

## **CLP PROGRAMMING(Control Language)**

Functionalities of a CL program:

- To call program interactively or in batch mode.**
- To control the sequencing of those programs.**
- Process files and other objects.**
- To monitor for all kind of messages.**
- To Make communication between different jobs.**
- Change the attribute of an object and then process it.**

### **CL source program segments**

#### **1) PGM command**

**PGM PARM (&A)**

PGM is an Optional part for the beginning of the CL program. But it becomes mandatory if we need to pass parameter to the CL program.

#### **2) Declare commands**

**(DCL, DCLF, COPYRIGHT)**

The declare commands must precede all other commands except the PGM command.

#### **3) INCLUDE command**

CL command to embed additional CL source commands at compile time

#### **4) CL processing commands**

**CHGVAR, SNDPGMMMSG, OVRDBF, DLTF**

#### **5) Logic control commands**

**IF, THEN, ELSE, DO, ENDDO, DOWHILE, DOUNTIL, DOFOR, LEAVE, ITERATE, GOTO, SELECT, ENDSELECT, WHEN, OTHERWISE, ENDSUBR**

#### **6) Built-in functions**

**%SUBSTRING (%SST), %BINARY (%BIN), %ADDRESS (%ADDR), %CAT()**

#### **7) Program control commands**

**CALL, RETURN, TFRCTL**

#### **8) Procedure control commands**

**CALLPRC**

CL command to pass control to another procedure

#### **9) ENDPGM command(optional)**

**ENDPGM**

### **Limitations of CL (comparison to RPG)**

- We cannot use CL program to ADD or UPDATE records in database files.
- It has limited printing capability.

- It cannot use Program described files.
- It doesn't support subfile (to display more than one record), but a single output message subfile is a special type of subfile that is supported well in CL.
- We cannot declare more than one object (file) in a CL program ( In CLLE, we can use more than 1 file to read with OPENID concept).

### **OPENID in DCLF**

- Open id is an identifier for any opened file in the CL program.

```
DCLF      FILE (AMIT/ACCOUNT)  OPNID (ID1)
```

### **\*CAT/\*BCAT/\*TCAT**

```
*CAT or || Joins two strings "as is"
*BCAT or |> Joins two strings with a single blank space between
*TCAT or |< Joins two strings trimming first string
```

```
'String One ' *CAT 'String Two ' Yields 'String One String Two '
'String One ' *BCAT 'String Two ' Yields 'String One String Two '
'String One ' *TCAT 'String Two ' Yields 'String OneString Two '
```

### **Multiple file read operation in CL**

If we want to use multiple files in the CL program, we need to use OPEN ID for this.

### **PASSING THE PARAMETER IN CL**

PGM	PARM (&IPARM)
-----	---------------

**DSPMSG →** to view the msg

CLR

### **Open query file**

- OPNQRYF command creates a temporary access path for a file and after its use the access path is discarded.
- We change the access path of a file dynamically with the help of this command.
- It can be used to select a subset of the available records, order the records, group the records, join the records.
- Before we go through open query file we should understand open data path also.

### **Open Data Path**

- This is a temporary object which is used to create dynamic access path for a file.
- Access path describes the order in which records are to be read.
- Access paths can be kept on the system permanently (such as physical or logical file) or temporarily (OPNQRYF). OPNQRYF command creates a temporary access path for use one time, and then discard the access path.
- The open data path contains the information like file name, format name, current record pointer, record selection information etc.
- Open data path has only one cursor. Each program that shares the same ODP has only one image of the cursor. So if the cursor is repositioned in the called program then the cursor will still be pointing at the same repositioned record.

- Languages other than CL cannot make dynamic access path or in other words cannot perform dynamic record selection. If we want other programs like RPG also to perform dynamic record selection, we can share the access path of the file (that we create in CL) with the RPG program.
- To share the access path we create an open data path in our CL program with OPNQRYF command and then we share this ODP with the called RPG program. This sharing is done with the help of SHARE (\*YES) in the OVRDBF command.

### Steps to create and use an OPNQRYF

#### I. OVRDBF

FILE (file PF) TOFILE (LIB/ PF) SHARE (\*Yes)

#### II. OPNQRYF

```
OPNQRYF      FILE(lib name/file name          +
                  Member-name           +
                  Record-format-name) +
                  OPTION(open-option)   +
                  FORMAT(lib name/database file name + +
                         Record-format name) +
                  QRYSLT(query selection) +
                  KEYFLD(field name)
```

The attribute of OPNQRYF is described below:

- **FILE** : File to be processed.
- **OPTION:** In which mode the file is to be processed.

**OPTION(\*INP \*OUT \*UPD \*DLT \*ALL)**

\*ALL is combination of first four.

- **FORMAT:** File format
- **QRYSLT:**

**\*ALL**

**FILE (LIB / PF) QRYSLT(\*ALL)**

It includes all records.

**\*BCAT**

**FILE (LIB / PF) QRYSLT ('EMPNO \*EQ ' \*BCAT &A)**

We can use \*BCAT to insert the variable's value in an expression with blanks in between.

**%WLDCRD**

It is similar to %LIKE in SQL

**QRYSLT ('PARTY\_NAME \*EQ %WLDCRD ("S\* ")')**

It will fetch all the records where party name starts from S.

**\*CT**

It will fetch all the records, which contains the particular character.

**QRYSLT ('PARTY\_NAME \*CT "S" ')**

It will fetch all the records where party name contains the character 'S'.

## %RANGE

It will fetch the records within the specific range

**QRYSLT ('PARTY\_NUM \*EQ %RANGE (10000001 20000001)')**

It will fetch all the records where party number falls in the range 10000001 and 20000001.

