Unit 5. Program Control

Objectives

- **Flow of control among the CICS application programs**
- **LINK**
- **XCTL**
- **RETURN**
- **LOAD**
- **RELEASE**

Figure: 5-1. Objectives

Notes :

Task

- **A Task is a unit of work which is scheduled by the operating system.**
- **CICS can be one of the many tasks under the OS.**
- **A unit of work scheduled by CICS is called a CICS Task.**
- **A Task may be accomplished by executing one or more programs.**

Figure: 5-2. Task.

Notes :

Transaction

- **A Transaction is an entity to initiate a task.**
- **A CICS Transaction is a CICS task which is initiated through a Transaction identifier(Transid).**
- **The Transid (1-4 chars) must be registered in the program control table.**
- **Since task is a single execution of a transaction in the single event environment both task and transaction means the same.**

Figure: 5-3. Transaction.

Notes :

Program

- **A program is a set of instructions to achieve a work.**
- **Under CICS an application program (max. 524, 152 bytes) is a set of instructions to perform some CICS task.**
- **Each CICS application Program may perform more than one task.**
- **Each CICS application Program must be registered in the processing program table.**

Figure: 5-4. Program.

Notes :

Application Program Logical Levels

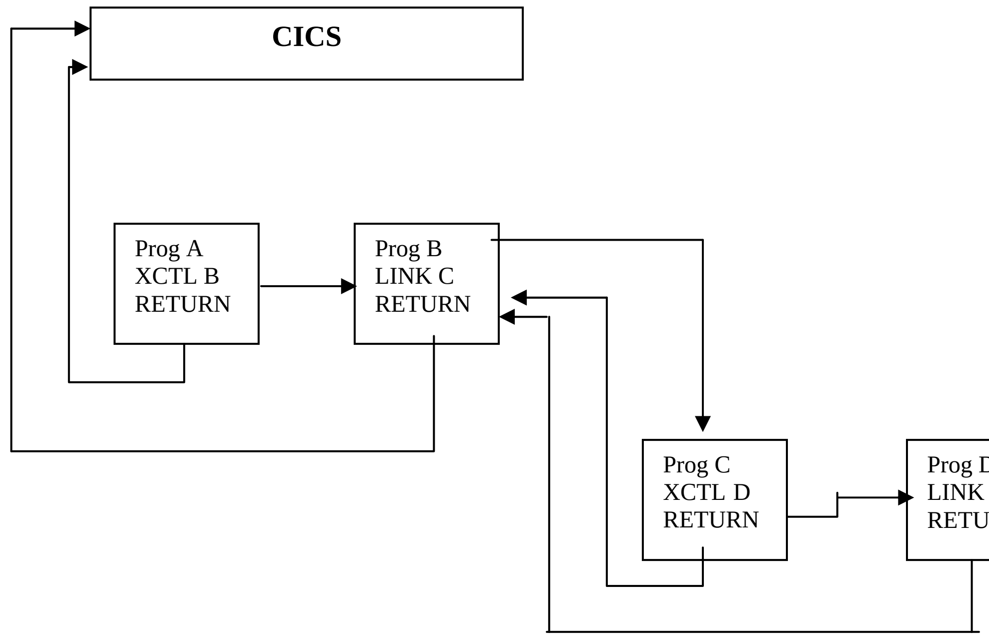


Figure: 5-5. Application Programming logical levels.

Notes :

LINK Command

- **LINK Command passes control from the original program to another program at one logical level lower than the current.**

- **It expects control to be returned to the originating program upon the successful execution of the return command in the linked – to program.**

- **Control is returned to the original calling program to the next sequential executable instruction.**

Figure:5-6. LINK Command

Notes :

LINK Command Syntax

```
EXEC CICS LINK
            PROGRAM(name)
            [COMMAREA(data-name)]
            [LENGTH(data_value)]
END-EXEC.
```

Figure: 5-7. LINK Command Syntax.

