

Author	Jan Sadek
Contact	jansadek@outlook.com
Date	2014/11/27
Project title	COND to IF converter
Category	REXX, JCL

- **Description**

Script reads COND statement entered by user, checks its syntax and if COND statement is correct it is converted into two versions of IF statements. Script can be used to learn how COND parameter works, to test your condition before testing it in your jobs or to find out how it works in jobs that you are not familiar with.

Input - user is prompted to enter COND parameter, all rules that apply to this parameter in JCL are also valid here. Input is not case sensitive. Full COND statement should be entered, for example:  
COND=((4,EQ),(12,GT,STEP1))

Output - script processes input, in case in which user enters incorrect statement, he will be informed about error with additional informations about it. If entered statement is correct program will convert it to two kinds of IF statements, one which is equivalent of COND statement and other which is negation of it, for example:

```
IF (RC EQ 4 ! STEP1.RC LT 12) THEN DO NOT EXECUTE...
IF (RC NE 4 & STEP1.RC GE 12) THEN EXECUTE...
```

- **Examples**

```
PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=(4,EQ)
IF (RC EQ 4) THEN DO NOT EXECUTE...
IF (RC NE 4) THEN EXECUTE...

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=((8,LT,STEP1),EVEN)
IF (STEP1.RC GT 8) THEN DO NOT EXECUTE...
IF (STEP1.RC LE 8) & (ABEND ! NOT ABEND) THEN EXECUTE...

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=((8,LT,STEP1),ONLY)
IF (STEP1.RC GT 8) ! NOT ABEND THEN DO NOT EXECUTE...
IF (STEP1.RC LE 8) & ABEND THEN EXECUTE...

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=((4,EQ),ONLY,(44,LT,STEP3))
IF (RC EQ 4 ! STEP3.RC GT 44) ! NOT ABEND THEN DO NOT EXECUTE...
IF (RC NE 4 & STEP3.RC LE 44) & ABEND THEN EXECUTE...

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=((2,GE,STEP3),EVEN,(12,LT))
IF (STEP3.RC LE 2 ! RC GT 12) THEN DO NOT EXECUTE...
IF (STEP3.RC GT 2 & RC LE 12) & (ABEND ! NOT ABEND) THEN EXECUTE...

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=(EVEN,(33,EQ,STEP3))
IF (STEP3.RC EQ 33) THEN DO NOT EXECUTE...
IF (STEP3.RC NE 33) & (ABEND ! NOT ABEND) THEN EXECUTE...

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=ONLY
IF NOT ABEND THEN DO NOT EXECUTE...
IF ABEND THEN EXECUTE...

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=EVEN
IF NOT (ABEND ! NOT ABEND) THEN DO NOT EXECUTE...
IF ABEND ! NOT ABEND THEN THEN EXECUTE...

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=((3,EQ,STEP3),(44,LT,STEP2),(8,LT,STEPX))
IF (STEP3.RC EQ 3 ! STEP2.RC GT 44 ! STEPX.RC GT 8) THEN DO NOT EXECUTE...
IF (STEP3.RC NE 3 & STEP2.RC LE 44 & STEPX.RC LE 8) THEN EXECUTE...

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
Q
***
```

- **Error handling**

Script tests every entered COND parameter, it must meet following conditions:

- There must be 1 to 8 conditions
- RC must be a number
- RC must have value 0-4095
- Operator must be valid
- There can be only one EVEN or ONLY condition in statement
- Parenthesis must match
- COND statement must have correct syntax

```

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=((4,EQ),(4,EQ),(4,EQ),(4,EQ),(4,EQ),(4,EQ),(4,EQ),(4,EQ))
THERE CAN BE MAX 8 CONDITIONS

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=(-1,LT,STEP1)
RETURN CODE MUST HAVE VALUE 0-4095

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=(R,GE)
ENTERED RETURN CODE IS NOT A NUMBER

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=(4,LU,STEP1)
INCORRECT OPERATOR: LU

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=((4,GE,MYSTEP),EVEN,ONLY)
THERE CAN BE ONLY ONE EVEN OR ONLY CONDITION, ALSO YOU CANNOT USE BOTH OF THEM
IN ONE STATEMENT

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
COND=(ONLY,(5,LT,STEP5))
SYNTAX ERROR, PARENTHESIS DOES NOT MATCH

PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT
CANT=EVEN
SYNTAX ERROR, ENTERED COND STATEMENT IS NOT CORRECT

