOB  STATISTICS ----
        17 CARDS          READ
        42 SYSOUT         PUNCH RECORDS
         0 SYSOUT         PUNCH KBYTES
        0.00 MINUTES      EXECUTION TIME

-------------------------------------------------------
STMT.NO.          MESSAGE
3   IEFC6111      OVERRIDDEN STEP NOT FOUND IN PROCEDURE
```

A common procedure usage error is EXEC statement modifications that are not in correct sequence, that is, not in procedure step sequence. The messages below indicate the kind of system and error message that might result:

- The system messages listed at the beginning of the job log indicate that the job did not execute because of a JCL error.
- The detailed error message at the end of the job indicates that the step to be overridden was not found in the procedure. The offending statement is statement number 3 of effective JCL.

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

EXEC statement modifications – an example.

An example of the statement of effective JCL listing is on the right. The EXEC statement coded to invoke the procedure contains an error.

The TIME parameter addition for the PSTEP2 EXEC statement of the procedure is specified before the PARM parameter addition for the PSTEP1 EXEC statement. The modifications are not specified in procedure step sequence.

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

Interpreting the JCL.

How does the system interpret the JCL?

When interpreting the JCL for this job, the system scans the procedure step sequence. It passes through PSTEP1, then adds the TIME parameter to PSTEP2.

When it encounters the PARM parameter addition, it cannot find a procedure step named PSTEP1 that follows PSTEP2. It therefore issues the diagnostic that the step to be overridden cannot be found in the procedure.

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

Are we on track?

Review the statement used to invoke the TRANSACT procedure in the previous example:

3. //JSTEP EXEC TRANSACT,TIME.PSTEP2=1,PARM.PSTEP1=NOCHECK

Code a correct JCL statement to invoke the procedure with the specified EXEC statement additions.

//JSTEP EXEC TRANSACT,_____

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

Misspelling of keywords operands.

```
----JES2 JOB LOG - SYSTEM EPP1 - NODE SPC

10.06.51 JOB02702 IEFC452I LA$TEST5 - JOB NOT RUN - JCL ERROR
---JES2 JOB STATITICS---
        20 CARDS          READ
        34 SYSOUT         PRINT RECORDS
         0 SYSOUT         PUNCH RECORDS
         2 SYSOUT         SPOOL KBYTES
        0.00 MINUTES      EXECUTION TIME

STMT NO. MESSAGE
     3 IEFC630I UNIDENTIFIED KEYWORD PATM
```

Another common JCL error is misspelling of keyword operands. The parts of a job log for an in-stream procedure named TRANSACT is shown above. The following information is provided in the job log:

- The system messages listed at the beginning of the job log indicate that the job did not execute because of a JCL error.
- The detailed message at the end of the job indicates an unidentified keyword. The offending statement is statement number 3 of the effective JCL, where PARM is spelt as PATM.

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

Are we on track?

Examine statement number 3 of the effective JCL listing. Identify the error in the EXEC statement coded to invoke the procedure.

```
3. //JSTEP    EXEC   TRANSACT
   //         PARM.PSTEP1=CHECK
```

Code a statement to correct the error.

```
   //JSTEP    EXEC _____________________
```

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

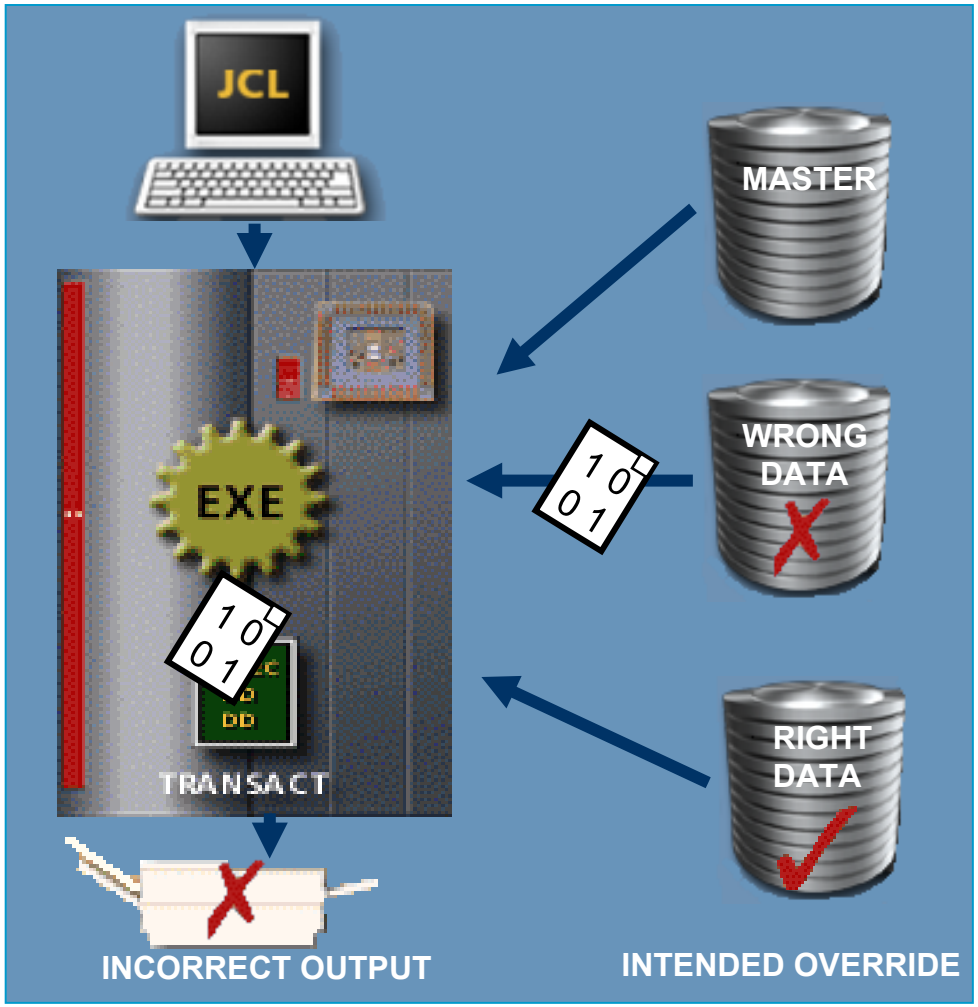
Common source for JCL errors.

Improperly sequenced addition and override statements are common source of JCL errors.

JCL error – An example

If an addition statement is coded before an override statement for the same procedure step, the system will interpret the override statement as another addition.

The procedure may execute, but with the wrong data as illustrated on the right.



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

Sequence of DD statements.

Improper sequence of override and addition DD statements is another common JCL error. They can be more difficult to diagnose than those for EXEC statements.

The specification of an addition DD statement before an override DD statement is a particularly difficult error to isolate. If both types of DD statement are required for the same procedure step, the system does not recognize this as an error. The program executes without detailed error messages. However, the error is reflected in the output of the program, which may be based on incorrect data.

Sequencing rules for coding override and addition DD statements within a procedure step is as follows:

1. Specify all override DD statements for a procedure step in same order as in the procedure.
2. Specify any addition DD statements for that step.

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

Sequence of DD statements – an example.

As an example of the effects of incorrect sequencing, consider the TRANSACT procedure definition, the data set INTRAN is related to DD1.