Notes :

XCTL Command

- **XCTL transfers program control from one application to another at the same logical level.**
- **The program that transfers control to another is released.**
- **If the program that is to receive control is not resident in main storage ,then it is loaded by CICS.**

Figure: 5-8. XCTL Command.

Notes :

XCTL Command Syntax

```
EXEC CICS XCTL
          PROGRAM(name)
          [COMMAREA(data-name)]
          [LENGTH(data_value)]
          [INPUTMSG(data-area)]
          [INPUTMSGLEN(data-value)]
END-EXEC.
```

Figure:5-9. XCTL Command Syntax.

Notes :

RETURN Command

The RETURN Command when issued by the application, program will return program control to either CICS or the next higher logical level application program.

Figure: 5-10. RETURN Command

Notes :

RETURN Command Syntax

```
EXEC CICS RETURN
          TRANSID(name)
          [COMMAREA(data-area)]
          [LENGTH(data_value)]
          [INPUTMSG(data-area)]
          [INPUTMSGLEN(data-value)]
END-EXEC.
```

Figure: 5-11. RETURN Command Syntax.

Notes :

LOAD Command

- The LOAD Command execution will fetch an application program, a table or a map.
- The command reduces system overhead by incrementing a counter by 1 each time the command is executed.

Figure: 5-12. LOAD Command.

Notes :

LOAD Command Syntax

```
EXEC CICS LOAD
          PROGRAM(name)
          [SET(ptr_ref)]
          [LENGTH(data_area)]
          [FLENGTH(data-area)
          [ENTRY(ptr_ref)]
          [HOLD]
END-EXEC.
```

Figure: 5-13.   LOAD Command Syntax.

Notes :

RELEASE Command

This command will release one of the following which is previously loaded by issuing the LOAD Command.

Loaded Program

Table

Mapset

Figure: 5-14. RELEASE Command.

Notes :

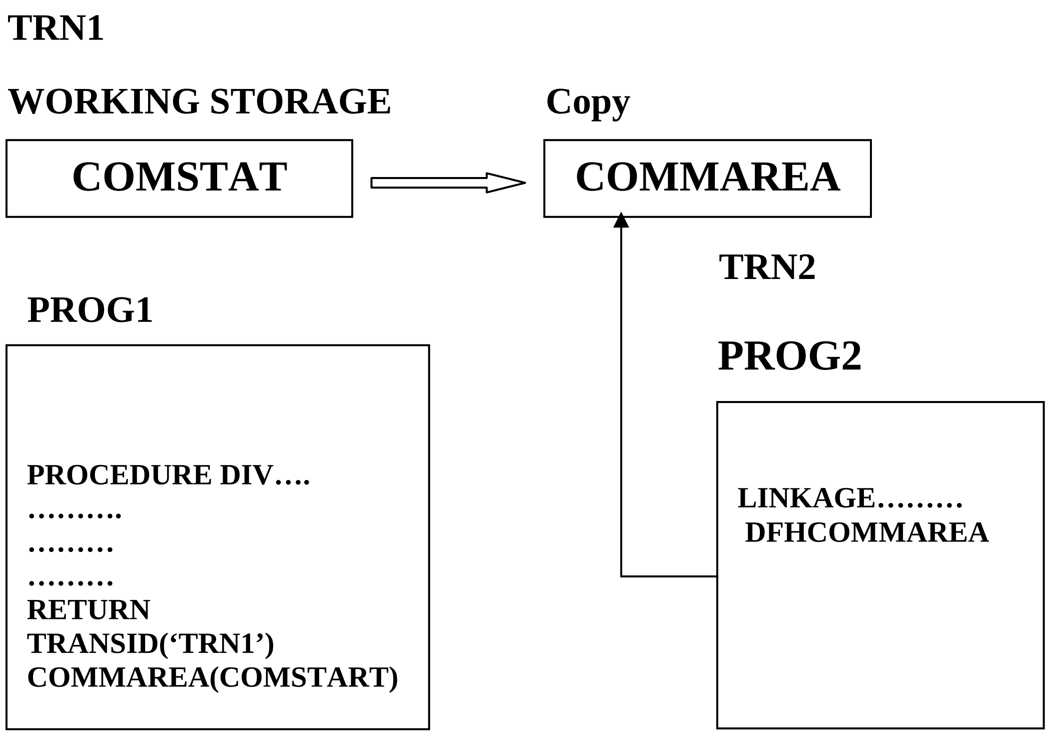
RELEASE Command Syntax

```
EXEC CICS LOAD
          PROGRAM(name)
END-EXEC.
```

