

**Multiple Virtual  
Storage**

Lesson 9 : Job  
Management Overview

## Lesson Objectives

- In this lesson, you will learn the following topics:
  - The basic concept and functions of Job
  - Various stages of a Job
  - Job Management
  - Job Scheduling



## Concept

- **JOB:**
  - A single unit of work in Batch Processing.
  - It is used to execute one or more programs / procedures as a series of steps.
- **JOB STEP:**
  - Each program to be executed by a Job is called a job step.
- **JCL:**
  - Job Control Language
  - The language used to code the job.
- **JES:**
  - An important Subsystem of MVS. It keeps track of jobs that enter the system, Presents them to MVS for processing. Sends spooled output to the correct destination.



Copyright © Capgemini 2015. All Rights Reserved 3

### What is a Job?

- JCL is the most dreaded word for a newcomer to IBM world.
- Job is execution of one or more related programs in pre-defined sequence.
  - Example: A job of creating an executable module (load module) from a source program consists of executing Compiler program and Linker program in sequence.

### Why JCL?

- Since the job is executed in the background, without user interaction, *all information required for the execution must be supplied in advance.*
- JCL is used to specify this control information.
- The most common information supplied through JCL is as follows:
  - To whom the job belongs (which user id)?
  - What is the program / utility that is to be executed?
  - Where (in which library / PDS) do you find the load module of the program or utility?
  - Where (which DASD volume / catalog, what data set name) do you find the input data files for the program / utility?
  - Where should (which DASD volume, what data set name) the output files be created?
  - The printer output should be directed to which printer?

### What is JCL?

- JCL stands for Job Control Language.
- Connotation is set for job commands stored as MEMBER in a PDS. For example: JCL to execute a batch program, JCL to compile and link a COBOL program, JCL to allocate a VSAM data set, JCL to SORT and MERGE two Physical Sequential Data Sets.
- Thus JCL is nothing but a set of commands.
- User keys-in commands using a editor, and saves as PDS Member. For example: PAYROLL.TEST.JCL(PROG1JCL)
- Good grasp of JCL is a must to be a versatile IBM programmer.

## Concept of Job Entry Subsystem (JES)

- Job Entry Subsystem (JES):

- JES shares the load on the operating system.
- It takes care of all inputs and outputs.
- It does a simple basic syntax checking.
- It performs resource Initialization.
- It creates address space.
- JES is also known as Job Scheduler.
- It is classified into JES2 and JES3
- Jobs are sent to MVS depending on the class priority schemes.



Copyright © Capgemini 2015. All Rights Reserved 4

### **Job Management:**

#### **Job entry Subsystem (JES):**

- JES is classified into JES2 and JES3:
  - **JES2** is primarily designed for single processor systems / uni-processor environment.
  - **JES3** is designed for multiple processor systems / multiprocessing environment (decided at the time of system initialization)
- Each MVS system uses either JES2 or JES3.
- JES3 has additional scheduler functions than JES2 (for example: schedule job at specific time of the day, interdependent job scheduling).

## Job Management Functions

- Job Management Functions are listed below:
  - Receive the job into operating system
  - Schedule the job for processing by O/S
  - Execute the Job
  - Process the output

## Stages of Job

- Various stages of a Job are as follows:
  - Job Preparation
  - Job Scheduling
  - Job Execution
  - End of execution (normal, erroneous)



Copyright © Capgemini 2015. All Rights Reserved 6

- Job Preparation:
  - User keys-in commands using Editor.
  - User saves it as a member in PDS.
- Job Scheduling:
  - It is initiated using TSO SUBMIT command.
  - It is not necessarily done on FIFO basis.
  - Prioritization is implemented using class and priority code.
- Job Execution:
  - It involves the actual execution of the job by a processor.
- End of execution (normal, erroneous):
  - Intimate the user
  - Job log management
  - Job output management
  - Printer output
  - Data set output

## Erroneous Termination of a Job

- Type of execution errors include the following:
  - Incorrect commands (command syntax errors)
  - Required resources (Data Sets, Program Library, Program Load Module) not available
  - Violation of access permissions for data sets, program load module, and so on.
  - Mismatch in data set status, as required by job and as it actually exists, for example: a create is issued for a data set which already exists.
  - Program errors
  - Mismatch for Dataset - between program definition and actual characteristics
  - Infinite loop
  - Data Type mismatch - numeric variable contains non-numeric data
- Any abnormal termination of program is called as “Abend”.