```
//DD1    DD   DSN=INTRAN,DISP=SHR
```

The user intends to invoke PSTEP1 of the procedure using a data set named NEWTRAN, rather than INTRAN. However, the user incorrectly codes an override statement for PSTEP1 after a valid addition statement.

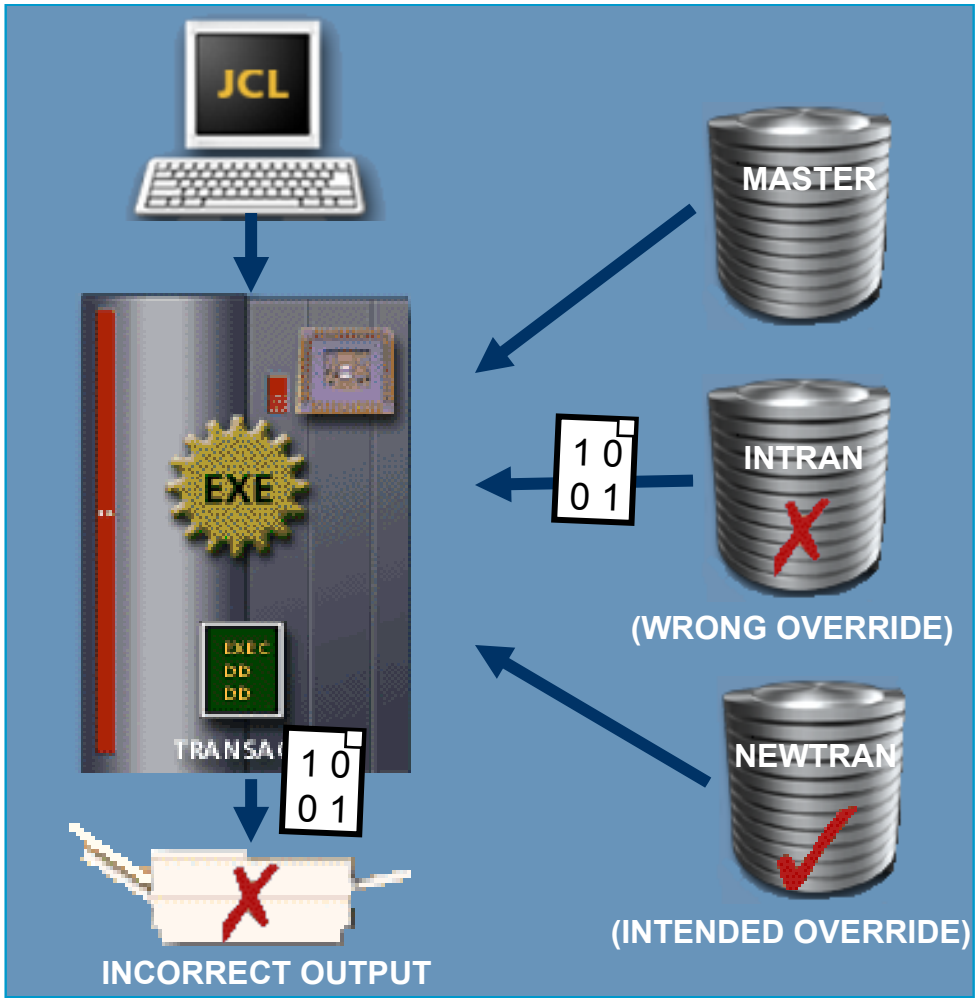
```
3. //JSTEP       EXEC TRANSACT,
   //                PARM.PSTEP1=CHECK
9. //PSTEP1.DD2 DD   DSN=MASTER,...
10.//PSTEP1.DD1 DD   DSN=NEWTRAN,...
```

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

Sequence of DD statements – an example.

The job executes successfully from the system's viewpoint. There are no detailed error messages in the job log. However, the program invoked in PSTEP1 executes using data from the data set related to DD1 in the procedure definition (INTRAN), rather than the data from NEWTRAN.

The system treats statement 10 as another addition statement, because it is specified after rather than before, a previous valid addition statement.



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

Effective JCL – job log.

The effective JCL portion of the job log shown on the right reinforces the conclusion in the example.

The DD1 DD statement (statement number 6) was not overridden. The +/ or X/ notations precede statements that are overridden. That is, the data set name associated with DD1 is still INTRAN.