Figure: 5-15.   RELEASE Command Syntax.

Notes :

Passing Data to Next Task



PCT ENTRIES

TRN1	PROG1
TRN2	PROG2

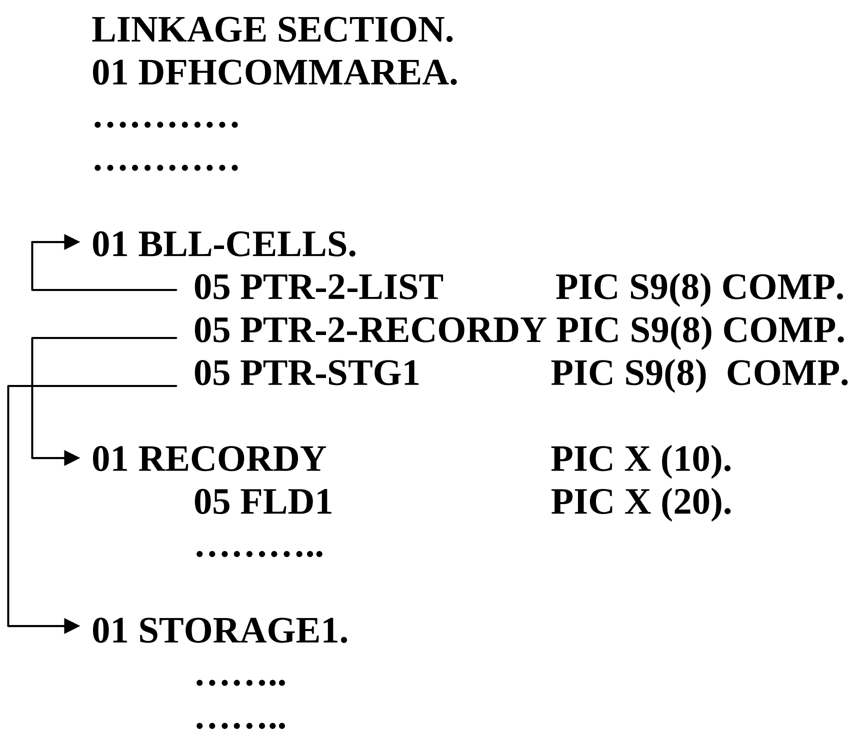
Figure: 5-16. Passing Data to Next Task.

Notes :

A COMMAREA is a CICS maintained unit of storage for passing and receiving data between CICS programs.

The first time a COMMAREA is passed, it must begin as in area of storage in the working storage section of the program passing the COMMAREA.

BLL Cells



```
LINKAGE SECTION.
01 DFHCOMMAREA.
…………
…………

01 BLL-CELLS.
        05 PTR-2-LIST          PIC S9(8) COMP.
        05 PTR-2-RECORDY PIC S9(8) COMP.
        05 PTR-STG1            PIC S9(8)  COMP.

01 RECORDY                     PIC X (10).
        05 FLD1                PIC X (20).
        ………..

01 STORAGE1.
        ……..
        ……..
```

Figure: 5-17. BLL Cells.

Notes :

The Base Locator for Linkage(BLL) cell is a special feature of the COBOL used by CICS.

