Session - 2

Exception Handling

Exceptional Condition

- An abnormal situation during execution of a command in CICS

POSSIBLE ERRORS

- Conditions that aren't normal from CICS's point of view but that are expected in the pgm.
- Conditions caused by user errors and input data errors.
- Conditions caused by omissions or errors in the application code.
- Errors caused by mismatches bet. applications and CICS tables, generation parameters & JCL
- Errors related to hardware or other system conditions beyond the control of an appl. pgm.

Error Handling Methods

When the error (exceptional conditions) occur, the program can do any of the following

- Take no action & let the program continue - Control returns to the next inst. following the command that has failed to execute. A return code is set in EIBRESP and EIBRCODE. This state occurs ‘cause of *NO* HANDLE /RESP/IGNORE conditions
- Pass control to a specified label - Control goes to a label in the program defined earlier by a HANDLE CONDITION command.
- Rely on the system default action - System will terminate or suspend the task depends on the exceptional condition occurred

Exception Handling Methods

- HANDLE CONDITION
- NOHANDLE
- IGNORE
- RESP
- PUSH & POP

HANDLE CONDITION

- Syntax :

```
EXEC CICS  HANDLE CONDITION
                ERROR(err-handle-para)
                LENGERR(len-err-handle-para)
                DUPREC(dup-rec-para)
END-EXEC
```

Sample Program – Handle condition

```
000100     IDENTIFICATION DIVISION.
000200     PROGRAM-ID. HANDL.
000300     ENVIRONMENT DIVISION.
000400     DATA DIVISION.
000500     WORKING-STORAGE SECTION.
000600     01 REC.
000700        02 TID   PIC X(4).
000800        02 A     PIC 99.
000900     77 B     PIC 99.
001000     77 C     PIC 99.
001010     77 MSG   PIC X(15).
001100     PROCEDURE DIVISION.
001200     PARA1.
001300          EXEC CICS HANDLE CONDITION
001400              LENGERR(EPARA)
001500          END-EXEC.
001600          EXEC CICS RECEIVE
001700              INTO(REC)
001800          END-EXEC.
```

```
001900        EXEC CICS RECEIVE
002000            INTO(B)
002100        END-EXEC.
002200        COMPUTE C = A + B.
002300        EXEC CICS SEND
002400            FROM(C)
002410            ERASE
002500        END-EXEC.
002501        EXEC CICS RETURN END-EXEC.
002510      EPARA.
002520        MOVE "LENGTH ERROR " TO MSG.
002530        EXEC CICS SEND
002540            FROM(MSG)
002550        END-EXEC.
002600        EXEC CICS
002700             RETURN
002800        END-EXEC.
002900
```

NOHANDLE

- Syntax

```
EXEC CICS …
   ……
   …….
   NOHANDLE
END-EXEC
```

Sample program for NOHANDLE

```
•    000100      IDENTIFICATION DIVISION.
•    000200      PROGRAM-ID. FHG.
•    000300      ENVIRONMENT DIVISION.
•    000400      DATA DIVISION.
•    000500      WORKING-STORAGE SECTION.
•    000600      01 REC.
•      000800        02 A     PIC 99.
•    000900      77 B     PIC 99.
•    001000      77 C     PIC 99.
•    001100      77 MSG   PIC X(15).
•    001110      77 LEN   PIC S9(4) COMP.
•    001200      PROCEDURE DIVISION.
•    001300      PARA1.
•    001400          EXEC CICS HANDLE CONDITION
•    001500              LENGERR(EPARA)
•    001600          END-EXEC.
•    001700          EXEC CICS RECEIVE
•    001800              INTO(REC)
•    001900              LENGTH(LENGTH OF REC)
•    001910              NOHANDLE
•    002000          END-EXEC.
•    002010          MOVE 2 TO LEN.
•
```

```
002100      EXEC CICS RECEIVE
002200        INTO(B)
002300        LENGTH(LEN)
002400      END-EXEC.
002500      COMPUTE C = A + B.
002600      EXEC CICS SEND
002700        FROM(C)
002800        LENGTH(LENGTH OF C)
002900        ERASE
003000      END-EXEC.
003100      EXEC CICS
003200         RETURN
003300      END-EXEC.
003310    EPARA.
003320      MOVE "LENGTH ERROR " TO MSG.
003330      EXEC CICS SEND
003340        FROM(MSG)
003350      END-EXEC.
003360      EXEC CICS
003370         RETURN
003380      END-EXEC.
003400      STOP RUN.
```