```
1.//LASTEST      JOB...ROSE,CLASS=B
2.//JOBLIB       DD DSN=TESTJECL,
  //                 DISP=SHR
  //TRANSACT     PROC
  //PSTEP1       EXEC PGM=PROG1
  //DD1          DD
   DSN=INTRAN,DISP=SHR

--------

3.//JSTEP        EXEC TRANSACT,
  //
   PARM.PSTEP1=CHECK
4.++TRANSACT     PROC
5.++PSTEP1       EXEC PGM=PROG1
6.//PSTEP1.DD1   DD DSN=NEWTRAN,
  //                 DISP=SHR
  +/DD1          DD
   DSN=INTRAN,DISP=SHR
```

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

Unit summary.

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

- **Identify the JCL in effect at job execution time by examining a job log.**
- **Specify the parts of a job log that can help you analyze the effective JCL.**
- **Identify and correct common JCL errors that can occur when a procedure is used.**

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

JCL

Chapter b5
Determinig the effective JCL

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

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

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

Chapter b5

Determining the effective JCL

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

Unit introduction.

This unit focuses on identifying and correcting common JCL errors that can occur when a procedure is used.

This unit discusses how to identify an effective JCL (the actual JCL resulting from the use of a procedure) in a message log.

This unit also explains how to look at system messages, error messages and the effective JCL to isolate and correct common JCL errors.

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

Course objectives.

Be able to:

- **Identify the JCL in effect at job execution time by examining a job log.**
- **Specify the parts of a job log that can help you analyze the effective JCL.**
- **Identify and correct common JCL errors that can occur when a procedure is used.**

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

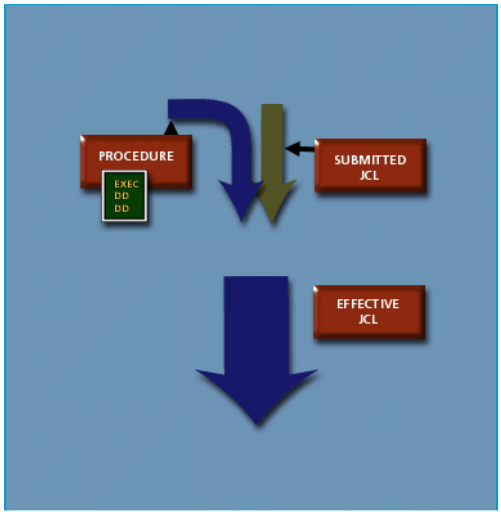
Using a job log.

Effective JCL.

What is an effective JCL?

When a procedure is invoked, the system combines both the submitted JCL and the JCL that is stored with the procedure. The resulting JCL is called the effective JCL.

Effective JCL is the JCL in effect at the time of job execution.



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

Submit MCOE.EDU.JCL.JCL(LASMCLG) and look at JESJCL part of the listing in SYSVIEW.

Examining effective JCL.

An effective JCL is examined for the following reasons:

- **To ensure that the effective JCL satisfies all the processing requirements, particularly if it contains a number of overrides and additions.**
- **To find the source of the problem if a job is not successfully executed.**

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

Using a job log.

Job log.

An effective JCL is checked by examining the job log.

In addition to the listing of effective JCL used during job execution, a job log can also contain the following:

- System messages.
- Detailed error messages for specific statements of effective JCL.
- Resource-allocation/job-step termination-status messages.

System Messages:

```
JES2 JOB LOG--SYSTEM EPP1 - NODE SPC

09.11.58 JOB0355 TEFC452I - JOB NOT RUN-
JCL ERROR
-----JES2 JOB STATISTICS-----
```

Detailed Error Message:

```
STMT NO. MESSAGE
3        IEFC630I UNIDENTIFIED KEYWORD
         PATM
```

Resource Allocation Messages:

```
IEF2371 4BB ALLOCATED TO DD1
```


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

System messages, such as job time and statistics, generally appear at the beginning of a job log. Detailed error messages, organized by statement number, appear at the end.

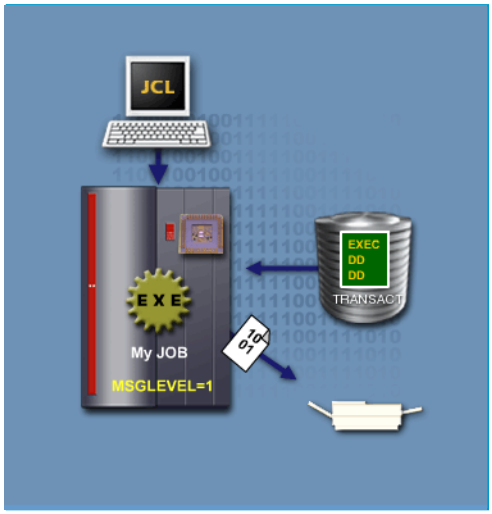
Using a job log.

Requesting a JCL listing.

A programmer can request that procedure statements be listed in a job log by coding the value 1 as the first MSGLEVEL subparameter of the JOB statement.