In COBOL, you declare BLL cells and manage the addresses they contain in order to properly address the fields in each 01-level structure. In VS COBOL II you do not define BLL cells.

Storage has to be defined for these areas via 01-level structures.

Storage has to be acquired for these areas either explicitly by a CICS GETMAIN or implicitly by the type of parameter coded on the CICS command.

In COBOL, the address of each storage area has to be placed in the appropriate BLL cell. In VS COBOL II the special register is used.

In COBOL, CICS automatically handles the BLL cell that points to itself.

THE INPUTMSG OPTION

We use the COMMAREA option on a LINK or XCTL command to pass data to the programs you’re invoking . The invoked program accesses the data through the DFHCOMMAREA field in its linkage section. When we use the INPUTMSG option on a LINK or XCTL command, CICS passes data to the invoked program as if that data were entered by the terminal user. The invoked program accesses the passed data by issuing a terminal control RECEIVE command.

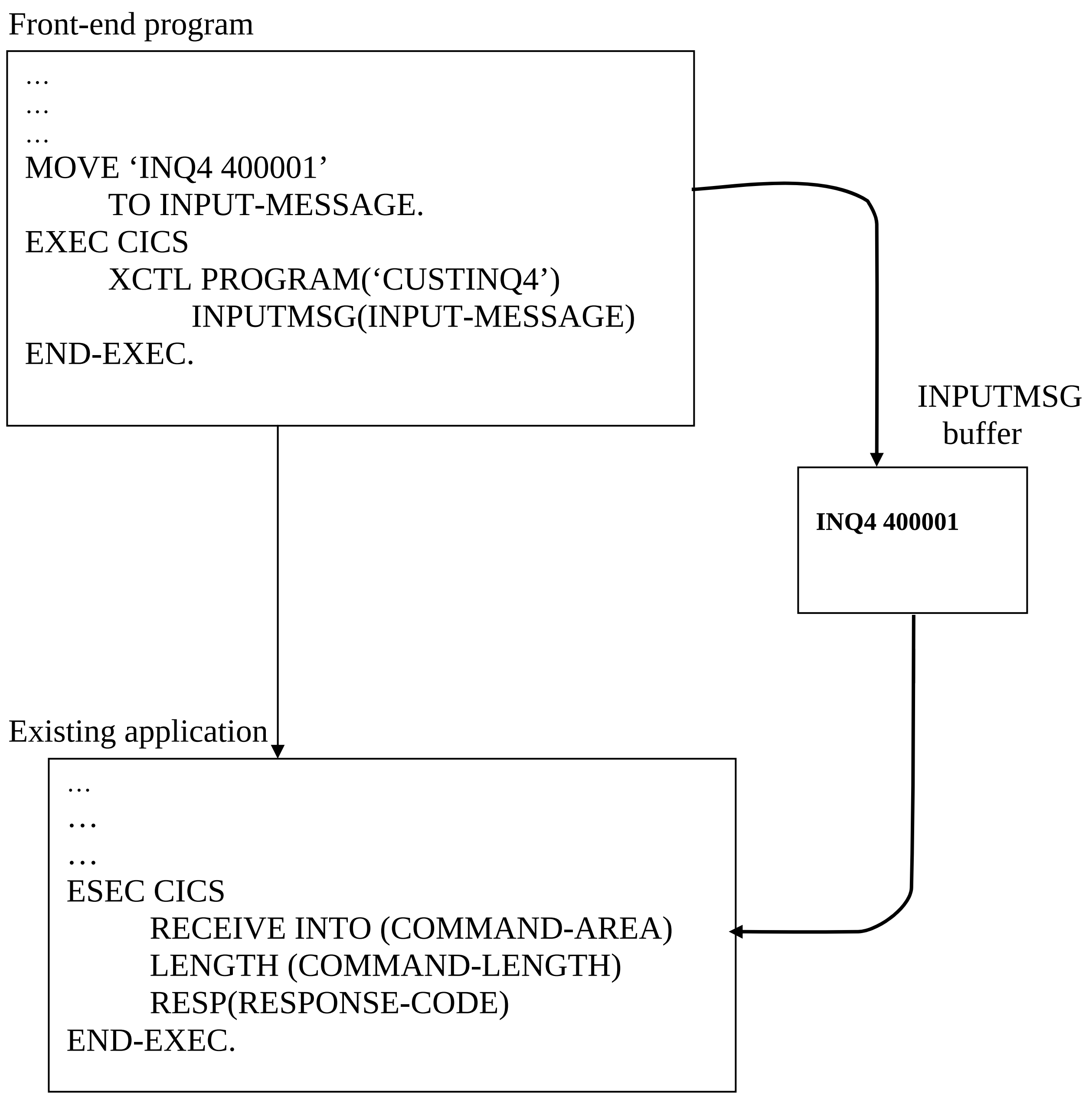
The INPUTMSG option let you create simple front-end programs. Then, these front-end programs can invoke existing application programs that obtain terminal input using the RECEIVE command.

