z/OS Introduction and Workshop

Job Control Language (JCL)

Unit Objectives

After completing this unit, you should be able to:

- Understand purpose of JCL
- Understand JCL JOB, EXEC, and DD statements
- Understand relationship of program file name to JCL DDNAME
- Locate JCL professional manuals, documentation, and online help

“In the beginning ...”

Mainframes prior to S/360 were designed for scientific application number crunching.

The original S/360 hardware was designed from the ground up to meet the needs of business where data throughput capability was greater than the speed of number crunching.

The original OS/360 needed to work with many newly planned Input and Output devices, aka “peripherals”, to handle data throughput, I/O.

Business applications needed to be independent of any peripheral I/O device.

The S/360 and OS/360 design required Device-independent I/O methods.

JCL provided for the requirement of business applications to be independent of the I/O devices.

Job Control Language, JCL

Fred Brooks managed development of System 360 which evolved into today's mainframe

Fred Brooks jokes about JCL saying,

- *“I always tell my students OS/360 Job Control Language is the worst programming language ever designed anywhere by anybody for any purpose and it was done under my management.”*

Computer History Museum Event

https://www.youtube.com/watch?v=8c0_Lzb1CJw

OS/360 JCL – the Worst Language

Done under my management

- One job language for all programming languages
- Like Assembler language, rather than PL/I, etc.
- But not exactly like: card-column dependent
- Too few verbs
- Declarations do verbish things, via parameters
- Awkward branching
- No clean iteration
- No clean subroutine call

- Basic problem was pedestrian vision
  - We did not see it as a schedule-time programming language, but as a “few control cards”
  - **It was not *designed*, it just grew as needs appeared**.

Fred Brooks

JCL, **J**ob **C**ontrol **L**anguage

Computer code that tells the operating system what to do.

Job Control are the best words describing JCL.

The word "Language" in JCL could easily be replaced by "Syntax" or "Commands" or "Statements".

JCL tells the computer what program to execute.

JCL provides a mechanism for the program to read input and write output to requested physical resources.

A separation of internal program file name from the physical resource name

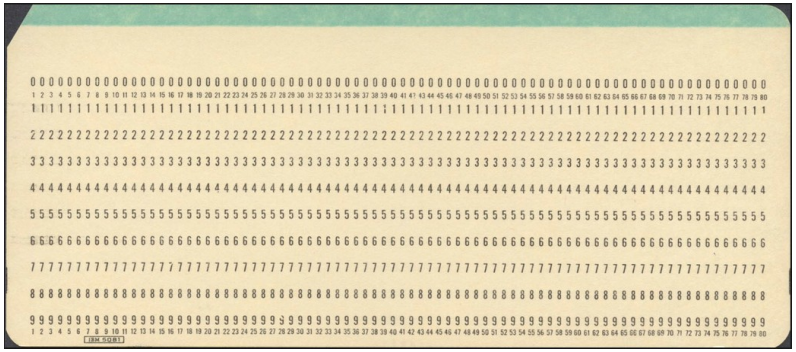
JCL connects internal file name to physical resource name

Sequential Stream of Statements

- Job Control Language (JCL) is a sequential collection of 80 character records beginning with ***//*** which the operating system reads and interprets

- JCL is used to
  - Assign name and authority level
  - Assign resources (programs, data, etc.) and services needed from the operating system to process a task

- JCL can be viewed as a list of statements to be ‘submitted’ for background (batch) processing or ‘started’ for foreground (started task) processing



JCL Statement Fields

80 Bytes

//NAME **OPERATION** **PARAMETER** **SEQ**

Parameter – Positional & Keyword details for the operation

Operation – Type of statement
Most common are **EXEC** and **DD**

Name – Identifies the statement so that other statements and system can refer to it. 8 bytes or less

Identifier starts in column 1
- // (followed by name and/or operation)
- /* (delimiter – end of data)
- //* (comment)
- // (followed by all blanks – null..end of job)

Ignored (73-80)
Sequence numbers

JCL Execute Program Statement

//MYSTEP EXEC PGM=

Parameter named program for the execute operation

Operation is to execute

Name is a user selected “STEPNAME”

STEPNAME label identifies a specific EXEC statement

JCL Execute Procedure

//MYSTEP EXEC PROC=

Parameter is a JCL procedure program for operation to execute. PROC= is optional

Operation is to execute

Name is a user selected “STEPNAME”

STEPNAME label identifies a specific EXEC statement

** JCL PROC creation and execution are discussed in detail in ***//STEP2*** session

JCL Data Definition (DD)

//DDNAME DD DISP=SHR,DSNAME=

Parameters describe the input or output resource

Operation is **D**ata **D**efinition

Name must match spelling of a program file name
Each ddname must be unique within EXEC stepname

Execution

Job Control Language (JCL) instructs z/OS as a result of "submit" or "start " command.

JCL is easily identified by // in column 1 and 2.

JCL is uppercase unless text is enclosed in quote marks such as unix file names.

Every batch JCL job must contain:

JOB statement
EXEC statement

JOB statement marks the beginning of a batch job and assigns a name to the job.

JCL started tasks do not require a JOB statement

EXEC (execute) statement marks the beginning of a job step, assigns a name to the step, and identifies the program or procedure to be executed in the step.

Every batch job and started task has at least one EXEC statement.

JCL (**J**ob **C**ontrol **L**anguage)

z/OS written application *programs* include *internal file names* which are *opened* for reading and writing during execution.

The program hard coded file names are only names that are not associated with any physical resources.

JCL *associates* the *program file name* with physical resources such has disk *data set names* or *unix file names*.

JCL is used to process programs in the background (aka 'batch') and to process programs in the foreground (aka 'started task').

JCL submit will result in batch processing of one or more programs.

JCL start will result in foreground processing of processing program.

Statement Stream

```
//STEP1      EXEC   PGM=MYPGM1
//PGMI       DD     DSN=MY.INPUT.DATA,DISP=SHR
//PGMO       DD     DSN=MY.OUTPUT.DATA,DISP=SHR
//*****
//*  End STEP1 execution and Begin STEP2 execution
//*****
//STEP2      EXEC   PGM=SYSPGM1
//SYSI       DD     DSN=SYS.INPUT.DATA,DISP=SHR
//SYSO       DD     DSN=SYS.OUTPUT.DATA,DISP=SHR
```

PROGRAM INPUTS and OUTPUTS

MYPGM1

```
OPEN FILE PGMI
OPEN FILE PGMO
READ FILE PGMI
WRITE FILE PGMO
```

SYSPGM1

```
OPEN FILE SYSI
OPEN FILE SYSO
READ FILE SYSI
WRITE FILE SYSO
```

```
//STEP1   EXEC   PGM=MYPGM1
//PGMI    DD     DSN=MY.INPUT.DATA,DISP=SHR
//PGMO    DD     DSN=MY.OUTPUT.DATA,DISP=SHR
//*****
//*  End STEP1 execution and Begin STEP2 execution
//*****
//STEP2   EXEC   PGM=SYSPGM1
//SYSI    DD     DSN=SYS.INPUT.DATA,DISP=SHR
//SYSO    DD     DSN=SYS.OUTPUT.DATA,DISP=SHR
```

When you want the program to read from or write to a different physical resource, changing JCL DD statement eliminated need to change program and recompile.

JCL DD Concatentation & Continuation

```
//STEP1      EXEC   PGM=MYPGM1
//PGMI       DD   DSN=MY.INPUT.DATA,DISP=SHR
//           DD   DSN=YOUR.DATA,DISP=SHR
//           DD   DSN=SYS.DATA,DISP=SHR
//PGMO       DD   DSN=MY.OUTPUT.DATA,DISP=SHR
```

DD statement with a blank DDNAME is owned by previous DDNAME
MYPGM1 reads all 3 data sets associated with DDNAME PGMI

```
//STEP1      EXEC   PGM=MYPGM1
//PGMI       DD   DSN=MY.INPUT.DATA,
//   DISP=SHR
//PGMO       DD   DSN=MY.OUTPUT.DATA,
//    DISP=SHR
```

Continuation of JCL operation statement is a comma followed by a space, then the next line begins with ***//*** - one space followed by additional parameters for the JCL operation

JCL JOB Statement – Batch Processing

```
//MYJOB      JOB
//STEP1      EXEC   PGM=MYPGM1
//PGMI       DD     DSN=MY.INPUT.DATA,DISP=SHR
//PGMO       DD     DSN=MY.OUTPUT.DATA,DISP=SHR
//*****
//*  End STEP1 execution and Begin STEP2 execution
//*****
//STEP2      EXEC   PGM=SYSPGM1
//SYSI       DD     DSN=SYS.INPUT.DATA,DISP=SHR
//SYSO       DD     DSN=SYS.OUTPUT.DATA,DISP=SHR
```

- A **job** is a **collection of related job steps** - identified by a **JOB** statement.
- When JCL is submitted using **submit** command, the JCL needs a ***JOB*** statement
- **JOB** statement can be **coded** or system will prompt to **generate** a **JOB** statement

JCL Statement Field Summary

Identifier

Name **Operation** **Parameters**

JOB parameters can consist of local customized accounting information and processing control

```
//MYJOB      JOB
//STEP1      EXEC   PGM=MYPGM1
//PGMI       DD     DSN=MY.INPUT.DATA,DISP=SHR
//PGMO       DD     DSN=MY.OUTPUT.DATA,DISP=SHR
//*****
//*  End STEP1 execution and Begin STEP2 execution
//*****
//STEP2      EXEC   PGM=SYSPGM1
//SYSI       DD     DSN=SYS.INPUT.DATA,DISP=SHR
//SYSO       DD     DSN=SYS.OUTPUT.DATA,DISP=SHR
```