```
//   JOB    ...MSGLEVEL=1
```

The installation may include procedure statements on a job log by default. If so, using the MSGLEVEL parameter to specify that procedure statements be listed is not required.



11

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

The MSGLEVEL parameter is used to control the printing of JCL statements and allocation messages and listing of the job log for a job. The following job log elements can be controlled:

o The JOB statement.

o All JCL in the job's input stream including all JCL statements and JES2 control (JECL) statements.

o In-stream and cataloged procedure statements for a procedure invoked by a job step.

o JCL substitution and processing messages.

o JES and MVS operator messages concerning the job's processing, including those for allocations of devices/volumes, start/stop of job steps and the job, and the disposition of the data sets used.

Syntax:

MSGLEVEL=({statements}{,messages})

Values for 'statements' in the MSGLEVEL keyword can be one of these:

0 - only JOB statement prints.

1 - all JCL and JES statements & messages print.

2 - only JCL & JES statements print.

Values for 'messages' in the MSGLEVEL keyword can be one of these:

0 - only JCL messages print; if the job ABENDs, then JES messages print too.

1 - JCL and JES messages print.

Using a job log.

Are we on track?

Enter the subparameter that you would code on a job statement to obtain a listing of procedure statements, if they are not included by default:

//MYJOB JOB 123,D.GREEN_____

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

The correct answer is ,MSGLEVEL=1 or ,MSGLEVEL=(1)

Identifying the JCL in effect.

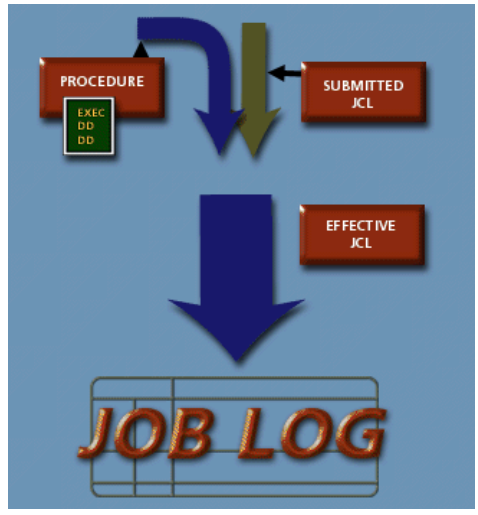
Job log listing.

The listing of a job log can be used to verify if the resulting JCL for the job is what is required.

The job log is also useful to verify the JCL and to distinguish between:

- JCL statements submitted.
- Statements stored in the procedure.
- JCL in effect.

In a job log listing, special notations in columns 1, 2, and 3 distinguish between different categories of statements as shown on the next slide.



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

Identifying the JCL in effect.

Notations in a job log listing.

Notations in Columns 1, 2, and 3		Statement identified
Cataloged Procedure	In-stream Procedure	
//	//	Statements you submit with the job, including in-stream procedure definitions (if applicable) and any alterations to DD statements.
XX	++	A statement in a procedure definition that is used during a job execution.
X/	+/	A DD statement in a procedure definition that you have overridden.
XX*	++*	A statement in a procedure definition, other than a comment statement, that the system considers to be a comment.
***	***	A comment or job entry subsystem control statement.

The table above illustrates the notations used on a job log to identify JCL statements. Special notation in columns 1,2 and 3 of the JCL listing distinguish the different categories of JCL statements.

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

Notations for cataloged procedures are in the first column and notations for in-stream procedures are in the second column.

Identifying the JCL in effect.

Are we on track?

Match the job log notations below with the kinds of statements they identify.

1. +/	**A. JCL statements you submit.**
2. XX	**B. A DD statement in an in-stream procedure that is overridden.**
3. //	**C. A comment or control statement.**
4. ***	**D. A statement from a cataloged procedure that is used.**

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

The correct answer is 1-B, 2-D, 3-A, 4-C

Identifying the JCL in effect.

JCL in effect – In-stream procedure.

The JCL on the right illustrates how to identify the JCL in effect at job execution time when using an in-stream procedure.

In the example, the JCL is for a job LA$TEST2 that invokes an in-stream procedure TRANSACT.

The JCL for LA$TEST2 comprises the following:

- The JOBLIB statement identifies the library storing the programs to be executed.
- The TRANSACT procedure definition appears between the PROC and PEND statements.
- The final statement executes the procedure.