## How does a Job enter into the System?

- To enter, or submit, the job into the system, the terminal user issues a SUBMIT command.
- When you submit the job, JES reads the job stream (sequence of JCL commands) from a DASD file and copies it to a job queue, which is a part of a special DASD file called JES SPOOL.



Copyright © Capgemini 2015. All Rights Reserved 8

**Note:** The JES component that processes the input job stream is called an **internal reader**.



## How is a Job Scheduled for Execution

- MVS does not necessarily process jobs in the order in which they are submitted.
  - Instead, JES examines the jobs in the job queue and selects the most important jobs for execution. That way JES can prioritize its work, giving preference to more important jobs.
- JES uses two characteristics to classify a job's importance, both of which can be specified in the job's JCL:
  - Job Class
  - Job Priority

## How is a Job Scheduled for Execution?

- If two or more jobs are waiting to be executed, then the JES scheduler selects the one with higher priority.
- Each job class is represented by a single character, either a letter (A-Z) or a digit (0-9).
  - Job classes are assigned based on the processing characteristics of the job.

## How is a Job Scheduled for Execution?...contd

- **INITIATOR:** An initiator is a program that runs in the system region of an address space.
  - Each initiator can handle one job at a time.
    - It examines the JES spool, selects an appropriate job for execution, executes the job in its address space, and returns to the JES spool for another job.
  - The number of active initiators on a system and as a result the number of address spaces eligible for batch job processing determine the number of batch jobs that can be multiprogrammed at once.

## How is a Job Scheduled for Execution?...contd

- Each initiator has one or more job classes associated with it. It executes jobs only from those classes.

Initiator	Eligible Job Classes
1	A
2	B,C,D,H,L,T
3	B,C,D,H,L,T
4	B,C
5	B,C
6	C

## How is a Job Scheduled for Execution?...contd

- Within a job class, initiator selects jobs for execution based on their priorities, which can range from 0 to 15.
- If two or more jobs have same class and priority, then they are executed in the order in which they are submitted.

## How is a Job Scheduled for Execution?...contd

- Once an initiator has selected a job for execution, it invokes a program called the interpreter.
- The interpreter's job is to examine the job information passed to it by JES and create a series of control blocks in the SWA, a part of the address space's private area.
- Control blocks describe all the datasets the job needs.

## How is a Job Executed?

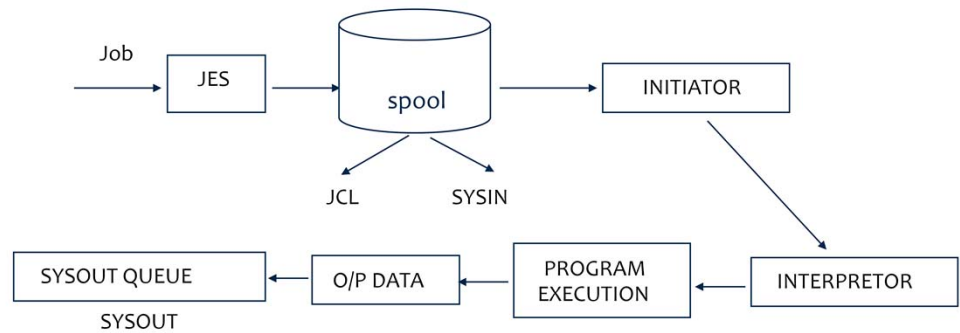
- Initiator goes through three phases for each step of job:
  1. Allocation (required resources are allocated)
  2. Processing (region is created and program is loaded and executed)
  3. De-allocation (resources are released)
- This continues until there are no more job steps to process.

## How is a Job Executed?

- Then, the initiator releases the job and searches the spool again for another job from the proper class to execute.
- As a user's program to execute, it can retrieve data that was included as part of job stream and stores in the JES spool.