DD Operation Parameters

Examples of Commonly Used Parameters

```
DD   DSN=DATA.SET.NAME,DISP=SHR

DD   DSN=DATA.SET.NAME,DISP=(NEW,CATLG,DELETE),
          SPACE=(CYL,(1,1)),UNIT=3390,VOL=SER=DISK01,
          DCB=(LRECL=80,RECFM=FB,DSORG=PS)

DD   PATH='/u/mypath/myfile',PATHOPTS=(ORDWR,OAPPEND)

DD   PATH='/u/mypath/myfile',PATHOPTS=(OWRONLY,OCREAT),
          PATHMODE=(SIRWXU,SIRGRP,SIROTH)
```

Minimum JCL batch JOB example:

```
//MYJOB  JOB
//       EXEC    PGM=IEFBR14
```

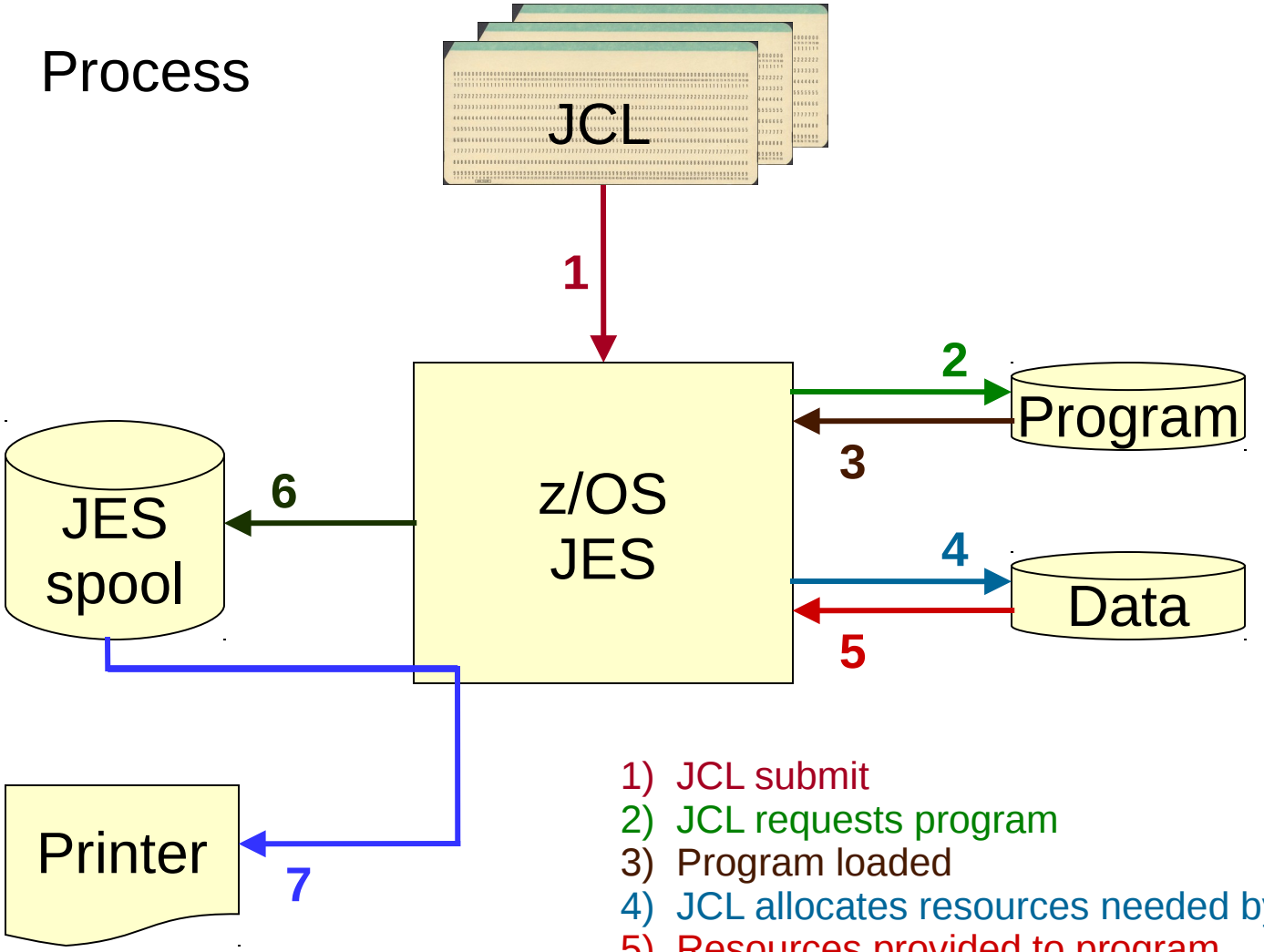
JCL batch job example with stepname of STEP1:

```
//MYJOB  JOB
//STEP1  EXEC    PGM=IEFBR14
```

JCL batch job example with multiple steps:

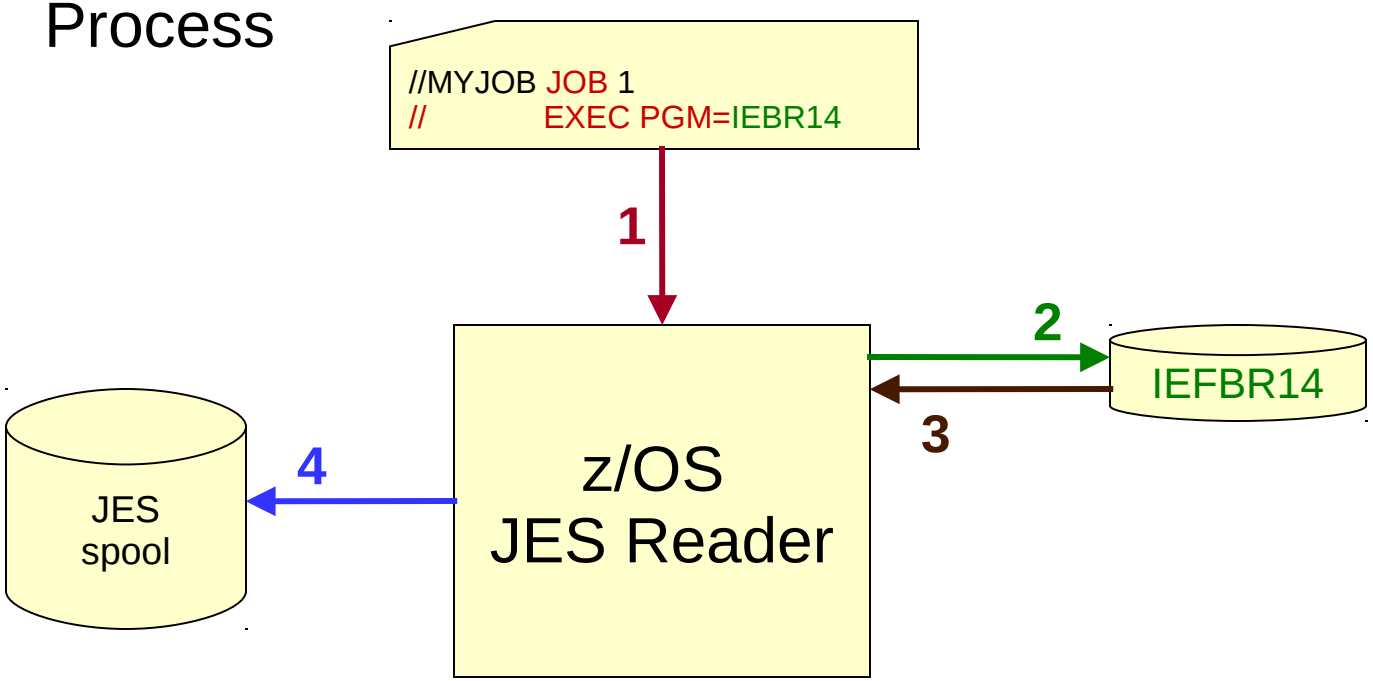
```
//MYJOB  JOB
//STEP1  EXEC    PGM=IEFBR14
//STEP2  EXEC    PGM=IEFBR14
//STEP3  EXEC    PGM=IEFBR14
```

Process



1) JCL submit
2) JCL requests program
3) Program loaded
4) JCL allocates resources needed by program
5) Resources provided to program
6) Program writes output to JES Spool
7) Output to printer as requested

Process



1) JCL submit
2) JCL requests program
3) Program loaded
4) Output written to JES spool

DD (Data Definition) Statements

The program opens DD names as input, output, or both.

The program has an internal file name that will match the JCL DD name.

This association allows different data set names or unix file names to be used by the same program without changing the internal program file name.

When JCL batch job executes, the system writes output to the system controlled JES output queue, data sets and/or unix files as directed by the JCL DD statements

DD Parameters

DD 'parameters' reference z/OS controlled physical resources such as unix file name, data set name and data set status

Examples:

PATH=‘*/unixpath/filename*’ <<<< unix file name reference

DSN=*DATA.SET.NAME* <<<< data set name reference

DISP=(*start*,*end*,*abnormal_end*) <<<< disposition status of data set

Disposition

DISP is an operand of the DD statement

DISP indicates what to do with the data set (the disposition) at step start, end, or abnormal end (if the job fails)

DISP helps to prevent unwanted simultaneous access to data sets, which is very important for general system operation.