```
//LA$TEST2 JOB 31SPC090156,ROSE,CLASS=B
//JOBLIB   DD DSN=TSOCHIS.TESTJCL.LOAD,
//            DISP=SHR
//TRANSACT PROC
//PSTEP1   EXEC PGM=PROG1
//DD1      DD  DSN=TSOCHIS.INTRAN,
//             DISP=SHR
//DD2      DD  DSN=TSOCHIS.MASTER,
//             DISP=SHR
//DD3      DD  SYSOUT=A
//DD4      DD  DSN=&&VALID,UNIT=SYSDA,
//             DISP=(NEW,PASS),
//             SPACE=(TRK,(1,1))
//PSTEP2   EXEC PGM=PROG2
//DD5      DD  DSN=&&VALID,
//             DISP=(OLD,DELETE)
//DD6      DD  SYSOUT=A
//         PEND
//JSTEP    EXEC TRANSACT
```


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

You can invoke the PROCedure more than once with different parameters. See MCOE.EDU.JCL.JCL(LASMCLG) and its PARM.L='XREF' parameter.

JOBLIB – A special DD statement that identifies in a private library, where programs reside.

Identifying the JCL in effect.

JCL statements in effect.

```
1.  //LA$TEST2     JOB       (31SPCO3090156W),ROSE,CLASS=B
2.  //JOBLIB       DD        DSN=TSOCHIS.TESTJCL.LOAD,DISP=SHR
    //TRANSACT     PROC
    //PSTEP1       EXEC      PGM=PROG1
    //DD1          DD        DSN=TSOCHIS.INTRAN,DISP=SHR
    //DD2          DD        DSN=TSOCHIS.MASTER,DISP=SHR
    //DD3          DD        SYSOUT=A
    //DD4          DD        DSN=&&VALID,UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(1,1))
    //PSTEP2       EXEC      PGM=PROG2
    //DD5          DD        DSN=&&VALID,DISP=(OLD,DELETE)
    //DD6          DD        SYSOUT=A
    //             PEND
3.  //JSTEP        EXEC      TRANSACT
4.  ++TRANSACT     PROC
5.  ++PSTEP1       EXEC      PGM=PROG1
6.  ++DD1          DD        DSN=TSOCHIS.INTRAN,DISP=SHR
7.  ++DD2          DD        DSN=TSOCHIS.MASTER,DISP=SHR
8.  ++DD3          DD        SYSOUT=A
9.  ++DD4          DD        DSN=&&VALID,UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(1,1))
10. ++PSTEP2       EXEC      PGM=PROG2
11. ++DD5          DD        DSN=&&VALID,DISP=(OLD,DELETE)
12. ++DD6          DD        SYSOUT=A
```

The JCL portion of the job log for LA$TEST2 is shown above.

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

The numbers in the left margin (1 through 12) identify the sequence of JCL statements in effect. Notice that the statements that make up the TRANSACT procedure definition (//) are not numbered where they first appear in the job stream. That is because they are not executed at the point at which they are submitted. The system merges the procedure statements into the executable JCL stream (++) at numbers 4 through 12. They follow the EXEC statement (number 3), which invokes the procedure.

Identifying the JCL in effect.

Are we on track?

Review the effective JCL for job LA$TEST2, on the previous page. The // notation identifies the JCL statements that are submitted with the job that invokes the TRANSACT procedure. Which of the following are included?

A. JOB statement.

B. JOBLIB DD statement.

C. DD override statement.

D. TRANSACT procedure definition.

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.

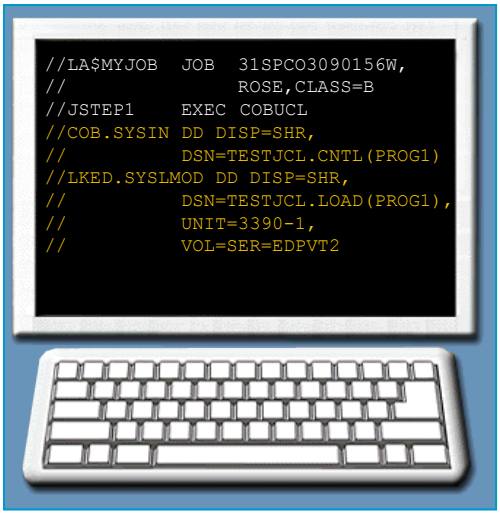
Identifying the JCL in effect.

JCL in effect – Cataloged procedure.

An example of JCL that invokes a cataloged procedure named COBUCL is shown on the right. COB step of COBUCL compiles a COBOL program and LKED link-edits the resulting COBOL object program.

There are addition and override statements for JSTEP1 as follows:

- COB.SYSIN is an addition DD statement that identifies the source module to be compiled.
- LKED.SYSLMOD is an override DD statement that specifies the data set and member name.



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

SYSLMOD – in this data set the link-edited program is passed for execution.

Identifying the JCL in effect.

Effective JCL in a job log.

