

Reviewing the JOB and EXEC Statements



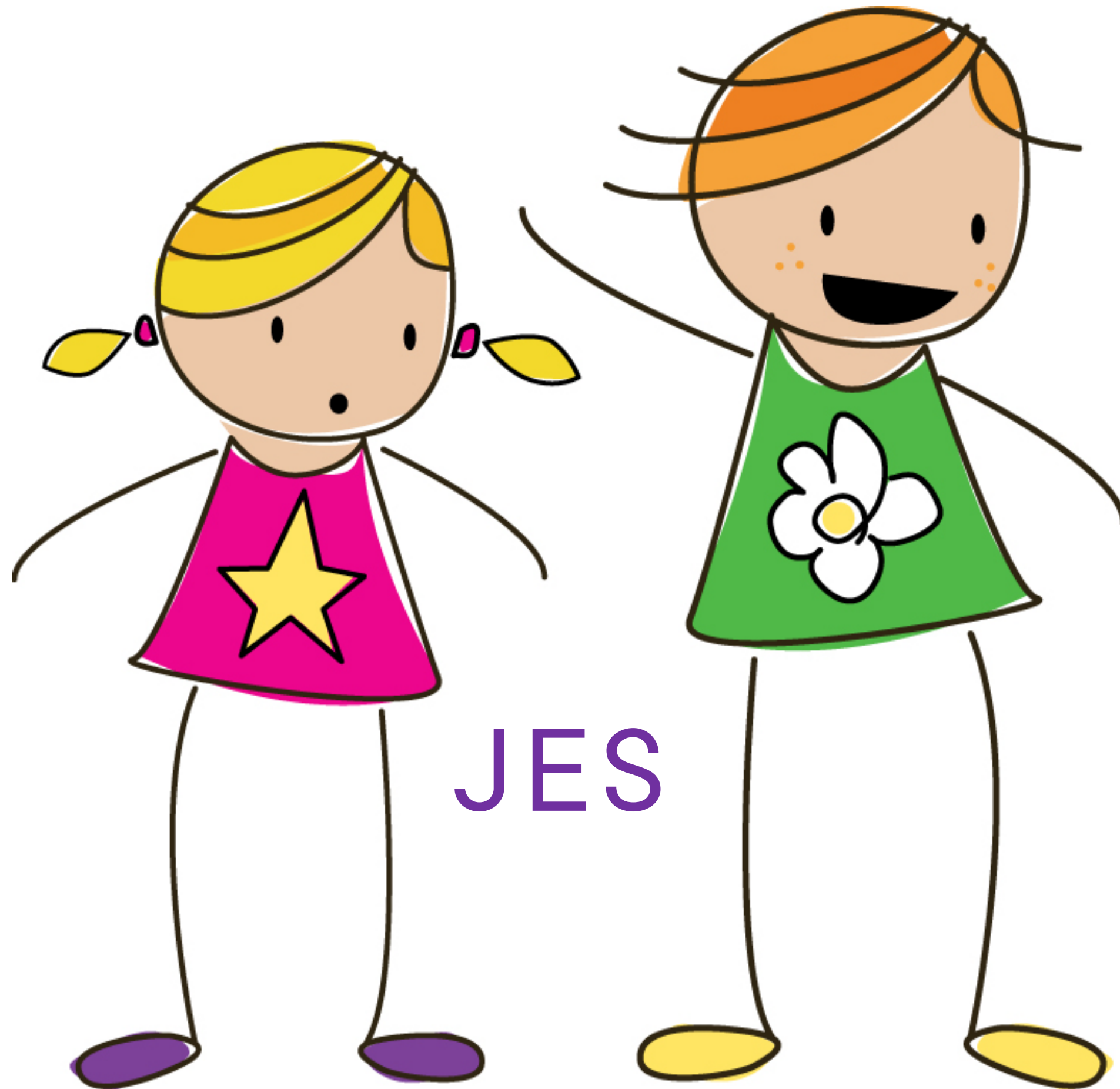
Dave Nicolette

Software Developer

@davenicolette neopragma.com

Module Intro and Overview of JCL and JECL

JES ≠ Jess,
like Jessica



JES = Jez,
like Jezebel

Three Kinds of JCL

**JCL: Job Control
Language**

**JECL: Job Entry
Control Language
(JES2)**

**JECL: Job Entry
Control Language
(JES3)**

**Designed and built
within IBM**



**z/OS
components
subsystems
applications**

**Field-Developed
by customers
and/or IBM field
reps to support
customers**



Overview



- JCL Statement Format
- JOB Statement Parameters
- EXEC Statement Parameters
- JCL Comments

JCL Statement Format

Punched Card with JOB Statement

[illegible]

1										2										3										4										5										6										7										8									
1234567890										1234567890										1234567890										1234567890										1234567890										1234567890										1234567890										1234567890									
//name										operation										parameter										comment																														Xnnnnnnnnnn																			

JCL Statement – General Format

JCL statements must follow strict formatting guidelines.

