

Application program needs an interface to CICS. EIB is known as Execute Interface Block which contains the useful information required during execution of program.

## Restricted COBOL Verbs

While coding a CICS program we cannot use the commands which give control directly to MVS. If we code these COBOL verbs, it will not give any compilation error but we may get unpredictable results. Following are the COBOL verbs which should not be used in a CICS program:

- File I/O statements like Open, Read, Write, Rewrite, Close, Delete and Start. All file I/O in CICS is handled by the file control module and they have their own set of statements like READ, WRITE, REWRITE and DELETE which we will be discussing in upcoming modules.
- File Section and Environment Division is not required.
- COBOL statements that invoke operating system functions like Accept, Date/Time cannot be used.
- Do not use DISPLAY, MERGE, STOP RUN and GO BACK.

## Execute Interface Block

The fundamental explanation of Execute Interface Block is as follows:

- Execute Interface Block is a control block which is loaded automatically by CICS for every program.
- The EIB is unique to a task and it exists for the duration of task. It contains a set of system related information corresponding to the task.
- EIB contains information about transaction identifier, time, date, etc which is used by CICS during execution of application program.
- Every program that executes as part of the task has access to the same EIB.
- The data in EIB at run time can be viewed by executing the program in CEDF mode.

## EIB Fields

Following table shows the lists of fields which are present in EIB:

EIB Field	PIC Clause	Description
EIBAID	X1	Aid key Pressed
EIBCALEN	S94 COMP	It contains length of DFHCOMMAREA
EIBDATE	S97 COMP-3	It contains Current System Date
EIBRCODE	X6	It contains Return code of the last transaction
EIBTASKN	S97 COMP-3	It contains Task number
EIBTIME	S97 COMP-3	It contains Current System Time
EIBTRMID	X4	Terminal Identifier
EIBTRNID	X4	Transaction Identifier

## CICS Programs Classification

CICS Programs are classified in following three categories which we will discuss one by one:

- Non Conversion Programs
- Conversion Programs
- Pseudo-conversion Programs - We will discuss in next module

### Non Conversion Programs

The fundamental explanation of Non Conversion Programs is as follows:

- While executing Non conversion programs no human intervention is required.
- All the necessary inputs are provided when the program is started.
- They are similar to batch programs that runs in the batch mode. So in CICS they are rarely developed.
- We can say they are just used for displaying a sequence of screens at regular intervals of time.