DD Resource Disposition Parameter

```
DISP=status
DISP=(status,normal_end)
DISP=(status,normal_end,abnormal_end)
```

where ‘status’ can be

- NEW
- OLD
- SHR
- MOD

where ‘normal_end’ can be:

- DELETE
- KEEP
- PASS
- CATLG
- UNCATLG

where ‘abnormal_end’ can be:

- DELETE
- KEEP
- CATLG
- UNCATLG

JCL DD Operation Parameters

In addition to the JOB and EXEC statements, jobs may contain one or more DD (Data Definition) statements used to identify and characterize the program input and output.

Example:

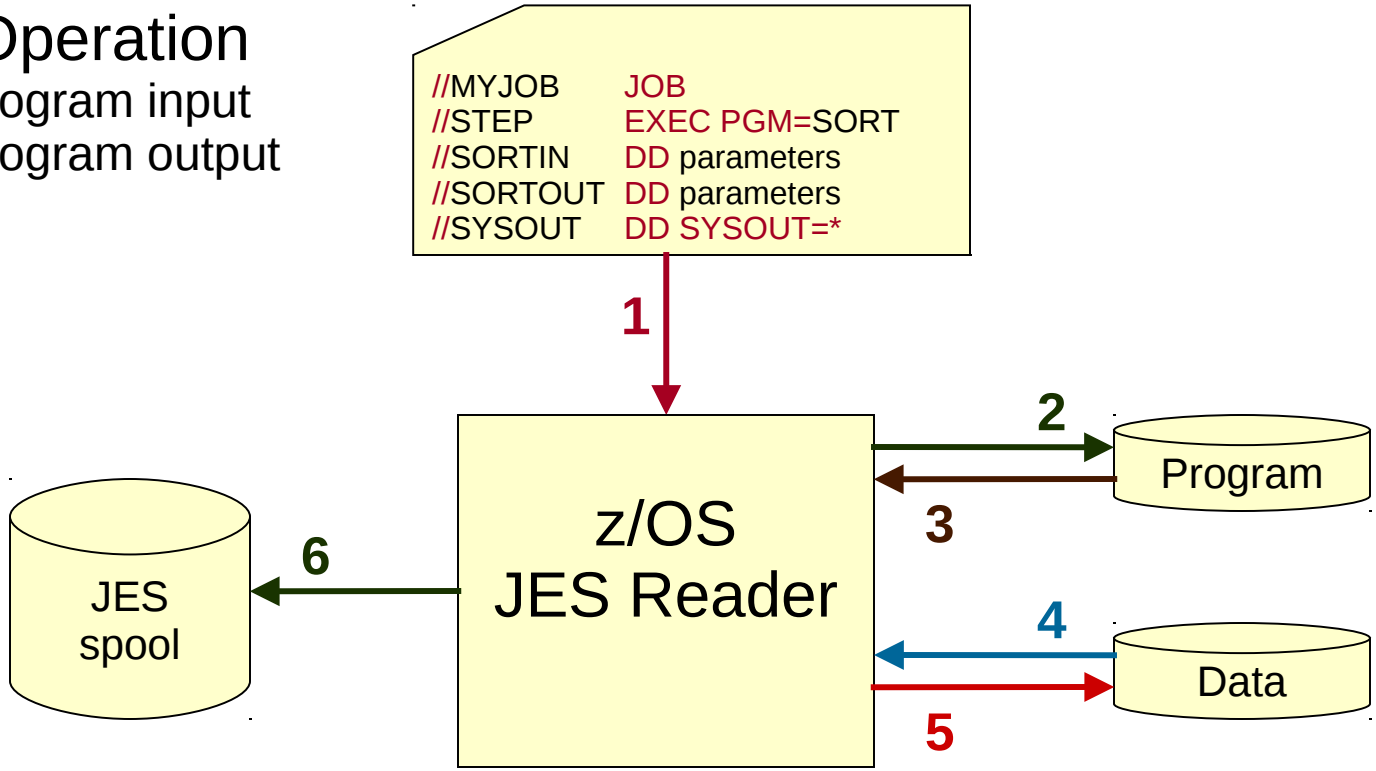
```
//MYJOB      JOB
//STEP       EXEC PGM=SORT
//SORTIN     DD   parameters
//SORTOUT    DD   parameters
//SYSIN      DD   parameters
//SYSOUT     DD   parameters
```

JCL keyword DD is preceded by a 'DD name'.

The above JCL example has 4 'DD names',
SORTIN
SORTOUT
SYSIN
SYSOUT

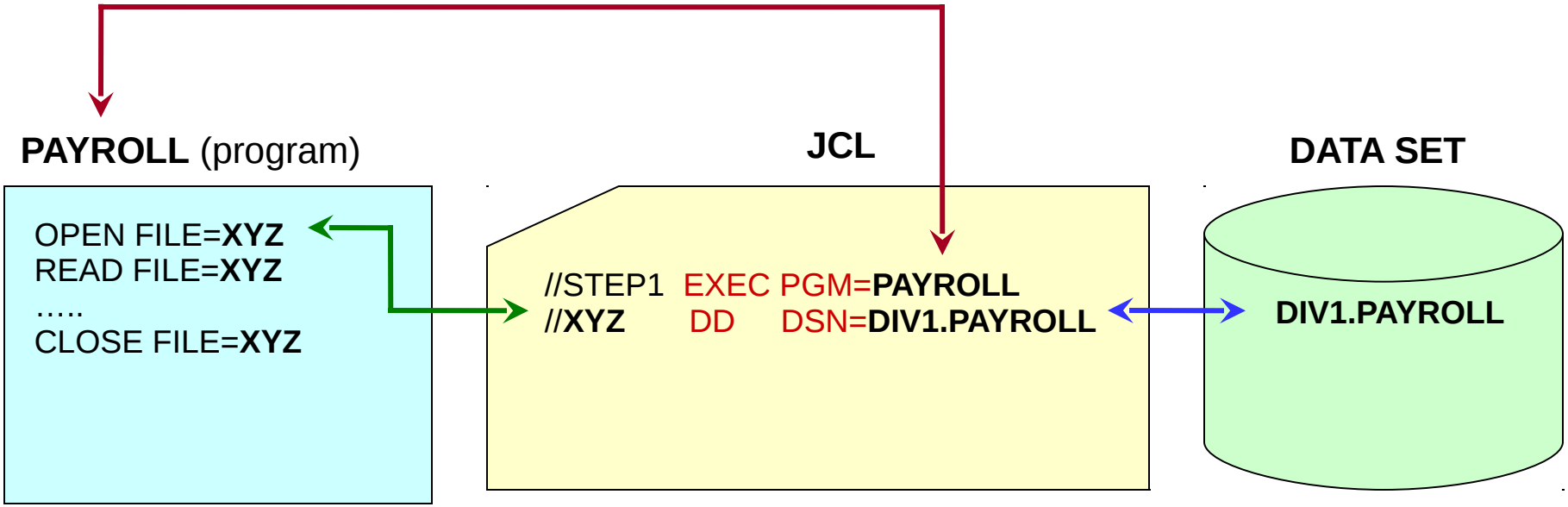
DD Operation

program input
program output



1) JCL submit
2) JCL requests program
3) Program loaded
4) JCL //SORTIN DD
5) JCL //SORTOUT DD
6) JCL //SYSOUT DD SYSOUT=*

JCL Referenced DDNAME

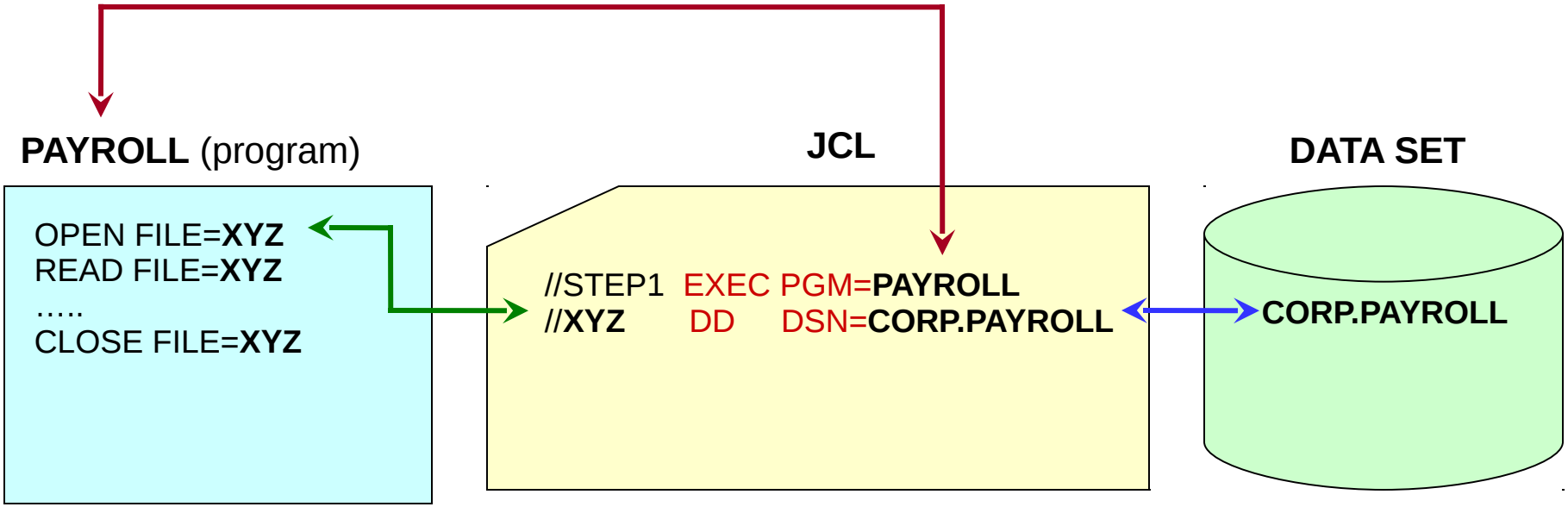


JCL is used to **connect** program **file name** to a z/OS **physical resource** such as a data set name, unix file name, JES spool, printer, network device, etc.

