TATA CONSULTANCY SERVICES

TATA

OS/390 & z/OS – Special Datasets and Condition Parameters in JCL

Experience certainty. IT Services
Business Solutions
Outsourcing

Course Objectives

- To understand the
  - Private Program Libraries
  - Concatenation of Datasets
  - Temporary Datasets
  - COND, IF/ELSE, SET parameters

DD Statements for Execution related Data sets

- Specifying private program library
  - JOBLIB
  - STEPLIB
- Specifying dump data set

Specifying Private Program libraries

- Private libraries are specified to execute programs that do not reside in the system libraries
- Job library applies to entire job
- Step libraries applies to only one step in a job
- If both JOBLIB and STEPLIB DD statements are coded, step library is searched first

(Job library ignored for that job step)

JOBLIB : Example

```
//MM01E JOB (36512),'LE MENENDEZ',NOTIFY=MM01
//JOBLIB DD DSNAME=MMA.TEST.LOADLIB,DISP=SHR
//STEP1 EXEC PGM=IEBGENER
.
.
//STEP2 EXEC PGM=ORD2200A
.
.
```

STEPLIB : Example

```
//MM01F JOB (36512),'LE MENENDEZ',NOTIFY=MM01
//STEP1 EXEC PGM=IEBGENER
.
.
//STEP2 EXEC PGM=ORD2200A
//STEPLIB DD DSNAME=MM01.TEST.LOADLIB,DISP=SHR .
.
.
```

Specifying Dump data set

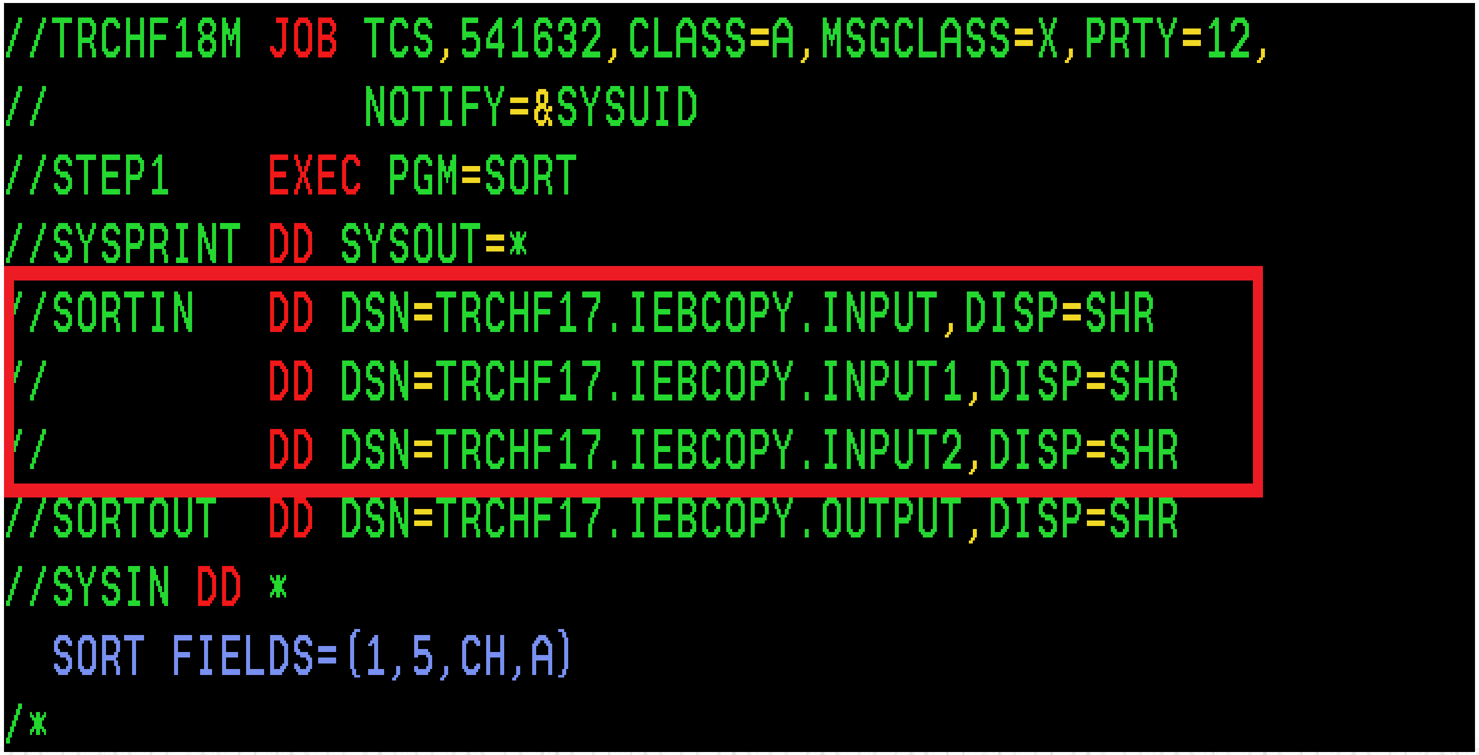
- Storage dump is the content of the virtual storage used by the program before the job terminates with error
- To obtain storage dump, code DD statement named SYSUDUMP, SYSABEND, or SYSMDUMP
- Syntax:

  //SYSDUMP DD SYSOUT=*
- If a job step ends normally, the dump DD statement is ignored

Dataset Concatenation

- Concatenation is used to append files in the JCL for a single ddname.
- The order of appending will be from top FILE to bottom FILE.
- MERGing files differ from concatenation, in a way that merging will also SORT after appending.
- Concatenation can be done upto 255 sequential files and 16 PDS files with a sing ddname/Joblib/Steplib.

Dataset Concatenation



```
//TRCHF18M JOB TCS,541632,CLASS=A,MSGCLASS=X,PRTY=12,
//             NOTIFY=&SYSUID
//STEP1    EXEC PGM=SORT
//SYSPRINT DD SYSOUT=*
//SORTIN   DD DSN=TRCHF17.IEBCOPY.INPUT,DISP=SHR
//         DD DSN=TRCHF17.IEBCOPY.INPUT1,DISP=SHR
//         DD DSN=TRCHF17.IEBCOPY.INPUT2,DISP=SHR
//SORTOUT  DD DSN=TRCHF17.IEBCOPY.OUTPUT,DISP=SHR
//SYSIN DD *
  SORT FIELDS=(1,5,CH,A)
/*
```

Temporary Dataset

- Syntax:

  DSN=&&<*data set name*>

- Dataset exists only during the execution of job. Usually called as Work files

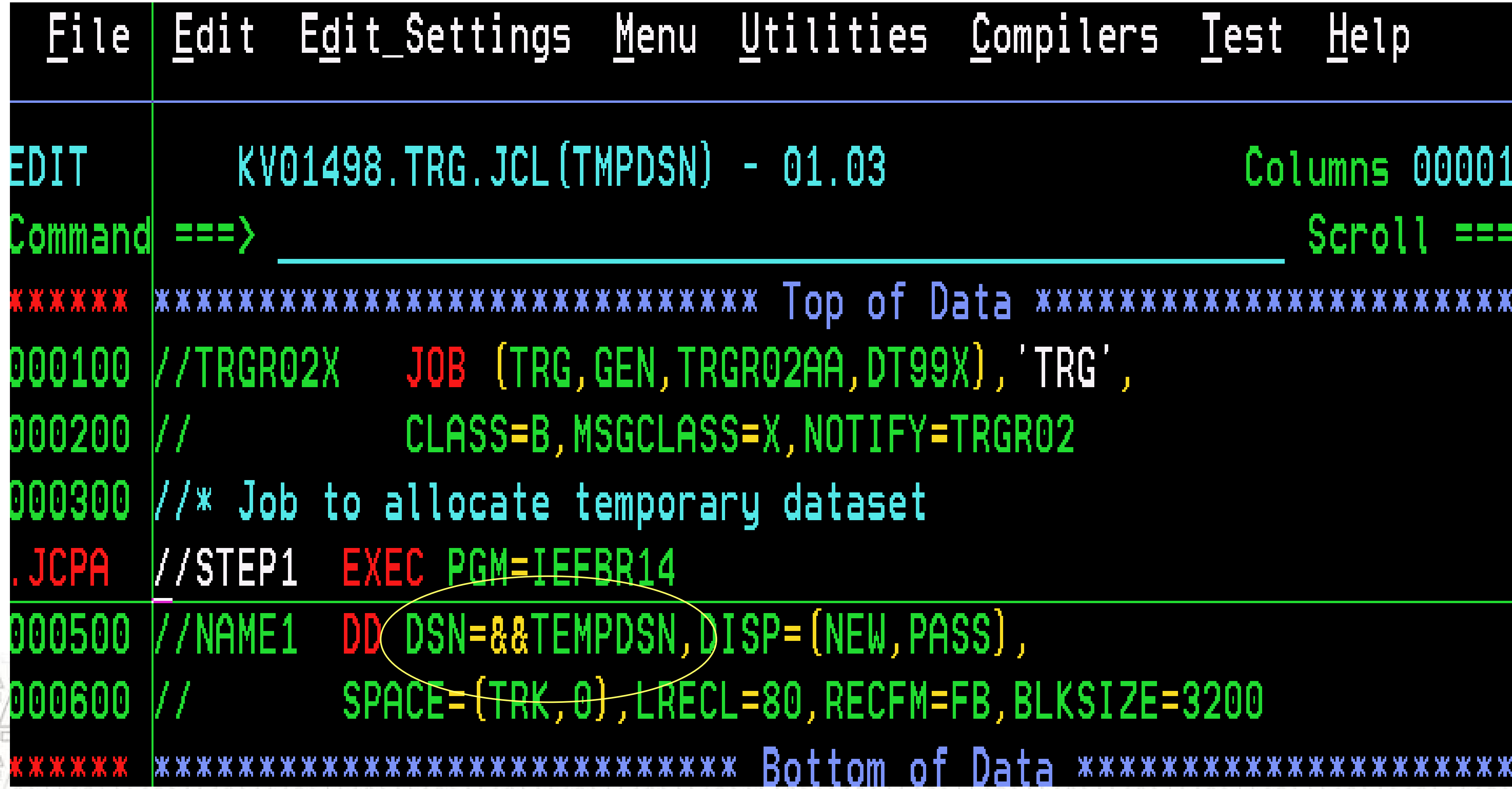
- Always deleted at the end of job

- Resides in Virtual Storage

- Compound DSN, joined by period not allowed in temporary

  dataset

Temporary Dataset : Example



```
