

JCL - BASE LIBRARY

http://www.tutorialspoint.com/jcl/jcl_base_library.htm

Copyright © tutorialspoint.com

Base Library is the Partitioned Dataset *PDS*, which holds the load modules of the program to be executed in the JCL or the catalogued procedure, which is called in the program. Base libraries can be specified for the whole JCL in a **JOBLIB** library or for a particular job step in a **STEPLIB** statement.

JOBLIB Statement

A **JOBLIB** statement is used in order to identify the location of the program to be executed in a JCL. The JOBLIB statement is specified after the JOB statement and before the EXEC statement. This can be used only for the in stream procedures and programs.

Syntax

Following is the basic syntax of a JCL JOBLIB statement:

```
//JOBLIB DD DSN=dsname, DISP=SHR
```

The JOBLIB statement is applicable to all the EXEC statements within the JCL. The program specified in the EXEC statement will be searched in the JOBLIB library followed by the system library.

For example, if the EXEC statement is executing a COBOL program, the load module of the COBOL program should be placed within the JOBLIB library.

STEPLIB Statement

A **STEPLIB** statement is used in order to identify the location of the program to be executed within a Job Step. The STEPLIB statement is specified after the EXEC statement and before the DD statement of the job step.

Syntax

Following is the basic syntax of a JCL STEPLIB statement:

```
//STEPLIB DD DSN=dsname, DISP=SHR
```

The program specified in the EXEC statement will be searched in the STEPLIB library followed by the system library. STEPLIB coded in a job step overrides the JOBLIB statement.

Example

The following example shows the usage of JOBLIB and STEPLIB statements:

```
//MYJCL JOB , , CLASS=6, NOTIFY=&SYSUID
//*
//JOBLIB DD DSN=MYPROC.BASE.LIB1, DISP=SHR
//*
//STEP1 EXEC PGM=MYPROG1
//INPUT1 DD DSN=MYFILE.SAMPLE.INPUT1, DISP=SHR
//OUTPUT1 DD DSN=MYFILES.SAMPLE.OUTPUT1, DISP=(, CATLG, DELETE),
//          RECFM=FB, LRECL=80
//*
//STEP2 EXEC PGM=MYPROG2
//STEPLIB DD DSN=MYPROC.BASE.LIB2, DISP=SHR
//INPUT2 DD DSN=MYFILE.SAMPLE.INPUT2, DISP=SHR
//OUTPUT2 DD DSN=MYFILES.SAMPLE.OUTPUT2, DISP=(, CATLG, DELETE),
//          RECFM=FB, LRECL=80
```

Here, the load module of the program MYPROG1 in STEP1 is searched in the MYPROC.SAMPLE.LIB1.

If not found, it is searched in the system library. In STEP2, STEPLIB overrides JOBLIB and load module of the program MYPROG2 is searched in MYPROC.SAMPLE.LIB2 and then in the system library.

INCLUDE Statement

A set of JCL statements coded within a member of a PDS can be included to a JCL using an **INCLUDE** statement. When the JES interprets the JCL, the set of JCL statements within the INCLUDE member replaces the INCLUDE statement.

Syntax

Following is the basic syntax of a JCL INCLUDE statement:

```
//name INCLUDE MEMBER=member -name
```

The main purpose of INCLUDE statement is reusability. For example, common files to be used across many JCLs can be coded as DD statements within INCLUDE member and used in a JCL.

Dummy DD statements, data card specifications, PROCs, JOB, PROC statements cannot be coded within an INCLUDE member. An INLCUDE statement can be coded within an INCLUDE member and further nesting can be done up to 15 levels.

JCLLIB Statement

A **JCLLIB** statement is used to identify the private libraries used in the job. It can be used both with instream and cataloged procedures.

Syntax

Following is the basic syntax of a JCL JCLLIB statement:

```
//name JCLLIB ORDER=(library1, library2....)
```

The libraries specified in the JCLLIB statement will be searched in the given order to locate the programs, procedures and INCLUDE member used in the job. There can be only one JCLLIB statement in a JCL; specified after a JOB statement and before EXEC and INCLUDE statement but it cannot be coded within an INCLUDE member.

Example

In the following example, the program MYPROG3 and INCLUDE member MYINCL is searched in the order of MYPROC.BASE.LIB1, MYPROC.BASE.LIB2, system library.

```
//MYJCL JOB ,,CLASS=6,NOTIFY=&SYSUID  
//*  
//MYLIB JCLLIB ORDER=(MYPROC.BASE.LIB1,MYPROC.BASE.LIB2)  
//*  
//STEP1 EXEC PGM=MYPROG3  
//INC INCLUDE MEMBER=MYINCL  
//OUTPUT1 DD DSN=MYFILES.SAMPLE.OUTPUT1, DISP=(,CATLG,DELETE),  
//          RECFM=FB,LRECL=80  
//*
```

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