//STEP1	EXEC	PGM=PAYROLL *results in open file=xyz*
//XYZ	DD	DSN=DIV1.PAYROLL *is xyz content read by the program*

DD is abbreviation for **D**ata **D**efinition
XYZ in this example is a program file name
XYZ in this example is also known as the JCL **DDNAME**

JCL Referenced DDNAME



JCL enables ability for same program to read a different z/OS physical resource without changing the program source code

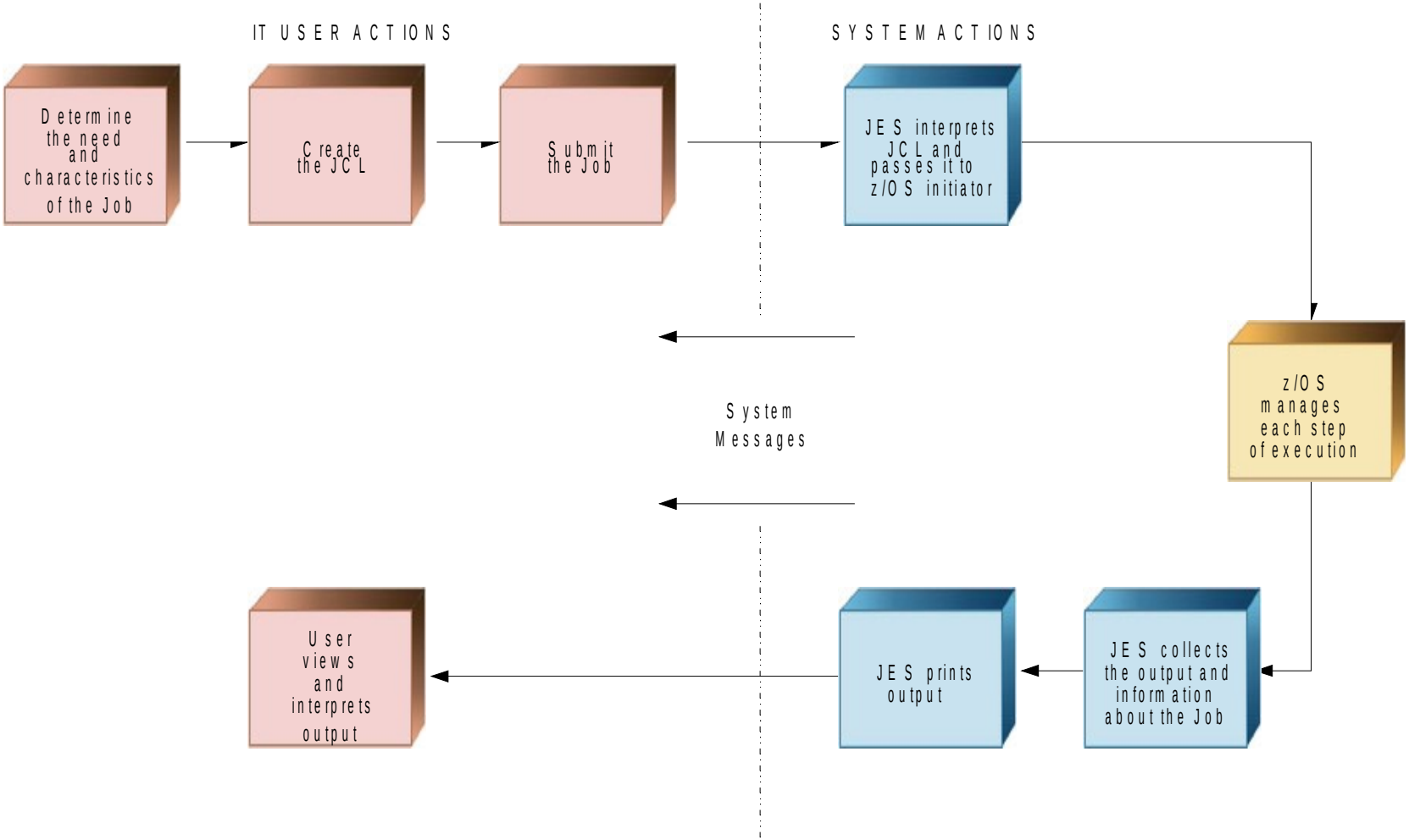
JCL - What is JES? Job Entry Subsystem

In the z/OS operating system, JES manages the input and output job queues and data.

JES handles the following aspects of JCL processing for z/OS:

- Reads JCL job into the operating system
- Interprets the JCL (variable substitution, etc.)
- Schedules job for processing
- Controls job output processing

JCL, JES & Batch Processing



JCL - Basic Syntax Review

```
//JOBNAME      JOB
//STEPNAME     EXEC
//DDNAME       DD
//*            … this is a comment statement
/*             … this indicates end of data
//             … this indicates end for JCL
```

JCL – Example

```
//MYJOB      JOB 1
//MYSORT     EXEC   PGM=SORT
//SORTIN     DD DSN=ZIBM000.JCL(AREACODE),DISP=SHR
//SORTOUT    DD SYSOUT=*
//SYSOUT     DD SYSOUT=*
//SYSIN      DD  *
 SORT FIELDS=(1,3,CH,A)
/*
```

MYJOB is the **jobname**
MYSORT is the **stepname**
SORTIN is program **input**
SORTOUT is program **output**
SYSOUT is system **output** messages
SYSIN is control or data program **input**

JCL JOB Output Listing

```
SDSF STATUS DISPLAY ALL CLASSES
COMMAND INPUT ===>
NP    JOBNAME  JobID    Owner    Prty Queue
      IBMUSER  TSU00858 IBMUSER    15 EXECUTION
s     TEST     JOB00867 IBMUSER     1 PRINT
?     TEST     JOB00869 IBMUSER     1 PRINT
```

S … select all the JCL JOB output

? .... list all the JCL JOB DDNAMEs

View and Understand JCL Job Output

JES2 Dynamically Allocates a few DDNAMEs for each JOB

JESJCLIN
JCL submitted

JESMSGLG
System messages for this job

JESJCL
All job control statements in the input stream

JESYSMSG
JES and operator messages about the job's processing
- allocation of devices and volumes
- execution and termination of job steps and the job
- disposition of data sets

JCL JOB Dynamically Allocated DDNAMEs

```
SDSF JOB DATA SET DISPLAY - JOB TEST     (JOB00867)
COMMAND INPUT ===>
NP   DDNAME   StepName ProcStep DSID Owner    C Dest
s    JESJCLIN                    1 IBMUSER   W
s    JESMSGLG JES2               2 IBMUSER   W LOCAL
s    JESJCL   JES2               3 IBMUSER   W LOCAL
s    JESYSMSG JES2               4 IBMUSER   W LOCAL
```

JESJCLIN Output .. w/JCL error

```
 SDSF OUTPUT DISPLAY TEST     JOB00867  DSID
 COMMAND INPUT ===>
********************************** TOP OF DATA ***
//TEST  JOB 1
//S1    EXEC PGM=IEFBR14
//D1    DD DSN=&SYSUID..JCL,DISP=SHR
//*
//S2    EXEC pgm=IEFBR14
//D2    DD DSN=&SYSUID..OUTPUT,DISP=SHR
//*
//S3    EXEC PGM=IEFBR14
//D3    DD DSN=&SYSUID..LOAD,DISP=SHR
//*
********************************** BOTTOM OF DATA *
```

JESMSGLG Output .. w/JCL error

```
 SDSF OUTPUT DISPLAY TEST     JOB00867  DSID     2 LINE  DATA SET DIS
 COMMAND INPUT ===>                                            SCROLL
********************************** TOP OF DATA ************************
                  J E S 2  J O B  L O G  --  S Y S T E M  S 0 W 1  -

08.28.27 JOB00867 ---- SUNDAY,    24 JUN 2018 ----
08.28.27 JOB00867  IRR010I  USERID IBMUSER  IS ASSIGNED TO THIS JOB.
08.28.27 JOB00867  IEFC452I TEST - JOB NOT RUN - JCL ERROR  530
------ JES2 JOB STATISTICS ------
           10 CARDS READ
           29 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            1 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
********************************** BOTTOM OF DATA *********************
```

JESJCL Output .. w/JCL error

```
 SDSF OUTPUT DISPLAY TEST     JOB00867  DSID     3 LINE  DATA SET I
 COMMAND INPUT ===>                                            SCRO
********************************** TOP OF DATA ********************
        1 //TEST  JOB 1
        2 //S1    EXEC PGM=IEFBR14
        3 //D1    DD DSN=&SYSUID..JCL,DISP=SHR
          //*
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.JCL,DISP=SHR
        4 //S2    EXEC pgm=IEFBR14
        5 //D2    DD DSN=&SYSUID..OUTPUT,DISP=SHR
          //*
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.OUTPUT,DISP=SHR
        6 //S3    EXEC PGM=IEFBR14
        7 //D3    DD DSN=&SYSUID..LOAD,DISP=SHR
          //*
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.LOAD,DISP=SHR
********************************** BOTTOM OF DATA *****************
```