## KEYFLD:

**OPNQRYF FILE(ACCOUNT)**

**QRYSLT(' CURRENCY = "USD" ')**

**KEYFLD((ORG-CODE \*DESCEND) (ACCOUNT))**

II. CALL PGM (LIB/PGM-NAME) PARM ()

IV. DLTOVR

DLTOVR FILE (OPNPF)

V. CLOF

CLOF OPNID (OPNPF)

1. Calculator program - Program that accepts 2 numbers and an operator as input and display the result. Also validates the operator.

```
Columns . . . : 1 80          Browse          GOPSLIB/QCLSRC
SEU=>                                     CONDITION2
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
      ***** Beginning of data *****
0001.00      PGM      PARM(&NUM1 &NUM2 &OPERATOR)           250113
0001.01
0002.00 /* VARIABLE DECLARATION*/
0003.00      DCL      VAR(&NUM1) TYPE(*DEC) LEN(5 0)           250113
0004.00      DCL      VAR(&NUM2) TYPE(*DEC) LEN(5 0)           250113
0005.00      DCL      VAR(&RES) TYPE(*DEC) LEN(7 0)           250113
0006.01      DCL      VAR(&MSG) TYPE(*CHAR) LEN(20)           250113
0007.00      DCL      VAR(&OPERATOR) TYPE(*CHAR) LEN(1)           250113
0007.01
0008.00 /* IF CONDITION USING %CHAR() FOR RESULT */
0008.01 /* *AND, *OR, AND *NOT OPERATORS */
0008.02      IF       COND(&OPERATOR *NE '+') *AND &OPERATOR *NE +
0008.03          '-' *AND &OPERATOR *NE '*' *AND &OPERATOR +
0008.04          '*NE '/') THEN(DO)
0008.05      CHGVAR   VAR(&MSG) VALUE('Invalid operator')        250113
0008.06      SNDMSG   MSG(&MSG) TOUSR(WING15)                  250113
0008.07
0008.08 /* GOTO ENDPGM IF INVALID OPERATOR*/
0008.09 /* IT TAKES THE CONTROL TO A SPECIFIED TAG-LABEL OR END SOB-ROUTINE LABEL*/

```

```
Columns . . . : 1 80          Browse          GOPSLIB/QCLSRC
SEU=>                                     CONDITION2
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
0008.10      GOTO    CMDBLBL(END)           250113
0008.11      ENDDO
0008.12
0008.13      IF       COND(&OPERATOR = '+') THEN(DO)
0008.14          CHGVAR VAR(&RES) VALUE(&NUM1 + &NUM2)           250113
0008.15          CHGVAR   VAR(&MSG) VALUE('ADDITION :'||%CHAR(&RES)) 250113
0008.16          ENDDO
0014.00
0014.01      IF       COND(&OPERATOR = '-') THEN(DO)
0014.02          CHGVAR VAR(&RES) VALUE(&NUM1 - &NUM2)           250113
0014.03          CHGVAR   VAR(&MSG) VALUE('SUBTRACTION :'||%CHAR(&RES)) 250113
0014.04          ENDDO
0014.05
0014.06
0014.07      IF       COND(&OPERATOR = '*') THEN(DO)
0014.08          CHGVAR VAR(&RES) VALUE(&NUM1 * &NUM2)           250113
0014.09          CHGVAR   VAR(&MSG) VALUE('PRODUCT :'||%CHAR(&RES)) 250113
0014.10          ENDDO
0014.11
0014.12
0014.13      IF       COND(&OPERATOR = '/') THEN(DO)
0014.14          CHGVAR VAR(&RES) VALUE(&NUM1 / &NUM2)           250113

```

## SNDMSG

RCVMSG

```

Columns . . . : 1 80                                Browse                               GOPZLIB/QCLSRC
SEU==>                                         CPYFERR1
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****
0001.00      PGM       PARM(&SRCFILE &DSTFILE)                           250119
0002.00      DCL       VAR(&SRCFILE) TYPE(*CHAR) LEN(10)                   250119
0003.00      DCL       VAR(&DSTFILE) TYPE(*CHAR) LEN(10)                   250119
0003.01      DCL       VAR(&DATA1) TYPE(*CHAR) LEN(100)                  250120
0003.02      DCL       VAR(&DATA2) TYPE(*CHAR) LEN(100)                  250120
0003.03      DCL       VAR(&ID) TYPE(*CHAR) LEN(7)                      250120
0004.00
0004.01      CPYF      FROMFILE(&SRCFILE) TOFILE(GOPZLIB/&DSTFILE) +
0004.02              MBROPT(*REPLACE)                                     250120
0005.00      MONMSG   MSGID(CPF0000) EXEC(DO)                            250120
0005.01      RCVMSG   MSGQ(*PGMQ) MSGID(&ID) MSG(&DATA1) +
0005.02              SECLEVEL(&DATA2)                                     250120
0005.03      SNDMSG   MSG(&id)||' '|&data1||&data2) TOUSR(WING15)        250120
0005.04      ENDDO
0010.00      ENDFGM
***** End of data *****

```

Give msgq as \*PGMQ

Receive Message (RCVMSG)

Type choices, press Enter.

Coded character set ID . . . . .	<input checked="" type="checkbox"/> JOB	*JOB, *HEX...
Reject default reply . . . . .	<input type="checkbox"/> NOALWRJT	*NOALWRJT, *ALWRJT
CL var for KEYVAR (4) . . .	<input type="checkbox"/>	Character value
CL var for 1st level text . . . >	<input type="checkbox"/> &DATA1	Character value
CL var for MSGLEN (5 0) . . .	<input type="checkbox"/>	Number
CL var for 2nd level text . . . >	<input type="checkbox"/> &DATA2	Character value
CL var for SECLVLLEN (5 0) . .	<input type="checkbox"/>	Number
CL var for msg data . . . . .	<input type="checkbox"/>	Character value
CL var for MSGDTALEN (5 0) . .	<input type="checkbox"/>	Number
CL var for MSGID (7) . . >	<input type="checkbox"/> &ID	Character value
CL var for SEV (2 0) . .	<input type="checkbox"/>	Number
CL var for SENDER (80) . .	<input type="checkbox"/>	Character value
Sender format . . . . . . . .	<input type="checkbox"/> *SHORT	*SHORT, *LONG
CL var for RTNTYPE (2) . .	<input type="checkbox"/>	Character value
CL var for ALROPT (9) . .	<input type="checkbox"/>	Character value
CL var for MSGF (10) . .	<input type="checkbox"/>	Character value

Give 1<sup>st</sup> lvl text 2<sup>nd</sup> lvl and msgid

SNDBRKMSG

## SNDUSRMSG & SNDPGMMMSG

**SNDUSRMSG:** This command is used to send a message to a specific user or workstation. It can also be used to display a message on the user's screen.

