

CICS

Lab Book

Document Revision History

Date	Revision No.	Author	Summary of Changes
23-Jan-2006	1.0		Content Creation
13-Nov-2009	2.0	Vandana Mistry	Content Upgradation
11-Dec-2009		CLS Team	Review
30-Jun-2011	3.0	Rajita Dhumal	Content Upgradation
8 th -Aug-2012	3.1	Rajita Dhumal	Revamped after Assignment Review
2-Aug-2016	3.2	Veena K	Revamp Post integration

Table of Contents

<i>Document Revision History</i>	<i>2</i>
<i>Table of Contents</i>	<i>3</i>
<i>Getting Started</i>	<i>5</i>
<i>Overview.....</i>	<i>5</i>
<i>Setup Checklist for CICS.....</i>	<i>5</i>
<i>Instructions</i>	<i>5</i>
<i>Lab 1. Create Maps</i>	<i>6</i>
1.1: <i>Compile and Submit the Map</i>	<i>7</i>
<<To Do>>	7
<i>Assignments (Part I):.....</i>	<i>7</i>
<<To Do>>	10
<i>Lab 2. Translating and Compiling COBOL programs.....</i>	<i>14</i>
2.1: <i>Code the following COBOL program.....</i>	<i>14</i>
<i>Lab 3. Display Current Date and Time</i>	<i>17</i>
3.1: <i>Creating a map to display current date and time</i>	<i>17</i>
<i>Lab 4. File Handling.....</i>	<i>18</i>
<i>Assignments 1:.....</i>	<i>18</i>
<i>Assignments 2:.....</i>	<i>19</i>
<i>Lab 5. Analysis, Enhancement and Debugging Assignments.....</i>	<i>21</i>
<i>Lab 6. Analysis, Enhancement and Debugging Assignments.....</i>	<i>23</i>
<i>Functional components of the case study:.....</i>	<i>23</i>
<i>Lab 7. Browse records.....</i>	<i>24</i>
5.1: <i>Write a program for Customer Inquiry Module.</i>	<i>24</i>
<i>Lab 8. Temporary Storage Queue.....</i>	<i>25</i>
6.1: <i>Using TSQ in the MODIFY routine.....</i>	<i>25</i>
<i>Lab 9. Menu Handling.....</i>	<i>26</i>
7.1: <i>Creating a Menu driven application.....</i>	<i>26</i>
<i>Lab 10. Bank Application</i>	<i>27</i>
1. Preface.....	27
2. Project Requirements.....	27
3. Design Details	27
3.1 <i>File Layout.....</i>	<i>27</i>
3.2 <i>Screen Design</i>	<i>28</i>
3.3 <i>Report Layout.....</i>	<i>29</i>
<i>Appendices</i>	<i>34</i>

<i>Appendix A: CEBR, CECI, CEMT, CEDF</i>	<i>34</i>
<i>Appendix A-1: CEBR:</i>	<i>34</i>
<i>Appendix A-2: CECI</i>	<i>38</i>
<i>Appendix A-3: CEMT</i>	<i>42</i>
<i>Appendix A-4: CEDF</i>	<i>51</i>
<i>Create Maps using CA-Realia Simulator</i>	<i>54</i>
<i>Translating and compiling COBOL programs using Ca-Realia</i>	<i>63</i>
<i>File Handling using CA-Realia</i>	<i>71</i>
<i>Appendix B: Table of Figures</i>	<i>74</i>

Getting Started

Overview

This lab book is a guided tour for learning CICS. It comprises solved examples and 'To Do' assignments. Follow the steps provided in the solved examples and work out the 'To Do' assignments given.

Setup Checklist for CICS

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

- Network PCs with Mainframe Connectivity

Please ensure that the following is done:

- PASSPORT PC-TO-HOST (mainframe terminal simulator) is installed
- Connectivity to the Mainframe
- CA-Realia Software

Instructions

- Create a directory by your name in drive <drive>. In this directory, create a subdirectory cics_assgn. For each lab exercise create a directory as lab <lab number>.

Lab 1. Create Maps

Goals	<ul style="list-style-type: none"> Learn to create, edit, and compile Maps.
Time	90-120 minutes

<<To Do>>

Create a Map with the following fields (constants):

ABC LTD.

1. File Maintenance
2. Inquiry Program

Enter your Choice: _

Ctrl+Enter = Process, F10 = EXIT

Message: _____

Operator Message: **Enter**-Process, **F3**-Exit, **F5**-Refresh

- a. Create a data entry field named **Choice** with the following attributes:

LENGTH =1

INITIAL=CHAR

ATTRB=UNPROT, IC, FSET

- b. Create a **Stopper** field for the **Choice** field, with the following attributes:
- c. Create the **Message** field (Display-only) with the following attributes:
- d. Define a **Stopper** field for the **Message** field.
- e. Define a **Dummy** field with the following attributes.

1.1: Compile and Submit the Map

<<To Do>>

Assignments (Part I):

Create the following Maps:

1. Menu Screen:

Customer Information System	
Hindustan Pvt. Ltd.	Date: 99/99/99
Menu	
1.	Customer Maintenance
2.	Customer Inquiry
Enter Valid Choice: _	
Message: _____	
Operator Message: Enter -Process, F3 -Exit, F5 -Refresh	

Note: The **Choice** field is a data entry field and the **Message** field is a display-only field. The remaining fields are constants.

2. Key Screen:

Customer Information System	
Hindustan Pvt. Ltd.	Date: 99/99/99
Enter Customer code: _____	
Message: _____	
Operator Message: Enter -Process, F3 -Menu, F5 -Refresh	

Note: The **Customer code** field is a data entry field and the **Message** field is a display-only field. The remaining fields are constants.

3. Detail Screen:

Customer Information System	
Hindustan Pvt. Ltd.	Date: 99/99/99
<div style="text-align: center; margin-bottom: 20px;">Modification/Inquiry/Data-entry</div> <div style="margin-bottom: 10px;">Customer Code: _____</div> <div style="margin-bottom: 10px;">Name: _____</div> <div style="margin-bottom: 10px;">Address: _____</div> <div style="margin-bottom: 10px;">City: _____</div> <div style="margin-bottom: 10px;">Zip-Code: _____</div> <div style="margin-bottom: 10px;">Message: _____</div> <div>Operator Message: _____</div>	

Note: The Name, Address, City, and zip-code fields are data entry fields. The Customer code, Message, and the Operator Message fields are display-only fields. The remaining fields are constants.

Use following names for maps and mapsets:

Screen	Map Name	Mapset Name
Menu	menumap	menumap
Key	keymap	keymap
Detail	detmap	detmap

Case Study 1: Voting Eligibility

Here we are working with one case study-Voting Eligibility which accept birth details online and writing them into a Birth register with unique SSN thereby generating the VOTER ID.

Major tasks are as mentioned below:

- Accept the SSN of a person
- Check eligibility criteria to vote
- Generate voter-ID based on eligibility criteria.
- Modification of the birth details is also done.

Accept the person's details depending on the choice.

1 – BIRTH REGISTRATION

2 – DEATH REGISTRATION

3 – BIRTH MODIFICATION

4 – VOTER REGISTRY

If choice is 1, BIRTH DETAILS are entered.

If choice is 2, DEATH DETAILS are entered.

If choice is 3, the details of birth can be modified.

If choice is 4, the person can check whether he is eligible for applying the Voter ID and if some other choice is made other than these 4 options, an error message is displayed.

Solve the given problems based on the above case study.

<<To Do>>

Problem 1:

Refer to shared file named 'LOGIMAP' for BMS coding

- The participants have to analyze the program, remove the error and successfully execute the program.
- The participants have to document the error in the excel sheet along with steps taken to resolve the errors.

Problem 2:

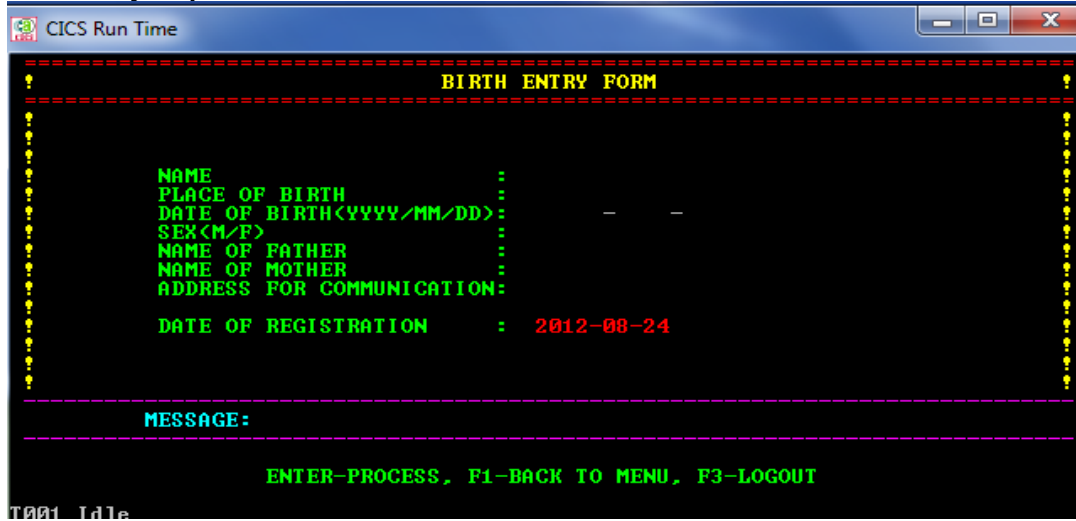
Refer to shared file named 'MENMAP' for BMS coding

- The participants have to analyze the program, remove the error and successfully execute the program.
- BIRTH REGISTRATION choice is coded in the given program. Do enhance the program for remaining choices such as DEATH REGISTRATION, BIRTH MODIFICATION, VOTER REGISTRY, etc
- The participants have to document the error in the excel sheet along with steps taken to resolve the errors.