JESYSMSG Output .. w/JCL error

```
SDSF OUTPUT DISPLAY TEST     JOB00867  DSID     4 LINE  DATA SET DIS
COMMAND INPUT ===>                                              SCROLL
********************************* TOP OF DATA ************************
STMT NO. MESSAGE
       4 IEFC620I UNIDENTIFIABLE CHARACTER p ON THE EXEC STATEMENT
       4 IEFC620I UNIDENTIFIABLE CHARACTER g ON THE EXEC STATEMENT
       4 IEFC620I UNIDENTIFIABLE CHARACTER m ON THE EXEC STATEMENT
******************************** BOTTOM OF DATA ***********************
```

JCL JOB Output .. w/JCL error

```
 SDSF OUTPUT DISPLAY TEST     JOB00867  DSID     1 LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
********************************* TOP OF DATA ***********************************
//TEST  JOB 1                                                          JOB00867
//S1    EXEC PGM=IEFBR14
//D1    DD DSN=&SYSUID..JCL,DISP=SHR
//*
//S2    EXEC pgm=IEFBR14
//D2    DD DSN=&SYSUID..OUTPUT,DISP=SHR
//*
//S3    EXEC PGM=IEFBR14
//D3    DD DSN=&SYSUID..LOAD,DISP=SHR
//*
                  J E S 2  J O B  L O G  --  S Y S T E M  S 0 W 1  --  N O D E

08.28.27 JOB00867 ---- SUNDAY,    24 JUN 2018 ----
08.28.27 JOB00867  IRR010I  USERID IBMUSER  IS ASSIGNED TO THIS JOB.
08.28.27 JOB00867  IEFC452I TEST - JOB NOT RUN - JCL ERROR  530
------ JES2 JOB STATISTICS ------
           10 CARDS READ
           29 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            1 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
        1 //TEST  JOB 1
        2 //S1    EXEC PGM=IEFBR14
        3 //D1    DD DSN=&SYSUID..JCL,DISP=SHR
          //*
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.JCL,DISP=SHR
        4 //S2    EXEC pgm=IEFBR14
        5 //D2    DD DSN=&SYSUID..OUTPUT,DISP=SHR
          //*
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.OUTPUT,DISP=SHR
        6 //S3    EXEC PGM=IEFBR14
        7 //D3    DD DSN=&SYSUID..LOAD,DISP=SHR
          //*
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.LOAD,DISP=SHR
 STMT NO. MESSAGE
        4 IEFC620I UNIDENTIFIABLE CHARACTER p ON THE EXEC STATEMENT
        4 IEFC620I UNIDENTIFIABLE CHARACTER g ON THE EXEC STATEMENT
        4 IEFC620I UNIDENTIFIABLE CHARACTER m ON THE EXEC STATEMENT
```

JESJCLIN Output

```
 SDSF OUTPUT DISPLAY TEST      JOB00869  DSID
 COMMAND INPUT ===>
********************************** TOP OF DATA ***>
//TEST   JOB 1
//S1     EXEC PGM=IEFBR14
//D1     DD DSN=&SYSUID..JCL,DISP=SHR
//*
//S2     EXEC PGM=IEFBR14
//D2     DD DSN=&SYSUID..OUTPUT,DISP=SHR
//*
//S3     EXEC PGM=IEFBR14
//D3     DD DSN=&SYSUID..LOAD,DISP=SHR
//*
********************************** BOTTOM OF DATA *>
```

JESMSGLG Output

```
 SDSF OUTPUT DISPLAY TEST     JOB00869  DSID     2 LINE  DATA SET DISPLAYED
 COMMAND INPUT ===>                                          SCROLL ===> CSR
********************************** TOP OF DATA **********************************
                  J E S 2  J O B  L O G  --  S Y S T E M  S 0 W 1  --  N O D E

08.38.36 JOB00869 ---- SUNDAY,    24 JUN 2018 ----
08.38.36 JOB00869  IRR010I  USERID IBMUSER  IS ASSIGNED TO THIS JOB.
08.38.36 JOB00869  ICH70001I IBMUSER  LAST ACCESS AT 08:27:48 ON SUNDAY, JUNE 24
08.38.36 JOB00869  $HASP373 TEST     STARTED - INIT 1    - CLASS A        - SYS
08.38.36 JOB00869  -                                     -----TIMINGS (MINS.)--
08.38.36 JOB00869  -STEPNAME PROCSTEP    RC   EXCP   CONN       TCB       SRB  C
08.38.36 JOB00869  -S1                   00      1      0       .00       .00
08.38.36 JOB00869  -S2                   00      2      0       .00       .00
08.38.37 JOB00869  -S3                   00      2      0       .00       .00
08.38.37 JOB00869  -TEST     ENDED.  NAME-                     TOTAL TCB CPU TIM
08.38.37 JOB00869  $HASP395 TEST     ENDED - RC=0000
------ JES2 JOB STATISTICS ------
  24 JUN 2018 JOB EXECUTION DATE
           10 CARDS READ
           78 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
           10 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
********************************** BOTTOM OF DATA ********************************
```

JESJCL Output

```
 SDSF OUTPUT DISPLAY TEST     JOB00869  DSID     3 LINE  DATA SET DISP
 COMMAND INPUT ===>                                          SCROLL
********************************* TOP OF DATA ************************
        1 //TEST  JOB 1
        2 //S1    EXEC PGM=IEFBR14
        3 //D1    DD DSN=&SYSUID..JCL,DISP=SHR
          //*
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.JCL,DISP=SHR
        4 //S2    EXEC PGM=IEFBR14
        5 //D2    DD DSN=&SYSUID..OUTPUT,DISP=SHR
          //*
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.OUTPUT,DISP=SHR
        6 //S3    EXEC PGM=IEFBR14
        7 //D3    DD DSN=&SYSUID..LOAD,DISP=SHR
          //*
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.LOAD,DISP=SHR
********************************* BOTTOM OF DATA *********************
```

JESYSMSG Output

```
 SDSF OUTPUT DISPLAY TEST     JOB00869  DSID     4 LINE  DATA SET DISPLAYED
 COMMAND INPUT ===>                                          SCROLL ===> CSR
********************************** TOP OF DATA ***********************************
ICH70001I IBMUSER  LAST ACCESS AT 08:27:48 ON SUNDAY, JUNE 24, 2018
IEFA111I TEST IS USING THE FOLLOWING JOB RELATED SETTINGS:
         SWA=ABOVE,TIOT SIZE=32K,DSENQSHR=DISALLOW,GDGBIAS=JOB
IEF236I ALLOC. FOR TEST S1
IGD103I SMS ALLOCATED TO DDNAME D1
IEF142I TEST S1 - STEP WAS EXECUTED - COND CODE 0000
IGD104I IBMUSER.JCL                                  RETAINED,  DDNAME=D1
IEF373I STEP/S1      /START 2018175.0838
IEF032I STEP/S1      /STOP  2018175.0838
        CPU:     0 HR  00 MIN  00.00 SEC    SRB:     0 HR  00 MIN  00.00 SEC
        VIRT:     4K  SYS:   228K  EXT:        0K  SYS:    10888K
        ATB- REAL:                   12K  SLOTS:                     0K
             VIRT- ALLOC:      10M SHRD:       0M
IEF236I ALLOC. FOR TEST S2
IEF237I 0D31 ALLOCATED TO D2
IEF142I TEST S2 - STEP WAS EXECUTED - COND CODE 0000
IEF285I   IBMUSER.OUTPUT                               KEPT
IEF285I   VOL SER NOS= VPWRKB.
IEF373I STEP/S2      /START 2018175.0838
IEF032I STEP/S2      /STOP  2018175.0838
        CPU:     0 HR  00 MIN  00.00 SEC    SRB:     0 HR  00 MIN  00.00 SEC
        VIRT:     4K  SYS:   228K  EXT:        0K  SYS:    10884K
        ATB- REAL:                   12K  SLOTS:                     0K
             VIRT- ALLOC:      10M SHRD:       0M
IEF236I ALLOC. FOR TEST S3
IGD103I SMS ALLOCATED TO DDNAME D3
IEF142I TEST S3 - STEP WAS EXECUTED - COND CODE 0000
IGD104I IBMUSER.LOAD                                 RETAINED,  DDNAME=D3
IEF373I STEP/S3      /START 2018175.0838
IEF032I STEP/S3      /STOP  2018175.0838
        CPU:     0 HR  00 MIN  00.00 SEC    SRB:     0 HR  00 MIN  00.00 SEC
        VIRT:     4K  SYS:   228K  EXT:        0K  SYS:    10884K
        ATB- REAL:                   12K  SLOTS:                     0K
             VIRT- ALLOC:      10M SHRD:       0M
IEF375I  JOB/TEST    /START 2018175.0838
IEF033I  JOB/TEST    /STOP  2018175.0838
        CPU:     0 HR  00 MIN  00.00 SEC    SRB:     0 HR  00 MIN  00.00 SEC
********************************** BOTTOM OF DATA ********************************
```

Advanced JCL Features

- JCL Procedures (PROCs)
- PROC Overrides
- Temporary Data Sets
- Referback
- IF, THEN, ELSE, ENDIF
- SET
- JCLLIB
- INCLUDE
- COMMAND
- XMIT, OUTPUT
- In-Stream Data Variable Substitution
- Impact of Storage Management Subsystem, SMS
- Useful JOB statement parameters
- DCB
- JECL

JCL Procedures

// PROC

Begin JCL procedure

- in-stream
- cataloged

// PEND

End JCL procedure

JCL Procedures (PROC to PEND)

```
//MYJOB      JOB 1
//MYPROC     PROC
//MYSORT     EXEC PGM=SORT
//SORTIN     DD   DSN=&SORTDSN,DISP=SHR
//SORTOUT    DD   SYSOUT=*
//SYSOUT     DD   SYSOUT=*
//           PEND
```