**SNDPGMMMSG:** This command is used to send a message to a program message queue. This is often used within applications to communicate information or errors between programs.

If msg is taken from some file we need to specify the message identifier and message file. Else, text is enough.

```
Send User Message (SNDUSRMSG)

Type choices, press Enter.

Message data field values . . . [REDACTED]

Valid reply values . . . . . *NONE
                           + for more values
Default reply value . . . . . *MSGDFT
Message type . . . . . . . > *INQ          *INQ, *INFO
To message queue . . . . . . . *           Name, *, *EXT, *SYSOPR
Library . . . . . . . . . . . *           Name, *LIBL, *CURLIB
To user profile . . . . . . . > WING15    Name, *SYSOPR, *REQUESTER
CL var for message reply . . . . > &RPLY    Character value
```

If we need to validate the replies we can provide the list in valid reply values field.

To get reply ,ie. For mandatory reply we need to give msg type as \*INQ ,else you can give \*INFO.

Specify the msg variable also

2. Calculator program - Program that accepts 2 numbers and an operator as input from screen and display the result. Also validates the operator.

```
CALCULATOR

ENTER 1ST NUMBER : 99999-
ENTER 2ND NUMBER : 99999-
ENTER THE OPERATOR : BBBB

RESULT : Oooooooooooooooo
```

Columns . . . :	1 80	Browse	GOPZLIB/QCLSRC CALCCLP
SEU==>			
FMT **	.....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8		
***** Beginning of data *****			
0001.00	PGM	250114	
0001.01	DCLF FILE(CALCULATOR)	250114	
0001.02	DOWHILE COND(&IN03 = '0')	250114	
0001.03	SNDRCVF RCDFMT(CALCENTRY)	250114	
0001.05	IF COND(&IN03 = '1') THEN(LEAVE)	250114	
0001.06	IF COND(&OPERATOR *NE '+' *AND &OPERATOR *NE '+-' *AND &OPERATOR *NE '**' *AND +	250114	
0001.07	&OPERATOR *NE '/') THEN(CHGVAR +	250114	
0001.08	VAR(&SMMSG) VALUE('INVALID OPERATOR'))	250114	
0001.09			
0001.11	ELSE CMD(DO)	250114	
0001.12	CHGVAR VAR(&SMMSG) VALUE(' ')	250114	
0001.13	IF COND(&OPERATOR = '+') THEN(CHGVAR +	250114	
0001.14	VAR(&RESULT) VALUE(&SNUM1 + &SNUM2))	250114	
0001.15	IF COND(&OPERATOR = '-') THEN(CHGVAR +	250114	
0001.16	VAR(&RESULT) VALUE(&SNUM1 - &SNUM2))	250114	
0001.17	IF COND(&OPERATOR = '**') THEN(CHGVAR +	250114	
0001.18	VAR(&RESULT) VALUE(&SNUM1 * &SNUM2))	250114	
0001.19	IF COND(&OPERATOR = '/') THEN(CHGVAR +	250114	
0001.20	VAR(&RESULT) VALUE(&SNUM1 / &SNUM2))	250114	
0001.21	ENDDO	250114	
0001.22	ENDDO	250114	
0029.00	ENDPGM	250114	
***** End of data *****			

3. Finding greatest among the three numbers accepted as parameters

```

Columns . . . : 1 80
SEU==> Browse
GOPZLIB/QCLSRC
CONDITION3
***** Beginning of data *****
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
0001.00 PGM      PARM(&NUM1 &NUM2 &NUM3)                                250114
0002.00 /* VARIABLE DECLARATION*/
0003.00   DCL      VAR(&NUM1) TYPE(*DEC) LEN(5 0)                         250113
0004.00   DCL      VAR(&NUM2) TYPE(*DEC) LEN(5 0)                         250113
0004.01   DCL      VAR(&NUM3) TYPE(*DEC) LEN(5 0)                         250113
0005.00   DCL      VAR(&RES) TYPE(*DEC) LEN(7 0)                         250113
0006.01   DCL      VAR(&MSG) TYPE(*CHAR) LEN(20)                          250113
0008.00 /* IF CODITION USING %CHAR() FOR RESULT */
0008.01 /* RELATIONAL (*EQ, =, *GT, >, *LT, <, *GE, >=, */          250114
0008.02 /* , *LE, <=, *NE, -=, *NG, ->, *NL, -<) */
0008.03   IF       COND(&NUM1 *GT &NUM2 *AND &NUM1 *GT &NUM3) +        250113
0008.04     THEN(DO)                                         250113
0008.05   CHGVAR  VAR(&RES) VALUE(&NUM1)                           250114
0008.08   ENDDO
0008.09
0008.10   ELSE    CMD(DO)                                         250113
0014.18
0014.19   IF       COND(&NUM2 *GT &NUM1 *AND &NUM2 *GT &NUM3) +        250113
0014.21     THEN(DO)                                         250113
0014.22   CHGVAR  VAR(&RES) VALUE(&NUM2)                           250114
0014.25   ENDDO
0014.26
0014.27   ELSE    CMD(DO)                                         250114
0014.28   CHGVAR  VAR(&RES) VALUE(&NUM3)                           250114
0014.31   ENDDO
0014.32   ENDDO
0027.00 /* SENDING MESSAGE */
0027.01   CHGVAR  VAR(&MSG) VALUE('Greatest number +           250114
0027.02           :'||%char(&RES))                               250114
0028.00   SNDMSG  MSG(&MSG) TOUSR(WING15)                         250113
0029.00   ENDPGM
***** End of data *****