Problem 3:

Create the following maps:

Birth Entry Map



BIRTH ENTRY FORM

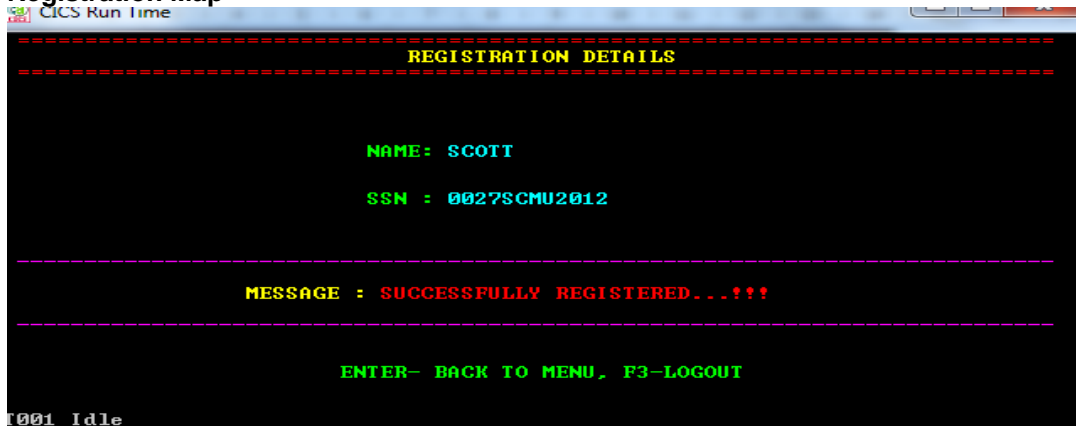
NAME :
 PLACE OF BIRTH :
 DATE OF BIRTH<YYYY/MM/DD> : - -
 SEX<M/F> :
 NAME OF FATHER :
 NAME OF MOTHER :
 ADDRESS FOR COMMUNICATION :
 DATE OF REGISTRATION : 2012-08-24

MESSAGE:

ENTER-PROCESS, F1-BACK TO MENU, F3-LOGOUT

T001 Idle

Registration Map



REGISTRATION DETAILS

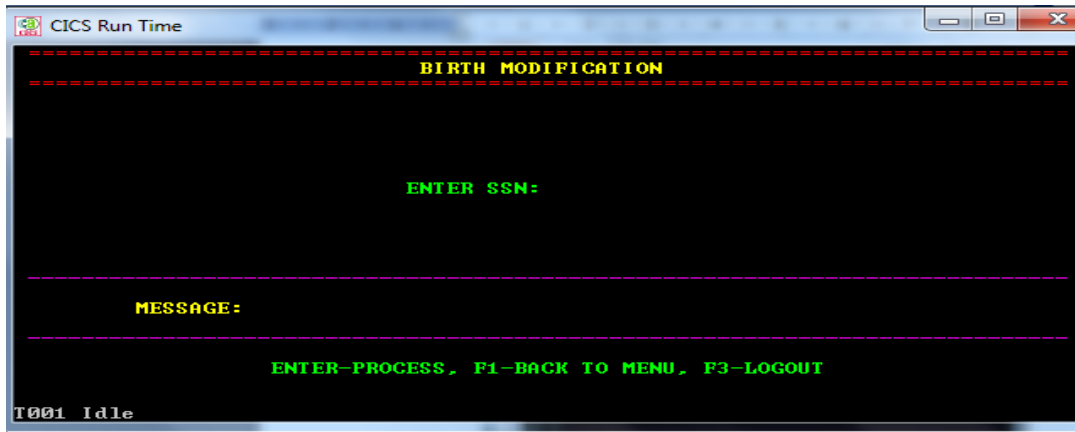
NAME: SCOTT
 SSN : 0027SCMU2012

MESSAGE : SUCCESSFULLY REGISTERED...!!!

ENTER- BACK TO MENU, F3-LOGOUT

T001 Idle

Birth Details Modification Maps



CICS Run Time

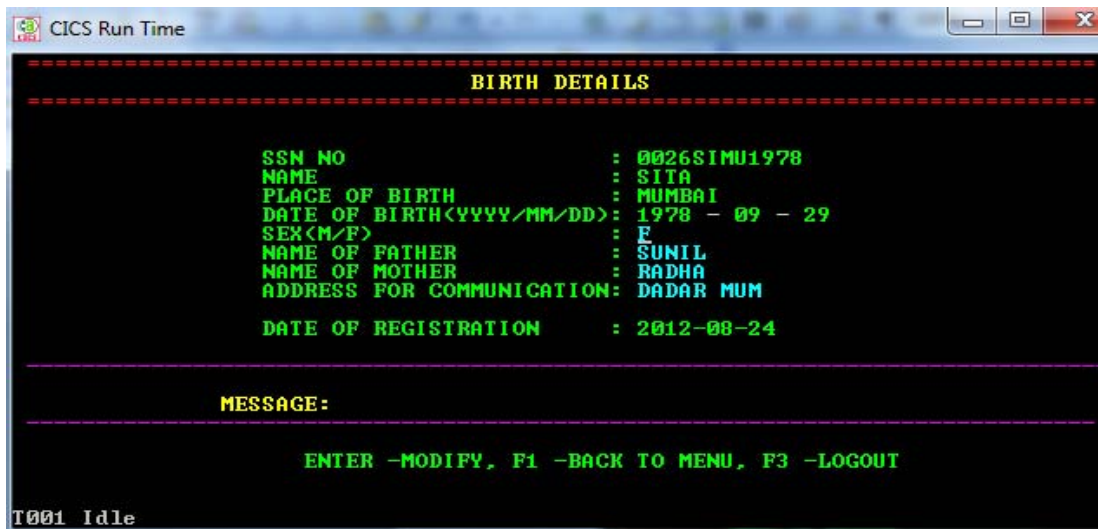
```

=====
BIRTH MODIFICATION
=====

ENTER SSN:

MESSAGE:

=====
ENTER-PROCESS, F1-BACK TO MENU, F3-LOGOUT
=====
T001 Idle
  
```



CICS Run Time

```

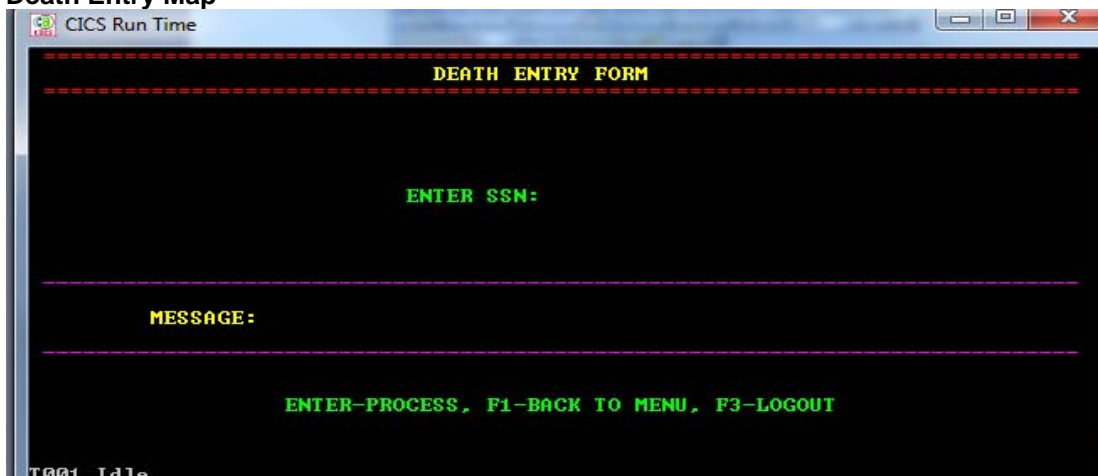
=====
BIRTH DETAILS
=====

SSN NO           : 0026SIMU1978
NAME             : SITA
PLACE OF BIRTH   : MUMBAI
DATE OF BIRTH<YYYY/MM/DD>: 1978 - 09 - 29
SEX<M/F>         : F
NAME OF FATHER   : SUNIL
NAME OF MOTHER   : RADHA
ADDRESS FOR COMMUNICATION: DADAR MUM
DATE OF REGISTRATION : 2012-08-24

MESSAGE:

=====
ENTER -MODIFY, F1 -BACK TO MENU, F3 -LOGOUT
=====
T001 Idle
  
```

Death Entry Map



CICS Run Time

```

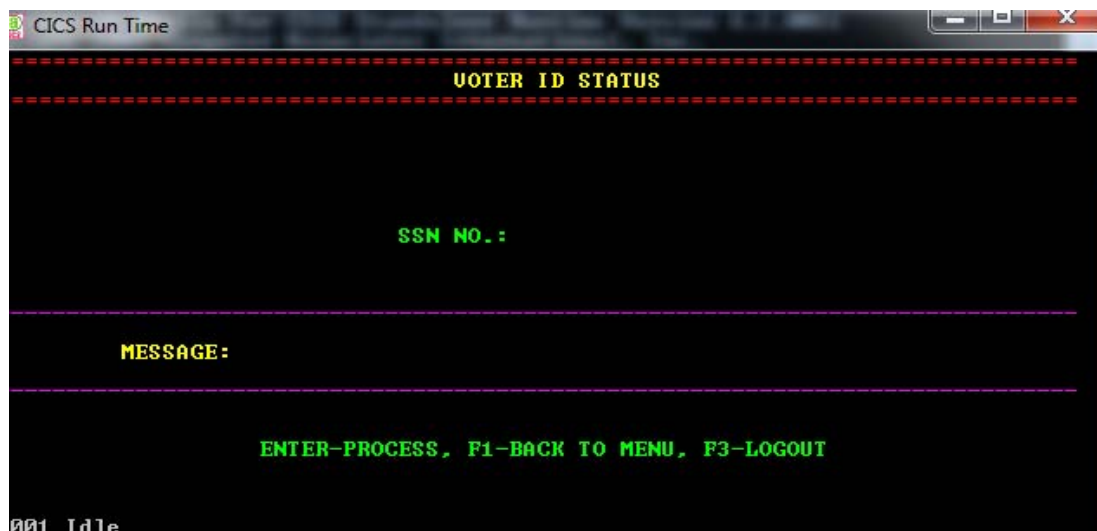
=====
DEATH ENTRY FORM
=====

ENTER SSN:

MESSAGE:

=====
ENTER-PROCESS, F1-BACK TO MENU, F3-LOGOUT
=====
T001 Idle
  