	1	2	3	4	5	6	7	8
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
//x								
//								

JCL Statement – Identifier Field

Two slashes followed by a space or any character *other than asterisk (*)* identifies this as a JCL statement or the continuation of a JCL statement from the previous line.

1

2

3

4

5

6

7

8

12345678901234567890123456789012345678901234567890123456789012345678901234567890

```

/ / *****
/ /* COMMENTS IN A BOX
/ / *****

```

 *Space or asterisk in column 4!*

JCL Statement – Identifier Field

Two slashes in columns 1 and 2, asterisk (*) in column 3, *and* space or asterisk in column 4 identify this as a JCL comment line.


```

//CICSRUN  JOB accounting info,name,CLASS=A,
//          MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=userid
/*JOBPARM  SYSAFF=sysid
//CICS     EXEC PGM=DFHSIP,REGION=240M,
//          PARM=( 'SIT=6$',
//          'DSALIM=6M,EDSALIM=120M',
//          'RENTPGM=PROTECT,STGPROT=YES',
//          'START=AUTO,SI' )
//SYSIN    DD *
GRP LIST=(DFHLIST,userlist1,userlist2),
LPA=YES,
APP ID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INIT PARM=(DFHDBCON='01',DFHD2INI=('MYDB')),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF  DD SYSOUT=*
//LOGUSR   DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR   DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=140)
//COUT     DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=137)
//CEEMSG   DD SYSOUT=A
//CEEOUT   DD SYSOUT=A
//DFHLCD   DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD,DISP=SHR
//DFHGCD   DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD,DISP=SHR,
//          AMP=( 'BUFND=33,BUFNI=32,BUFSP=1114112' )
//DFHCXRF  DD SYSOUT=A
etc.

```

◀ **Slash slash character other than asterisk: JCL statement with a non-space in 1-3: It's a JES2 statement**

◀ **No slash in column 1: It's in-stream data**

◀ **Slash, asterisk, space in 1-3: End of in-stream data**


```
//CICSRUN JOB accounting info,name,CLASS=A,
//          MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=userid
/*JOBPARM SYSAFF=sysid
//CICS EXEC PGM=DFHSIP,REGION=240M,
//          PARM=( 'SIT=6$' ,
//          'DSALIM=6M,EDSALIM=120M' ,
//          'RENTPGM=PROTECT,STGPROT=YES' ,
//          'START=AUTO,SI' )
//SYSIN DD *
GRPLIST=(DFHLIST,userlist1,userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INITPARM=(DFHDBCON='01' ,DFHD2INI=('MYDB' )),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF DD SYSOUT=*
//LOGUSR DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=140)
//COUT DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=137)
//CEEMSG DD SYSOUT=A
//CEEOUT DD SYSOUT=A
//DFHLCD DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD,DISP=SHR
//DFHGCD DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD,DISP=SHR,
//          AMP=( 'BUFND=33,BUFNI=32,BUFSP=1114112' )
//DFHCXRF DD SYSOUT=A
etc.
```

◀ Job name: **CICSRUN**

◀ Step name: **CICS**

◀ DD name: **SYSIN**


	1	2	3	4	5	6	7	8
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
//name		operation	parameter	comment				



Rules for Names

The name must begin in column 3, immediately after the two slashes, with no intervening spaces.

1										2										3										4										5										6										7										8									
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0										
//A										operation										parameter										comment																																																	
//ABCDEFGH										operation										parameter										comment																																																	



Rules for Names

The name must consist of between 1 and 8 characters.

	1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890123456789012345678901234567890								
//A		operation	parameter	comment				
//ABCDEFGH		operation	parameter	comment				
//\$JNAME		operation	parameter	comment				
//#JB@23		operation	parameter	comment				

Rules for Names

You can use characters from ISO-8859-1 including letters, numbers, and the characters \$, #, and @.