```

#### 4. Retrieve data from data area with and without positions

To create data area → CRTDTAARA+F4

To view data area → DSPDTAARA+F4

```

Columns . . . : 1 80
SEU==> Browse
***** Beginning of data *****
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
0001.00 PGM
0001.01   DCL      VAR(&FULLDATA) TYPE(*CHAR) LEN(20)                  250114
0001.02   DCL      VAR(&SINGLEDATA) TYPE(*CHAR) LEN(1)                   250114
0001.03 /* RETRIEVE DATA AREA WITH ATTRIBUTES --> DTAARA(DATA AREA) AND */
0001.04 /* RTNVAR --> CL VARIABLE FOR RETURNED VALUE */                  250117
0001.05
0001.06   RTVDTAARA DTAARA(FIRSTDA *ALL) RTNVAR(&FULLDATA)            250114
0001.07   RTVDTAARA DTAARA(FIRSTDA (19 1)) RTNVAR(&SINGLEDATA)          250114
0001.08
0001.09   SNDMSG  MSG(&FULLDATA) TOUSR(WING15)                         250117
0001.10   SNDMSG  MSG(&SINGLEDATA) TOUSR(WING15)                         250117
0002.00 ENDPGM
***** End of data *****

```

#### 5. Program to retrieve system values

```

Columns . . . : 1 80
Browse
GOPZLIB/QCLSRC
SEU=> RTNSYSVAL
FMT ** . .+... 1 . .+... 2 . .+... 3 . .+... 4 . .+... 5 . .+... 6 . .+... 7 . .+... 8
***** Beginning of data *****
0001.00 PGM
0001.01     DCL      VAR(&DATE1) TYPE(*CHAR) LEN(6)          250114
0001.02     DCL      VAR(&TIME1) TYPE(*CHAR) LEN(6)          250115
0001.03     DCL      VAR(&CENT) TYPE(*CHAR) LEN(1)          250115
0001.04     DCL      VAR(&SYSNAME) TYPE(*CHAR) LEN(10)         250115
0001.05     DCL      VAR(&JOB) TYPE(*CHAR) LEN(10)          250115
0001.06     DCL      VAR(&USER) TYPE(*CHAR) LEN(10)          250115
0001.07     DCL      VAR(&COUNT) TYPE(*DEC) LEN(10 0)        250115
0001.08 /* 6 DIGITS CURRENT DATE */
0001.09     RTVSYSVAL SYSVAL(QDATE) RTNVAR(&DATE1)          250115
0001.10     RTVSYSVAL SYSVAL(OTIME) RTNVAR(&TIME1)          250115
0001.11     RTVSYSVAL SYSVAL(QCENTURY) RTNVAR(&CENT)        250115
0001.12     RTVNETA  SYSNAME(&SYSNAME)                      250115
0001.13     RTVJOBA  JOB(&JOB) USER(&USER)                  250115
0001.14     RTVMBRD FILE(EMPPF) NBRCURRCD(&COUNT)          250115
0002.00 ENDPGM
***** End of data *****

```

## 6. Program to implement the usage of string functions in CLP

```

Columns . . . : 1 80
Browse
GOPZLIB/QCLSRC
FUNCTIONCL
SEU=>
FMT ** . .+... 1 . .+... 2 . .+... 3 . .+... 4 . .+... 5 . .+... 6 . .+... 7 . .+... 8
***** Beginning of data *****
0001.00 PGM
0002.00     DCL      VAR(&RES)    TYPE(*DEC) LEN(2 0)        250115
0003.00     DCL      VAR(&STRING)  TYPE(*CHAR) LEN(20) +
0004.00           VALUE('INITIAL VALUE')                   250115
0005.00     DCL      VAR(&SUBSTR) TYPE(*CHAR) LEN(2)          250115
0006.00 /****SUBSTRING**/
0007.00     CHGVAR   VAR(&SUBSTR) VALUE(%SST(&STRING 10 5))  250115
0008.00 /****LENGTH **/
0009.00     CHGVAR   VAR(&RES)    VALUE(%LEN(&STRING))        250115
0010.00 /****SCAN **/
0011.00     CHGVAR   VAR(&RES)    VALUE(%SCAN('N' &STRING))  250115
0012.00
0013.00 ENDPGM
***** End of data *****

```

## 7. Program to implement PARM, GOTO, concatenation with pipes, \*TCAT

■ DEMO

File Edit View Tools Help

Columns . . . : 1 80 Browse GOPZLIB/QCLSRC  
SEU=> PARAMDEMO

```
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****
0000.01 /*TO GET PARAMETERS FIRST DEFINE PARAMLIST --GOTO PGM PRESS F4--LABEL */
0001.00 PARM:      PGM      PARM(&A &B)          250113
0001.02      DCL      VAR(&A) TYPE(*CHAR) LEN(10) 250113
0001.03      DCL      VAR(&B) TYPE(*CHAR) LEN(10) 250113
0001.04      DCL      VAR(&C) TYPE(*CHAR) LEN(20) 250102
0001.05      CHGVAR   VAR(&C) VALUE(&B)          250102
0001.07      SNDMSG   MSG(&C) TOUSR(WING15)       250102
0001.08 /*TO CONATENATE VALUES WITH TRIM */
0001.09      CHGVAR   VAR(&C) VALUE(%TRIM(&A)||%TRIM(&B)) 250113
0001.10      SNDMSG   MSG(&C) TOUSR(WING15)       250113
0001.11 /*TO CONATENATE VALUES WITHOUT TRIM AND USING PIPES*/
0001.12      CHGVAR   VAR(&C) VALUE(&A||&B)          250113
0001.13      SNDMSG   MSG(&C) TOUSR(WING15)       250113
0001.14 /*TO CONATENATE VALUES WITHOUT TRIM AND USING *TCAT*/
0001.15      CHGVAR   VAR(&C) VALUE(&A *TCAT &B) 250113
0001.16      SNDMSG   MSG(&C) TOUSR(WING15)       250113
0002.00 ENDPGM
***** End of data *****
```

## 8. Program to read deptname from deptpf

Columns . . . : 1 80 Browse GOPZLIB/QCLSRC  
READFILE

SEU=> \*\*\*\*\* Beginning of data \*\*\*\*\*

```
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****
0001.00 PGM
0002.00      DCLF     FILE(DEPTPF)          250114
0003.00 READ:    RCVF   /*READ A DATABASE FILE*/
0003.01          /*ERROR MSG FOR EOF()-->CPP0864*/
0004.00      MONMSG   MSGID(CPP0864) EXEC(GOTO CMDLBL(END)) 250114
0004.01      SNDMSG   MSG(&DEPTNAME) TOUSR(WING15)       250114
0004.02      GOTO     CMDLBL(READ)          250114
0005.00 END:    ENDPGM
***** End of data *****
```