```

- **Testing results with JCL**

In order to make sure that conditions works exactly like intended it is recommended to additionally test those in various scenatios. Following JCL can be used for this task.

```

//CONDTEST JOB
//STEP1 EXEC PGM=S806
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
SET MAXCC=0
/*****
//STEP2 EXEC PGM=IDCAMS,COND=EVEN
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
SET MAXCC=0
/*****
//STEP3 EXEC PGM=IDCAMS,COND=EVEN
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
SET MAXCC=15
/*****
//STEPT1 EXEC PGM=IEBGENER,COND=((4,EQ),ONLY,(44,LT,STEP3))
//SYSPRINT DD SYSOUT=*

```

```

//SYSIN      DD DUMMY
//SYSUT1     DD *
STEP T1 EXECUTED
//SYSUT2     DD SYSOUT=*
//*****
// IF (RC EQ 4 ! STEPC3.RC GT 44) ! NOT ABEND THEN
//STEPT2     EXEC PGM=IEBGENER
//SYSPRINT   DD SYSOUT=*
//SYSIN      DD DUMMY
//SYSUT1     DD *
STEP T1 SHOULD NOT EXECUTE
//SYSUT2     DD SYSOUT=*
// ENDIF
//*****
// IF (RC NE 4 & STEPC3.RC LE 44) & ABEND THEN
//STEPT3     EXEC PGM=IEBGENER
//SYSPRINT   DD SYSOUT=*
//SYSIN      DD DUMMY
//SYSUT1     DD *
STEP T1 SHOULD EXECUTE
//SYSUT2     DD SYSOUT=*
// ENDIF

```

Steps STEPC1, STEPC2 and STEPC3 are used to set various conditions such as different return codes or abends. Simplest way to cause an abend is to execute non-existing program (S806 in this example).

Step STEPT1 is used to check original COND parameter. If any condition here is true step will not execute. Abend conditions (EVEN and ONLY) works differently and they define in what case step will execute.

Step STEPT2 is used to test IF statement that is negation of COND parameter, it should never execute when STEPT1 does.

Step STEPT3 is used to test IF statement that is acts identically like COND parameter, it should always execute when STEPT1 does.

Note that COND parameters can test for RC from 0 to 4095 but IDCAMS can set up MAXCC from 0 to 16.

#### Output from presented JCL:

STEPNAME	RC	EXCP	CPU
STEPC1	*S806	11	.02
STEPC2	00	44	.00
STEPC3	15	49	.00
STEPT1	00	55	.00
STEPT2	FLUSH	0	.00
STEPT3	00	50	.00
...			
PROCESSING ENDED AT EOD			
STEP T1 EXECUTED			
DATA SET UTILITY - GENERATE			
IEB352I WARNING: ONE OR MORE OF THE OUTPUT DCB PARMS COPIED FROM INPUT			