IGNORE CONDITION

- Syntax :

```
EXEC CICS    IGNORE CONDITION
              ITEMERR
               LENGERR
END-EXEC
```

Sample Program – IGNORE condition

```
000100        IDENTIFICATION DIVISION.
 000200        PROGRAM-ID. FHG.
 000300        ENVIRONMENT DIVISION.
 000400        DATA DIVISION.
 000500        WORKING-STORAGE SECTION.
 000600        01 REC.
 000700           02 TID   PIC X(4).
 000800           02 A     PIC 99.
 000900        77 B     PIC 99.
 001000        77 C     PIC 99.
 001100        77 MSG   PIC X(15).
 001200        PROCEDURE DIVISION.
 001300        PARA1.
 001400             EXEC CICS IGNORE CONDITION
 001500                 LENGERR
 001600             END-EXEC.
```

```
•  001700          EXEC CICS RECEIVE
•  001800              INTO(REC)
•  001810              LENGTH(LENGTH OF REC)
•  001900          END-EXEC.
• 002000          EXEC CICS RECEIVE
•  002100              INTO(B)
•  002110              LENGTH(LENGTH OF B)
•  002200          END-EXEC.
•  002300          COMPUTE C = A + B.
•  002400          EXEC CICS SEND
•  002500              FROM(C)
•  002510              LENGTH(LENGTH OF C)
•  002600              ERASE
•  002700          END-EXEC.
•  003400          EXEC CICS
•  003500               RETURN
•  003600          END-EXEC.
•  003700          STOP RUN.
```

RESP

- Syntax

```
EXEC CICS …..
    ……..
    ……..
    RESP(resp-variable)
END-EXEC
```

Sample Program - RESP

```
•   000001     IDENTIFICATION DIVISION.
•   000002     PROGRAM-ID. FFF.
•   000003     ENVIRONMENT DIVISION.
•   000004     DATA DIVISION.
•   000005     WORKING-STORAGE SECTION.
•   000006     01 INP.
•   000007        02 TID     PIC X(4).
•   000008        02 F       PIC X.
•   000009        02 A       PIC 9(4).
•   000010     01 OUP.
•   000011        02 MSG     PIC X(20).
•   000012        02 A1      PIC 9(4).
•   000013     77 WS-RESP    PIC S9(8) COMP.
•   000014     PROCEDURE DIVISION.
•   000015     PARA1.
•   000016          EXEC CICS RECEIVE
•   000017               INTO(INP)
•   000018               LENGTH(LENGTH OF INP)
•   000019               RESP(WS-RESP)
•   000020          END-EXEC.
```

```
000021        IF WS-RESP = DFHRESP(LENGERR)
000022            MOVE " LENGTH ERROR " TO MSG
000023            MOVE 0 TO A1
000024        ELSE
000025            MOVE " NUMBER IS " TO MSG
000026            MOVE A TO A1
000027        END-IF.
000028        EXEC CICS SEND
000029            FROM(OUP)
000030            LENGTH(LENGTH OF OUP)
000031            ERASE
000032        END-EXEC.
000033        EXEC CICS
000034            RETURN
000035        END-EXEC.
000036        STOP RUN.
```

PUSH & POP

- Suspend and reactivate HANDLE condition in effect

- EXEC CICS PUSH HANDLE END-EXEC
- EXEC CICS POP HANDLE END-EXEC

Sample Program – PUSH & POP

```
•   000001      IDENTIFICATION DIVISION.
•   000002      PROGRAM-ID. PPP.
•   000003      ENVIRONMENT DIVISION.
•   000004      DATA DIVISION.
•   000005      WORKING-STORAGE SECTION.
•   000006      01 INP.
•   000007         02 TID    PIC X(4).
•   000008         02 F      PIC X.
•   000009         02 A      PIC 9(4).
•   000010      77 MSG       PIC X(20).
•   000011      PROCEDURE DIVISION.
•   000012      PARA1.
•   000013          EXEC CICS HANDLE CONDITION
•   000014              LENGERR(EPARA)
•   000015          END-EXEC.
•   000016          EXEC CICS
•   000017               PUSH HANDLE
•   000018          END-EXEC.
```

```
000019       EXEC CICS RECEIVE
000020            INTO(INP)
000021            LENGTH(LENGTH OF INP)
000022        END-EXEC.
000023        EXEC CICS
000024            POP HANDLE
000025        END-EXEC.
000026        EXEC CICS SEND
000027            FROM(A)
000028            LENGTH(LENGTH OF A)
000029        END-EXEC.
000030        EXEC CICS
000031            RETURN
000032        END-EXEC.
000033    EPARA.
000034        MOVE "LENGTH ERROR " TO MSG.
000035        EXEC CICS SEND
000036            FROM(MSG)
000037            LENGTH(LENGTH OF MSG)
000038        END-EXEC.
000039        EXEC CICS
000040            RETURN
000041        END-EXEC.
000042        STOP RUN.
```