1										2										3										4										5										6										7										8									
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
//STEPLIB DD										parameter										comment																																																											
//SYSPRINT DD										parameter										comment																																																											
//SYSUDUMP DD										parameter										comment																																																											
//etc. DD										parameter										comment																																																											

Rules for Names

Some DD names have special meaning to the system. Don't use them for your own datasets.
There is a list in the course handouts.

	1	2	3	4	5	6	7	8
1234567890123456789012345678901234567890123456789012345678901234567890								
//MYFILE	DD	DSN=first.in.concat, DISP=SHR						
//	DD	DSN=second.in.concat, DISP=SHR						
//	DD	DSN=third.in.concat, DISP=SHR						
//	DD	DSN=fourth.in.concat, DISP=SHR						

Rules for Names

When concatenating datasets under the same DDNAME, code the name only on the 1st DD statement.

```
//CICSRUN JOB accounting info,name,CLASS=A,
//          MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=userid
/*JOBPARM SYSAFF=sysid
//CICS EXEC PGM=DFHSIP,REGION=240M,
//          PARM=( 'SIT=6$' ,
//          'DSALIM=6M,EDSALIM=120M' ,
//          'RENTPGM=PROTECT,STGPROT=YES' ,
//          'START=AUTO,SI' )
//SYSIN DD *
GRPLIST=(DFHLIST,userlist1,userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INITPARM=(DFHDBCON='01' ,DFHD2INI=('MYDB' ) ),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF DD SYSOUT=*
//LOGUSR DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=140)
//COUT DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=137)
//CEEMSG DD SYSOUT=A
//CEEOUT DD SYSOUT=A
//DFHLCD DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD,DISP=SHR
//DFHGCD DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD,DISP=SHR,
//          AMP=( 'BUFND=33,BUFNI=32,BUFSP=1114112' )
//DFHCXRF DD SYSOUT=A
etc.
```

◀ Operation field: **JOB**

◀ Operation field: **EXEC**

◀ Operation field: **DD**


```

//CICSRUN JOB accounting, 'NAME', CLASS=A,
//          MSGCLASS=A, MSGLEVEL=(1,1), NOTIFY=userid
/*JOBPARM SYSAFF=sysid
//CICS      EXEC PGM=DFHSIP, REGION=240M,
//          PARM=( 'SIT=6$',
//                'DSALIM=6M, EDSALIM=120M',
//                'RENTPGM=PROTECT, STGPROT=YES',
//                'START=AUTO, SI' )
//SYSIN      DD *
GRPLIST=(DFHLIST, userlist1, userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INITPARM=(DFHDBCON='01', DFHD2INI=('MYDB')),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF   DD SYSOUT=*
//LOGUSR    DD SYSOUT=*, DCB=(DSORG=PS, RECFM=V, BLKSIZE=136)
//MSGUSR    DD SYSOUT=*, DCB=(DSORG=PS, RECFM=V, BLKSIZE=140)
//COUT      DD SYSOUT=*, DCB=(DSORG=PS, RECFM=V, BLKSIZE=137)
//CEEMSG    DD SYSOUT=A
//CEEOUT    DD SYSOUT=A
//DFHLCD    DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD, DISP=SHR
//DFHGCD    DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD, DISP=SHR,
//          AMP=('BUFND=33, BUFNI=32, BUFSP=1114112')
//DFHCXRF   DD SYSOUT=A
etc.

```

◀ JOB Parameters

◀ Step Parameters

◀ DD Parameters

```
//CICSRUN JOB accounting, 'NAME', CLASS=A,
//          MSGCLASS=A, MSGLEVEL=(1,1), NOTIFY=userid
/*JOBPARM SYSAFF=sysid
//CICS      EXEC PGM=DFHSIP, REGION=240M,
//          PARM=( 'SIT=6$',
//          'DSALIM=6M, EDSALIM=120M',
//          'RENTPGM=PROTECT, STGPROT=YES',
//          'START=AUTO, SI' )
//SYSIN      DD *
GRPLIST=(DFHLIST, userlist1, userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INITPARM=(DFHDBCON='01', DFHD2INI=('MYDB')),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF    DD SYSOUT=*
//LOGUSR     DD SYSOUT=*, DCB=(DSORG=PS, RECFM=V, BLKSIZE=136)
//MSGUSR     DD SYSOUT=*, DCB=(DSORG=PS, RECFM=V, BLKSIZE=140)
//COUT       DD SYSOUT=*, DCB=(DSORG=PS, RECFM=V, BLKSIZE=137)
//CEEMSG     DD SYSOUT=A
//CEEOUT     DD SYSOUT=A
//DFHLCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD, DISP=SHR
//DFHGCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD, DISP=SHR,
//          AMP=( 'BUFND=33, BUFNI=32, BUFSP=1114112' )
//DFHCXRF    DD SYSOUT=A
etc.
```