JCL Procedures (continued)

```
//MYJOB       JOB 1
//*-------------------------------------------------------------------------------------------*
//MYPROC      PROC
//MYSORT      EXEC PGM=SORT
//SORTIN      DD   DSN=&SORTDSN,DISP=SHR
//SORTOUT     DD   SYSOUT=*
//SYSOUT      DD   SYSOUT=*
//            PEND
//*-------------------------------------------------------------------------------------------*
//STEP1       EXEC MYPROC,SORTDSN=ZIBM000.JCL(AREACODE)
//SYSIN       DD   *
 SORT FIELDS=(1,3,CH,A)
```

JCL Procedures – Statement Override

```
//MYJOB        JOB 1
//*
//MYPROC       PROC
//MYSORT       EXEC   PGM=SORT
//SORTIN       DD     DSN=&SORTDSN,DISP=SHR
//SORTOUT      DD     SYSOUT=*
//SYSOUT       DD     SYSOUT=*
//             PEND
//*
//STEP1        EXEC MYPROC,SORTDSN=IBMUSER.AREA.CODES
//MYSORT.SORTOUT  DD DSN=IBMUSER.MYSORT.OUTPUT,
//                   DISP=(NEW,CATLG),
//                   SPACE=(CYL,(1,1)),
//                   UNIT=SYSDA,VOL=SER=VPWRKA
//SYSIN        DD  *
 SORT FIELDS=(1,3,CH,A)
```

View and Understand JCL Job Output

JES2 Dynamically Allocated DDNAMEs for each JOB

JESJCLIN
JCL submitted

JESMSGLG
System messages for this job

JESJCL
All job control statements in the input stream

JESYSMSG
JES and operator messages about the job's processing
- allocation of devices and volumes
- execution and termination of job steps and the job
- disposition of data sets

In-stream JCL Procedure Statements in (JESJCL)

++ ... DD statement that was **not overridden** and all other JCL statements, except the JCL comment statement. Each statement appears in the listing exactly as it appears in the procedure.

+/ ... DD statement that was **overridden** (preceded by the overriding DD statement)

++* – **Comment** statement or considered comment

In-stream JCL Procedure (JESJCLIN)

```
 SDSF OUTPUT DISPLAY SORTJOB  JOB00874  DSID     1 LINE 0
 COMMAND INPUT ===>
********************************** TOP OF DATA **************
//SORTJOB JOB 1,NOTIFY=&SYSUID
//*---------------------------*/
//MYPROC  PROC
//MYSORT  EXEC PGM=SORT
//SYSOUT   DD SYSOUT=*
//SORTOUT  DD SYSOUT=*
//SORTIN   DD DISP=SHR,DSN=&SORTDSN
//        PEND
//*---------------------------*/
//STEP1   EXEC MYPROC,SORTDSN=CLASS.LAB.JCL(AREACODE)
//MYSORT.SORTOUT DD DSN=&SYSUID..SORT.OUTPUT,
//          DISP=(NEW,CATLG),SPACE=(CYL,(1,1)),UNIT=SYSDA,
//          DCB=(LRECL=20,BLKSIZE=0,RECFM=FB,DSORG=PS)
//SYSIN    DD *
********************************** BOTTOM OF DATA ************
```

In-stream JCL Procedure (JESMSGLG)

```
 SDSF OUTPUT DISPLAY SORTJOB  JOB00874  DSID     2 LINE  DATA SET DISPLAYED
 COMMAND INPUT ===>                                          SCROLL ===> CSR
********************************** TOP OF DATA **********************************
                  J E S 2  J O B  L O G  --  S Y S T E M  S 0 W 1  --  N O D E

10.11.07 JOB00874 ---- SUNDAY,    24 JUN 2018 ----
10.11.07 JOB00874  IRR010I  USERID IBMUSER  IS ASSIGNED TO THIS JOB.
10.11.07 JOB00874  ICH70001I IBMUSER  LAST ACCESS AT 08:38:36 ON SUNDAY, JUNE 24
10.11.07 JOB00874  $HASP373 SORTJOB  STARTED - INIT 1    - CLASS A        - SYS
10.11.08 JOB00874  -                                     -----TIMINGS (MINS.)--
10.11.08 JOB00874  -STEPNAME PROCSTEP    RC   EXCP   CONN       TCB       SRB  C
10.11.08 JOB00874  -STEP1    MYSORT      00     43      3       .00       .00
10.11.08 JOB00874  -SORTJOB  ENDED.  NAME-                     TOTAL TCB CPU TIM
10.11.08 JOB00874  $HASP395 SORTJOB  ENDED - RC=0000
------ JES2 JOB STATISTICS ------
  24 JUN 2018 JOB EXECUTION DATE
           16 CARDS READ
          111 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
           10 SYSOUT SPOOL KBYTES
         0.01 MINUTES EXECUTION TIME
********************************** BOTTOM OF DATA ********************************
```

In-stream JCL Procedure (JESJCL)

```
 SDSF OUTPUT DISPLAY SORTJOB  JOB00874  DSID     3 LINE  DATA SET DISPLAYED
 COMMAND INPUT ===>                                          SCROLL ===> CSR
********************************** TOP OF DATA **********************************
        1 //SORTJOB JOB 1,NOTIFY=&SYSUID
          //*---------------------------*/
          IEFC653I SUBSTITUTION JCL - 1,NOTIFY=IBMUSER
        2 //MYPROC  PROC
          //MYSORT  EXEC PGM=SORT
          //SYSOUT   DD SYSOUT=*
          //SORTOUT  DD SYSOUT=*
          //SORTIN   DD DISP=SHR,DSN=&SORTDSN
          //        PEND
          //*---------------------------*/
        3 //STEP1   EXEC MYPROC,SORTDSN=CLASS.LAB.JCL(AREACODE)
        4 ++MYPROC  PROC
        5 ++MYSORT  EXEC PGM=SORT
        6 ++SYSOUT   DD SYSOUT=*
        7 //MYSORT.SORTOUT DD DSN=&SYSUID..SORT.OUTPUT,
          //          DISP=(NEW,CATLG),SPACE=(CYL,(1,1)),UNIT=SYSDA,
          //          DCB=(LRECL=20,BLKSIZE=0,RECFM=FB,DSORG=PS)
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.SORT.OUTPUT,DISP=(NEW,CATLG),S
          DCB=(LRECL=20,BLKSIZE=0,RECFM=FB,DSORG=PS)
          +/SORTOUT  DD SYSOUT=*
        8 ++SORTIN   DD DISP=SHR,DSN=&SORTDSN
          IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=CLASS.LAB.JCL(AREACODE)
        9 //SYSIN    DD *
******************************** BOTTOM OF DATA *********************************
```

In-stream JCL Procedure (JESYSMSG)

```
 ------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY SORTJOB  JOB00874  DSID     4 LINE  DATA SET DISPLAYED
 COMMAND INPUT ===>                                          SCROLL ===> CSR
********************************** TOP OF DATA ***********************************
 STMT NO. MESSAGE
        3 IEFC001I PROCEDURE MYPROC WAS EXPANDED USING INSTREAM PROCEDURE DEFINI
ICH70001I IBMUSER  LAST ACCESS AT 08:38:36 ON SUNDAY, JUNE 24, 2018
IEFA111I SORTJOB IS USING THE FOLLOWING JOB RELATED SETTINGS:
         SWA=ABOVE,TIOT SIZE=32K,DSENQSHR=DISALLOW,GDGBIAS=JOB
IEF236I ALLOC. FOR SORTJOB MYSORT STEP1
IEF237I JES2 ALLOCATED TO SYSOUT
IGD100I 0D32 ALLOCATED TO DDNAME SORTOUT  DATACLAS (        )
IEF237I 0D30 ALLOCATED TO SORTIN
IEF237I JES2 ALLOCATED TO SYSIN
IEF237I 0D30 ALLOCATED TO SYS00001
IEF285I   CLASS.LAB.JCL                                KEPT
IEF285I   VOL SER NOS= VPWRKA.
IEF142I SORTJOB MYSORT STEP1 - STEP WAS EXECUTED - COND CODE 0000
IEF285I   IBMUSER.SORTJOB.JOB00874.D0000102.?          SYSOUT
IEF285I   IBMUSER.SORT.OUTPUT                          CATALOGED
IEF285I   VOL SER NOS= VPWRKC.
IEF285I   CLASS.LAB.JCL                                KEPT
IEF285I   VOL SER NOS= VPWRKA.
IEF285I   IBMUSER.SORTJOB.JOB00874.D0000101.?          SYSIN
IEF373I STEP/MYSORT  /START 2018175.1011
IEF032I STEP/MYSORT  /STOP  2018175.1011
        CPU:     0 HR  00 MIN  00.01 SEC    SRB:     0 HR  00 MIN  00.00 SEC
        VIRT:  1068K  SYS:   268K  EXT:     6160K  SYS:    24140K
        ATB- REAL:                   48K  SLOTS:                     0K
             VIRT- ALLOC:      16M SHRD:       0M
IEF375I  JOB/SORTJOB /START 2018175.1011
IEF033I  JOB/SORTJOB /STOP  2018175.1011
        CPU:     0 HR  00 MIN  00.01 SEC    SRB:     0 HR  00 MIN  00.00 SEC
******************************** BOTTOM OF DATA ***********************************
```

Cataloged JCL Procedure Statement in (JESJCL)