PROCESSING ENDED AT EOD  
STEP T1 SHOULD EXECUTE

- **Source code**

```
/* REXX - COND TO IF CONVERTER */
ERRORMSG = "SYNTAX ERROR, ENTERED COND STATEMENT IS NOT CORRECT"
ERRPRMSG = "SYNTAX ERROR, PARENTHESIS DOES NOT MATCH"
SAY "WELCOME TO 'COND TO IF CONVERTER'!!!"
DO FOREVER
  SAY ''
  SAY "PLEASE ENTER A COND STATEMENT OR 'Q' TO QUIT"
  DO J=1 TO 10
    DROP IFCOND.J
  END
  EVEN=0
  ONLY=0

  PARSE UPPER PULL UIN
  UPPER UIN
  UIN=SPACE(UIN)
  IF UIN="Q" THEN RETURN 0
  IF SUBSTR(UIN,1,5)<>"COND=" THEN
  DO
    SAY ERRORMSG
    ITERATE
  END
  UIN=SUBSTR(UIN,6)

  IF SUBSTR(UIN,1,2)="(" ! SUBSTR(UIN,1,2)="(" ! SUBSTR(UIN,1,2)="(" ! SUBSTR(UIN,1,2)="(" THEN
  DO
    IF SUBSTR(UIN,LENGTH(UIN),1)<>")" THEN
    DO
      SAY ERRPRMSG
      ITERATE
    END
    ELSE
    DO
      UIN=SUBSTR(UIN,2)
      UIN=SUBSTR(UIN,1,LENGTH(UIN)-1)
    END
  END
  ERR=0
  CALL EXTRACTCONDS UIN
  IF EVEN+ONLY>1 THEN
  DO
    SAY "THERE CAN BE ONLY ONE EVEN OR ONLY CONDITION, ALSO YOU" ,
      "CANNOT USE BOTH OF THEM IN ONE STATEMENT"
  END
  ELSE
  IF ERR<>1 THEN
  DO
    CALL CREATEIF
```

```

    SAY IFS
    SAY IFS2
END
END
RETURN 0

CREATEIF:
IFS = "IF ("
IFS2 = "IF ("
IF IFCOND.1="IFCOND."1 THEN
DO
    IF ONLY=1 THEN
    DO
        IFS="IF NOT ABEND THEN DO NOT EXECUTE..."
        IFS2="IF ABEND THEN EXECUTE..."
    END
    IF EVEN=1 THEN
    DO
        IFS="IF NOT (ABEND ! NOT ABEND) THEN DO NOT EXECUTE..."
        IFS2="IF ABEND ! NOT ABEND THEN THEN EXECUTE..."
    END
    RETURN 0
END
DO J=1 TO 10
    IF IFCOND.J="IFCOND."J THEN LEAVE
    IFS=IFS!!IFCOND.J' ! '
    IFS2=IFS2!!IFCOND2.J' & '
END
IFS=SUBSTR(IFS,1,LENGTH(IFS)-3)
IFS2=SUBSTR(IFS2,1,LENGTH(IFS2)-3)
IF ONLY=1 THEN
DO
    IFS=IFS") ! NOT ABEND THEN DO NOT EXECUTE..."
    IFS2=IFS2") & ABEND THEN EXECUTE..."
    RETURN 0
END
IF EVEN=1 THEN
DO
    IFS=IFS") THEN DO NOT EXECUTE..."
    IFS2=IFS2") & (ABEND ! NOT ABEND) THEN EXECUTE..."
    RETURN 0
END
IFS=IFS") THEN DO NOT EXECUTE..."
IFS2=IFS2") THEN EXECUTE..."
RETURN 0

EXTRACTCONDS:
PARSE ARG CONDS
J=1
DO 10
    IF J>8 THEN
    DO
        SAY "THERE CAN BE MAX 8 CONDITIONS"
        ERR=1
        RETURN 0
    
```

```

END
IF SUBSTR(CONDS,1,4)="EVEN" THEN
DO
  EVEN=EVEN+1
  CONDS=SUBSTR(CONDS,5)
  IF SUBSTR(CONDS,1,1)=',' THEN
  DO
    CONDS=SUBSTR(CONDS,2)
    ITERATE
  END
  ELSE LEAVE
END
IF SUBSTR(CONDS,1,4)="ONLY" THEN
DO
  ONLY=ONLY+1
  CONDS=SUBSTR(CONDS,5)
  IF SUBSTR(CONDS,1,1)=',' THEN
  DO
    CONDS=SUBSTR(CONDS,2)
    ITERATE
  END
  ELSE RETURN 0
END
IF POS(',')',CONDS)<>0 THEN
  C=SUBSTR(CONDS,2,POS(',')',CONDS)-2)
ELSE
DO
  SAY ERRPRMSG
  ERR=1
  RETURN 0
END
PARSE VAR C CC ',' OPER ',' STEP

IFCOND.J=''
IFCOND2.J=''
IF DATATYPE(CC,N)<>1 THEN
DO
  SAY "ENTERED RETURN CODE IS NOT A NUMBER"
  ERR=1
  RETURN 0
END
IF CC<0 ! CC>4095 THEN
DO
  SAY "RETURN CODE MUST HAVE VALUE 0-4095"
  ERR=1
  RETURN 0
END
IF STEP<>' ' THEN
  IFCOND.J=STEP'.'

SELECT
  WHEN OPER='GE' THEN OP='LE'
  WHEN OPER='GT' THEN OP='LT'
  WHEN OPER='LE' THEN OP='GE'
  WHEN OPER='LT' THEN OP='GT'

```

```

    WHEN OPER='EQ' THEN OP='EQ'
    WHEN OPER='NE' THEN OP='NE'
    OTHERWISE
    DO
        SAY "INCORRECT OPERATOR: "OPER
        ERR=1
        RETURN 0
    END
END

IFCOND2.J=IFCOND.J
SELECT
    WHEN OP='GE' THEN OP2='LT'
    WHEN OP='GT' THEN OP2='LE'
    WHEN OP='LE' THEN OP2='GT'
    WHEN OP='LT' THEN OP2='GE'
    WHEN OP='EQ' THEN OP2='NE'
    WHEN OP='NE' THEN OP2='EQ'
    OTHERWISE
    DO
        SAY "INCORRECT OPERATOR: "OP
        ERR=1
        RETURN 0
    END
END

IFCOND.J=IFCOND.J'RC 'OP' 'CC
IFCOND2.J=IFCOND2.J'RC 'OP2' 'CC
CONDS=SUBSTR(CONDS,POS(')',CONDS)+1)

IF SUBSTR(CONDS,1,1)=',' THEN CONDS=SUBSTR(CONDS,2)
ELSE LEAVE
J=J+1
END
RETURN 0

```