

# JCL - DD STATEMENT

Datasets are mainframe files with records organised in a specific format. Datasets are stored on the Direct Access Storage Device *DASD* or Tapes of the mainframe and are basic data storage areas. If these data are required to be used/created in a batch program, then the file *i. e.* , *dataset* physical name along with the file format and organisation are coded in a JCL.

The definition of each dataset used in the JCL is given using the **DD statement**. The input and output resources required by a job step needs to be described within a DD statement with information such as the dataset organisation, storage requirements and record length.

## Syntax

Following is the basic syntax of a JCL DD statement:

```
//DD-name DD Parameters
```

## Description

Let us see the description of the terms used in above DD statement syntax.

### DD-NAME

A DD-NAME identifies the dataset or input/output resource. If this is an input/output file used by a COBOL/Assembler program, then the file is referenced by this name within the program.

### DD

This is the keyword to identify it as an DD statement.

## PARAMETERS

Following are the various parameters for DD statement. You can use one or more parameters based on requirements and they are separated by comma:

Parameter	Description
<b>DSN</b>	<p>The DSN parameter refers to the physical dataset name of a newly created or existing dataset. The DSN value can be made up of sub-names each of 1 to 8 characters length, separated by periods and of total length of 44 characters <i>alphanumeric</i>. Following is the syntax:</p> <p><b>DSN=Physical Dataset Name</b></p> <p><b>Temporary datasets</b> need storage only for the job duration and are deleted at job completion. Such datasets are represented as <b>DSN=&amp;name</b> or simply without a DSN specified.</p> <p>If a temporary dataset created by a job step is to be used in the next job step, then it is referenced as <b>DSN=*.stepname.ddname</b>. This is called <b>Backward Referencing</b>.</p>
<b>DISP</b>	<p>The DISP parameter is used to describe the status of the dataset, disposition at the end of the job step on normal and abnormal completion. DISP is not required in a DD statement only when the dataset gets created and deleted in the same job step <i>likethetemporarydatasets</i>. Following is the syntax:</p>

## **DISP**=*status, normal – disposition, abnormal – disposition*

Following are valid values for **status**:

- **NEW** : The dataset is newly created by the job step. OUTPUT1 in the example above.
- **OLD** : The dataset is already created and will be overwritten in the job step. The job step gains exclusive access on the dataset and no other job can access this dataset until the completion of the job step.
- **SHR** : The dataset is already created and will be read in the job step. The dataset can be read by multiple jobs at the same time. Example: INPUT1 and INPUT2.
- **MOD** : The dataset is already created. This disposition will be used when there is a need to append new records to the existing dataset *existing records will not be overwritten*.

A **normal-disposition** parameter can take one of the following values

- CATLG, UNCATLG, DELETE, PASS and KEEP

A **abnormal-disposition** parameter can take one of the following values

- CATLG, UNCATLG, DELETE and KEEP

Here is the description of CATLG, UNCATLG, DELETE, PASS and KEEP parameters:

- **CATLG** : The dataset is retained with a entry in the system catalog.
- **UNCATLG** : The dataset is retained but system catalog entry is removed.
- **KEEP** : The dataset is retained without changing any of the catalog entries. KEEP is the only valid disposition for VSAM files. This is to be used only for permanent datasets.
- **DELETE** : Dataset is deleted from user and system catalog.
- **PASS** : This is valid only for normal disposition. This is used when the dataset is to be passed and processed by the next job step in a JCL

When any of the sub-parameters of DISP are not specified, the default values are as follows:

- **status** : NEW is the default value.
- **normal-disposition** : If status is NEW, default normal-disposition is DELETE, else it is KEEP.
- **abnormal-disposition** : Same as normal disposition.

## **DCB**

The Data Control Block *DCB* parameter details the physical characteristics of a dataset. This parameter is required for datasets that are newly created in the job step.

LRECL is the length of each record held within the dataset.

RECFM is the record format of the dataset. RECFM can hold values FB, V or VB. FB is a fixed block organisation where one or more logical

records are grouped within a single block. V is variable organisation where one variable length logical record is placed within one physical block. VB is Variable Block organisation where one or more variable length logical records are placed within one physical block.

BLKSIZE is the size of the physical block. The larger the block, greater is the number of records for a FB or VB file.

DSORG is the type of dataset organisation. DSORG can hold values PS *PhysicalSequential*, PO *PartitionedOrganisation* and DA *DirectOrganisation*.

When there is a need to replicate the DCB values of one dataset to another within the same jobstep or JCL, then it is specified as DCB=\*.stepname.ddname where stepname is the name of the job step and ddname is the dataset from which the DCB is copied.

Check below example where RECFM=FB,LRECL=80 forms the DCB of dataset OUTPUT1.

## SPACE

The SPACE parameter specifies the space required for the dataset in the DASD *DirectAccessStorageDisk*. Following is the syntax:

**SPACE=spcunits, (pri, sec, dir, RLSE)**

Here is the description of all the used parameters:

- **spcunits** : This can be one of the CYL*Cylinder*, TRK*Tracks* or BLKSIZE*BlockSize*.
- **pri** : This is the primary space required for the dataset.
- **sec** : This is the additional space required, when the primary space is not being sufficient.
- **ir** : This is the directory blocks required, if the dataset is a PDS *PartitionedDataset* with members within it.
- **RLSE** : This is used to release the unused space at job completion.

## UNIT

The UNIT and VOL parameters are listed in the system catalog for catalogued datasets and hence can be accessed with just the physical DSN name. But for uncataloged datasets, the DD statement should include these parameters. For new datasets to be created, the UNIT/VOL parameters can be specified or the Z/OS allocates the suitable device and volume.

The UNIT parameter specifies the type of device on which the dataset is stored. The device type can be identified using Hardware Address or Device type group. Following is the syntax:

**UNIT=DASD | SYSDA**

Where DASD stands for Direct Access Storage Device and SYSDA stands for System Direct Access and refers to the next available disk storage device.

## VOL

The VOL parameter specifies the volume number on the device identified by the UNIT parameter. Following is the syntax:

**VOL=SER=v1, v2**

Where v1, v2 are volume serial numbers. You can use the following syntax as well:

### **VOL=REF=\*.DDNAME**

Where REF is the backward reference to the volume serial number of a dataset in any of the preceding job steps in the JCL.

## **SYSOUT**

The DD statement parameters discussed so far corresponds to data being stored in a dataset. The SYSOUT parameter directs the data to output device based on the class specified. Following is the syntax

### **SYSOUT=class**

Where if class is A then it directs output to printer, and if class is \* then it directs output to same destination as that of the MSGCLASS parameter in the JOB statement.

## **Example**

Following is an example, which makes use of DD statements along with various parameters explained above:

```
//TTYYSAMP JOB 'TUTO', CLASS=6, MSGCLASS=X, REGION=8K,  
//          NOTIFY=&SYSUID  
//*  
//STEP010 EXEC PGM=ICETOOL, ADDRSPC=REAL  
//*  
//INPUT1 DD DSN=TUTO.SORT.INPUT1, DISP=SHR  
//INPUT2 DD DSN=TUTO.SORT.INPUT2, DISP=SHR, UNIT=SYSDA,  
//          VOL=SER=(1243, 1244)  
//OUTPUT1 DD DSN=MYFILES.SAMPLE.OUTPUT1, DISP=(, CATLG, DELETE),  
//          RECFM=FB, LRECL=80, SPACE=(CYL, (10, 20))  
//OUTPUT2 DD SYSOUT=*
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js