```
1.  //LA$MYJOB     JOB      3ISPCO3090156W,ROSE,MSGCLASS=T,CLASS=T,MSGLEVEL(1,1)
2.  //JSTEP1       EXEC     COBUCL
3.  XXCOBUCL       PROC
4.  XXCOB          EXEC     PGM=IKFCBL00
5.  XXSYSPRINT     DD       SYSOUT=*
6.  XXSYSUT1       DD       UNIT=SYSDA,SPACE=(CYL,(1,1))
7.  XXSYSUT2       DD       UNIT=SYSDA,SPACE=(CYL,(1,1))
8.  XXSYSUT3       DD       UNIT=SYSDA,SPACE=(CYL,(1,1))
9.  XXSYSUT4       DD       UNIT=SYSDA,SPACE=(CYL,(1,1))
10. XXSYSLIN       DD       DSN=**LOADSET,UNIT=SYSDA,
    XX                      DISP=(MOD,PASS),SPACE=9TRK,93,300,DCB=BLKSIZE=800
11. //COB.SYSIN    DD       DSN=TESTJCL.CNTL(PROG1),DISP=SHR
12. XXLKED         EXEC     PGM=IEW,PARM='LIST,MAP',COND=(5,LT,COB),
. . .
15. //LKED.SYSLMOD DD       DSN=TESTJCL.LOAD(PROG1),UNIT=SYDA,DISP=SHR
    X/SYSLMOD      DD       DSN=&&GOSET,DISP=(,PASS),NIT=SYSDA,SPACE=(CYL,(1,1,1,)0
```

A portion of the effective JCL for LA$MYJOB is shown above. The JCL statements that are submitted are identified by the // notation. The statements from the procedure definition are identified by the XX notation. The addition DD statement for procedure step COB is statement 11. The override statement for procedure step LKED (statement 15) appears at the point where the DD statement to be overridden appears. The statement to be overridden is preceded by the X/ notation.

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

Identifying the JCL in effect.

Are we on track?

Review the effective JCL in job log for LA$MYJOB, on the previous slide. Notice that the system merges the cataloged procedure statements into the job stream at the appropriate places. Order the following statements to reflect the sequence of effective JCL in the example.

A. **Statements of procedure step COB.**

B. **EXEC statement to invoke the procedure.**

C. **Addition statement for COB.**

D. **JOB statement.**

E. **DD statement that is overridden.**

F. **Statement of procedure step LKED.**

G. **Override statement for procedure step LKED.**

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

JCL statements causing error messages.

Information provided in the job log can also be used to isolate JCL statements causing error messages. The errors can be corrected in the EXEC and DD statements that are submitted when you invoke the procedure.

To isolate and correct JCL errors, you may have to examine the following portions of the job log, illustrated in the next slide:

- **The system messages.**
- **The listing of effective JCL.**
- **The detailed error messages for specific statements.**
- **The resource-allocation messages for specific statements.**

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

Interpreting error messages.

JCL error – system messages.

```
JES2 JOB LOG - SYSTEM EPP1 - NODE SPC

09.11.58 JOB0355  TEFC452I - JOB NOT RUN - JCL ERROR
----JES2  JOB  STATISTICS ----
        17 CARDS         READ
        42 SYSOUT        PUNCH    RECORDS
         0 SYSOUT        PUNCH    KBYTES
                         0.00 MINUTES             EXECUTION        TIME
TIME
```

An example of a system message is shown above. System messages appear at the beginning of a job log. They give information such as time of the job, the job number assigned internally, and job statistics.

If the job is not run because of a JCL error, a system message indicating the error will appear in this part of the job log.

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

Interpreting error messages.

JCL error messages.

```
STMT NO. MESSAGE

19 IEF6861 DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
35 IEF6861 DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
----------
----------
IEF2371 DMY ALLOCATED TO
```

The example above shows detailed error messages for LA$TEST. These types of messages appear at the end of the job log.

The two warning messages that appear here to draw the user's attention to unresolved DDNAME keyword operands are JCL statements 19 and 35. In this case, the unresolved operands do not prevent the job from executing. A later resource allocation statement indicates that the data set was assigned dummy status.

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

Intepreting error messages.

Are we on track?

Match the items from a job log with the kind of information they can give you about a procedure:

1. Resource allocation messages.	**A. Whether or not a job has run.**
2. Effective JCL.	**B. A specific problem and the statement in which it may appear.**
3. Detailed Error Messages.	**C. The JCL the system has executed.**
4. System Messages.	**D. How system resources are used.**

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

The correct answer is 1-D, 2-C, 3-B, 4-A

Interpreting error messages.

JCL error – statement number.

```
Error Message:

19 IEF6861 DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVEI

Effective JCL:

18XX             DD  DDNAME=SYSIN
19//JSTEP2       EXEC COBUCL
```

Sometimes the statement number associated with an error message is not the actual statement to which the message applies. In this example, statement 19 is associated with the error message. However, the error actually occurs in statement 18 as shown in the job log.

As the system interpreted the JCL for this job, it expected statement 19 to be a DD statement to assign a value to the DDNAME operand in statement 18. When the system encountered an EXEC statement instead, it created a message for statement 19. However, the reason for the diagnostic actually appears in statement 18.

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

Can I say „dragged“ error?

Interpreting error messages.

System message – an example.

```
                                          JES2  JOB LOG==SYSTEM EPPI—MODE
   SPC
16.31.28         JOBO3361         LASTEST9 STARTED—INIT 37—CLASST—SYS EPPI
16.31.28         JOBO3361         IEF4581  LASTEST9 - STARTED
16.31.29         JOBO3361         IEC1301  DD2      DD STATEMENT MISSING
16.31.29         JOBO3361         LASTEST9 ENDED
```

As another example of interpreting error messages, examine the system messages for a job named LASTEST9 as shown above to answer the question on the next slide.

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

Intepreting error messages.

Are we on track?

In the example on the previous slide, which one of the following problems is identified by the system messages:

A. The job did not execute.

B. DD statement DD2 is missing.

C. There was a JCL error.

D. An operand was unresolved.

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

The correct answer is B.

JCL statements causing error messages.

Examine the JCL listing for LASTEST9 on the right. Note that the procedure definition does not include a DD statement named DD2.

However, a later addition statement (statement 9) refers to a data set named DDIN in PSTEP1.