```

Voter Status Map:



Lab 2. Translating and Compiling COBOL programs

Goals	<ul style="list-style-type: none"> To execute a simple pseudo conversational program
Time	30 minutes

2.1: Code the following COBOL program

```

IDENTIFICATION DIVISION.
PROGRAM-ID. SAMP1.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 WS-SWITCHES.
   05 WS-END-SESSION          PIC X(01) VALUE 'N'.
01 WS-END-SESSION-MSG PIC X(50) VALUE SPACES.
01 WS-COMM-AREA          PIC X(01) VALUE SPACES.
*
COPY SAMPMAP.
COPY DFHAID.

LINKAGE SECTION.
01 DFHCOMMAREA          PIC X(01).
**

PROCEDURE DIVISION.
0000-MAIN SECTION.
   IF EIBCALEN = ZEROS
      PERFORM 1000-FIRST-TIME
   ELSE
      MOVE DFHCOMMAREA TO WS-COMM-AREA
      PERFORM 2000-SUBSEQUENT.
*
   IF WS-END-SESSION = 'Y'
      EXEC CICS
         SEND TEXT
            FROM(WS-END-SESSION-MSG)
            LENGTH(+50)
            ERASE
            FREEKB
      END-EXEC
      EXEC CICS
         RETURN
      END-EXEC
   ELSE
      EXEC CICS
         RETURN
         TRANSID('TRN1')
         COMMAREA(WS-COMM-AREA)
         LENGTH(+1)
      END-EXEC.

```

```
0000-EXIT.
EXIT.
*

1000-FIRST-TIME SECTION.
MOVE LOW-VALUES TO SAMPMAP.
MOVE -1 TO CHOICE1.
EXEC CICS
    SEND MAPSET('SAMPMAP')
    MAP('SAMPMAP')
    FROM(SAMPMAP)
    ERASE
    FREEKB
    CURSOR
END-EXEC.
1000-EXIT.
EXIT.
*

2000-SUBSEQUENT SECTION.
IF EIBRID = DFHENTER
    PERFORM 2100-PROCESS-CHOICE
ELSE IF EIBRID = DFHPF10
    MOVE 'SESSION ENDED' TO WS-END-SESSION-MSG
    MOVE 'Y' TO WS-END-SESSION
ELSE
    MOVE 'INVALID ATTENTION KEY PRESSED' TO MESSAGE1.
*

IF WS-END-SESSION NOT = 'Y'
    PERFORM 2500-SEND-MAP.
2000-EXIT.
EXIT.
2100-PROCESS-CHOICE SECTION.
PERFORM 2110-RECEIVE-MAP.
IF CHOICE1 = ZEROS OR CHOICE1 = SPACES
    MOVE 'YOU MUST ENTER CHOICE' TO MESSAGE0
ELSE
    IF CHOICE1 = 1
        MOVE 'ASSIGN1 NOT READY' TO MESSAGE0
    ELSE IF CHOICE1 = 2
        MOVE 'ASSIGN2 NOT READY' TO MESSAGE0
    ELSE
        MOVE 'INVALID CHOICE ' TO MESSAGE1.
2100-EXIT.
EXIT.
2500-SEND-MAP SECTION.
MOVE -1 TO CHOICE1.
EXEC CICS
    SEND MAPSET('SAMPMAP')
    MAP('SAMPMAP')
    FROM(SAMPMAP)
    FREEKB
    CURSOR
```

```
END-EXEC.  
2500-EXIT.  
EXIT.  
  
2110-RECEIVE-MAP SECTION.  
EXEC CICS HANDLE CONDITION  
      MAPFAIL(2110-MAP-FAIL)  
END-EXEC.  
  
EXEC CICS  
      RECEIVE MAPSET('SAMPMPAP')  
      MAP('SAMPMPAP')  
      INTO(SAMPMPAPI)  
END-EXEC.  
GO TO 2110-EXIT.  
2110-MAP-FAIL  
      MOVE 'MAP FAIL OCCURRED' TO WS-END-SESSION-MSG.  
      MOVE 'Y' TO WS-END-SESSION.  
2110-EXIT.  
EXIT.
```

Example 1: Sample Program

Lab 3. Display Current Date and Time

Goals	<ul style="list-style-type: none"> To Create a Map to display current date and time
Time	1.5 Hrs

3.1: Creating a map to display current date and time

1. Create the following map:

DATE:

TIME:

MESSAGE:

F12=DISPLAY F3=EXIT F5=REFRESH

- When user presses **F12** AID key, the Date and Time should be displayed.



Hint: Use the ABSTIME and the FORMATTIME function.

- If user presses **F3** AID key, session should get terminated.
- If user presses **F5** AID key, screen should get refreshed.
- If user presses any other key, screen should display the appropriate error message.

Lab 4. File Handling

Goals	<ul style="list-style-type: none"> Getting familiar with the programs based on file handling concepts.
Time	8 hours

Assignments 1:

Refer to shared file named 'File Read Prog' for CICS-COBOL coding for business logic

- The participants have to analyze the program, structure the same as per the coding standards, remove the error and successfully execute the program.
- The participants have to document the errors in the excel sheet along with steps taken to resolve the errors.
- 'EMPS' is a transaction used to return information pertaining to an employee when the "EMPID" is entered on the screen.
- Create VSAM Cluster along with the employees data.
- The map as shown below and the working storage section of the emp-info are given for your reference. Code BMS file for the given map.
- If the employee id is found the information has to be sent to the screen (Status field) with the message "Emp Id: XXX found."
- If the emp-id key is not found then status field should array the message "Key not found." and the 'EMP ID' field should be set to bright.
- If the Exit option is set to "Y" then the task has to terminated.

Structure of the VSAM KSDS dataset.

Working-Storage Section.

```

01 EMP-IOAREA.
    05 EMP-REC.
        10 EMP-KEY  PIC XXX.
        10 EMP-NAME PIC X(32).
        10 EMP-SEX  PIC X.
        10 EMP-DEPT PIC X(10)
        10 EMP-DESIG PIC X(5).
        10 EMP-SAL  PIC 9(7).
```

EMPLOYEE INFORMATION FORM

Employee ID :
Employee Name :
Employee Designation:
Sex:
Department :
Salary :
Status :

EXIT : X

Assignments 2:

1. Write a program for Customer File Maintenance Module.

- Type the customer code on the **Key** screen.
- If customer code is not present in the data file, then perform the following to add the record:
 - Display the **DETAIL** screen with the following operator message:
F2-ADD, F3-KEY-SCREEN, F5-REFRESH
 - Unprotect all the fields except the customer code field, with the cursor on the Name field.
 - Key in data in all fields, and validate as follows:
 - If any of the fields is blank, position the cursor on the invalid field and display appropriate error message.
 - If all the data is valid, press the **F2** key to add the record in the dataset and the control should come back to the Key screen.
 - After the control comes to the Key screen, the Message 'Record added' should appear in the **Message** field of the Key screen.
 - In case the user presses the **F3** key, no action should be taken. The message 'No record added' should appear in the Message field of the Key screen.
- In case the record exists in the file, display following Operator Message on the Detail screen.
F2-MODIFY, F4-DELETE, F3-KEY-SCREEN, F5-REFRESH
 - To modify the record, key in data in all fields (except customer code), and validate as follows:
 - If any of the fields is blank, position the cursor on the invalid field and display appropriate error message.
 - If all the data is valid, press the **F2** key to modify the record in the dataset and the control should come back to the Key screen.
 - After the control comes to the Key screen, the Message 'Record Modified' should be displayed in the Message field of the Key screen.
 - In case the user presses the **F3** key, no action should be taken. The message 'No record Modified' should be displayed in the Message field of the Key screen.
 - To delete the record, press the **F4** key.

- The record should get deleted, and the control should go back to the Key screen with the appropriate message.
- After processing is done for add / modify / delete, the Key screen should be displayed. The customer code, which got added / modified / deleted, should get displayed on the Key screen with appropriate message 'Record added / Record modified / Record deleted'.
- For any map, if the user presses a wrong key, then display INVALID KEY message in the error-message box.
- Make use of some variable of WS-COMM-AREA, to determine whether to process KEY map, ADD map, or MODIFY map.
- Make use of sections while writing program code.
- Perform all necessary fields validations before proceeding further.

Lab 5. Analysis, Enhancement and Debugging Assignments

Goals	Solve Analysis, Enhancement and Debugging Assignments-Case Study 1
Time	8 Hrs

Case Study 1: Voting Eligibility

Solve the given problems based on the case study referred in Lab 1 i.e. Voting Eligibility.

Problem 1:

Refer to shared program named NYPROJ.txt for CICS-COBOL partial coding.