EDIT       KV01498.TRG.JCL(TMPDSN) - 01.03                Columns 00001
Command ===> ____________________________________________ Scroll ===
****** ***************************** Top of Data *************************
000100 //TRGR02X    JOB (TRG,GEN,TRGR02AA,DT99X),'TRG',
000200 //           CLASS=B,MSGCLASS=X,NOTIFY=TRGR02
000300 //* Job to allocate temporary dataset
.JCPA  //STEP1   EXEC PGM=IEFBR14
000500 //NAME1   DD DSN=&&TEMPDSN,DISP=(NEW,PASS),
000600 //           SPACE=(TRK,0),LRECL=80,RECFM=FB,BLKSIZE=3200
****** **************************** Bottom of Data ***********************
```

Keyword Parameter in JOB and EXEC - COND

- Syntax:

  COND=(*code,Operator*[,*step name*]) [,EVEN | ONLY]

- Examples:

  //STP1 EXEC PGM=P1,COND=(0,NE)

If any preceding step returns non zero return code this step is bypassed

//STP2 EXEC PGM=P2,COND=(4,LT,STP1)

If step STP1 returns a code less than or equal to 4 then this step is executed else bypassed.

Keyword Parameter- COND

- Examples:

  //STP1 EXEC PGM=P1,COND=EVEN

  STP1 is executed even if a preceding step abends

  //STP2 EXEC PGM=P3,COND=((4,LE,STP1),ONLY)

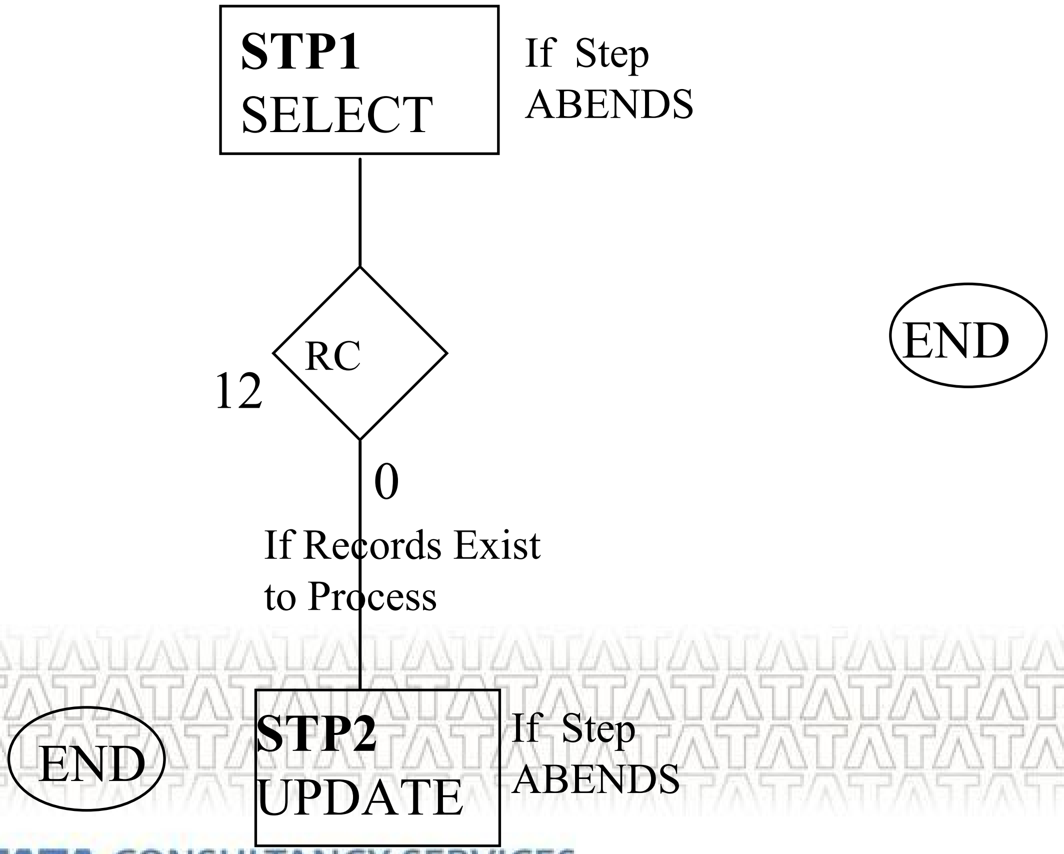
  STP2 is executed only if a preceding step abends and return code of STP1 is less than 4