## How is a Job Executed?...contd



Copyright © Capgemini 2015. All Rights Reserved 17

### **Job Execution:**

#### **How Job Enters the System?**

- Job can enter the system:
  1. From local or remote card readers (now obsolete).
  2. By starting a cataloged JCL procedure (for example: when user logs in, a predefined set of commands are executed as a batch job. These commands are stored as cataloged JCL procedure)
  3. By interactive users 'thru' SUBMIT command. Users can create a PDS member in which commands are specified. On issuing SUBMIT command, these are executed as a job.
- We will focus on the third approach.

**Job Execution (contd.):****Input:**

- On **SUBMIT**, the **internal reader** reads the JCL and creates an **input stream**.
- **JES2** reads the **input stream**, assigns a **Job Number** and places input stream in SPOOL data set (a message is sent to TSO user about the job number).
- Job is put in the conversion queue.

**Conversion:**

- Converter program analyzes JCL statements.
- It then converts into converter / interpreter text.
- It then checks for Syntax errors.
  - If any error, Job is queued for output processing.
  - If no error, Job is queued for processing.

**Processing:**

- Selection is based on job class and priority.
- Selected job is passed to **Initiator**.
- Initiator invokes **Interpreter**.
- Interpreter builds **control blocks** from converter / interpreter text in a **Scheduler Work Area (SWA)**. SWA is part of address space's private area.
- Control blocks describe the data sets required by the job.
  - Initiator allocates resources required by the Job.
  - Initiator starts the program to be executed.
  - It builds the user region.
  - It loads the program in the user region.
  - It transfers control to the program.
  - On completion of the program execution, initiator de-allocates the resources.
- The process of allocation / execution and de-allocation is repeated for each job step.

**Initiator Characteristics:**

- Each initiator can handle one job at a time.
- There can be multiple initiators.
- Each initiator has a job class associated with it.
- System Operators can control the number of initiators and the class/es associated with each initiator.

**Job Execution (contd.):****Input Data:**

- Input data to the user's program can be specified in the job.
- It is called **in-stream data** or **SYSIN data**.
- SYSIN data is read and stored in JES spool.
- SYSIN data is treated like a data coming from card reader.

**Output:**

- It is concerned with management of System Messages, User Data Sets that need to be Printed / Punched.
- It is organized by output class and device set-up requirements.
- User's program can produce output data that is stored in a JES spool. It is called SYSOUT data.
- Each SYSOUT data is assigned an output class.
- Output class indicates the printer selection.
- "Held" output: Special class (usually Z) is assigned to "hold" the output.
- "Held" output remains in the SYSOUT indefinitely. It is usually used to verify before printing. User can change the class and thus release the "held" output.

**Hard-Copy:**

- It involves local or remote processing.
- It involves device selection.
- You can queue the output for print /punch.

**Purge:**

- It involves releasing SPOOL and Job Queue space.
- It further involves intimating TSO user about job completion.

**Job Output:**

- Output is produced at each stage of job processing.
- It includes output produced by:
  - JES
  - MVS
  - User's program where SYSOUT is allocated as output device.
- Job output is available to user (you can see it dynamically).
- It can be viewed using ISPF.

**Job Execution (contd.):****Components of Job Output:****Component 1- Separator Page:**

- It is the first and last page of job output.
- It is Inserted by JES.
- It helps operator to the segregate the job outputs when directed to printer.

**Component 2 – part I Job Log:**

- It contains messages produced by JES.
- It is also displayed on operator's console.
- If the job Abends, then error messages are logged in Job Log.

**Component 2 – part II Job Statistics:**

- It contains the summary information of system resources used by the job for example:
  - Number of JCL cards
  - Spool usage
  - Execution time

**Component 3 - JCL Listing:**

- It contains the list of JCL that was processed for the job.
- It should be same as what user has created.

**Component 4 - Message Log:**

- It includes messages regarding job execution.
- It includes messages produced by MVS.
- It includes details of:
  - Resource Allocation
  - Program Execution
  - Resource De-allocation
- It consists of Message label and message text. Message label starting with IEF indicates a MVS message.
- It contains installation specific messages.