- The participants have to analyze the program, remove the error and successfully execute the program.
- In order to make it modular, divide the business logic into subprograms.
- The participants have to document the error in the excel sheet along with steps taken to resolve the errors.
- Wherever required, enhance the given program to fulfill the given requirements by developing the correct business logic.
- In the shared program, authorized user credentials are hard coded. Create a separate dataset which will take care of user credentials and refer to the same in the code as a good practice.
- Some of the constraints that need to take care are:
 - If the person is dead, then the respective record in the VOTER register will be deleted only if that entry is present in voter list and will be updated in the DEATH file.
 - The status is given whether the person is eligible for the voter registry.
 - If the person is alive and his age is greater than 18yrs, then that person is eligible to be in voter list.
 - If he/she is dead or not having the required age then respective message is displayed.
 - The details of the people are taken and are updated in the corresponding birth and death files.
 - If the person is eligible then those are updated in the voter registry.

Stretched Assignment based on the given Case Study:

Do implement COBOL Batch applications to generate the given reports.

- **Generate BIRTH certificate with unique SSN.**
- **Accept the SSN, checks eligibility criteria to vote and generate voter-ID. Sample Report for Voter Identity Card is provided for your reference.**

```
*****
                        GOVERNMENT OF INDIA
*****
                        VOTER IDENTITY CARD
-----
CONSTITUENCY:TIRUPATI          SSN NO:0002ABTI1989
-----

ELECTORS NAME                  : ABHILASH B
SEX                             : M
DATE OF BIRTH                  : 1989-07-15
AGE                             : 23
NAME OF THE FATHER             : B VENKATA NARAYANA
ADDRESS                        : FLAT NO: 502,MRVS AP
                               ARTMENT,KBLAYOUT,TPT
REGISTERED ON (YYYY-DD-MM)     : 2012-07-02

                               SIGNATURE OF THE REGISTRAR
*****
```

Lab 6. Analysis, Enhancement and Debugging Assignments

Goals	Solve Analysis, Enhancement and Debugging Assignments-Case Studies
Time	4 Hrs

Case Study: The Census management system

The Census management system will be used to produce statistics that are relevant to data users and that every action in a census must be directed towards producing relevant output that meets the needs of public.

Functional components of the case study:

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed.

- ◆ Data processing (Data capture, Data editing, Coding, Tabulation)
- ◆ Dissemination (publication of printed tables and reports, dissemination on computer media, on-line dissemination)
- ◆ Evaluation (demographic analysis for census evaluation, post-enumeration survey)
- ◆ Analysis of the results
- ◆ Publication of census results (descriptive reports, basic statistical reports)

Refer to shared files for BMS coding as well for COBOL Coding.

- The participants have to analyze the program, remove the error and successfully execute the program.
- File named **census.cbl** contains incomplete COBOL coding which you need to understand and complete as per the requirements of the system
- The participants have to document the error in the excel sheet along with steps taken to resolve the errors.

Case Study: Insurance Policy Processing System (IPPS)

Refer to shared file 'INSURANCE POLICY PROCESSING SYSTEM' for the problem.

Online Processing:

This is the initial step which will allow the Agent/Customer to create a new policy or the customer can edit his existing policy. CICS would be used to create this OLTP (Online Transaction Processing) System

Refer to the file structure and all other required details used in familiarization to MF course.

Lab 7. Browse records

Goals	• Write program for browsing records of the customer file
Time	3 Hrs

5.1: Write a program for Customer Inquiry Module.

Step 1: Write a program for **Customer Inquiry Module**.

1. Display the DETAIL screen.
2. Allow the user to enter specific CUST-CODE or go for various AID keys.
3. Provide the following keys to the user to perform an inquiry:
 - ENTER-PROCESS
 - F1-FIRST
 - F2-LAST
 - F7-PREVIOUS
 - F8-NEXT
 - F5-REFRESH
 - F3-EXIT
4. Display appropriate error messages in the error message box.

Lab 8. Temporary Storage Queue

Goals	<ul style="list-style-type: none">• To create a TSQ and use it in modification
Time	2 hours

6.1: Using TSQ in the MODIFY routine

Use TSQ in the MODIFY routine (Assignment of Lab 4) to determine whether record has been changed during the current task's pseudo-conversational cycle. For this use the following procedure:

Step 1: READ the record for specified customer code.

Step 2: If record is found, store it in a TSQ.

Step 3: Send the DETAIL map to perform MODIFY or DELETE routine.

Step 4: For processing detail screen for modification, perform read with update. After this I/O, again read the record with UPDATE option and now compare this record, which is read, and the one, which is there in the TSQ. If both are same, it implies that no other task has modified that record. So continue processing. If the comparison fails, then display the message "Record has already been modified by someone. Enter the changes again".

Lab 9. Menu Handling

Goals	<ul style="list-style-type: none"> To create an Integrated Menu driven application
Time	3 Hours

7.1: Creating a Menu driven application

Step 1: Integrate Assignments of lab 1, lab 4, lab 5 to create a Menu driven application.

- Accept the choice from the MENU program.
 - In case the operator selects option 1, pass control to the Maintenance (Assignment part IV) program, which will perform File Maintenance. After exiting from the File Maintenance program, the control should come back to the Menu program.
 - In case the operator selects option 2, pass control to the Inquiry (Assignment 5) program, which will perform the Inquiry. After exiting from the Inquiry program, the control should come back to the Menu program.
 - Ensure that Assignment Part IV and Assignment Part V get executed only through the Menu program. It should not get invoked through transaction code.

Note: Use XCTL to invoke Assignment part IV and Assignment V.



Hint: General Guidelines for all Assignments:

- Check for EIBCALEN=0 for invalid entry to INQUIRY and MAINTENANCE program. (The entry should be always from MENU program).
- Check the CA-PROCESS-SW in the main section to branch out to different sections.
- Use "GO TO" to take control to the EXIT paragraph in the section. (Code EXIT at the end of every section).
- Code CICS command and the associated HANDLE CONDITION in the same section.
- All working-storage variables must be initialized.
- Each section must be preceded with proper documentation.

Lab 10. Bank Application

Goals	<ul style="list-style-type: none"> To create an Integrated Bank application
Time	4 Hours

1. Preface

The purpose of this document is to provide specification description of the module exercise & document the project results on completion.

2. Project Requirements

ABC Bank of India wants to create an application to automate their Business. This application should allow them to maintain their customer information and daily transactions of debit or credit. Application needs to be created in CICS and VSAM.

3. Design Details

3.1 File Layout

❖ Input/Output:

Following files needs to be created

File1: ABCCUST – To store the customer information

Field	Type	Length	Value/Format/Criteria
ABCCUST-ACCOUNT-NO	X	6	VSAM Key
ABCCUST-ACC-TYPE	X	1	S-Savings, C-Current
ABCCUST-NAME	X	20	Customer Name
ABCCUST-DOB	X	10	YYYY-MM-DD – Date of birth
ABCCUST-ADDRESS	X	30	Includes Address, City, State
ABCCUST-PHONE	X	10	
ABCCUST-JOIN-DATE	X	10	YYYY-MM-DD
ABCCUST-BALANCE	9	7v99	Comp-3 – Current Balance
Filler	X	18	
Total record length		110	

File2: ABCTTRAN – To store the daily transaction

Field	Type	Length	Value/Format/Criteria
ABCTTRAN-DATE	X	8	VSAM Key – YYYY-MM-DD
ABCTTRAN-TRAN-NO	X	12	VSAM Key – Account number + Current time HHMMSS
ABCTTRAN-TRAN-TYPE	X	1	C – Credit, D-Debit

ABCTTRAN-AMOUNT	9	7V99	COMP-3
Filler	X	24	
Total record length		50	

Note: Key of the file is combination of date and transaction number. Transaction number is created based on the xxxxxxHHMMSS (Account number x(06) and current time stamp X(06))

3.2 Screen Design

Screen 1: This screen is the main screen and customer entry/view and updates are managed in this screen. All the fields are mandatory except Balance field. Balance field is for view only

```

      is for view only

      ABC BANK OF INDIA
      CUSTOMER ENTRY SCREEN

      Account No   : _____
      Account Type : _

      Name        : _____
      Date of Birth : ____-__-__

      Address     : _____

      Phone       : _____

      Joining Date : ____-__-__

      Balance      : 00000000.00

      Message:
      _____
      _____
      =====
      =====
      PF2: REFRESH      PF3: EXIT      PF4: SAVE      PF5: UPDATE
      PF12: DELETE
      PF13 – TRANS SCREEN  PF14-REPORT1  PF15-REPORT2
  
```

Screen 2: This screen can be initiated from Customer entry screen by pressing PF13. This screen is used to enter the transactions (Credit or Debit)

ABC BANK OF INDIA

TRANSACTION ENTRY SCREEN

Account No : _____

Transaction type : _

Amount : 000000.00

Current Balance : 0000000.00

Message:

PF2: REFRESH
PF13 – CUST SCREEN

PF3: EXIT
PF14-REPORT1

PF4: SAVE
PF15-REPORT2

3.3 Report Layout

Report Screen 1:

ABC BANK OF INDIA Daily Report Screen

Date : ____-____-____

Account No	Transaction Time	Transaction Type	Amount
------------	------------------	------------------	--------

xxxxxxx	xx:xx:xx		x
000000.00			
xxxxxxx	xx:xx:xx		x
000000.00			
xxxxxxx	xx:xx:xx		x
000000.00			
xxxxxxx	xx:xx:xx		x
000000.00			
xxxxxxx	xx:xx:xx		x

```
000000.00
XXXXXX      XX:XX:XX      X
000000.00
XXXXXX      XX:XX:XX      X
000000.00
XXXXXX      XX:XX:XX      X
000000.00
XXXXXX      XX:XX:XX      X
000000.00
XXXXXX      XX:XX:XX      X
000000.00

Message:
=====
=====
PF2: REFRESH      PF3: EXIT      PF7: UP      PF8:
Down
PF13 – CUST SCREEN      PF14-TRANS SCREEN      PF15-
REPORT2
```

Report Screen 2:

ABC BANK OF INDIA

Customer Report Screen

Account No : _____

Customer Name : _____

Current Balance : 0000000.00

Date	Transaction Time	Transaction Type	Amount
XXXX-XX-XX	XX:XX:XX		X
000000.00			
XXXX-XX-XX	XX:XX:XX		X
000000.00			
XXXX-XX-XX	XX:XX:XX		X
000000.00			
XXXX-XX-XX	XX:XX:XX		X
000000.00			
XXXX-XX-XX	XX:XX:XX		X
000000.00			
XXXX-XX-XX	XX:XX:XX		X
000000.00			
XXXX-XX-XX	XX:XX:XX		X
000000.00			
XXXX-XX-XX	XX:XX:XX		X
000000.00			
XXXX-XX-XX	XX:XX:XX		X
000000.00			

Message:

=====

=====

PF2: REFRESH

PF13 – CUST SCREEN

PF14-TRANS SCREEN

PF15-REPORT1

PF3: EXIT

PF7: UP

PF8:

Pseudo code

Screen 1: - Program 1: Customer Entry screen

1. User enters Account number and presses ENTER key: Account information needs to be displayed based on the customer data available in the File.
 - a. If the customer information NOT FOUND then allow the user to type the rest of the information.
 - b. If the customer information FOUND then allow the user to change the details, (Except Balance) based on PF5 Key press update the changes. Based on PF12 Key press delete the Account details
2. Enter all the information (Except Balance field – Protected) and press PF4: Insert the record into ABCCUST file. If the account number already exists, display the error message
3. PF13 Key press: Transfer the control to program 2 (Transaction Entry)
4. PF14 Key press: Transfer the control to Report program 1 (Daily report screen)
5. PF15 Key press: Transfer the control to Report program 2 (Customer report screen)

Screen 2: - Program 2: Transaction Entry screen

1. User enters Account number and presses ENTER key: Account Balance will be displayed in the protected field.
2. Enter all the information (Except Balance field – Protected) and press PF4: Insert the record into ABCTAN file. If the account number not exists, display the error message
3. PF13 Key press: Transfer the control to program 1 (Customer Entry)
4. PF14 Key press: Transfer the control to Report program 1 (Daily report screen)
5. PF15 Key press: Transfer the control to Report program 2 (Customer report screen)

Report 1: - Program 3: Daily Report screen

1. User enters Date and presses ENTER key: All the transaction belongs to that date needs to be displayed.
 - a. Start the ABCTAN file using key (Date + low values) and read all the records and write into TSQ till the date ends.
 - b. From the TSQ read the records and display into Screen.
 - c. If the date is changed then delete the TSQ and reload the new records
 - d. If no records are available for that date, display error message
2. PF7 – Read the previous page of records from TSQ and display
3. PF8 – Read the next page of records from TSQ and display
4. PF13 Key press: Transfer the control to program 1 (Customer Entry)
5. PF14 Key press: Transfer the control to Program 2 (Transaction Entry)
6. PF15 Key press: Transfer the control to Report program 2 (Customer report screen)

Report 2: - Program 4: Customer Report screen

1. User enters Customer number and presses ENTER key: Customer information and all the transaction belongs to that customer needs to be displayed.
 - a. Read the ABCCUST file and based on the customer number and display to the screen (Save this to Commarea).
 - b. Start the ABCTAN file using key (low values) and read all the records, and write the matching customer records into TSQ till the file ends.
 - c. From the TSQ read the records and display into Screen.
 - d. If the Customer number is changed then delete the TSQ and reload the new records
 - e. If no records are available for that Customer, display error message

5. PF7 – Read the previous page of records from TSQ and display
6. PF8 – Read the next page of records from TSQ and display
7. PF13 Key press: Transfer the control to program 1 (Customer Entry)
2. PF14 Key press: Transfer the control to Program 2 (Transaction Entry)
3. PF15 Key press: Transfer the control to Report program 1 (Daily report screen)

Appendices

Appendix A: CEBR, CECI, CEMT, CEDF

Appendix A-1: CEBR:

The **Temporary Storage Browse (CEBR)** is a CICS supplied transaction, which browses **Temporary Storage Queue (TSQ)**. It is a convenient tool if you need to display the content to TSQ when you are monitoring an application program through EDF.

Invoking CEBR is simple. Type **CEBR**, and press **Enter** key. Then you can type the command **QUEUE <TSQ Name>**, and press **Enter** key. The content of the TSQ will be displayed.

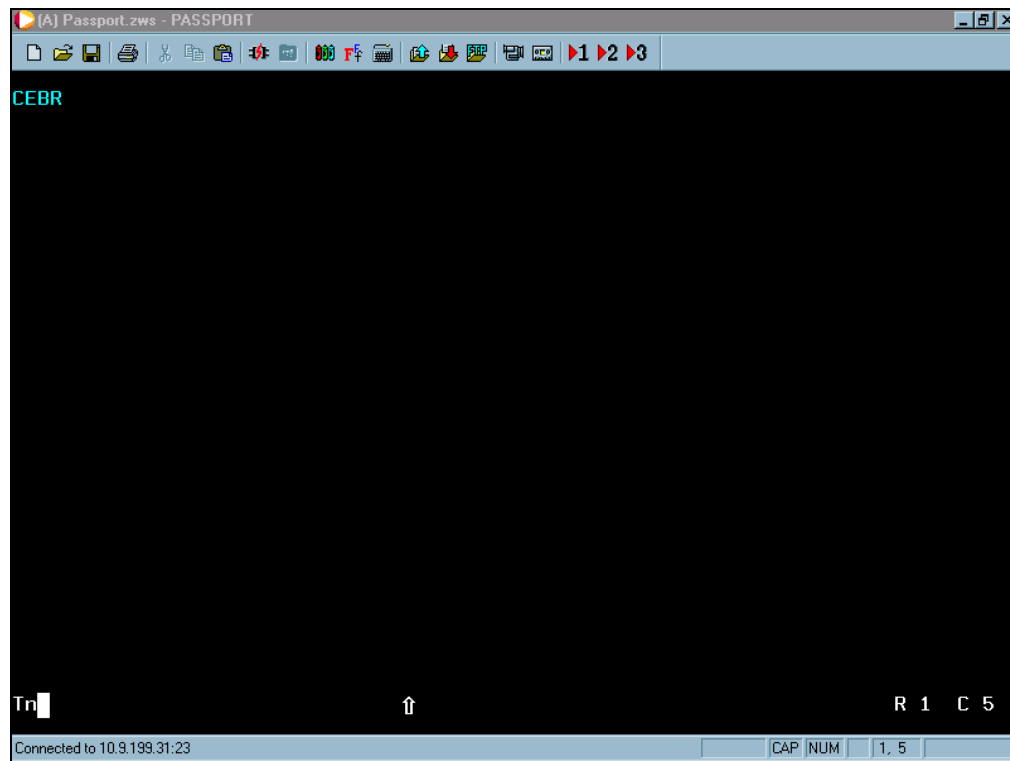
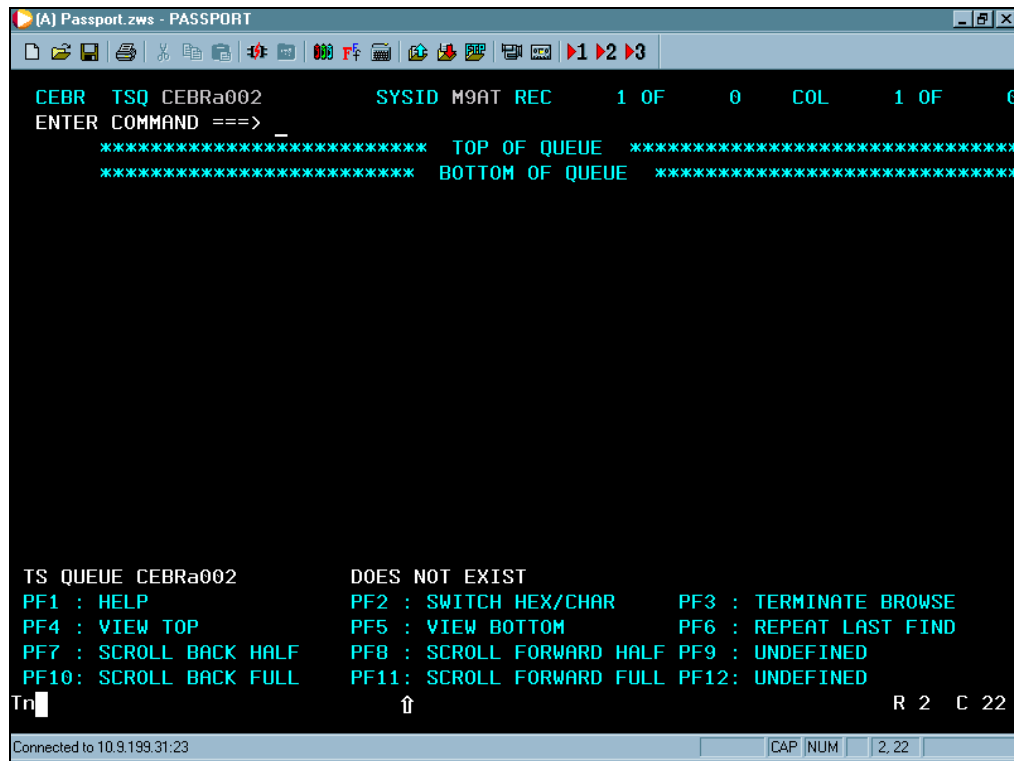