XX ... DD statement that was **not overridden** and all other JCL statements,except the JCL comment statement. Each statement appears in the listing exactly as it appears in the procedure

X/ ... DD statement that was **overridden** (preceded by the overriding DD statement)

XX* ... JCL **comment** statement or consider comment

Cataloged JCL Procedure (JESJCLIN)

```
 SDSF OUTPUT DISPLAY SORTJOB  JOB00875  DSID     1 LINE 0
 COMMAND INPUT ===>
********************************** TOP OF DATA ***********
//SORTJOB JOB 1,NOTIFY=&SYSUID
//STEP1   EXEC MYPROC,SORTDSN=CLASS.LAB.JCL(AREACODE)
//MYSORT.SORTOUT DD DSN=&SYSUID..SORT.OUTPUT,DISP=SHR
//SYSIN    DD *
********************************** BOTTOM OF DATA *********
```

Cataloged JCL Procedure (JESMSGLG)

```
 SDSF OUTPUT DISPLAY SORTJOB  JOB00875  DSID     2 LINE  DATA SET DISPLAYED
 COMMAND INPUT ===>                                          SCROLL ===> CSR
********************************** TOP OF DATA ***********************************
                  J E S 2  J O B  L O G  --  S Y S T E M  S 0 W 1  --  N O D E

10.15.17 JOB00875 ---- SUNDAY,    24 JUN 2018 ----
10.15.17 JOB00875  IRR010I  USERID IBMUSER  IS ASSIGNED TO THIS JOB.
10.15.17 JOB00875  ICH70001I IBMUSER  LAST ACCESS AT 10:11:07 ON SUNDAY, JUNE 24
10.15.17 JOB00875  $HASP373 SORTJOB  STARTED - INIT 1    - CLASS A        - SYS
10.15.18 JOB00875  -                                     -----TIMINGS (MINS.)--
10.15.18 JOB00875  -STEPNAME PROCSTEP    RC   EXCP   CONN       TCB       SRB  C
10.15.18 JOB00875  -STEP1    MYSORT      00     60      3       .00       .00
10.15.18 JOB00875  -SORTJOB  ENDED.  NAME-                     TOTAL TCB CPU TIM
10.15.18 JOB00875  $HASP395 SORTJOB  ENDED - RC=0000
------ JES2 JOB STATISTICS ------
  24 JUN 2018 JOB EXECUTION DATE
           6 CARDS READ
         105 SYSOUT PRINT RECORDS
           0 SYSOUT PUNCH RECORDS
          10 SYSOUT SPOOL KBYTES
        0.00 MINUTES EXECUTION TIME
********************************** BOTTOM OF DATA ********************************
```

Cataloged JCL Procedure (JESJCL)

```
 ------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY SORTJOB  JOB00875  DSID     3 LINE  DATA SET DISPLAYED
 COMMAND INPUT ===>                                          SCROLL ===> CSR
********************************* TOP OF DATA ***********************************
        1 //SORTJOB JOB 1,NOTIFY=&SYSUID
          IEFC653I SUBSTITUTION JCL - 1,NOTIFY=IBMUSER
        2 //STEP1   EXEC MYPROC,SORTDSN=CLASS.LAB.JCL(AREACODE)
        3 XXMYPROC  PROC
        4 XXMYSORT  EXEC PGM=SORT
        5 XXSYSOUT   DD SYSOUT=*
        6 XXSORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5,1))
        7 //MYSORT.SORTOUT DD DSN=&SYSUID..SORT.OUTPUT,DISP=SHR
          IEFC653I SUBSTITUTION JCL - DSN=IBMUSER.SORT.OUTPUT,DISP=SHR
          X/SORTOUT  DD SYSOUT=*
        8 XXSORTIN   DD DISP=SHR,DSN=&SORTDSN
          IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=CLASS.LAB.JCL(AREACODE)
        9 //SYSIN    DD *
       10 XX        PEND
********************************* BOTTOM OF DATA ********************************
```

Cataloged JCL Procedure (JESYSMSG)

```
 SDSF OUTPUT DISPLAY SORTJOB  JOB00875  DSID     4 LINE  DATA SET DISPLAYED
 COMMAND INPUT ===> [ ]                                   SCROLL ===> CSR
********************************** TOP OF DATA ***********************************
 STMT NO. MESSAGE
        2 IEFC001I PROCEDURE MYPROC WAS EXPANDED USING SYSTEM LIBRARY VENDOR.PRO
ICH70001I IBMUSER  LAST ACCESS AT 10:11:07 ON SUNDAY, JUNE 24, 2018
IEFA111I SORTJOB IS USING THE FOLLOWING JOB RELATED SETTINGS:
         SWA=ABOVE,TIOT SIZE=32K,DSENQSHR=DISALLOW,GDGBIAS=JOB
IEF236I ALLOC. FOR SORTJOB MYSORT STEP1
IEF237I JES2 ALLOCATED TO SYSOUT
IGD100I VIO ALLOCATED TO DDNAME SORTWK01 DATACLAS (        )
IEF237I 0D32 ALLOCATED TO SORTOUT
IEF237I 0D30 ALLOCATED TO SORTIN
IEF237I JES2 ALLOCATED TO SYSIN
IEF237I 0D30 ALLOCATED TO SYS00001
IEF285I   CLASS.LAB.JCL                                KEPT
IEF285I   VOL SER NOS= VPWRKA.
IEF142I SORTJOB MYSORT STEP1 - STEP WAS EXECUTED - COND CODE 0000
IEF285I   IBMUSER.SORTJOB.JOB00875.D0000102.?          SYSOUT
IEF285I   SYS18175.T101517.RA000.SORTJOB.R0101290      DELETED
IEF285I   IBMUSER.SORT.OUTPUT                          KEPT
IEF285I   VOL SER NOS= VPWRKC.
IEF285I   CLASS.LAB.JCL                                KEPT
IEF285I   VOL SER NOS= VPWRKA.
IEF285I   IBMUSER.SORTJOB.JOB00875.D0000101.?          SYSIN
IEF373I STEP/MYSORT  /START 2018175.1015
IEF032I STEP/MYSORT  /STOP  2018175.1015
        CPU:     0 HR  00 MIN  00.01 SEC    SRB:     0 HR  00 MIN  00.00 SEC
        VIRT:  1072K  SYS:   268K  EXT:     6160K  SYS:    24204K
        ATB- REAL:                    36K  SLOTS:                      0K
             VIRT- ALLOC:      14M SHRD:       0M
IEF375I  JOB/SORTJOB /START 2018175.1015
IEF033I  JOB/SORTJOB /STOP  2018175.1015
        CPU:     0 HR  00 MIN  00.01 SEC    SRB:     0 HR  00 MIN  00.00 SEC
********************************** BOTTOM OF DATA ********************************
```

Temporary Data Sets

A temporary data set is a data set that is created and deleted in the same job, and is identified by coding one of the following:

DSNAME=&&dsname

For a temporary data set

DSNAME=&&dsname(member)

For a member of a temporary PDS or PDSE

No DSNAME parameter

For a temporary data set to be named by the system

Referback to an earlier DD statement

If a data set name is used several times in a job, copy it from the DD statement that uses it first.

It can be copied whether it is specified in the DSNAME parameter or assigned by the system.

Use copying to make changing data sets from job to job easier and to eliminate having to assign names to temporary data sets.

Copy a data set name by coding:

//ddname DD DSNAME=*.ddname

//ddname DD DSNAME=*.stepname.ddname

//ddname DD DSNAME=*.stepname.procstepname.ddname

Other Commonly Used JCL Operations

- *//name* IF *(condition)* THEN
- *//name* ELSE
- *//name* ENDIF
- *//name* SET
- *//name* JCLLIB
- *//name* INCLUDE
- *//name* COMMAND
- *//name* XMIT
- more exist

IF, THEN, ELSE, ENDIF

```
//START       EXEC PGM=MYPGM1
//    IF RC=0 THEN
//SUCCESS  EXEC PGM=MYPGM2
//    ELSE
//FAILURE     EXEC PGM=MYPGM3
//    ENDIF
```

//*name* **SET**

Defines and assigns values to symbolic parameters
used when processing JCL statements.