## 9. Program to count number of employees in a given city passed via param

■ DEMO

File Edit View Tools Help

Columns . . . : 1 80 Browse GOPZLIB/QCLSRC  
READFILE1

SEU=> \*\*\*\*\* Beginning of data \*\*\*\*\*

```
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****
0001.00      PGM      PARM(&P_EMPCITY)          250114
0002.00      DCLF     FILE(EMPPF)          250114
0002.01      DCL      VAR(&P_EMPCITY) TYPE(*CHAR) LEN(10) 250114
0002.02      DCL      VAR(&COUNT) TYPE(*DEC) LEN(2 0) VALUE(0) 250114
0002.03
0002.04 /* (LOGICAL) DATA TYPE--TWO POSSIBLE VALUES: *ON (TRUE) OR *OFF (FALSE) */
0002.05
0002.06      DCL      VAR(&LOOP) TYPE(*LGL) LEN(1) VALUE('1') 250114
0002.07      DOWHILE COND(&LOOP = '1')          250114
0003.00 READ:    RCVF   /*READ A DATABASE FILE*/
0003.01          /*ERROR MSG FOR EOF()-->CPP0864*/
0004.00      MONMSG   MSGID(CPP0864) EXEC(LEAVE)          250114
0004.01      IF       COND(&EMPCITY *EQ &P_EMPCITY) THEN(DO)
0004.02      CHGVAR   VAR(&COUNT) VALUE(&COUNT + 1)          250114
0004.03      ITERATE
0004.04      ENDDO
0004.06      ENDDO
0004.07      SNDMSG   MSG(%CHAR(&COUNT)) TOUSR(WING15) 250114
0005.00 END:    ENDPGM
***** End of data *****
```

## 10. Program to count num of employees in each city

DEMO

File Edit View Tools Help

Clear Erase Attn Sysreq Help

Columns . . . : 1 80 Browse GOPZLIB/QCLSRC  
SEU==> READFILE2

FMT \*\* . . .+.... 1 . . .+.... 2 . . .+.... 3 . . .+.... 4 . . .+.... 5 . . .+.... 6 . . .+.... 7 . . .+.... 8

\*\*\*\*\* Beginning of data \*\*\*\*\*

0001.00 PGM 250114  
0002.00 DCLF FILE(EMPLF2) 250114  
0002.01 DCL VAR(&P\_EMPCITY) TYPE(\*CHAR) LEN(10) 250114  
0002.02 DCL VAR(&C\_EMPCITY) TYPE(\*CHAR) LEN(10) 250114  
0002.03 DCL VAR(&COUNT) TYPE(\*DEC) LEN(2 0) VALUE(1) 250114  
0002.04 250114  
0002.05 250114  
0002.06 RCVF 250114  
0002.07 CHGVAR VAR(&P\_EMPCITY) VALUE(&EMPCITY) 250114  
0002.08 DOWHILE COND('1' = '1') 250114  
0003.01 250114  
0003.02 RCVF 250114  
0004.00 MONMSG MSGID(CPF0864) EXEC(LEAVE) 250114  
0004.01 250114  
0004.04 CHGVAR VAR(&C\_EMPCITY) VALUE(&EMPCITY) 250114  
0004.05 IF COND(&C\_EMPCITY \*EQ &P\_EMPCITY) THEN(DO) 250114  
0004.06 CHGVAR VAR(&COUNT) VALUE(&COUNT + 1) 250114  
0004.08 ENDDO 250114  
0004.09 250114

\*\*\*\*\*

FMT \*\* . . .+.... 1 . . .+.... 2 . . .+.... 3 . . .+.... 4 . . .+.... 5 . . .+.... 6 . . .+.... 7 . . .+.... 8

0004.10 ELSE CMD(DO) 250114  
0004.11 SNDMSG MSG(%TRIM(&P\_EMPCITY) || %CHAR(&COUNT)) TOUSR(WING15) 250114  
0004.12 CHGVAR VAR(&P\_EMPCITY) VALUE(&C\_EMPCITY) 250114  
0004.13 CHGVAR VAR(&COUNT) VALUE(1) 250114  
0004.14 ENDDO 250114  
0004.15 250114  
0004.16 ITERATE 250114  
0004.17 ENDDO 250114  
0004.18 SNDMSG MSG(%TRIM(&P\_EMPCITY) || %CHAR(&COUNT)) TOUSR(WING15) 250114  
0005.00 END: ENDPGM 250114

\*\*\*\*\* End of data \*\*\*\*\*

11. create citypf with field city , create cl program to read city pf and it should generate report for each city by submitting rpgle report program in batch using sbmjob command.  
Job name should be seperate for each city

```

Columns . . . :   6  80          Browse                               GOPZLIB/QRPGLESRC
SEU==> _____ REPORTPGM
FMT F  FFilename++IPEASFRlen+LKlen+ADevice+Keywords+++++*****+
***** Beginning of data *****
0001.00 FEMPPRTF  O   F  100      PRINTER OFLIND(*IN99)           250116
0002.00 FEMPPF    IF  E          K DISK                         250116
0002.01 DCITY     S            10                           250116
0003.00 *
0003.01 C   *ENTRY      PLIST                         250116
0003.02 C   PARM          CITY                         250116
0003.03 C   EXCEPT HEADER                         250110
0004.00 C   EXCEPT HEADING                        250110
0004.02 C   *LOVAL      SETLL  EMPPF                         250116
0004.03 C   READ          EMPPF                         250116
0005.00 C   DOW          NOT %EOF()                      250110
0006.01 C   if           %eof()                        250110
0006.02 C   leave          250110
0006.03 C   endif          250110
0007.00 C   IF           *IN99 = '1'                  250110
0008.00 C   EXCEPT HEADING                        250110
0008.01 C   ELSE          250116
0008.02 C   EVAL          *IN99 = '0'                  250116
0009.00 C   ENDIF          250110