```
1.//LASTEST9    JOB 12345,ROSE,...
2.//JOBLIB      DD  DSN=TEST.JCL,
  //                DISP=SHR
  //TRANSACT    PROC
  //PSTEP1      EXEC PROG=PROG1
  //DD1         DD  DSN=INTRAN,
  //                DISP=SHR
  //DD3         DD  SYSOUT=A
 . . .
9.//PSTEP1.DDIN DD  DSN=MASTER,
  //                DISP=SHR
```


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

Intepreting error messages.

Are we on track?

Refer to the effective JCL for LASTEST9 on the previous slide. Based on this JCL, which of the following do you think is the likely cause of the error?

A. An incorrectly sequenced addition statement.

B. An invalid name for an addition statement (DDIN, not DD2).

C. An incorrectly sequenced override statement.

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

The correct answer is B.

Glossary.

Unresloved
Not assigned a value at procedure execution.

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

Common JCL errors.

It can be helpful to know the way the system interprets JCL, while tracking the source of JCL errors. The actual cause of the error may differ from the cause identified in a detailed error message.

Common JCL errors made when invoking procedures are:

- **Specifying EXEC statement modifications in an incorrect sequence.**
- **Misspelling keyword parameters.**
- **Specifying override and addition DD statements in an incorrect sequence.**
- **Specifying an invalid name for an override or addition DD statements.**
- **Violating JCL syntax rules.**

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

EXEC statement modifications.

```
09.11.58 JOB0355  TEFC452I - JOB NOT RUN - JCL ERROR
----JES2  JOB  STATISTICS ----
        17 CARDS         READ
        42 SYSOUT        PUNCH RECORDS
         0 SYSOUT        PUNCH KBYTES
        0.00 MINUTES     EXECUTION TIME

-------------------------------------------------------
STMT.NO.         MESSAGE
3   IEFC6111     OVERRIDDEN STEP NOT FOUND IN PROCEDURE
```

A common procedure usage error is EXEC statement modifications that are not in correct sequence, that is, not in procedure step sequence. The messages below indicate the kind of system and error message that might result:

- The system messages listed at the beginning of the job log indicate that the job did not execute because of a JCL error.
- The detailed error message at the end of the job indicates that the step to be overridden was not found in the procedure. The offending statement is statement number 3 of effective JCL.

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

EXEC statement modifications – an example.

An example of the statement of effective JCL listing is on the right. The EXEC statement coded to invoke the procedure contains an error.

The TIME parameter addition for the PSTEP2 EXEC statement of the procedure is specified before the PARM parameter addition for the PSTEP1 EXEC statement. The modifications are not specified in procedure step sequence.

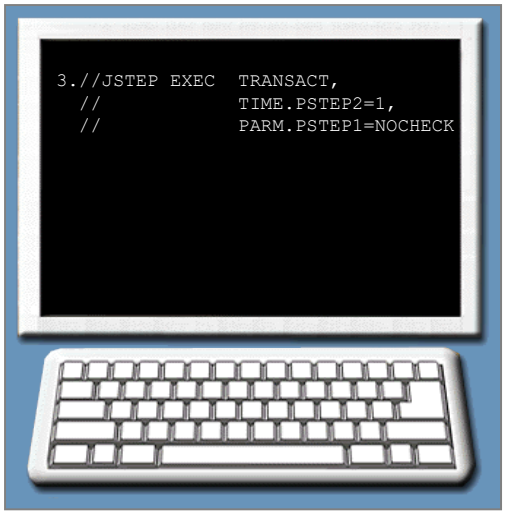
Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Interpreting the JCL.

How does the system interpret the JCL?

When interpreting the JCL for this job, the system scans the procedure step sequence. It passes through PSTEP1, then adds the TIME parameter to PSTEP2.

When it encounters the PARM parameter addition, it cannot find a procedure step named PSTEP1 that follows PSTEP2. It therefore issues the diagnostic that the step to be overridden cannot be found in the procedure.



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

Correcting JCL errors.

Are we on track?

Review the statement used to invoke the TRANSACT procedure in the previous example:

3. //JSTEP EXEC TRANSACT,TIME.PSTEP2=1,PARM.PSTEP1=NOCHECK

Code a correct JCL statement to invoke the procedure with the specified EXEC statement additions.

//JSTEP EXEC TRANSACT,_____

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

The correct answer is PARM.PSTEP1=NOCHECK,TIME.PSTEP2=1

Correcting JCL errors.

Misspelling of keywords operands.

```
----JES2 JOB LOG - SYSTEM EPP1 - NODE SPC

10.06.51 JOB02702 IEFC452I LA$TEST5 - JOB NOT RUN - JCL ERROR
---JES2 JOB STATITICS---
         20 CARDS         READ
         34 SYSOUT        PRINT RECORDS
          0 SYSOUT        PUNCH RECORDS
          2 SYSOUT        SPOOL KBYTES
         0.00 MINUTES     EXECUTION TIME

STMT NO. MESSAGE
     3 IEFC630I UNIDENTIFIED KEYWORD PATM
```

Another common JCL error is misspelling of keyword operands. The parts of a job log for an in-stream procedure named TRANSACT is shown above. The following information is provided in the job log:

- The system messages listed at the beginning of the job log indicate that the job did not execute because of a JCL error.
- The detailed message at the end of the job indicates an unidentified keyword. The offending statement is statement number 3 of the effective JCL, where PARM is spelt as PATM.

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

Correcting JCL errors.

Are we on track?

Examine statement number 3 of the effective JCL listing. Identify the error in the EXEC statement coded to invoke the procedure.