I HAVE A FEW
COMMENTS ABOUT
THIS JOB.

◀ Comments

Continuing a JCL Statement

```
//CICSRUN JOB accounting,'NAME',CLASS=A,
//          MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=userid
/*JOBPARM SYSAFF=svsid
//CICS      EXEC PGM=DFHSIP,REGION=240M,
//          PARM=( 'SIT=6$',
//                'DSALIM=6M,EDSALIM=120M',
//                'RENTPGM=PROTECT,STGPROT=YES',
//                'START=AUTO,SI' )
//SYSIN      DD *
GRPLIST=(DFHLIST,userlist1,userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INITPARM=(DFHDBCON='01',DFHD2INI=('MYDB')),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF    DD SYSOUT=*
//LOGUSR     DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR     DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=140)
//COUT       DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=137)
//CEEMSG     DD SYSOUT=A
//CEEOUT     DD SYSOUT=A
//DFHLCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD,DISP=SHR
//DFHGCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD,DISP=SHR,
//          AMP=('BUFND=33,BUFNI=32,BUFSP=1114112')
//DFHCXRF    DD SYSOUT=A
etc.
```

◀ Parameters on the EXEC statement


```
//CICSRUN JOB accounting,'NAME',CLASS=A,
//          MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=userid
/*JOBPARM SYSAFF=sysid
//CICS      EXEC PGM=DFHSIP,REGION=240M,
//          PARM=( 'SIT=6$',
//          'DSALIM=6M,EDSALIM=120M',
//          'RENTPGM=PROTECT,STGPROT=YES',
//          'START=AUTO,SI' )
//SYSIN      DD *
GRPLIST=(DFHLIST,userlist1,userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INITPARM=(DFHDBCON='01',DFHD2INI=('MYDB')),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF    DD SYSOUT=*
//LOGUSR     DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR     DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=140)
//COUT       DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=137)
//CEEMSG     DD SYSOUT=A
//CEEOUT     DD SYSOUT=A
//DFHLCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD,DISP=SHR
//DFHGCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD,DISP=SHR,
//          AMP=( 'BUFND=33,BUFNI=32,BUFSP=1114112' )
//DFHCXRF    DD SYSOUT=A
etc.
```

◀ **Break the line after at least one parameter value – not before the parameters begin**

```
//CICSRUN JOB accounting,'NAME',CLASS=A,
//          MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=userid
/*JOBPARM SYSAFF=sysid
//CICS      EXEC PGM=DFHSIP,REGION=240M, ←
//          PARM=( 'SIT=6$', ←
//          'DSALIM=6M,EDSALIM=120M', ←
//          'RENTPGM=PROTECT,STGPROT=YES', ←
//          'START=AUTO,SI' )
//SYSIN      DD *
GRPLIST=(DFHLIST,userlist1,userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INITPARM=(DFHDBCON=' 01 ' ,DFHD2INI=(' MYDB' )),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF    DD SYSOUT=*
//LOGUSR     DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR     DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=140)
//COUT       DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=137)
//CEEMSG     DD SYSOUT=A
//CEEOUT     DD SYSOUT=A
//DFHLCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD,DISP=SHR
//DFHGCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD,DISP=SHR,
//          AMP=( ' BUFND=33,BUFNI=32,BUFSP=1114112' )
//DFHCXRF    DD SYSOUT=A
etc.
```