Handle AID

- Syntax :

  EXEC CICS **HANDLE AID**

  Option (label)

  END-EXEC

Sample Program – HANDLE AID

```
000001    IDENTIFICATION DIVISION.
000002    PROGRAM-ID. SND.
000003    ENVIRONMENT DIVISION.
000004    DATA DIVISION.
000005    WORKING-STORAGE SECTION.
000006    01 REC.
000007       02 TID  PIC X(4).
000008    77 A       PIC X(25).
000009    PROCEDURE DIVISION.
000010    MPARA.
000011        MOVE LOW-VALUES TO A.
000012        MOVE "PRESS F1 F2 OR F3" TO A.
000013        EXEC CICS SEND
000014            FROM(A)
000015            LENGTH(LENGTH OF A)
000016        END-EXEC.
000017        EXEC CICS HANDLE AID
000018            PF1(PARA1)
000019            PF2(PARA2)  PF3(CPARA)
000020        END-EXEC.
```

```
000021     EXEC CICS RECEIVE
000022         INTO(REC)
000023         LENGTH(LENGTH OF REC)
000024     END-EXEC.
000025 PARA1.
000026     MOVE "WELCOME TO CICS" TO A.
000027     EXEC CICS SEND
000028         FROM(A)
000029         LENGTH(LENGTH OF A)
000030         ERASE
000031     END-EXEC.
000032     GO TO CPARA.
000033 PARA2.
000034     MOVE "THANK U" TO A.
000035     EXEC CICS SEND
000036         FROM(A)
000037         LENGTH(LENGTH OF A)
000038         ERASE
000039     END-EXEC.
000040 CPARA.
000041     EXEC CICS
000042         RETURN
000043     END-EXEC.
000044     STOP RUN.
```

z/OS MVS CICS BASIC MAPPING SUPPORT

- **EIB (EXECUTE INTERFACE BLOCK)**
- The Execute Interface Block (EIB) is a CICS area that contains information related to the current task such as the date and time the task was started and the transaction-id that was used to start it.
- The definition of this area is inserted into the Linkage Section of the program when the program is prepared for execution.

- The EIBCALEN field contains the length of the data passed to the program through its communication area (DFHCOMMAREA)

- When the user presses an Attention Identifier Key (AID) CICS passes a one-byte value to the program through the EIBAID field in the EIB.

z/OS MVS CICS BASIC MAPPING SUPPORT

- **COMMON EIB FIELDS**
- EIBDATE Contains the system date when the transaction started
- EIBHTIME Contains the system time when the transaction started
- EIBTRNID Contains the Trans-id 4 characters
- EIBTRMID Contains the Terminal-id where the user has signed on 4 Char
- EIBCALEN The Length of the Communication area (DFHCOMMAREA)
- EIBTASKNO The Task number assigned to a particular task on the terminal
-

EXEC Interface Block (EIB)

- CICS provides some system-related information to each task as EXEC Interface Block (EIB)
- unique to the CICS command level

EIBAID	Attention- Id (1 Byte)
EIBCALEN	Length of DFHCOMMAREA (S9(4) comp)
EIBDATE	Date when this task started (S9(7) comp-3)
EIBFN	Function Code of the last command (2 Bytes)
EIBRCODE	Response Code of the last command (6 Bytes)
EIBTASKN	Task number of this task (S9(7) comp-3)
EIBTIME	Time when this task started (S9(7) comp-3)
EIBTRMID	Terminal-Id (1 to 4 chars)
EIBTRNID	Transaction-Id (1 to 4 chars)

Recap

- Name some of the common exceptions.
- What are the various exception handling techniques?
- What is the use of PUSH & POP?
- What is the use of HANDLE AID?

Try Yourself !

- Write a program in CICS to implement all the exception handling techniques in the same program and realize the difference!

Possible Condition Names

- ENDFILE
- INVREQ
- QZERO
- QIDERR
- DUPKEY
- PGMIDERR
- MAPFAIL
- NOSPACE
- NOTOPEN
- NOTFND