Figure 1: CEBR



```

(A) Passport.zws - PASSPORT
CEBR TSQ CEBRa002          SYSID M9AT REC 1 OF 0 COL 1 OF 0
ENTER COMMAND ==>
***** TOP OF QUEUE *****
***** BOTTOM OF QUEUE *****

TS QUEUE CEBRa002          DOES NOT EXIST
PF1 : HELP                 PF2 : SWITCH HEX/CHAR      PF3 : TERMINATE BROWSE
PF4 : VIEW TOP             PF5 : VIEW BOTTOM        PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF    PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL    PF11: SCROLL FORWARD FULL PF12: UNDEFINED
Tn
                               ↑
R 2 C 22
Connected to 10.9.199.31:23
CAP NUM 2, 22

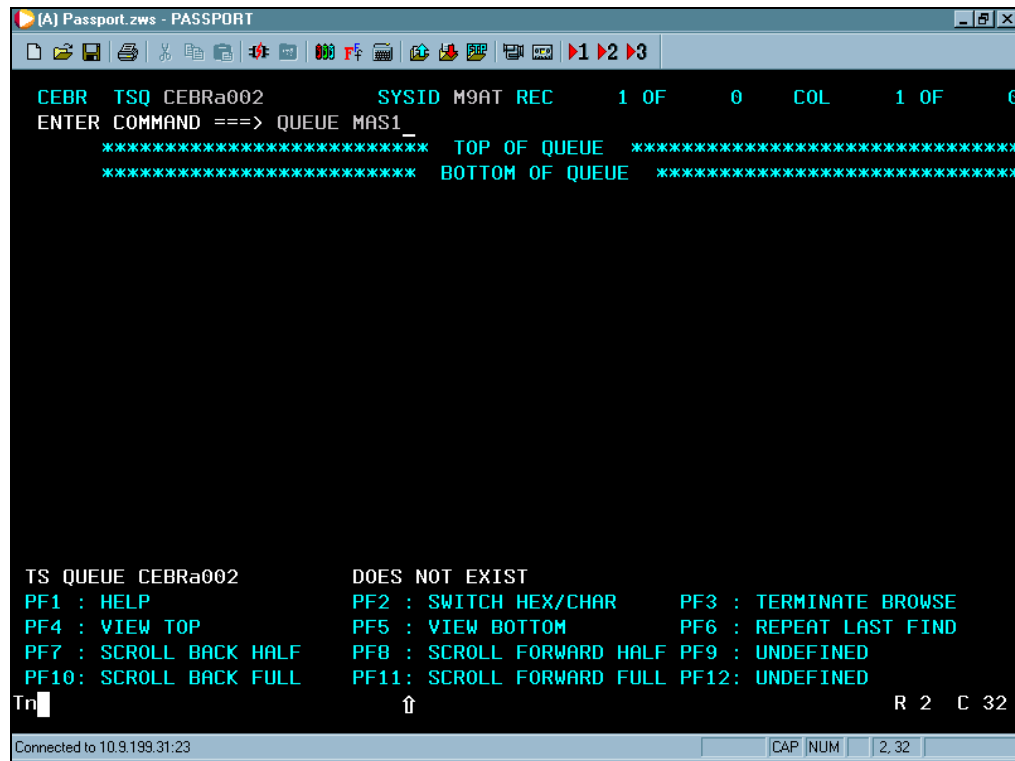
```

Figure 2: CEBR

Press **PF3** key to end the session.



Figure 3: Browse ended message



```

(A) Passport.zws - PASSPORT
CEBR TSQ CEBRa002          SYSID M9AT REC 1 OF 0 COL 1 OF 0
ENTER COMMAND ==> QUEUE MAS1_
***** TOP OF QUEUE *****
***** BOTTOM OF QUEUE *****

TS QUEUE CEBRa002          DOES NOT EXIST
PF1 : HELP                 PF2 : SWITCH HEX/CHAR    PF3 : TERMINATE BROWSE
PF4 : VIEW TOP             PF5 : VIEW BOTTOM      PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF    PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL    PF11: SCROLL FORWARD FULL PF12: UNDEFINED

Tn ↑                        R 2 C 32
Connected to 10.9.199.31:23 CAP NUM 2, 32

```

Figure 4: CEBR

Appendix A-2: CECI

CECI is **Command Level Interpreter (CECI)**. It is a CICS-supplied transaction which performs interactive patching into application System.

In CICS, you need to expedite modules. These can be instances wherein the module gets trapped and even after ending the expediter session the module is not released.

Objective: To Release a Program:

Step 1: Type the command **CECI REL PROG (H2537QAC)**

H2537QAC is the module name.

Argument used for module is PROG (Program).

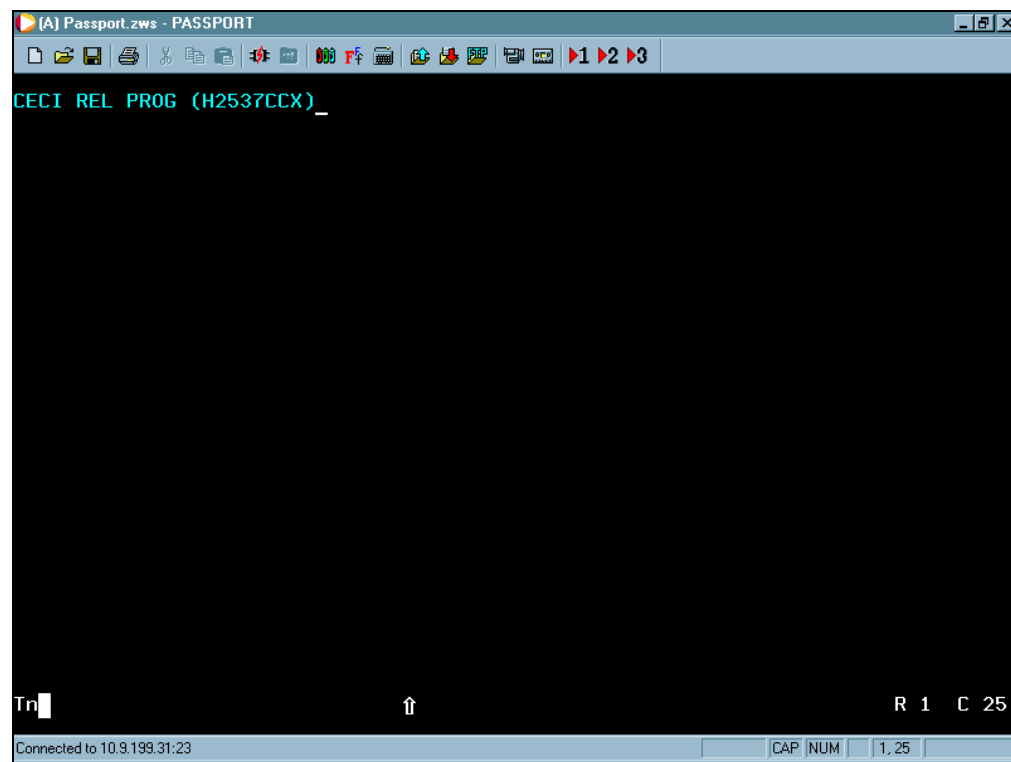
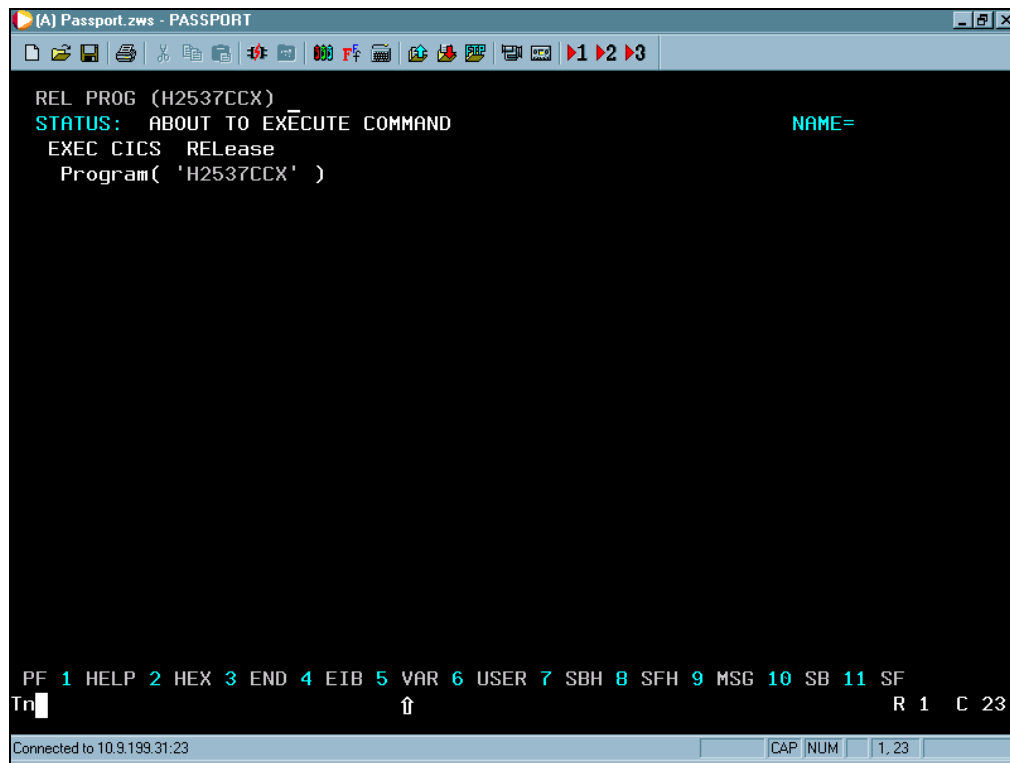


Figure 5: Typing the command

Output is as shown below:



The screenshot shows a terminal window titled "(A) Passport.zws - PASSPORT". The main display area is black with white text. The text reads: "REL PROG (H2537CCX)", "STATUS: ABOUT TO EXECUTE COMMAND", "EXEC CICS RELease", and "Program('H2537CCX')". To the right of "STATUS:" is the text "NAME=" in red. At the bottom of the terminal, there is a command line with "PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF" and a cursor pointing to the first character of the command line. Below the command line, it says "Connected to 10.9.199.31:23".