```

```

Columns . . . :   6  80          Browse                               GOPZLIB/QRPGLESRC
SEU==> _____ REPORTPGM
FMT *  *. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
0010.00 *
0011.00 C   EXCEPT DETAIL                         250110
0011.01 C   READ   EMPPF                         250116
0012.00 C   ENDDO          250110
0013.00 C   EVAL          *INLR = '1'                  250110
0014.00 ****
0015.00 OEMPPRTF  E   HEADER                         250110
0016.00 O   40 'EMPLOYEE REPORT'                 250110
0017.00 O   75 'PAGE :'
0018.00 O   page          +2s                          250110
0018.01 O   E   HEADING                         250110
0018.02 O   17 'EMPNAME'                         250110
0018.03 O   +15 'EMPCITY'                         250110
0018.05 O   E   DETAIL   2                         250110
0018.06 O   EMPNAME          30                         250110
0018.07 O   EMPCITY          +2                         250110
0019.00 O   E   END                           250110
***** End of data *****

```

12. create a cl program to accept source file and destination file as parameters ,program should copy data from source file to destination file , if destination file doesn't exist cpy command should be executed by creating file.

```

Columns . . . :   1  80          Browse                               GOPZLIB/QCLSRC
SEU==> _____ CPYFERR
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****
0001.00   PGM      PARM(&SRCFILE &DSTFILE)           250119
0002.00   DCL      VAR(&SRCFILE) TYPE(*CHAR) LEN(10)        250119
0003.00   DCL      VAR(&DSTFILE) TYPE(*CHAR) LEN(10)        250119
0004.00
0005.00   MONMSG  MSGID(CPF9999) EXEC(GOTO CMDLBL(END))  250119
0009.03
0009.04   CPYF      FROMFILE(&SRCFILE) TOFILE(GOPZLIB/&DSTFILE) +
               MBROPT(*ADD) CRTFILE(*YES)                   250119
0009.05
0009.06   SNDPGMMMSG MSG('File copied successfully')      250119
0009.07   END:    SNDPGMMMSG MSG('Unexpected error occurred while +
               referring file')                            250119
0009.08
0010.00   ENDPGM
***** End of data *****

```

13. create a cl program to accept source file and destination file as parameters ,program should copy data from source file to destination file , if destination file doesn't exist cpy command should be executed by creating file – flatfiles



16. Create a file to implement command level error mgmt

## 17. Handle error in cpyf command using default way

(Here, program level error)

```

Columns . . . : 1 80                                Browse                               GOPZLIB/QCLSRC
SEU==>                                         CPYFERR
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****
0001.00      PGM      PARM(&SRCFILE &DSTFILE)                      250119
0002.00      DCL      VAR(&SRCFILE) TYPE(*CHAR) LEN(10)                250119
0003.00      DCL      VAR(&DSTFILE) TYPE(*CHAR) LEN(10)                250119
0004.00
0005.00      MONMSG   MSGID(CPF9999) EXEC(GOTO CMDLBL(END))        250119
0009.03
0009.04      CPYF     FROMFILE(&SRCFILE) TOFILE(GOPZLIB/&DSTFILE) +
0009.05          MBROPT(*ADD) CRTFILE(*YES)                         250119
0009.06      SNDPGMMMSG MSG('File copied successfully')           250119
0009.07  END:    SNDPGMMMSG MSG('Unexpected error occurred while +
0009.08          referring file')                           250119
0010.00      ENDPGM
***** End of data *****

```

## 18. Handle error in cpyf command for flatfiles

(Here, command level error)

```

Columns . . . : 1 80                                Browse                               GOP2LIB/QCLSRC
SEU==> [ ]                                         CPYFLATERR

FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****
0001.00      PGM      PARM(&SRCFILE &DSTFILE)                      250119
0002.00      DCL      VAR(&SRCFILE) TYPE(*CHAR) LEN(10)                250119
0003.00      DCL      VAR(&DSTFILE) TYPE(*CHAR) LEN(10)                250119
0004.00
0004.01      CPYF     FROMFILE(&SRCFILE) TOFILE(&DSTFILE) +
                  MBROPT('ADD') FMTOPT('NOCHK')                     250119
0004.02
0005.00      MONMSG   MSGID(CPF9999) EXEC(DO)                         250119
0009.03
0009.04
0009.05      CRTPF   FILE(&DSTFILE) RCDLEN(500)                      250119
0009.06      CPYF     FROMFILE(&SRCFILE) TOFILE(&DSTFILE) +
                  MBROPT('ADD') FMTOPT('NOCHK')                     250119
0009.07
0009.08      SNDMSG   MSG('File copied successfully') TOUSR(WING15)    250119
0009.09      ENDDO
0009.10
0010.00      ENDPGM
***** End of data *****

```

19. Employee names where city name starts with 'T'

```
Columns . . . : 1 80                                         Browse                                     GOPZLIB/QCLSRC
SEU==> [***** Beginning of data *****]
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
0000.04      DCL      VAR(&COND) TYPE(*CHAR) LEN(50)           250117
0000.06
0000.08      CHGVAR   VAR(&COND) VALUE('EMPCITY *EQ %WLDCRD("T*")')
0000.10
0000.13
0001.00      OVRDBF   FILE(EMPPF) SHARE(*YES)                 250116
0002.05      OPNQRYF  FILE((EMPPF)) OPTION(*ALL) QRYSLT(&COND)  250117
0002.06
0004.00      CALL     PGM(EMPREAD)                           250116
0005.00      CLOF    OPNID(EMPPF)                           250116
0006.00      DLTOVR   FILE(*ALL)                            250116
***** End of data *****
```

20. Employee names where city contains 'T'

```
Columns . . . : 1 80                                         Browse                                     GOPZLIB/QCLSRC
SEU==> [***** Beginning of data *****]
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
0000.04      DCL      VAR(&COND) TYPE(*CHAR) LEN(50)           250117
0000.06
0000.08      CHGVAR   VAR(&COND) VALUE('EMPCITY *CT ("T")')
0000.10
0000.13
0001.00      OVRDBF   FILE(EMPPF) SHARE(*YES)                 250116
0002.05      OPNQRYF  FILE((EMPPF)) OPTION(*ALL) QRYSLT(&COND)  250117
0002.06
0004.00      CALL     PGM(EMPREAD)                           250116
0005.00      CLOF    OPNID(EMPPF)                           250116
0006.00      DLTOVR   FILE(*ALL)                            250116
***** End of data *****
```