**Component 5 – SYSOUT:**

- It contains separate sub-component for each SYSOUT allocation.
- Each SYSOUT can have different characteristics, for example: class, record length, and so on.

## How is the Job's Output Processed?

- Like Jobs, SYSOUT data is assigned an output class that determines how the output will be handled.
- Common O/P classes are as follows:
  - A : Printer
  - B : Card Punch O/P
  - X : Held O/P



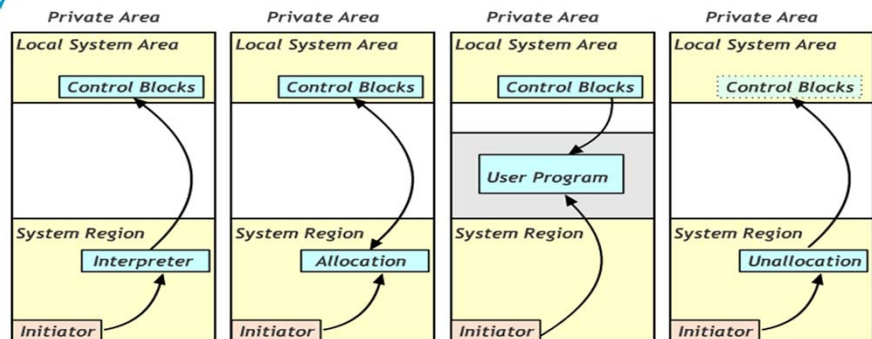
Copyright © Capgemini 2015. All Rights Reserved 21

**Note:** Held O/P stays on the sysout queue indefinitely. Usually, O/P is held so that it can be examined from a TSO terminal.

## How is a Job Purged?

- After the job's output has been processed, the job is purged from the system.
  - That is to say, JES spool space, the job used, is freed so it can be used by other jobs, and any JES control blocks associated with the job are deleted.
- Once a job has been purged, JES no longer knows of its existence.

## Overview of Allocation and Job Execution



After the initiator selects a job for execution, it invokes the interpreter, which builds the required control blocks in the address space's SWA.

For each job step, the initiator invokes allocation routines to allocate the units, volumes, and data sets required by the job step.

After the job step's resources have been allocated, the initiator creates a user region, loads the user program into it, and transfers control to the user program.

When the user program completes the initiator invokes unallocation routines to deallocate the resources used by the job step. Then if the job has more steps, the initiator repeats the allocation execution-unallocation process.

**Note:** When the user program completes, the initiator invokes **Unallocation** routine to deallocate the resources used by the job step.

## Using SDSF

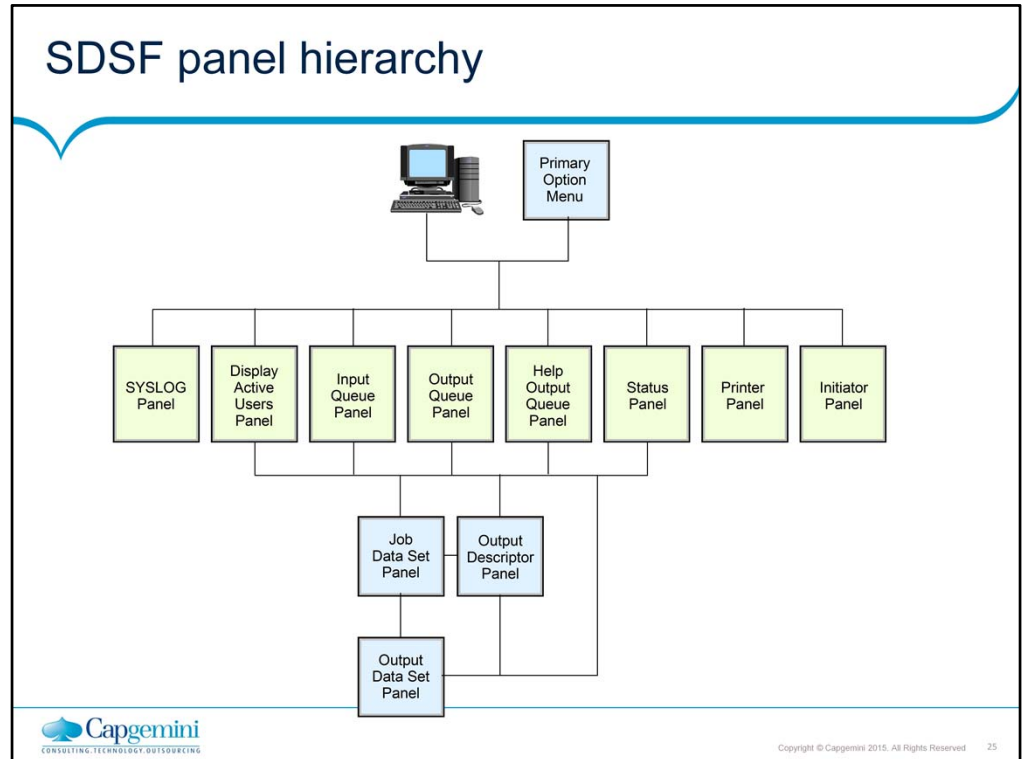
- After submitting a job, Z/OS users use System Display and Search Facility (SDSF) to review the job output for successful completion or JCL errors.
- SDSF lets the users to:
  - View and search the system log
  - Enter system commands
  - Hold, release, cancel, and purge jobs
  - Monitor jobs while they are processed
  - Display job output before deciding to print it
  - Control the order in which jobs are processed
  - Control the order in which output is printed
  - Control printers and initiators