```
3. //JSTEP    EXEC   TRANSACT
   //         PARM.PSTEP1=CHECK
```

Code a statement to correct the error.

```
   //JSTEP    EXEC _____________________
```


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

The correct answer is TRANSACT,PARM.PSTEP1=CHECK

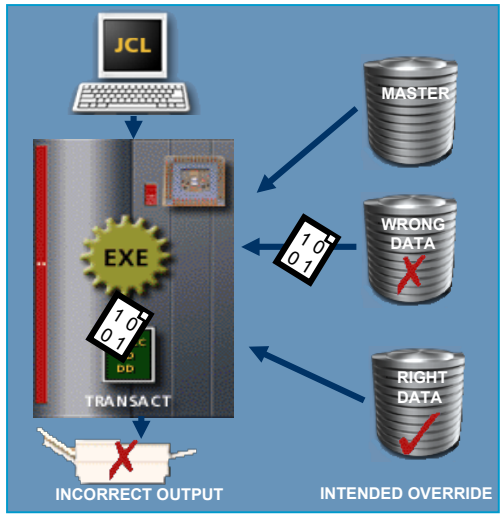
Common source for JCL errors.

Improperly sequenced addition and override statements are common source of JCL errors.

JCL error – An example

If an addition statement is coded before an override statement for the same procedure step, the system will interpret the override statement as another addition.

The procedure may execute, but with the wrong data as illustrated on the right.



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

Sequence of DD statements.

Improper sequence of override and addition DD statements is another common JCL error. They can be more difficult to diagnose than those for EXEC statements.

The specification of an addition DD statement before an override DD statement is a particularly difficult error to isolate. If both types of DD statement are required for the same procedure step, the system does not recognize this as an error. The program executes without detailed error messages. However, the error is reflected in the output of the program, which may be based on incorrect data.

Sequencing rules for coding override and addition DD statements within a procedure step is as follows:

1. **Specify all override DD statements for a procedure step in same order as in the procedure.**
2. **Specify any addition DD statements for that step.**

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

Sequence of DD statements – an example.

As an example of the effects of incorrect sequencing, consider the TRANSACT procedure definition, the data set INTRAN is related to DD1.

```
//DD1   DD  DSN=INTRAN,DISP=SHR
```

The user intends to invoke PSTEP1 of the procedure using a data set named NEWTRAN, rather than INTRAN. However, the user incorrectly codes an override statement for PSTEP1 after a valid addition statement.

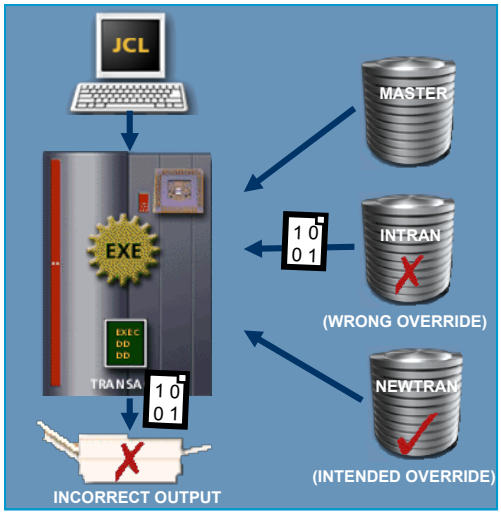
```
3. //JSTEP      EXEC TRANSACT,
   //              PARM.PSTEP1=CHECK
9. //PSTEP1.DD2 DD  DSN=MASTER,...
10.//PSTEP1.DD1 DD  DSN=NEWTRAN,...
```

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

Sequence of DD statements – an example.

The job executes successfully from the system's viewpoint. There are no detailed error messages in the job log. However, the program invoked in PSTEP1 executes using data from the data set related to DD1 in the procedure definition (INTRAN), rather than the data from NEWTRAN.

The system treats statement 10 as another addition statement, because it is specified after rather than before, a previous valid addition statement.



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

Correcting JCL errors.

Effective JCL – job log.

The effective JCL portion of the job log shown on the right reinforces the conclusion in the example.

The DD1 DD statement (statement number 6) was not overridden. The +/ or X/ notations precede statements that are overridden. That is, the data set name associated with DD1 is still INTRAN.

```
1.//LASTEST     JOB...ROSE,CLASS=B
2.//JOBLIB      DD DSN=TESTJECL,
  //                DISP=SHR
  //TRANSACT    PROC
  //PSTEP1      EXEC PGM=PROG1
  //DD1         DD
   DSN=INTRAN,DISP=SHR

--------

3.//JSTEP       EXEC TRANSACT,
  //
   PARM.PSTEP1=CHECK
4.++TRANSACT    PROC
5.++PSTEP1      EXEC PGM=PROG1
6.//PSTEP1.DD1  DD DSN=NEWTRAN,
  //                DISP=SHR
  +/DD1         DD
   DSN=INTRAN,DISP=SHR
```


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

As we can see in the job log, there is no +/ or X/ indication about any overriding – DD1 is still INTRAN.

Unit summary.

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

- **Identify the JCL in effect at job execution time by examining a job log.**
- **Specify the parts of a job log that can help you analyze the effective JCL.**
- **Identify and correct common JCL errors that can occur when a procedure is used.**

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