

RUNNING COBOL PROGRAMS USING JCL

Compiling COBOL Programs

In order to execute a COBOL program in batch mode using JCL, the program needs to be compiled and a load module is created with all the sub-programs. The JCL uses the load module and not the actual program at the time of execution. The load libraries are concatenated and given to the JCL at the time of execution using **JCLLIB** or **STEPLIB**.

There are many mainframe compiler utilities available to compile a COBOL program. Some corporate companies use Change Management tools like **Endevor**, which compiles and stores every version of the program. This is useful in tracking the changes made to the program.

```
//COMPILE JOB , CLASS=6, MSGCLASS=X, NOTIFY=&SYSUID
// *
//STEP1 EXEC IGYCRCTL, PARM=RMODE, DYNAM, SSRANGE
//SYSIN DD DSN=MYDATA.URMI.SOURCES(MYCOBB), DISP=SHR
//SYSLIB DD DSN=MYDATA.URMI.COPYBOOK(MYCOPY), DISP=SHR
//SYSLMOD DD DSN=MYDATA.URMI.LOAD(MYCOBB), DISP=SHR
//SYSPRINT DD SYSOUT=*
// *
```

IGYCRCTL is an IBM COBOL compiler utility. The compiler options are passed using PARM parameter. In the above example, RMODE instructs the compiler to use relative addressing mode in the program. The COBOL program is passed using SYSIN parameter and the copybook is the library used by the program in SYSLIB.

This JCL produces the load module of the program as output which is used as the input to the execution JCL.

Running COBOL Programs

Below a JCL example where the program MYPROG is executed using the input file MYDATA.URMI.INPUT and produces two output files written to the spool.

```
//COBBSTEP JOB CLASS=6, NOTIFY=&SYSUID
//
//STEP10 EXEC PGM=MYPROG, PARM=ACCT5000
//STEPLIB DD DSN=MYDATA.URMI.LOADLIB, DISP=SHR
//INPUT1 DD DSN=MYDATA.URMI.INPUT, DISP=SHR
//OUT1 DD SYSOUT=*
//OUT2 DD SYSOUT=*
//SYSIN DD *
//CUST1 1000
//CUST2 1001
/*
```

The load module of MYPROG is located in MYDATA.URMI.LOADLIB. This is important to note that the above JCL can be used for a non-DB2 COBOL module only.

Passing Data to COBOL Programs

Data input to COBOL batch program can be through files, PARAM parameter and SYSIN DD statement. In the above example:

- Data records are passed to MYPROG through file MYDATA.URMI.INPUT. This file will be referred in the program using the DD name INPUT1. The file can be opened, read and closed in the program.
- The PARM parameter data ACCT5000 is received in the LINKAGE section of the program MYPROG in a variable defined within that section.

- The data in the SYSIN statement is received through ACCEPT statement in the PROCEDURE division of the program. Every ACCEPT statement reads one whole record *i. e.* , CUST11000 into a working storage variable defined in the program.

Running a COBOL-DB2 program

For running COBOL DB2 program, specialised IBM utility is used in the JCL and program; DB2 region and required parameters are passed as input to the utility.

The below steps are followed in running a COBOL-DB2 program:

- When a COBOL-DB2 program is compiled, a DBRM *DatabaseRequestModule* is created along with the load module. The DBRM contains the SQL statements of the COBOL programs with its syntax checked to be correct.
- The DBRM is bound to the DB2 region *environment* in which the COBOL will run. This can be done using the IKJEFT01 utility in a JCL.
- After the bind step, the COBOL-DB2 program is run using IKJEFT01 *again* with the load library and DBRM library as the input to the JCL.

```
//STEP001 EXEC PGM=IKJEFT01
//*
//STEPLIB DD DSN=MYDATA.URMI.DBRMLIB,DISP=SHR
//*
//input files
//output files
//SYSPRINT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSDBOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DISPLAY DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(SSID)
    RUN PROGRAM(MYCOBB) PLAN(PLANNAME) PARM(parameters to cobol program) -
    LIB('MYDATA.URMI.LOADLIB')
END
/*
```

In the above example, MYCOBB is the COBOL-DB2 program run using IKJEFT01. Please note that the program name, DB2 Sub-System Id *SSID*, DB2 Plan name are passed within the SYSTSIN DD statement. The DBRM library is specified in the STEPLIB.

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