Relational Operators for COND

	Return code(RC) from just completed step	
cond parameter	continue job if	terminate job if
COND=(code,GT)	RC >=code	RC<code
COND=(code,GE)	RC>code	RC<=code
COND=(code,EQ)	RC !=code	RC=code
COND=(code,LT)	RC<=code	RC>code
COND=(code,LE)	RC<code	RC>=code
COND=(code,NE)	RC=code	RC!=code

COND Parameter with Return Codes

Job requiring **special** Conditional execution



ABOUT SELECT PROGRAM.

SELECT PROGRAM :

RETURNS - 0, IF AT LEAST ONE RECORD IS SELECTED FOR PROCESSING BY **UPDATE** PROGRAM.

RETURNS - 12, IF NO RECORDS ARE SELECTED FOR PROCESSING BY **UPDATE** PROGRAM.

COND Parameter with Return Codes (Contd.

- JCL for executing JOB described in the previous slide

```
//JOB1 JOB ACT1,TRG,CLASS=B,NOTIFY=TRGXXX
//STP1 EXEC PGM=SELECT
.....  ......  DD STATEMENTS......
//STP2  EXEC PGM=UPDATE,COND=(12,LE,STP1)
.....  ......  DD STATEMENTS......
//STP3 EXEC PGM=ABDMP,COND=ONLY
.....  ......  DD STATEMENTS......
//
```

JCL Statements - Optional

- IF / THEN / ELSE / ENDIF Statement for selective execution of a Job Ste
- SET Statement assigns symbolic parameter values
- INCLUDE Statement to copy JCL from a file into Job Stream
- PROC & PEND Statements mark the beginning and end of a Procedure (Cataloged / in-stream)
- JCLLIB Statement names the PROC & JCL Data set

IF/THEN/ELSE/ENDIF

- Used to conditionally execute job step with job
- Examples:

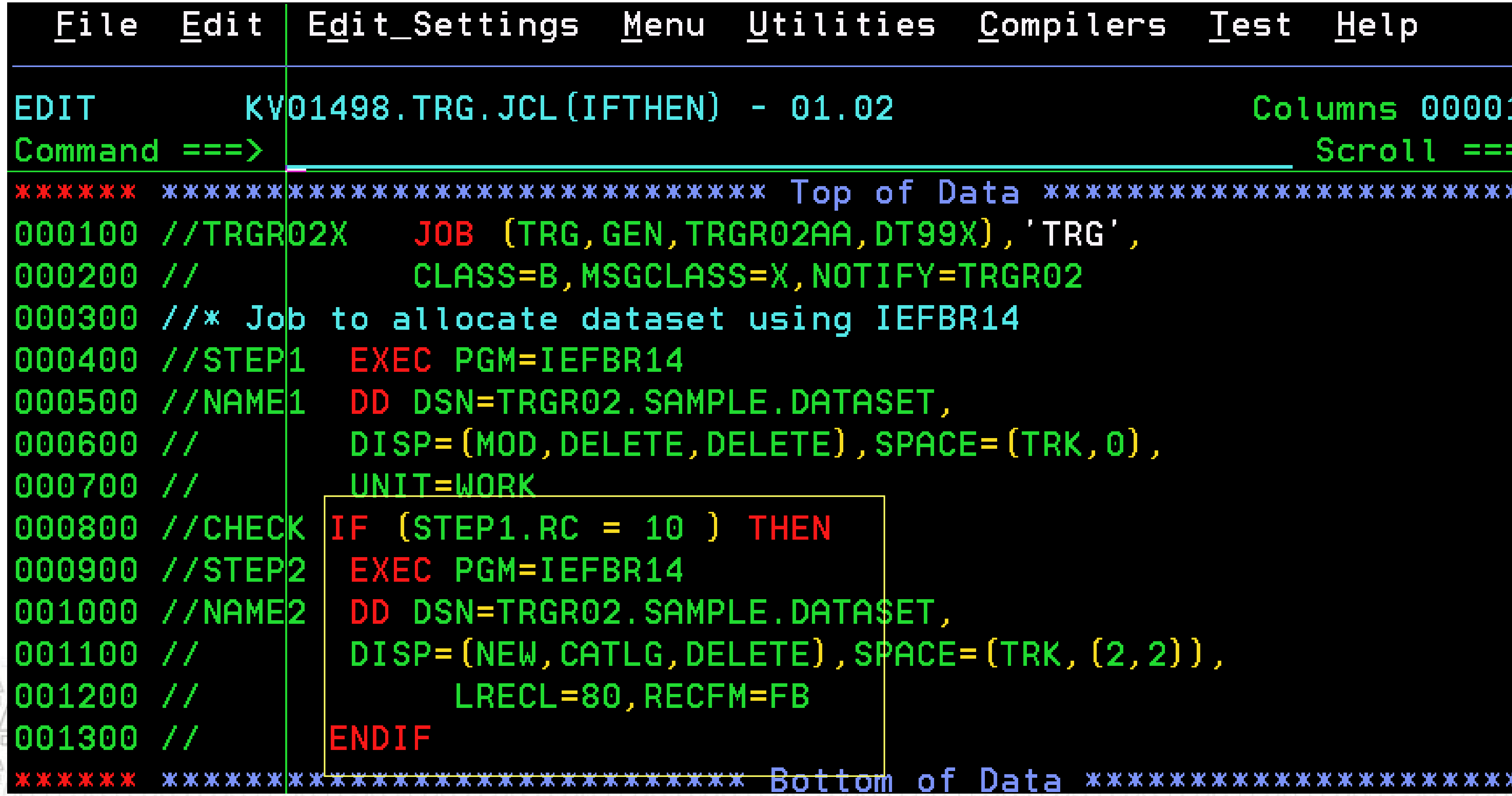
```
//NAME IF (RC GT 4) THEN
```

To Test if RETURN-CODE > 4

```
//NAME IF (STEP10.RC GT 4) THEN
```

To Test if RETURN-CODE from the Job Step STEP10 > 4

IF/THEN/ELSE/ENDIF : Example



```
EDIT       KV01498.TRG.JCL(IFTHEN) - 01.02                Columns 0000
Command ===>                                                 Scroll ===
****** ***************************** Top of Data ************************
000100 //TRGR02X   JOB (TRG,GEN,TRGR02AA,DT99X),'TRG',
000200 //          CLASS=B,MSGCLASS=X,NOTIFY=TRGR02
000300 //* Job to allocate dataset using IEFBR14
000400 //STEP1   EXEC PGM=IEFBR14
000500 //NAME1   DD DSN=TRGR02.SAMPLE.DATASET,
000600 //        DISP=(MOD,DELETE,DELETE),SPACE=(TRK,0),
000700 //        UNIT=WORK
000800 //CHECK  IF (STEP1.RC = 10 ) THEN
000900 //STEP2   EXEC PGM=IEFBR14
001000 //NAME2   DD DSN=TRGR02.SAMPLE.DATASET,
001100 //        DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(2,2)),
001200 //             LRECL=80,RECFM=FB
001300 //       ENDIF
****** **************************** Bottom of Data **********************
```

SET Statement

- Assign initial values to symbolic parameters
- Change or Nullify the values of symbolic parameter
- Syntax:

```
//[NAME]   SET symbolic-parameter=values
```

- If there are more than one parameter, separate them by comma
- Example:

```
//SP1 SET PARM1=OLD,PARM2=22201
```

- Parameter values containing special character enclosed by apostrophes
- Example:

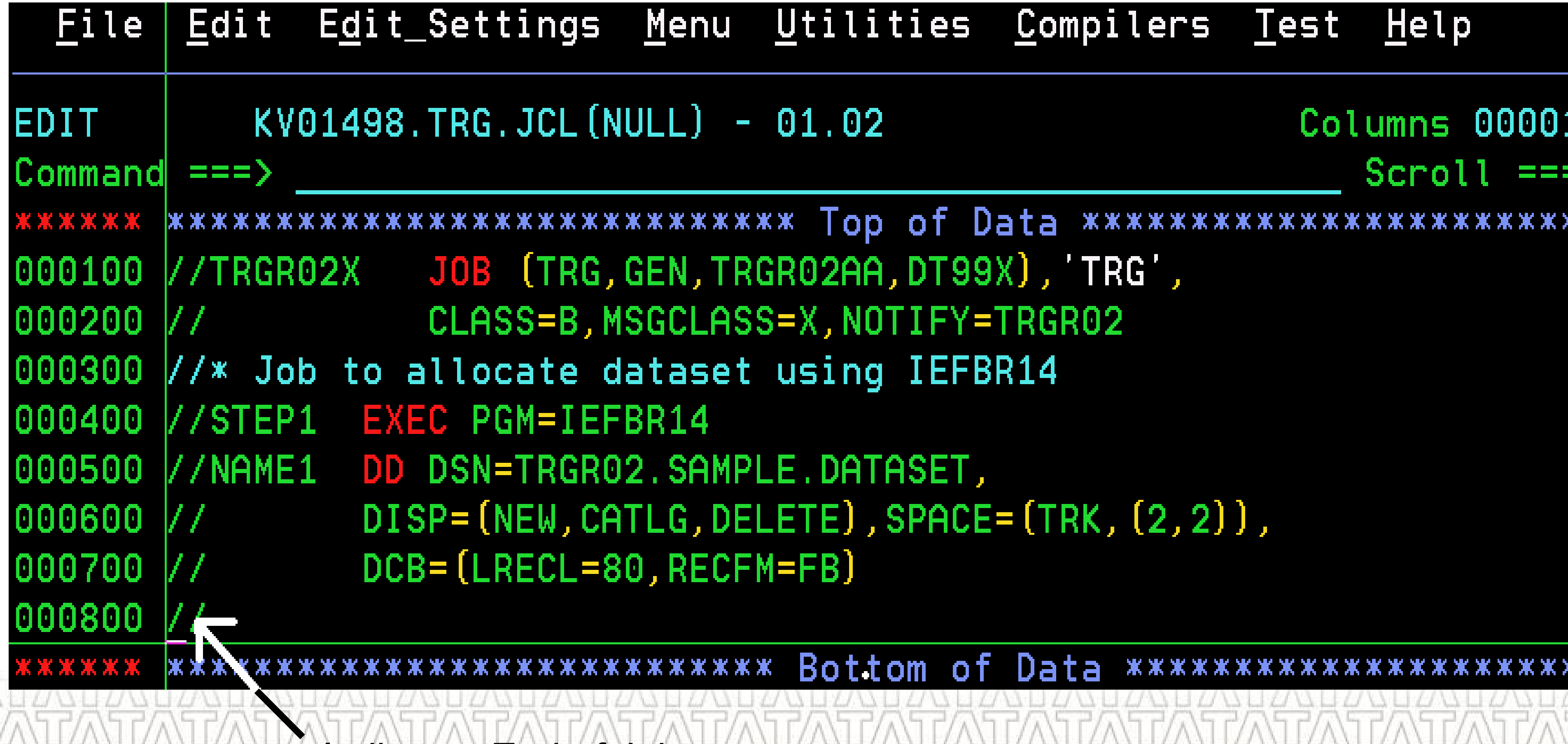
```
//SP3 SET PARM1=‘2004-10-13’
```

INCLUDE Statement

- Identify the member of a partitioned dataset containing the JCL statement
- Syntax:

  //[NAME] INCLUDE MEMBER=name
- INCLUDE group cannot contain
  - JOB Statement
  - PROC and PEND statement
  - JCLLIB statement
  - JES2 and JES3 statement
  - DD * and DD DATA statement

NULL STATEMENT



```
EDIT       KV01498.TRG.JCL(NULL) - 01.02                   Columns 0000
Command ===>                                                 Scroll ===
****** ***************************** Top of Data ***********************
000100 //TRGR02X   JOB (TRG,GEN,TRGR02AA,DT99X),'TRG',
000200 //             CLASS=B,MSGCLASS=X,NOTIFY=TRGR02
000300 //* Job to allocate dataset using IEFBR14
000400 //STEP1   EXEC PGM=IEFBR14
000500 //NAME1   DD DSN=TRGR02.SAMPLE.DATASET,
000600 //           DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(2,2)),
000700 //           DCB=(LRECL=80,RECFM=FB)
000800 //
****** **************************** Bottom of Data **********************
```

Indicates End of Job

JCL Statements - Optional

- IF / THEN / ELSE / ENDIF Statement for selective execution of a Job Ste
- SET Statement assigns symbolic parameter values
- INCLUDE Statement to copy JCL from a file into Job Stream
- PROC & PEND Statements mark the beginning and end of a Procedure (Cataloged / in-stream)
- JCLLIB Statement names the PROC & JCL Data set

IF/THEN/ELSE/ENDIF

- Used to conditionally execute job step with job
- Examples:

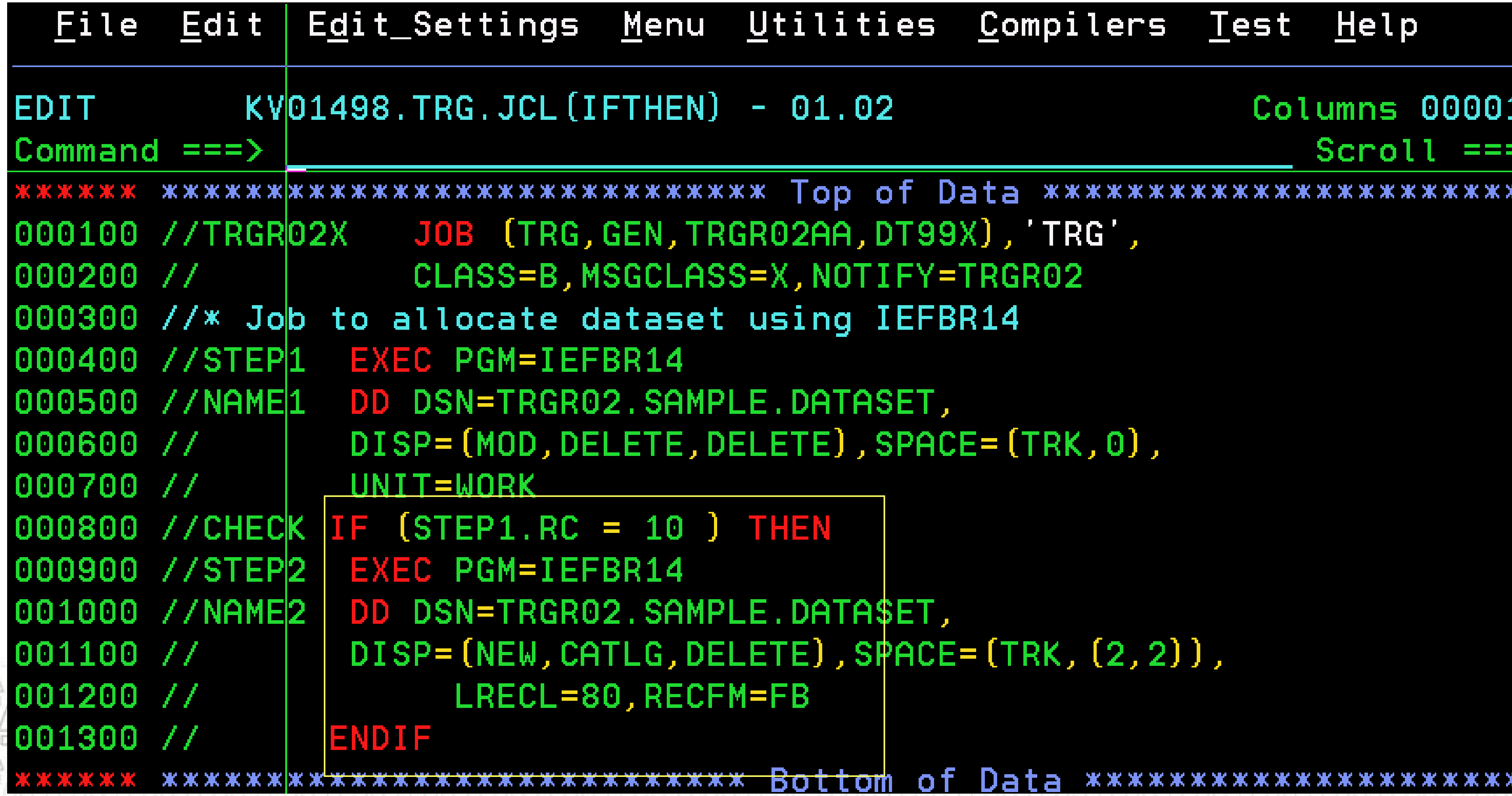
```
//NAME IF (RC GT 4) THEN
```

To Test if RETURN-CODE > 4

```
//NAME IF (STEP10.RC GT 4) THEN
```

To Test if RETURN-CODE from the Job Step STEP10 > 4

IF/THEN/ELSE/ENDIF : Example



```
EDIT       KV01498.TRG.JCL(IFTHEN) - 01.02                    Columns 0000
Command ===>                                                     Scroll ===
****** ***************************** Top of Data ************************
000100 //TRGR02X   JOB (TRG,GEN,TRGR02AA,DT99X),'TRG',
000200 //             CLASS=B,MSGCLASS=X,NOTIFY=TRGR02
000300 //* Job to allocate dataset using IEFBR14
000400 //STEP1   EXEC PGM=IEFBR14
000500 //NAME1   DD DSN=TRGR02.SAMPLE.DATASET,
000600 //           DISP=(MOD,DELETE,DELETE),SPACE=(TRK,0),
000700 //           UNIT=WORK
000800 //CHECK IF (STEP1.RC = 10 ) THEN
000900 //STEP2   EXEC PGM=IEFBR14
001000 //NAME2   DD DSN=TRGR02.SAMPLE.DATASET,
001100 //           DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(2,2)),
001200 //                LRECL=80,RECFM=FB
001300 //         ENDIF
****** **************************** Bottom of Data ***********************
```

SET Statement

- Assign initial values to symbolic parameters
- Change or Nullify the values of symbolic parameter
- Syntax:

```
//[NAME]   SET symbolic-parameter=values
```

- If there are more than one parameter, separate them by comma
- Example:

```
//SP1 SET PARM1=OLD,PARM2=22201
```

- Parameter values containing special character enclosed by apostrophes
- Example:

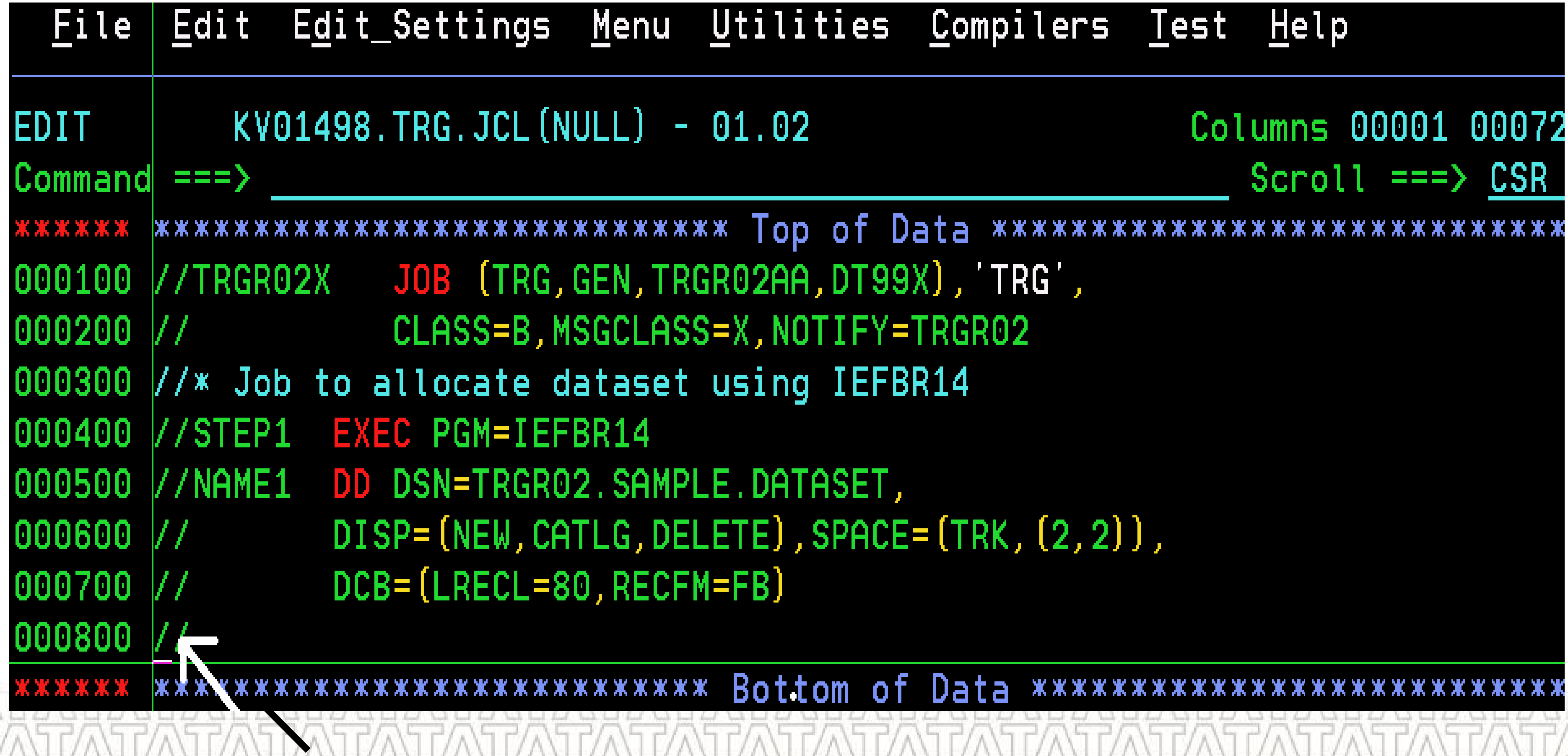
```
//SP3 SET PARM1=‘2004-10-13’
```

INCLUDE Statement

- Identify the member of a partitioned dataset containing the JCL statement
- Syntax:

  //[NAME] INCLUDE MEMBER=name
- INCLUDE group cannot contain
  - JOB Statement
  - PROC and PEND statement
  - JCLLIB statement
  - JES2 and JES3 statement
  - DD * and DD DATA statement

NULL STATEMENT



```
EDIT       KV01498.TRG.JCL(NULL) - 01.02                  Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ***************************** Top of Data ******************************
//TRGR02X   JOB (TRG,GEN,TRGR02AA,DT99X),'TRG',
//          CLASS=B,MSGCLASS=X,NOTIFY=TRGR02
//* Job to allocate dataset using IEFBR14
//STEP1  EXEC PGM=IEFBR14
//NAME1  DD DSN=TRGR02.SAMPLE.DATASET,
//       DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(2,2)),
//       DCB=(LRECL=80,RECFM=FB)
//
****** **************************** Bottom of Data ****************************
```

Indicates End of Job

THANK YOU