◀ Break the line after the comma that separates
 ◀ Break the line after the comma that separates
 ◀ Break the line after the comma that separates
 ◀ Break the line after the comma that separates
 one parameter from the next
 one parameter from the next

```
//CICSRUN JOB accounting,'NAME',CLASS=A,
//          MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=userid
/*JOBPARM SYSAFF=sysid
//CICS      EXEC PGM=DFHSIP,REGION=240M,
//          PARM=( 'SIT=6$',
//          'DSALIM=6M,EDSALIM=120M',
//          'RENTPGM=PROTECT,STGPROT=YES',
//          'START=AUTO,SI' )
//SYSIN      DD *
GRPLIST=(DFHLIST,userlist1,userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INITPARM=(DFHDBCON='01',DFHD2INI=('MYDB')),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF    DD SYSOUT=*
//LOGUSR     DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR     DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=140)
//COUT       DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=137)
//CEEMSG     DD SYSOUT=A
//CEEOUT     DD SYSOUT=A
//DFHLCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD,DISP=SHR
//DFHGCD     DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD,DISP=SHR,
//          AMP=( 'BUFND=33,BUFNI=32,BUFSP=1114112' )
//DFHCXRF    DD SYSOUT=A
etc.
```

◀ **Start each continuation line with two slashes followed by at least one space.**

```
//CICSRUN JOB accounting, 'NAME', CLASS=A,
//          MSGCLASS=A, MSGLEVEL=(1,1), NOTIFY=userid
/*JOBPARM SYSAFF=sysid
//CICS      EXEC PGM=DFHSIP, REGION=240M,
//          PARM=( 'SIT=6$',
//          'DSALIM=6M, EDSALIM=120M',
//          'RENTPGM=PROTECT, STGPROT=YES',
//          'START=AUTO, SI' )
//SYSIN     DD *
GRPLIST=(DFHLIST, userlist1, userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER,
MXT=30,
INITPARM=(DFHDBCON='01', DFHD2INI=('MYDB')),
ISC=YES,
IRCSTRT=YES,
.END
/*
//DFHCXRF   DD SYSOUT=*
//LOGUSR    DD SYSOUT=*, DCB=(DSORG=PS, RECFM=V, BLKSIZE=136)
//MSGUSR    DD SYSOUT=*, DCB=(DSORG=PS, RECFM=V, BLKSIZE=140)
//COUT      DD SYSOUT=*, DCB=(DSORG=PS, RECFM=V, BLKSIZE=137)
//CEEMSG    DD SYSOUT=A
//CEEOUT    DD SYSOUT=A
//DFHLCD    DD DSN=CICSTS55.CICS.CICSHTH1.DFHLCD, DISP=SHR
//DFHGCD    DD DSN=CICSTS55.CICS.CICSHTH1.DFHGCD, DISP=SHR,
//          AMP=('BUFND=33, BUFNI=32, BUFSP=1114112')
//DFHCXRF   DD SYSOUT=A
etc.
```

◀ **Resume the continued text between columns 4 and 16 – not after column 16**

1					2					3					4					5					6					7					8				
12345678901234567890123456789012345678901234567890123456789012345678901234567890																																							
//CICS										EXEC										PGM=DFHSIP,REGION=240M,PARM=('SIT=6\$',										'DSALIM=6M,EDSALIM=									
//										120M',										RENTPGM=PROTECT,STGPROT=YES',										START=AUTO,SI')									

Continuing a line in the middle of a quoted value

Continue the quoted value through column 71. Do not code anything in column 72.

1					2					3					4					5					6					7					8				
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
//CICS										EXEC										PGM=DFHSIP,REGION=240M,PARM=('SIT=6\$',										'DSALIM=6M,EDSALIM=									
//										120M',										RENTPGM=PROTECT,STGPROT=YES',										START=AUTO,SI')									

Continuing a line in the middle of a quoted value

Start the continuation line with two slashes in columns 1 and 2.

Resume the quoted value exactly in column 16.

1	2	3	4	5	6	7	8
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
//CICS EXEC PGM=DFHSIP,REGION=240M,							
//	PARM=(' SIT=6\$ ' ,			USE SYSTEM INIT TBL 6\$			
//	' DSALIM=6M,EDSALIM=120M ' ,			DSA MAX 31BIT WAS 800			
//	' RENTPGM=PROTECT,STGPROT=YES ' ,			WRITE/STG PROTECTION			
//	' START=COLD ')			AUTO,SI ')		COLD START BECAUSE OF X	
//				WHAT HAPPENED YESTERDAY			
//				RANDALL SAYS HIS DOG X			
//				ATE THE RECOVERY MANAGX			
//				ER CONTROL RECORD. X			
//				THIRD TIME THIS WEEK.			