21. Generate report for employees whose empno is greater than 101

```
Columns . . . : 1 80                                         Browse                                     GOPZLIB/QCLSRC
SEU==> [***** Beginning of data *****]
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
0001.00      PGM
0002.00      DCL      VAR(&COND) TYPE(*CHAR) LEN(50)           250116
0003.00      CHGVAR   VAR(&COND) VALUE('EMPNO *GT 101')
0004.00      OVRDBF   FILE(EMPPF) SHARE(*YES)                 250116
0005.00      OPNQRYF  FILE((EMPPF)) OPTION(*ALL) QRYSLT(&COND) + 250116
0006.00          KEYFLD(*FILE)
0007.00      CALL     PGM(EMPCL1RPG)                          250116
0008.00      CLOF    OPNID(EMPPF)                           250116
0009.00      DLTOVR   FILE(*ALL)                            250116
0010.00      ENDPGM
***** End of data *****
```

```
Columns . . . : 6 80                                         Browse                                     GOPZLIB/QRPGLESRC
SEU==> [***** Beginning of data *****]
FMT FX FFilename++IPEASF....L....A.Device+.Keywords+++++-----+
0002.00 FEMPPF    IF E      K DISK                         250116
0003.00 FEMPCL1SCN CF E      WORKSTN SPILE(EMPSFL:RRN)       250116
0004.00 DRRN      S      4 0                           250111
0007.00 *
0009.00 C          DOW    *IN03 = '0'                      250111
0009.01 C          EXFMT  ENTRYRCD                     250116
0009.02 C          IF     *IN03= '1'                      250116
0009.03 C          LEAVE
0009.04 C          ENDIF
0009.05 C
0010.00 +
0011.00 C          EXSR   CLRSFL                      250111
0011.01 C          EXSR   LOADSFL                      250111
0011.02 C          EXSR   DSPLYSFL                     250111
0012.00 C          EXSR   PROCSFL                      250111
0013.00 +
0014.00 C          ENDDO
0015.00 C          EVAL   *INLR = '1'                      250111
0016.00 *
```

```

Columns . . . :   6  80                                Browse                               GOPZLIB/QRPGLESRC
SEU==> ■
FMT C CL0N01Factor1++++++Opcode&ExtFactor2++++++Result++++++Len++D+HiLoEq....
0017.00 C    CLRSFL     BEGSR                                         250111
0017.01 C          EVAL      *IN93 = '1'                           250111
0017.02 C          WRITE     EMPCTL                         250116
0017.03 C          EVAL      *IN93 = '0'                           250111
0017.04 C          ENDSR                                         250111
0018.00 *****                                                 250111
0019.00 C    LOADSFL     BEGSR                                         250111
0019.01 C          EVAL      RRN = 0                           250111
0019.02 C    *LOVAL     SETLL     EMPPF                         250116
0019.03 C          READ     EMPPF                          250116
0019.05 C          DOW      NOT %EOF()                      250111
0019.06 C          IF       EMCITY = SEMPCITY                250117
0019.07 C          EVAL      RRN = RRN + 1                  250111
0019.08 C          WRITE     EMPSFL                         250116
0019.12 C          ENDIF                                         250117
0019.13 C          READ     EMPPF                          250117
0019.14 C          ENDDO                                         250117
0020.00 C          ENDSR                                         250111
0021.00 *****                                                 250111
0022.00 C    DSPLYSFL    BEGSR                                         250111

```

```

Columns . . . :   6  80                                Browse                               GOPZLIB/QRPGLESRC
SEU==>
FMT C CL0N01Factor1++++++Opcode&ExtFactor2++++++Result++++++Len++D+HiLoEq....
0022.01 C          WRITE     FOOTER                         250111
0022.02 C          EVAL      *IN92 = '1'                           250111
0022.03 C          IF       RRN > 0                           250111
0022.04 C          EVAL      *IN91 = '1'                           250111
0022.05 C          EVAL      *IN94 = '1'                           250111
0022.06 C          ENDIF                                         250111
0022.07 C          EXPMT     EMPCTL                         250116
0022.13 C          EVAL      *IN92 = '0'                           250111
0022.14 C          EVAL      *IN91 = '0'                           250111
0022.15 C          EVAL      *IN94 = '0'                           250111
0023.00 C          ENDSR                                         250111
0024.00 *****                                                 250111
0025.00 C    PROCSPFL    BEGSR                                         250111
0025.04 C          IF       RRN = 0                           250111
0025.05 C          LEAVESR                         250111
0025.06 C          ENDIF                                         250111
0025.07 C          ENDSR                                         250111
0026.00 C          ENDSR                                         250111
***** End of data *****

```

## 22. System date conversion

```

Columns . . . :   1  80                                Browse                               GOPZLIB/QCLSRC
SEU==> ■
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****
0001.00 PGM                                         250119
0002.00 /* DECLARE VARIABLES */
0003.00 DCL VAR(&SYSDATE) TYPE(*CHAR) LEN(6) /* SYSTEM DATE IN *MDY FORMAT */ 250119
0004.00 DCL VAR(&ISO_DATE) TYPE(*CHAR) LEN(10) /* ISO DATE FORMAT YYYY-MM-DD */ 250119
0005.00
0006.00 /* RETRIEVE THE SYSTEM DATE */
0007.00 RTVSYSTVAL SYSVAL(QDATE) RTNVAR(&SYSDATE) 250119
0008.00
0009.00 /* CONVERT DATE TO ISO FORMAT YYYY-MM-DD */
0010.00      CVDAT     DATE(&SYSDATE) TOVAR(&ISO_DATE) + 250119
0010.01          FROMFMT(*MDY) TOFMT(*ISO) TOSEP('-') 250119
0011.00
0012.00 /* SEND CONVERTED DATE AS A PROGRAM MESSAGE */
0013.00 SNDPGMMMSG MSG('SYSTEM DATE IN ISO FORMAT: ' *CAT &ISO_DATE) 250119
0014.00
0015.00 ENDPGM                                         250119
***** End of data *****

```

23. Screen1 (record format) will have only 1 field which is the Order date. When we enter an order date in the Screen 1 and hits Enter, screen 2 which is a subfile will display all the order details for the purchase orders which has the order date as the value entered in screen 1 and are shipped (ORDSHP = Y). Fields that should be displayed in screen 2 should be Order number, item number, item quantity and User ordered. Screen 1 should have the function key F3. Screen 2 should have the function keys F12 and F3. On hitting F12, should go back to Screen1. On hitting F3, should end the program.