Copyright © Capgemini 2015. All Rights Reserved 24

SDSF is a software product whose primary purpose is to display printed output held in the JES spool area. Much of the printed output sent to JES by batch jobs (and other jobs) is never actually printed. Instead it is inspected using SDSF and deleted or used as needed.





## SDSF: Primary Option Menu

```
Display Filter View Print Options Help
-----
HQPX7707 ----- SDSF PRIMARY OPTION MENU -----
-----
DA  Active users          INIT  Initiators
I   Input queue          PR   Printers
O   Output queue         PUN  Punches
H   Held output queue    RDR  Readers
ST  Status of jobs       LINE Lines
                               NODE Nodes
LOG  System log          SO   Spool offload
SR   System requests     SP   Spool volumes
MAS  Members in the MAS
JC   Job classes         ULOG  User session log
SE   Scheduling environments
RES  WLM resources
ENC  Enclaves
PS   Processes

END   Exit SDSF

COMMAND INPUT ==> _
F1=HELP  F2=SPLIT  F3=END    F4=RETURN  F5=IFIND  F6=BOOK
F7=UP    F8=DOWN   F9=SWAP   F10=LEFT  F11=RIGHT F12=RETRIEVE

                                SCROLL ==> CSR
```

This is the SDSF primary options menu. Some of the options shown are:

<b>DA</b> address spaces (jobs, started tasks,	The Display Active panel shows information about MVS
<b>I</b> started	and TSO users) that are running.
<b>O</b> SYSOUT data	The Input Queue panel displays information about jobs,
<b>H</b> SYSOUT data sets for	tasks, and TSO users on the JES2 input queue or
<b>ST</b> started tasks,	The Output Queue panel displays information about
<b>LOG</b> search it.	sets for jobs, started tasks, and TSO users on any
<b>PS</b> UNIX System Services processes.	The Held Output panel shows information about
<b>PR</b> printers printing	jobs, started tasks, and TSO users on any <i>held</i> JES2
	The Status panel displays information about jobs,
	and TSO users on the JES2 queues.
	The system Log panel displays the log and lets you
	The Processes panel displays information about z/OS
	The Printers panel displays information about JES2
	jobs, started task, and TSO user output.

