JCL - ENVIRONMENT SETUP

Installing JCL on Windows/Linux

There are many Free Mainframe Emulators available for Windows which can be used to write and learn sample JCLs.

One such emulator is Hercules, which can be easily installed in Windows by following few simple steps given below:

- Download and install the Hercules emulator, which is available from the Hercules' home site - : www.hercules-390.eu
- Once you installed package on Windows machine, it will create a folder like **C:\Mainframes**.
- Run Command Prompt *CMD* and reach the directory C:\Mainframes on CMD.
- The complete guide on various commands to write and execute a JCL can be found on URL **www.jaymoseley.com/hercules/installmvs/instmvs2.htm**

Hercules is an open source software implementation of the mainframe System/370 and ESA/390 architectures, in addition to the latest 64-bit z/Architecture. Hercules runs under Linux, Windows, Solaris, FreeBSD, and Mac OS X.

Running JCL on Mainframes

A user can connect to a mainframe server in a number of ways such a thin client, dummy terminal, Virtual Client System *VCS* or Virtual Desktop System *VDS*.

Every valid user is given a login id to enter into the Z/OS interface *TSO/EorISPF*. In the Z/OS interface, the JCL can be coded and stored as a member in a Partitioned Dataset *PDS*. When the JCL is submitted, it is executed and the output received as explained in the job processing section of previous chapter.

Structure of a JCL

The basic structure of a JCL with the common statements is given below:

```
//SAMPJCL JOB 1,CLASS=6,MSGCLASS=0,NOTIFY=&SYSUID          (1)
//*                                                        (2)
//STEP010  EXEC PGM=SORT                                   (3)
//SORTIN   DD DSN=JCL.SAMPLE.INPUT,DISP=SHR                (4)
//SORTOUT  DD DSN=JCL.SAMPLE.OUTPUT,                       (5)
//         DISP=(NEW,CATLG,CATLG),DATACLAS=DSIZE50
//SYSOUT   DD SYSOUT=*                                     (6)
//SYSUDUMP DD SYSOUT=C                                     (6)
//SYSPRINT DD SYSOUT=*                                     (6)
//SYSIN    DD *                                            (6)
  SORT FIELDS=COPY
  INCLUDE COND=(28,3,CH,EQ,C'XXX')
/*                                                         (7)
```

Program Description

The numbered JCL statements have been explained below:

1 JOB statement - Specifies the information required for SPOOLing of the job such as job id, priority of execution, user-id to be notified upon completion of the job.

2 //* statement - This is a comment statement.

3 EXEC statement - Specifies the PROC/Program to be executed. In the above example, a SORT program is being executed *i. e. , sortingtheinputdatainaparticularorder*

4 Input DD statement - Specifies the type of input to be passed to the program mentioned in 3. In the above example, a Physical Sequential *PS* file is passed as input in shared mode *DISP = SHR*.

5 Output DD statement - Specifies the type of output to be produced by the program upon execution. In the above example, a PS file is created. If a statement extends beyond the 70th position in a line, then it is continued in the next line, which should start with "//" followed by one or more spaces.

6 There can be other types of DD statements to specify additional information to the program *Intheaboveexample*: *TheSORTconditionisspecifiedintheSYSINDDstatement* and to specify the destination for error/execution log *Example*: *SYSUDUMP*/*SYSPRINT*. DD statements can be contained in a dataset *mainframefile* or as in stream data *informationhard − codedwithintheJCL* as given in above example.

7 /* marks the end of in stream data.

All the JCL statements except in stream data starts with //. There should be at least one space before and after JOB, EXEC and DD keywords and there should not be any spaces in rest of the statement.

JOB Parameter Types

Each of the JCL statements is accompanied by a set of parameters to help the Operating Systems in completing the program execution. The parameters can be of two types:

Positional Parameters

- Appears at pre-defined position and order in the statement. Example: Accounting information Parameter can appear only after the **JOB** keyword and before the programmer name parameter and the Keyword Parameters. If a positional parameter is omitted, it has to be replaced with a comma.
- Positional Parameters are present in JOB and EXEC statements. In the above example, PGM is a positional parameter coded after the **EXEC** keyword.

Keyword Parameters

- They are coded after the positional parameters, but can appear in any order. Keyword parameters can be omitted if not required. The generic syntax is KEYWORD= *value*. Example: MSGCLASS=X, i.e., the job log is redirected to the output SPOOL after the job completion.
- In the above example, CLASS, MSGCLASS and NOTIFY are keyword parameters of JOB statement. There can be keyword parameters in EXEC statement as well.

These parameters have been detailed out in the subsequent chapters along with appropriate examples.