Note: Add proper validations for the Date field in screen 1 and display appropriate error messages  
The 1<sup>st</sup> screen should be displayed using CLP

```

Columns . . . : 1 80                                         Browse                                     GOPZLIB/QCLSRC
SEU==> |                                                 EXM2ORDCLI
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****

0001.00      PGM                                         250116
0002.00      DCLF     FILE(EXAM2) RCDfmt(DSPLY)          250117
0003.00      DCL      VAR(&COND) TYPE(*CHAR) LEN(50)       250116
0004.00      DCL      VAR(&EXIT) TYPE(*CHAR) LEN(1)        250117
0005.00      /*****/
0006.00      DOWHILE (&IN03 = '0' *AND &EXIT = ' ')
0007.00      HANDLER: SNDRCVF RCDfmt(DSPLY)             250119
0008.00      IF      COND(&IN03 = '1') THEN(LEAVE)         250117
0008.01      CVTDATE DATE(&SORDDATE) TOVAR(&SORDDATE) +
0008.02          FROMFMT('YYMD') TOFMT('YYMD') TOSEP('-')    250119
0008.03      MONMSG  MSGID(CPF0000 CPF9999) EXEC(DO)       250119
0008.04      CHGVAR   VAR(&SMSSG) VALUE('date format invalid') 250119
0008.05      GOTO    CMDLBL(HANDLER)                   250119
0008.06      ENDDO
0008.07
0009.00      CHGVAR   VAR(&COND) VALUE('ORDSHP = "Y" *AND      250117
0010.00          ORDATE = "' || &SORDDATE|| '')           250117
0011.00      OVRDBF   FILE(ORDMSTPF) SHARE(*YES)          250117
0012.00      OPNQRYF  FILE((ORDMSTPF)) OPTION(*ALL) QRYSLT(&COND) + 250117

FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
KEYFLD(*FILE)                                         250116
0013.00
0014.00      CALL    PGM(EXAM2B) PARM((&EXIT))          250117
0015.00      CLOF    OPNID(ORDMSTPF)                  250117
0016.00      DLTOVR  FILE(*ALL)                      250116
0017.00      ENDDO
0018.00      ENDPGM
***** End of data *****

```

## 24. Overriding flatfiles

```

Columns . . . : 1 80                                         Browse                                     GOPZLIB/QCLSRC
SEU==> |                                                 MROVRPG
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****

0000.01 /* OVRDBF (override database file) */
0000.02 /* change the attributes of a database file for the duration of a job or */
0000.03 /* a specific CL program*/
0001.00      OVRDBF   FILE(FLATFILE1) TOFILE(FLATFILE1) MBR(MBR1) 250116
0002.00      CALL     PGM(FLATMBRPG2)                  250116
0002.01 /* DLTOVR (Delete override) command with the *ALL parameter */
0002.02 /* used to delete all active overrides for the current job in AS400 */
0003.00      DLTOVR   FILE(*ALL)                      250116
***** End of data *****

```

## 25. Overriding pf files

```

Columns . . . : 1 80                                         Browse                                     GOPZLIB/QCLSRC
SEU==> |                                                 OVERWRITE
FMT ** ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
***** Beginning of data *****

0001.00 /* Overwriting one file to other file*/
0002.00      OVRDBF   FILE(EMPPF) TOFILE(EMPPF1)          250115
0003.00      CALL     PGM(EMPREAD)                     250115
0004.00      DLTOVR   FILE(*ALL)                      250115
***** End of data *****

```

## 26. Override attr using OPNQRYF call empread

Columns . . . :	1 80	Browse	GOPZLIB/QCLSRC OVRATTR
SEU=>			
FMT **	...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8		
	***** Beginning of data *****		
0000.01 /* 2WAYS_QRYSLT-->1)DIRECTLY GIVE IN QRYSLT OR GIVE VAR TO STORE COND */		250116	
0000.02 /* 2ND WAY IS EFFICIENT */		250116	
0000.03 /* ALWAYS DCL SHOULD BE BEFORE OTHER OPERATIONS */		250116	
0000.04 DCL VAR(&QRYCOND) TYPE(*CHAR) LEN(50)		250116	
0000.05 DCL VAR(&QRYCOND1) TYPE(*CHAR) LEN(80)		250116	
0000.06		250116	
0000.07 /* ACTUALLY CHAR SHOULD BE GIVEN IN" BUT HERE WORKED WITH ' IE KOCHI */		250116	
0000.08 CHGVAR VAR(&QRYCOND) VALUE('EMPCITY *EQ ''KOCHI'' +		250116	
0000.09 *AND EMPNO *GT 101')		250116	
0000.10		250116	
0000.11 CHGVAR VAR(&QRYCOND1) VALUE('EMPCITY *EQ ''KOCHI'' +		250116	
0000.12 *OR EMCITY *EQ ''ALUVA'' *AND EMPNO *GT 102')		250116	
0000.13		250116	
0000.14 /* TO FILE IS NOT GIVEN BCOZ HERE ATTRIBUTES ARE GETTING OVERRIDEN, NOT FILE*/		250116	
0001.00 OVRDBF FILE(EMPPF) SHARE(*YES)		250116	
0001.01		250116	
0002.00 /* OPNQRYF FILE((EMPPF)) OPTION(*ALL) QRYSLT('EMPCITY + */		250116	
0002.01 /* *EQ ''KOCHI'' *AND EMPNO *GT 101') */		250116	
0002.02		250116	
0002.03 /* OPNQRYF FILE((EMPPF)) OPTION(*ALL) QRYSLT(&QRYCOND) */		250116	
0002.04		250116	
0002.05 OPNQRYF FILE((EMPPF)) OPTION(*ALL) QRYSLT(&QRYCOND1)		250116	
0002.06		250116	
0004.00 CALL PGM(EMPREAD)		250116	
0005.00 CLOF OPNID(EMPPF)		250116	
0006.00 DLTOVR FILE(*ALL)		250116	
	***** End of data *****		