**Cics**

Send Map

EXEC CICS SEND

MAP('map-name')

MAPSET('mapset-name')

[FROM(data-area)]

[LENGTH(data\_value)]

[DATAONLY]

[MAPONLY]

[CURSOR]

[ERASE/ERASEAUP]

[FREEKB]

[FRSET]

END-EXEC

## Receive Map

EXEC CICS RECEIVE

MAP('map-name')

MAPSET('mapset-name')

[INTO(data-area)]

[FROM(data-area)]

[LENGTH(data\_value)]

END-EXEC

# **READ**

EXEC CICS READ

FILE('name')

INTO(data-area)

RIDFLD(data-area)

LENGTH(data-value)

KEYLENGTH(data-value)

END-EXEC.

# **WRITE**

EXEC CICS WRITE

FILE(name)

FROM(data-area)

RIDFLD(data-area)

LENGTH(data-value)

KEYLENGTH(data-value)

END-EXEC.

# **REWRITE**

EXEC CICS REWRITE

FILE (name)

FROM (data-area)

LENGTH (data-value)

END-EXEC.

# **DELETE**

EXEC CICS DELETE

FILE('name')

END-EXEC.

EXEC CICS DELETE

FILE('name')

RIDFLD(data-value)

END-EXEC.

## EIB Fields

|  |  |  |
| --- | --- | --- |
| **EIB Field** | **PIC Clause** | **Description** |
| EIBAID | X(1) | Aid key Pressed |
| EIBCALEN | S9(4) COMP | It contains length of DFHCOMMAREA |
| EIBDATE | S9(7) COMP-3 | It contains Current System Date |
| EIBRCODE | X(6) | It contains Return code of the last transaction |
| EIBTASKN | S9(7) COMP-3 | It contains Task number |
| EIBTIME | S9(7) COMP-3 | It contains Current System Time |
| EIBTRMID | X(4) | Terminal Identifier |
| EIBTRNID | X(4) | Transaction Identifier |

### **COMMAREA**

EXEC CICS RETURN

TRANSID ('transaction-id')

COMMAREA (WS-COMMAREA)

LENGTH (WS-COMMAREA-LENGTH)

END-EXEC.

## Pseudo Code

MOVE DFHCOMMAREA TO WS-COMMAREA

IF EIBCALEN = 0

STEP1: SEND MAP

STEP2: MOVE <internal-transaction-id1> to WS-COMMAREA

STEP3: ISSUE CONDITIONAL RETURN

ELSE

IF WS-COMMAREA = <internal-transaction-id1>

STEP4: RECEIVE MAP

STEP5: PROCESS DATA

STEP6: SEND OUTPUT MAP

STEP7: MOVE <internal-transaction-ID2> to WS-COMMAREA

STEP8: ISSUE CONDITIONAL RETURN

END-IF

END-IF

STEP9: REPEAT STEP3 TO STEP7 UNTIL EXIT

### **ASKTIME**

EXEC CICS ASKTIME

[ABSTIME(WS-TIMESTAMP)]

END-EXEC.

### **FORMATTIME**

EXEC CICS FORMATTIME

ABSTIME(WS-TIMESTAMP)

[YYDDD(WS-DATE)]

[YYMMDD(WS-DATE)]

[YYDDMM(WS-DATE)]

[DATESEP(WS-DATE-SEP)]

[TIME(WS-TIME)]

[TIMESEP(WS-TIME-SEP)]

END-EXEC.

## WRITEQ TS

EXEC CICS WRITEQ TS

QUEUE ('queue-name')

FROM (queue-record)

[LENGTH (queue-record-length)]

[ITEM (item-number)]

[REWRITE]

[MAIN /AUXILIARY]

END-EXEC.

## READQ TS

EXEC CICS READQ TS

QUEUE ('queue-name')

INTO (queue-record)

[LENGTH (queue-record-length)]

[ITEM (item-number)]

[NEXT]

END-EXEC.

## DELETEQ TS

EXEC CICS DELETEQ TS

QUEUE ('queue-name')

END-EXEC.

## WRITEQ TD

EXEC CICS WRITEQ TD

QUEUE ('queue-name')

FROM (queue-record)

[LENGTH (queue-record-length)]

END-EXEC.

## READQ TD

EXEC CICS READQ TD

QUEUE ('queue-name')

INTO (queue-record)

[LENGTH (queue-record-length)]

END-EXEC.

## DELETEQ TD

EXEC CICS DELETEQ TD

QUEUE ('queue-name')

END-EXEC.

## ERROR’S

|  |  |
| --- | --- |
| **Sr.No** | **Code & Description** |
| 1 | **ASRA**  Program Check Exception |
| 2 | **AEI0**  Program ID Error |
| 3 | **AEI9**  Map Fail condition |
| 4 | **AEIO**  Duplicate Key |
| 5 | **AEIN**  Duplicate Record |
| 6 | **AEID**  End of file reached |
| 7 | **AEIS**  File is not open |
| 8 | **AEIP**  Invalid request condition |
| 9 | **AEY7**  Not authorized to use the resource |
| 10 | **APCT**  Program not found |
| 11 | **AFCA**  Dataset not found |
| 12 | **AKCT**  Time out error |
| 13 | **ABM0**  Specified map not found |
| 14 | **AICA**  Program in infinite loop |
| 15 | **AAOW**  Internal logic error |