Ex.
The user types the trans-id INQ4 followed by a space and a customer number, like this :

```
INQ4  400001
```

Here, the user is asking the inquiry program to retrieve data for customer 400001.

PASSING DATA TO AN APPLICATION VIA THE INPUTMSG OPTION



Front-end program

```
…
…
…
MOVE 'INQ4 400001'
        TO INPUT-MESSAGE.
EXEC CICS
        XCTL PROGRAM('CUSTINQ4')
                INPUTMSG(INPUT-MESSAGE)
END-EXEC.
```

Existing application

```
…
…
…
ESEC CICS
        RECEIVE INTO (COMMAND-AREA)
        LENGTH (COMMAND-LENGTH)
        RESP(RESPONSE-CODE)
END-EXEC.
```

THE INPUTMSG OPTION (Cont..)

The LINK command

```
EXEC  CICS
        LINK         ROGRAM (name)
                [ COMMAREA (data-area)
                        [ LENGTH (data-value) ]  ]
                [  INPUTMSG (data-area)
                        [  INPUTMSGLEN (data-value) ]
END-EXEC
```

Explanation :

PROGRAM Specifies the one-to-eight-character name of the program to be invoked. This name must be defined in the Processing Program Table (PPT).

COMMAREA Specifies a data area that's passed to the invoked program as a communication area. The invoked program accesses the communication area. via its DFHCOMMAREA field.

LENGTH Specifies a binary half word (PIC S9(4) COMP) or numeric literal that indicates the length of the data area specified in the COMMAREA option.

INPUTMSG Specifies a data area that's passed to the invoked program as an input message. The invoked program accesses the input message by issuing a RECEIVE command.

INPUTMSGLEN Specifies a binary half word (PIC S9(4) COMP) or numeric literal that indicates the length of the data area specified in the INPUTMSG option.

THE INPUTMSG OPTION (Cont..)

The LINK command

```
EXEC  CICS
        XCTL PROGRAM (name)
                [ COMMAREA (data-area)
                        [ LENGTH (data-value) ]  ]
                [  INPUTMSG (data-area)
                        [  INPUTMSGLEN (data-value) ]
END-EXEC
```

Explanation :

PROGRAM   Specifies the one-to-eight-character name of the program to be invoked. This name must be defined in the Processing Program Table (PPT).

COMMAREA   Specifies a data area that's passed to the invoked program as a communication area. The invoked program accesses the communication area. via its DFHCOMMAREA field.

LENGTH   Specifies a binary half word (PIC S9(4) COMP) or numeric literal that indicates the length of the data area specified in the COMMAREA option.

INPUTMSG   Specifies a data area that's passed to the invoked program as an input message. The invoked program accesses the input message by issuing a RECEIVE command.

INPUTMSGLEN   Specifies a binary half word (PIC S9(4) COMP) or numeric literal that indicates the length of the data area specified in the INPUTMSG option.