Continuing a comment that follows parameter values

Continue the quoted value through column 71.

Code a non-blank character in column 72.

Start subsequent lines with two slashes.

Leave column 3 blank.

Continue the comment text anywhere after column 3, through column 71

	1	2	3	4	5	6	7	8
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
	//CICS EXEC PGM=DFHSIP,REGION=240M,							
	//	PARM=(' SIT=6\$ ' ,			USE SYSTEM INITIALIZA-X			
	//				TION TABLE 6\$			
	//	' DSALIM=6M,EDSALIM=120M' ,			UPPER LIMIT OF DSA			X
	//				BELOW 16MB LINE,			X
	//				AND IN 31BIT STORAGE			
	//	' RENTPGM=PROTECT, STGPROT=YES' ,			REENTRANT PROGRAM			X
	//				WRITE PROTECTION AND			X
	//				STORAGE PROTECTION			
	//	' START=AUTO, SI ')						

Don't continue both the parameter list and the comments

The system will honor the parameter continuation and ignore the comment continuation.
You will probably get a JCL error.

1 2 3 4 5 6 7 8

```
1234567890123456789012345678901234567890123456789012345678901234567890
//*****
//* USING SIT TABLE 6$.
//* UPPER LIMIT FOR DSA ABOVE THE LINE WAS ORIGINALLY 800M.
//* REENTRANT PROGRAM WRITE PROTECTION YES.
//* STORAGE PROTECTION YES.
//*****
//CICS EXEC PGM=DFHSIP,REGION=240M,
//          PARM=( ' SIT=6$' ,
//          ' DSALIM=6M,EDSALIM=120M' ,
//          ' RENTPGM=PROTECT,STGPROT=YES' ,
//          ' START=AUTO,SI' ) AUTO START NOTWITHSTANDING RANDALL'S DOG
```

If you need lots of comments, code them on separate lines.

It's more readable. Comments on the right of each line are okay if they are very short.

The JOB Statement

1 2 3 4 5 6 7 8

```
123456789012345678901234567890123456789012345678901234567890
//AYT289X JOB 599, 'A. YAN', CLASS=A, MSGCLASS=A, MSGLEVEL=(2,1), TIME=1440,
// NOTIFY=&SYSUID
// . . .
```

JOB Statement with Single-Value Job Accounting Parameter

Job accounting parameter has a single value.
Programmer name parameter is used.

1 2 3 4 5 6 7 8

```
123456789012345678901234567890123456789012345678901234567890
//GMT443N JOB (599, 'T/HK'), 'GENNA', CLASS=A, MSGCLASS=A, MSGLEVEL=(1,1),
//          NOTIFY=&SYSUID
// . . .
```

JOB Statement with Two-Value Job Accounting Parameter

Job accounting parameter has two comma-delimited fields, all enclosed in parentheses.
Programmer name parameter is used.

1

2

3

4

5

6

7

8

```
1234567890123456789012345678901234567890123456789012345678901234567890
//GMT443N JOB (599, 'T/HK'), 'GENNA', CLASS=A, MSGCLASS=A, MSGLEVEL=(1,1),
//          NOTIFY=&SYSUID
// . . .
```

Programmer Name Parameter

The programmer name parameter is optional as far as the system is concerned, but may be required in your organization.

The length of the value is limited to 20 characters.

1

2

3

4

5

6

7

8

```
123456789012345678901234567890123456789012345678901234567890
//P0117A JOB , 'ORLOVSKY', CLASS=A, MSGCLASS=A, MSGLEVEL=(1,1),
//          NOTIFY=&SYSUID
// . . .
```

JOB Statement with No Job Accounting Parameter

Placeholder is coded for the omitted job accounting parameter.
Programmer name parameter is used.

1

2

3

4

5

6

7

8

```
1234567890123456789012345678901234567890123456789012345678901234567890
//JKT308B JOB CLASS=A,MSGCLASS=Q,MSGLEVEL=(1,1),NOTIFY=&SYSUID
// . . .
```

JOB Statement with Neither Job Accounting Nor Programmer Name

No placeholders are coded for the omitted positional parameters.

