

JCL - EXEC STATEMENT

Each JCL can be made of many job steps. Each job step can execute a program directly or can call a procedure, which in turn executes one or more programs *jobsteps*. The statement, which holds the job step program/procedure information is the **EXEC statement**.

The purpose of the EXEC statement is to provide required information for the program/procedure executed in the job step. Parameters coded in this statement can pass data to the program in execution, can override certain parameters of JOB statement and can pass parameters to the procedure if the EXEC statement calls a procedure instead of directly executing a program.

Syntax

Following is the basic syntax of a JCL EXEC statement:

```
//Step-name EXEC Positional-param, Keyword-param
```

Description

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

STEP-NAME

This identifies the job step within the JCL. It can be of length 1 to 8 with alphanumeric characters.

EXEC

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

POSITIONAL-PARAM

These are positional parameters, which can be of two types:

Positional Parameter	Description
PGM	This refers to the program name to be executed in the job step.
PROC	This refers to the procedure name to be executed in the job step. We will discuss it a separate chapter.

KEYWORD-PARAM

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

Keyword Parameter	Description
PARM	Used to provide parametrized data to the program that is being executed in the job step. This is a program dependant field and do not have definite rules, except that the PARM value has to be included within quotation in the event of having special characters. For example given below, the value "CUST1000" is passed as an alphanumeric value to the program. If the program is in COBOL, the value passed through a PARM parameter in a JCL is received in

the LINKAGE SECTION of the program.

ADDRSPC

This is used to specify whether the job step require virtual or real storage for execution. Virtual storage is pageable whereas real storage is not and is placed in the main memory for execution. Job steps, which require faster execution can be placed in real storage. Following is the syntax:

ADDRSPC=VIRT | REAL

When an ADDRSPC is not coded, VIRT is the default one.

ACCT

This specifies the accounting information of the job step. Following is the syntax:

ACCT=*userid*

This is similar to the positional parameter **accounting information** in the JOB statement. If it is coded both in JOB and EXEC statement, then the accounting information in JOB statement applies to all job steps where an ACCT parameter is not coded. The ACCT parameter in an EXEC statement will override the one present in the JOB statement for that job step only.

Common Keyword Parameters of EXEC and JOB Statement

Keyword Parameter	Description
ADDRSPC	ADDRSPC coded in JOB statement overrides the ADDRSPC coded in EXEC statement of any job step.
TIME	If TIME is coded in an EXEC statement, then it applies to that job step only. If it is specified in both JOB and EXEC statement, then both will be in effect and can cause time-out error due to either of it. It is not recommended to use TIME parameter in both the JOB and EXEC statement together.
REGION	If REGION is coded in an EXEC statement, then it applies to that job step only. REGION coded in JOB statement overrides the REGION coded in EXEC statement of any job step.
COND	Used to control the job step execution based on the return-code of the previous step. If a COND parameter is coded in an EXEC statement of a job step, then the COND parameter of the JOB statement <i>if present</i> is ignored. The various tests that can be performed using a COND parameter is explained in conditional Processing.

Example

Following is a simple example of JCL script along with JOB and EXEC statements:

```
//TTYYSAM  JOB 'TUTO',CLASS=6,MSGCLASS=X,REGION=8K,  
//          NOTIFY=&SYSUID  
//  
//STEP010 EXEC PGM=MYCOBOL,PARAM=CUST1000,  
//          ACCT=(XXXX).REGION=8K.ADDRSPC=RFA1.TIME=1440
```

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