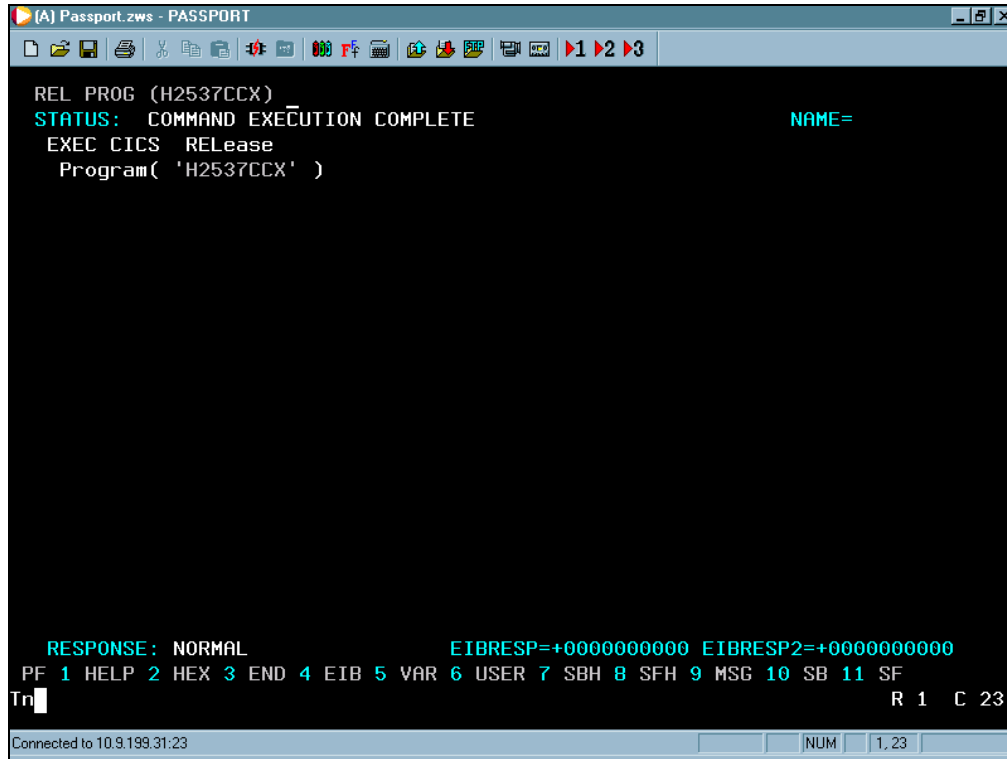
```
REL PROG (H2537CCX)
STATUS: ABOUT TO EXECUTE COMMAND
EXEC CICS RELease
Program( 'H2537CCX' )

PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF
Tn
↑
R 1 C 23

Connected to 10.9.199.31:23
```

Figure 6: Output

Step 2: Press **Enter** key to release the program.



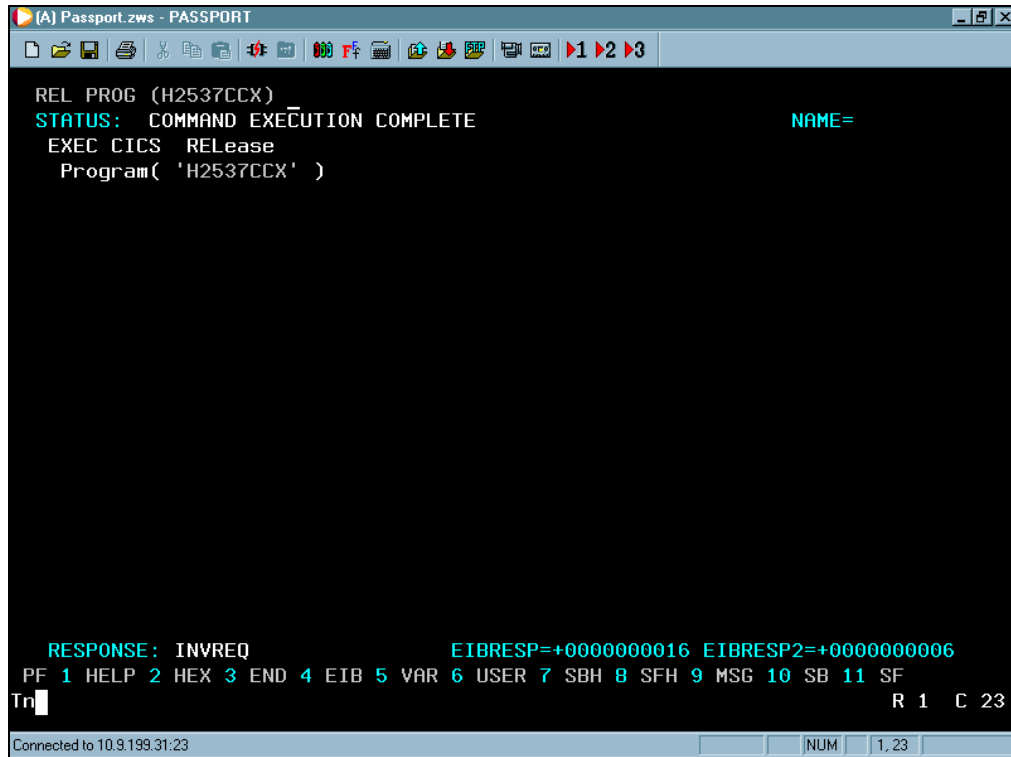
The screenshot shows a terminal window titled "(A) Passport.zws - PASSPORT". The main display area is black with green text. The text shown is: "REL PROG (H2537CCX)", "STATUS: COMMAND EXECUTION COMPLETE", "EXEC CICS RELease", and "Program('H2537CCX')". At the bottom, it shows "RESPONSE: NORMAL", "EIBRESP=+0000000000", "EIBRESP2=+0000000000", and a list of function keys: "PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF". Below this is a cursor and "Tn". At the very bottom, it says "Connected to 10.9.199.31:23" and "NUM 1,23".

```
(A) Passport.zws - PASSPORT
REL PROG (H2537CCX)
STATUS: COMMAND EXECUTION COMPLETE
EXEC CICS RELease
Program( 'H2537CCX' )

RESPONSE: NORMAL
EIBRESP=+0000000000 EIBRESP2=+0000000000
PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF
Tn
Connected to 10.9.199.31:23
NUM 1,23
```

Figure 7: Response: Normal

If the program is already released, the response is INVREQ as you can see in the screen shot shown below:



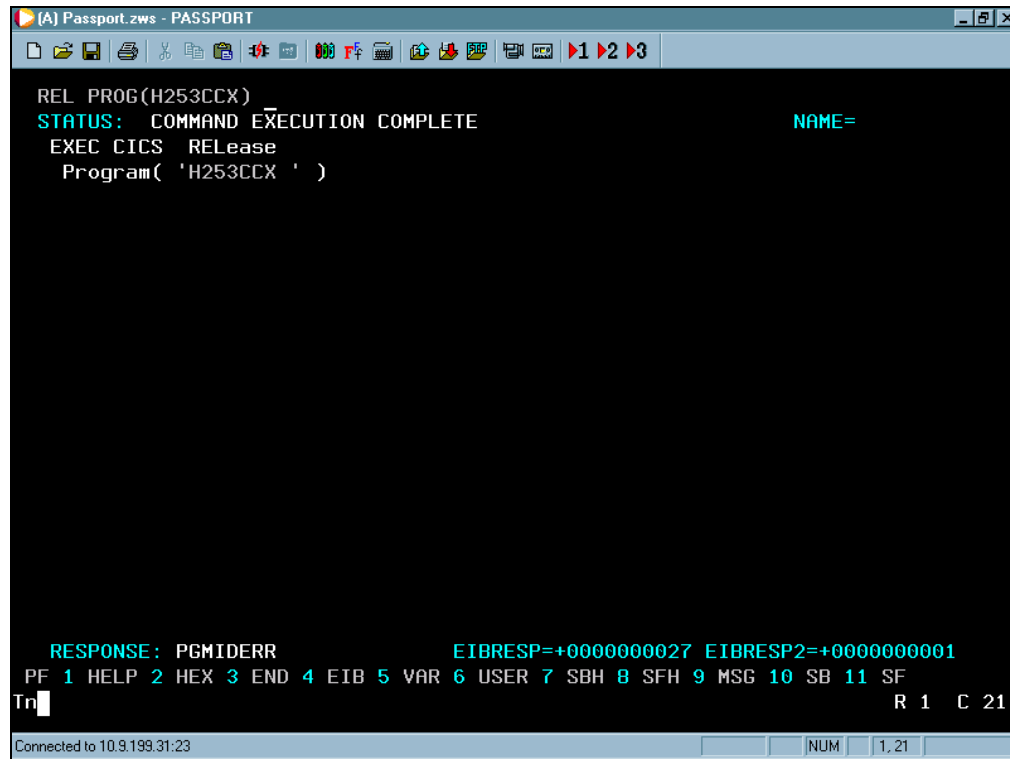
The screenshot shows a CICS terminal window titled "(A) Passport.zws - PASSPORT". The main display area is black with white and green text. The text indicates that a command execution is complete for program H2537CCX. At the bottom, a response message "RESPONSE: INVREQ" is shown, along with hexadecimal values for EIBRESP and EIBRESP2. A command menu is visible at the bottom, and the terminal status bar shows the connection to 10.9.199.31:23.

```
(A) Passport.zws - PASSPORT
REL PROG (H2537CCX)
STATUS:  COMMAND EXECUTION COMPLETE      NAME=
EXEC CICS  RELease
  Program( 'H2537CCX' )

RESPONSE: INVREQ      EIBRESP=+0000000016 EIBRESP2=+0000000006
PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF
Tn
R 1 C 23
Connected to 10.9.199.31:23
```

Figure 8: Response: INVREQ

Also if the program name is not valid, Response is PGMIDERR. It means that the module name entered is invalid or such a module does not exist.