### Example

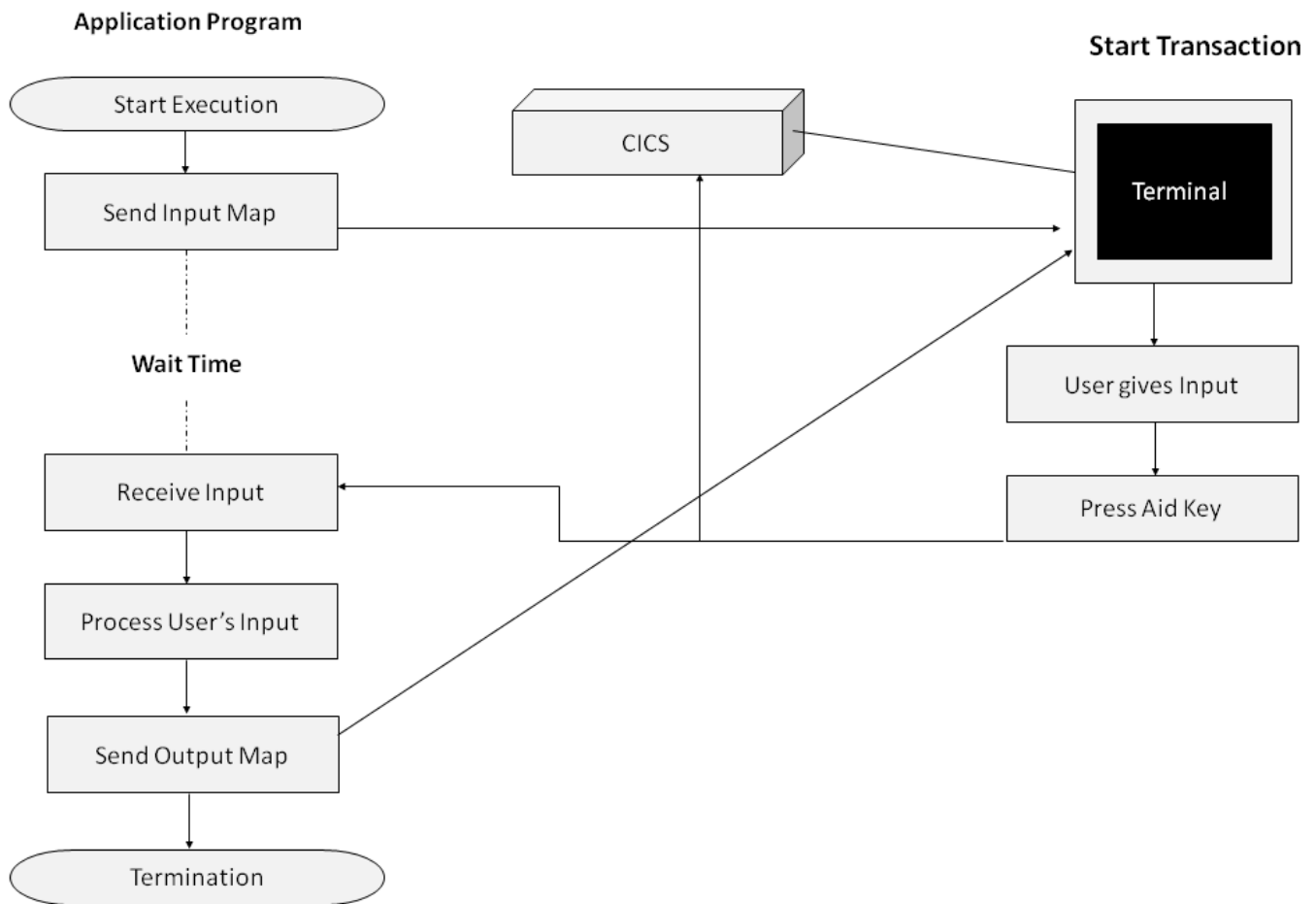
Following example shows a non conversion program which is simply displaying "HELLO WORLD" on CICS terminal as output:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 WS-MESSAGE          PIC X(30).
PROCEDURE DIVISION.
*****
* SENDING DATA TO SCREEN
*****
        MOVE 'HELLO WORLD' TO WS-MESSAGE
        EXEC CICS SEND TEXT
            FROM (WS-MESSAGE)
        END-EXEC
*****
* TASK TERMINATES WITHOUT ANY INTERACTION FROM THE USER*
*****
        EXEC CICS RETURN
        END-EXEC.
```

### Conversion Program

Sending a message to the terminal and receiving a response from the user is called a conversation. An online application achieves a conversation between user and application program by a pair of SEND and RECEIVE command. The key points of conversion program are as follows:

- The system sends a message to the screen and waits for the user's response.
- The time taken by user to respond is known as **Think Time**. This time is considerably high, so this is a major drawback of conversion programs.
- The user provides the necessary input and presses a AID key.
- The application processes the user's input and sends the output.
- The program is loaded into the main storage at the beginning and is retained till the task ends.



## Example

Following example shows a conversion program which is taking input from user and then simply displaying the same input on CICS terminal as output:

```

IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 WS-MESSAGE          PIC X(30) VALUE SPACES.
PROCEDURE DIVISION.
    MOVE 'ENTER MESSAGE' TO WS-MESSAGE
    *****
    * SENDING DATA FROM PROGRAM TO SCREEN                *
    *****
    EXEC CICS SEND TEXT
        FROM (WS-MESSAGE)
    END-EXEC
    *****
    * GETTING INPUT FROM USER                            *
    *****
    EXEC CICS RECEIVE
        INTO(WS-MESSAGE)
    END-EXEC
    EXEC CICS SEND TEXT
        FROM (WS-MESSAGE)
    END-EXEC
    *****
    * COMMAND TO TERMINATE THE TRANSACTION                *
    *****
    EXEC CICS RETURN
    END-EXEC.
  
```