//name **JCLLIB** ORDER=(*names of the libraries to be searched*)

```
//SET1        SET LIB=MY.JCLLIB,D=MY.INPUT.DATA,M=AA
//*
//PRIVATE     JCLLIB ORDER=(&LIB)
//*
//*   search for MYPROC first in MY.JCLLIB
//*
//COPY        EXEC MYPROC
//INDATA      DD     DSN=&D,DISP=SHR
//MOREJCL   INCLUDE MEMBER=&M
```

Standalone JCL Operations

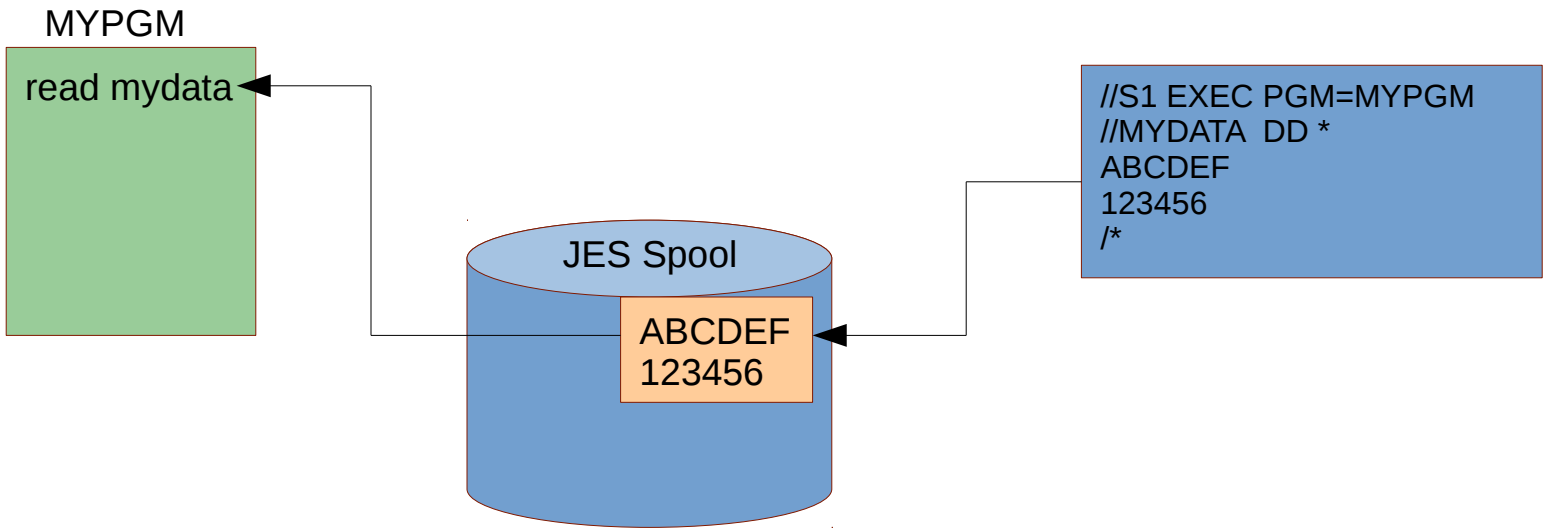
//name **COMMAND** *system_command*

Only if enabled and ID has authority

//name **XMIT** *parameters*

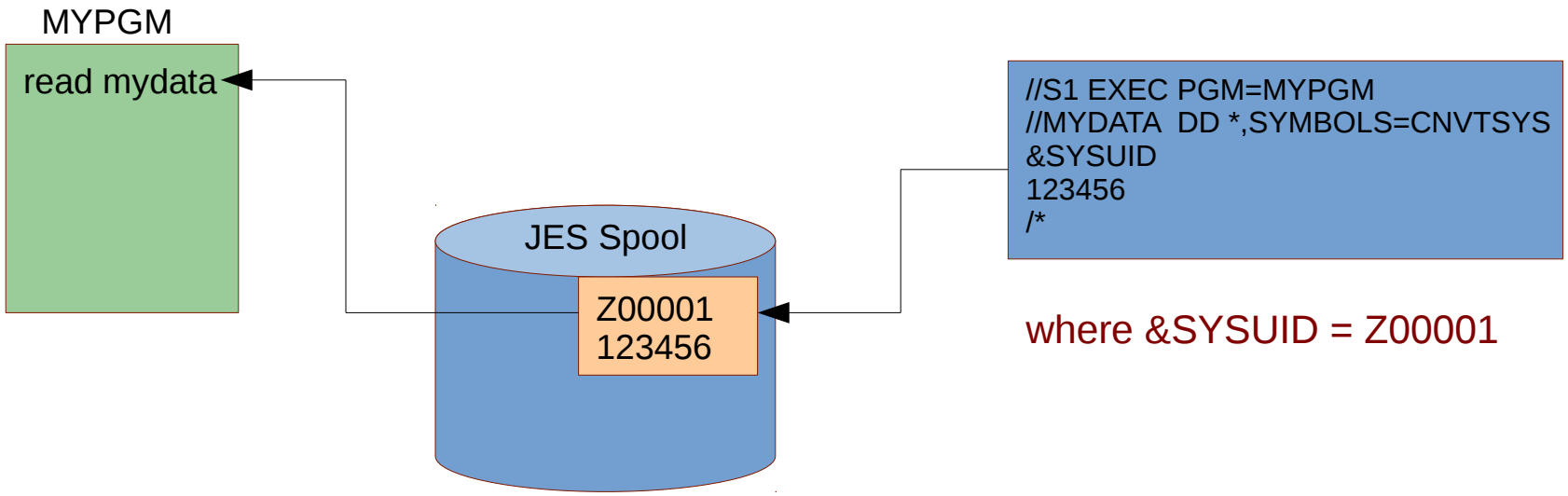
Transmit records to a defined location

JCL DD * uses JES Spool to store data



Variables in the
//name DD *
data stream
/*
is possible with JCL DD SYMBOLS parameter

JCL DD *,SYMBOLS= enable variable conversion



SYMBOLS=**JCLONLY**

JCL symbols and JES symbols found in the in-stream data set are replaced with their values

SYMBOLS=**EXECSYS**

JCLONLY and system system symbols

SYMBOLS=**CNVTSYS**

EXECSYS and substitute variables on the system where conversion occurred

SMS, Storage Management Subsystem

Enables Disk Storage Administrators to simplify JCL DD parameters

Locally documented JCL procedures and policy

ACS, Automatic Class Selection, routine parses and changes JCL

Routine assigns DD operands based upon any of the following:

1) Data Set Name
2) DATACLAS=
3) MGMTCLAS=
4) STORCLAS=

Routine discards user JCL DD operands and substitute different JCL DD parameters

Useful JOB Statement Parameters

TYPRUN=

SCAN	check JCL syntax
HOLD	hold until command to release
JCLHOLD	JES2 hold until command to release
COPY	copy JCL to output without processing

NOTIFY=

&SYSUID	any valid ID

TIME= modify default processing time

REGION= modify default processing memory
MEMLIMIT=

PAGES= modify default output volume
LINES=

EMAIL=

Numerous more

DD Operation DCB= parameter

Used to assign attributes to a resource such as a data set name

- Logical Record Length
- Record Format
- Data Set Organization

DCB, Data Control Block, operands

- LRECL=
- RECFM=
- DSORG=

Assembler Macro

LIKE= parameter exists for newly allocated data set names

JES JECL Statements

JES **J**ob **E**ntry **C**ontrol **L**anguage

Category of JCL used to control JOB setup and special processing

JES3 JECL stablized
- No new JECL advancements

JES2 JECL
- Now includes most JES3 JECL capabilities
- Recent advancements and future enhancements

JCL and System Utilities

z/OS includes a number of programs useful in batch processing called utilities.

Utilities provide many small, obvious, useful and often critical functions.

Some examples of system utilities:

- IEBGENER Copies a sequential data set
- IEBCOPY Copies a partitioned data set
- IDCAMS Works with VSAM data sets
- IKJEFT01 Run any TSO workload in batch
- SORT Data sequencing and formatting

JCL and Utilities Documentation

MVS JCL User's Guide

MVS JCL Reference

DFSMSdfp Utilities

Basic JCL Concepts

Reusable JCL Examples

hyperlinks

wikipedia.org

Job Control Language - Wikipedia, the free encyclopedia - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address http://en.wikipedia.org/wiki/JCL Go

WIKIPEDIA
The Free Encyclopedia

navigation
- Main Page
- Community Portal
- Featured articles
- Current events
- Recent changes
- Random article
- Help
- Contact Wikipedia
- Donations

search

Go Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Cite this article

in other languages
- Dansk
- Deutsch
- Español
- Français

Job Control Language

From Wikipedia, the free encyclopedia
(Redirected from JCL)

JCL redirects here. See National Junior Classical League for the student honor society.

Job Control Language (JCL) is a scripting language used on IBM mainframe operating systems to instruct the Job Entry Subsystem (that is, JES2 or JES3) on how to run a batch program or start a subsystem.

JCL is characterized by a pair of slashes "//" that indicate the start of each statement. The slashes date back from when punched cards were used to submit JCL code for execution. If the cards were mistakenly put back to front in the reader the slashes wouldn't be read first (instead, the sequence numbers would be), so the card deck could be rejected.

For backward compatibility, the basic syntax of JCL for z/OS hasn't changed since the 1960s. It is the same as JCL for OS/360.

DOS/VSE also has a JCL language. Its syntax is entirely different, the only similarity being that statements still start with two slashes: "//".

Contents [hide]

http://en.wikipedia.org/wiki/JCL#Conditional_processing Internet

tutorialspoint
SIMPLYEASYLEARNING

LEARN JCL
job control language

JCL Video Tutorials

Pre

JCL Tutorial

- JCL Home
- JCL - Overview
- JCL - Environment
- JCL - JOB Statement
- JCL - EXEC Statement
- JCL - DD Statement
- JCL - Base Library
- JCL - Procedures
- JCL - Conditional Processing
- JCL - Defining Datasets
- JCL - Input/Output Methods
- JCL - Run COBOL Programs
- JCL - Utility Programs
- JCL - Basic Sort Tricks

JCL Useful Resources

- JCL - Questions and Answers
- JCL - Quick Guide
- JCL - Useful Resources
- JCL - Discussion

https://www.tutorialspoint.com/jcl/index.htm

https://www.tutorialspoint.com/jcl/index.htm

Unit summary

Having completed this unit, you should be able to:

- ✓ Understand purpose of JCL
- ✓ Understand JCL JOB, EXEC, and DD statements
- ✓ Understand relationship of program file name to JCL DDNAME
- ✓ Locate JCL professional manuals, documentation, and online help

Lab #2

- Use JCL Sort Data
- Use JCL Procedure (PROC) to sort data
- Use JCL to Compile, Link and Execute COBOL
- Use JCL to Define VSAM data set and copy data to the VSAM data set
- Use JCL to Compile, Link and Execute COBOL program - VSAM input