1

2

3

4

5

6

7

8

```
1234567890123456789012345678901234567890123456789012345678901234567890
//JKT308B JOB CLASS=A,MSGCLASS=Q,MSGLEVEL=(1,1),NOTIFY=&SYSUID
// . . .
```

NOTIFY Parameter Specifying the &SYSUID Symbol

JES will replace &SYSUID with the user id of the account that submitted the job.

When the job starts and when it ends, JES will send a notification message *via* TSO to that account.

1

2

3

4

5

6

7

8

```
123456789012345678901234567890123456789012345678901234567890
//JKT308B JOB CLASS=Q,MSGCLASS=Q,MSGLEVEL=(1,1),NOTIFY=&SYSUID
// . . .
```

CLASS Parameter Specifying Predefined Value "A"

JES will handle job initiation according to the configuration defined for job class A. This is a system configuration setting not under control of application developers. You will be told which job classes to use for each type of job you work with.

1

2

3

4

5

6

7

8

```
1234567890123456789012345678901234567890123456789012345678901234567890
//JKT308B JOB CLASS=A,MSGCLASS=Q MSGLEVEL=(1,1), NOTIFY=&SYSUID
// . . .
```

MSGLEVEL Parameter

Controls which output datasets you want to be generated in the job log.

MSGLEVEL=(*statements,messages*)

statements

0 = Only the JOB statement

1 = JCL & procedure statements

2 = Only JCL statements

messages

0 = Only JCL messages

1 = JCL, JES, and operator
messages

```
//xxx JOB . . . MSGLEVEL=(0,0)
```

```
//xxx JOB . . . MSGLEVEL=(1,1)
```

```
//xxx JOB . . . MSGLEVEL=(2,0)
```

◀ **Output the JOB statement and JCL messages**

◀ **Output all JCL and procedure statements and all JCL, JES, and operator message (typical)**

◀ **Output only JCL statements and only JCL messages**

1

2

3

4

5

6

7

8

```
1234567890123456789012345678901234567890123456789012345678901234567890
//JKT308B JOB CLASS=A,MSGCLASS=Q,MSGLEVEL=(1,1),NOTIFY=&SYSUID
// . . .
```

MSGCLASS Parameter Specifying Predefined Value "Q"

JES will handle job log output according to the configuration defined for message class A. This is a system configuration setting not under control of application developers. You will be told which message classes to use for each type of job you work with.

The EXEC Statement

//xxx

JOB . . .

◀ **A job with no steps isn't very useful**



//xxx	JOB . . .
//STEP1	EXEC . . .
//STEP2	EXEC . . .
//STEP3	EXEC . . .
//STEP4	EXEC . . .
//STEP5	EXEC . . .
//STEP6	EXEC . . .
//STEP7	EXEC . . .
//STEP8	EXEC . . .
//STEP9	EXEC . . .
//STEP10	EXEC . . .
//STEP11	EXEC . . .
//STEP12	EXEC . . .
//STEP251	EXEC . . .
//STEP252	EXEC . . .
//STEP253	EXEC . . .
//STEP254	EXEC . . .
//STEP255	EXEC . . .



◀ **A job must have at least one step...**

◀ **...and *can* have up to 255 steps.**

```
//STEP1 EXEC PGM=program
```

```
//STEP2 EXEC PROC=procedure
```

```
//STEP3 EXEC procedure
```

◀ **PGM=** names an executable to be run.

◀ **PROC=** names a catalogued procedure to be inserted into the jobstream at this point.
The procedure may contain multiple steps.

◀ **PROC=** is the default, so you can omit the keyword.

Module Summary

1 2 3 4 5 6 7 8

```
123456789012345678901234567890123456789012345678901234567890
//GMT443N JOB (599, 'T/HK'), 'GENNA', CLASS=A, MSGCLASS=A, MSGLEVEL=(1,1),
//          NOTIFY=&SYSUID
// . . .
```

JOB Statement with Two-Value Job Accounting Parameter

Job accounting parameter has two comma-delimited fields, all enclosed in parentheses.
Programmer name parameter is used.


```
//STEP1 EXEC PGM=program
```

```
//STEP2 EXEC PROC=procedure
```

```
//STEP3 EXEC procedure
```

◀ **PGM=** names an executable to be run.

◀ **PROC=** names a catalogued procedure to be inserted into the jobstream at this point.
The procedure may contain multiple steps.

◀ **PROC=** is the default, so you can omit the keyword.

Up Next:

Breaking Down Data Sets & the DD Statement