## Viewing of JES2 output files

Screen 1

Display Filter View Print Options Help									
SDSF STATUS DISPLAY ALL CLASSES									
NP	JOBNAME	JobID	Name	Prg Queue	C	Pos	SRF	Step	Status
DSRP042	TSU01338	DSRP042	15	EXECUTION			DSRC	DSRC	
DSRP042H	J0801339	DSRP042	1	PRINT	A	3			
DSRP042H	J0801340	DSRP042	1	PRINT	A	4			
DSRP042H	J0801341	DSRP042	1	PRINT	A	5			
DSRP042H	J0801342	DSRP042	1	PRINT	A	6			

Screen 2

Display Filter View Print Options Help									
SDSF JOB DATA SET DISPLAY - JOB DSRP042H (J0801342) LINE 1-3 (3)									
NP	DDNAME	StepName	ProdStep	DDID	Owner	C	Dest	Rec	Cnt Page
JES2JCL	JES2		2	DSRP042	X	LOCAL		11	
JES2JCL	JES2		3	DSRP042	X	LOCAL		10	
JES2MSG	JES2		4	DSRP042	X	LOCAL		2	



The first screen shown on the slide displays a list of the jobs we submitted and whose status is directed to the spool region. In this case there are four job has been submitted and executed.

Issuing a ? command in the NP column displays the output files generated by job 1342.

The second screen displays three ddnames: the JES2 messages log file, the JES2 JCL file, and the JES2 system messages file.

This option is useful when you are seeing jobs with many files directed to SYSOUT and you want to display one associated with a specific step. You issue an S in the NP column to select a file you want.

To see all files, instead of a ?, type S in the NP column;

## SDSF: Display active users (DA)

SDSF	DA	DSRC	DSRC	PAG	0	SIO	4	CPU	4/	4	LINE	1-17	(71)
NP	JOBNAME	StepName	ProcStep	JobID	Owner	C	Pos	DP	Real	Paging	SIO		
	*MASTER*			STC08264	+MASTER+	NS	FF	11T	0.00	0.00			
	ALLOCAS	ALLOCAS				NS	FF	1343	0.00	0.00			
	ANTAS000	ANTAS000	IEFPROC			NS	FB	3168	0.00	0.00			
	ANTMAIN	ANTMAIN	IEFPROC			NS	FF	8696	0.00	0.00			
	BPX0INIT	BPX0INIT	BPX0INIT			IN	FF	166	0.00	0.00			
	CATALOG	CATALOG	IEFPROC			NS	FF	1917	0.00	0.00			
	CICS1	CICS1	CICS	STC05508	CIC1USR	NS	FB	9210	0.00	0.00			
	CICS2	CICS2	CICS	STC05511	CIC2USR	NS	FB	9888	0.00	0.00			
	CICS3	CICS3	CICS	STC05514	CIC3USR	NS	FB	9156	0.00	0.00			
	CICS4	CICS4	CICS	STC00768	CIC4USR	NS	FB	10T	0.00	0.00			
	CICS5	CICS5	CICS	STC05518	CIC5USR	NS	FB	9760	0.00	0.00			
	CICS6	CICS6	CICS	STC01013	CIC6USR	NS	FB	9625	0.00	0.00			
	CICS7	CICS7	CICS	STC05524	CIC7USR	NS	FB	8972	0.00	0.00			
	CNMPROC	CNMPROC	NETVIEW			NS	FB	12T	0.00	0.00			
	CNMPSSI	CNMPSSI	NETVIEW			NS	FB	190	0.00	0.00			
	CONSOLE	CONSOLE				NS	FF	1003	0.00	0.00			
	CSQ1CHIN	CSQ1CHIN	PROCSTEP	STC05497	RACFSTC	IN	FB	3333	0.00	0.00			
COMMAND INPUT ==> owner *										SCROLL ==> CSR			
F1=HELP	F2=SPLIT	F3=END	F4=RETURN	F5=IFIND	F6=BOOK								
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE								

## Demo

- Demo on:
  - COBOL program execution and JCL program
  - Output of COBOL program



## Summary

- In this lesson, you have learnt:
  - Job is an execution of one or more related programs in sequence.
  - JCL is a file containing control statements that provide the specifications necessary to process a job.
  - Job goes through various stages, such as:
    - Preparation
    - Scheduling
    - Execution
    - Termination



## Review Question

- Question 1: \_\_\_\_ selects job for execution.
  - Option 1: Initiator
  - Option 2: JES
  - Option 3: Scheduler
- Question 2: Job Output are always held.
  - True/ False
- Question 3: Job Priority ranges from \_\_\_\_ to \_\_\_\_.