The screenshot shows a terminal window titled "(A) Passport.zws - PASSPORT". The command executed is "REL PROG(H253CCX)". The status is "COMMAND EXECUTION COMPLETE". The response is "PGMIDERR". The terminal also shows a list of function keys: PF 1 HELP, 2 HEX, 3 END, 4 EIB, 5 VAR, 6 USER, 7 SBH, 8 SFH, 9 MSG, 10 SB, 11 SF. The terminal is connected to 10.9.199.31:23.

```

REL PROG(H253CCX)
STATUS:  COMMAND EXECUTION COMPLETE
EXEC CICS  RELease
  Program( 'H253CCX ' )

RESPONSE: PGMIDERR
EIBRESP=+0000000027 EIBRESP2=+0000000001
PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF
Tn R 1 C 21
Connected to 10.9.199.31:23
  
```

Figure 9: Response: PGMIDERR

Appendix A-3: CEMT

CEMT (Enhanced Master Terminal) transaction is a CICS supplied transaction, which manipulates the CICS environment, such as transactions, programs, files, TAQs, and tasks.

Major functions:

- **INQUIRE:** To inquire about the status of CICS environments
- **SET:** To update the status of the CICS environments
- **PERFORM:** For further system operations

Example:

Objective: Close a file in CICS environment.

Steps:

Step 1: Type Command: **CEMT SET FILE (H2537NBL)**

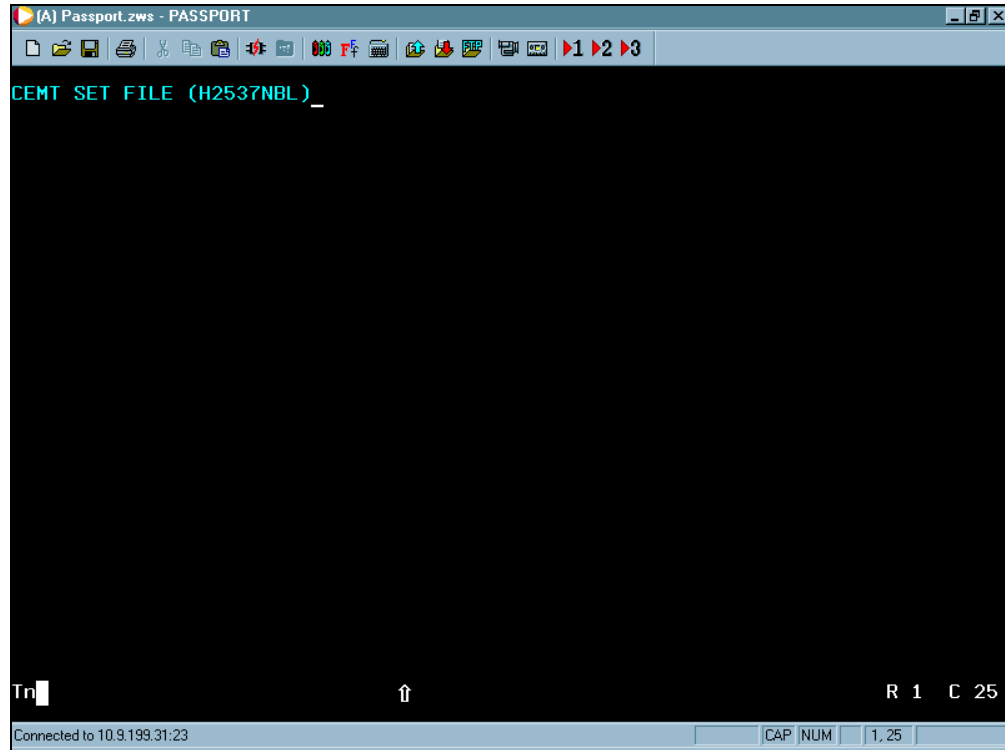
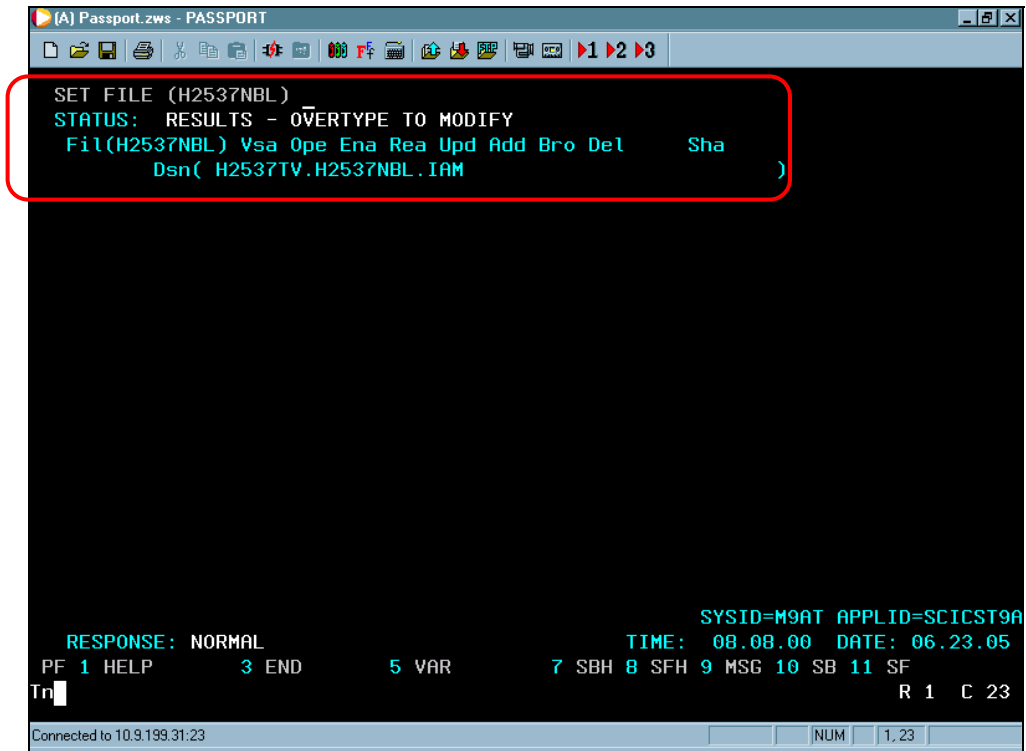


Figure 10: Typing the command

Output is as shown below:

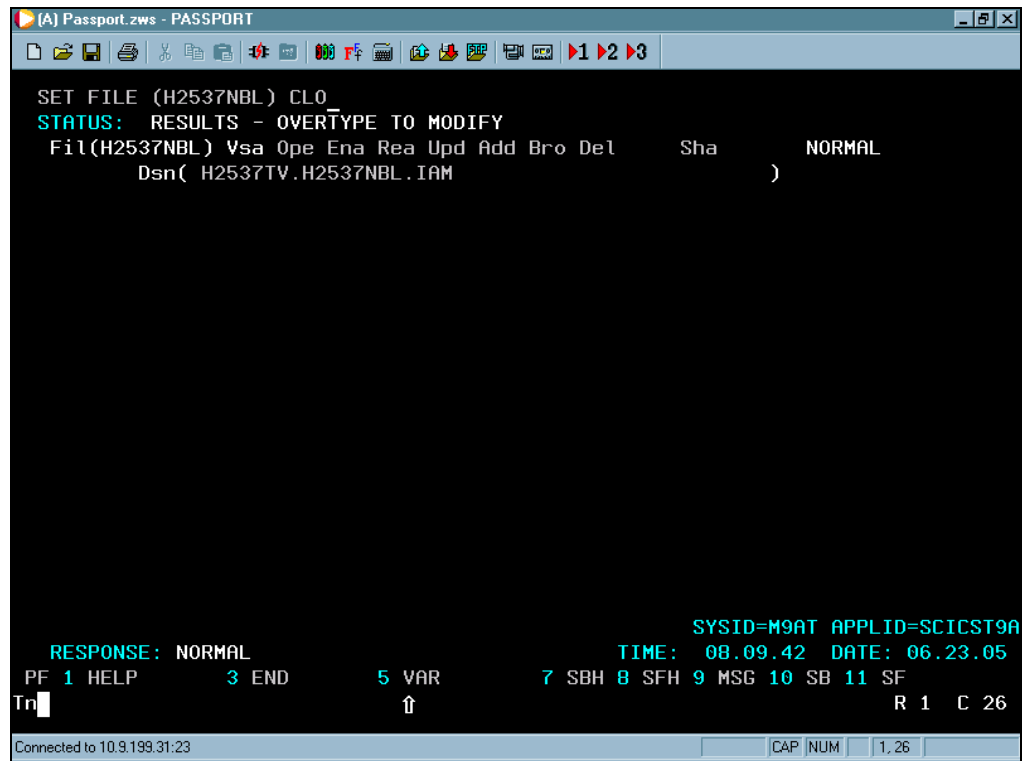


```
(A) Passport.zws - PASSPORT
SET FILE (H2537NBL)
STATUS: RESULTS - OVERTYPE TO MODIFY
Fil(H2537NBL) Vsa Ope Ena Rea Upd Add Bro Del Sha
Dsn( H2537TV.H2537NBL.IAM )

RESPONSE: NORMAL
TIME: 08.08.00 DATE: 06.23.05
PF 1 HELP 3 END 5 VAR 7 SBH 8 SFH 9 MSG 10 SB 11 SF
Trn
R 1 C 23
Connected to 10.9.199.31:23 NUM 1,23
```

Figure 11: Output

Step 2: Type **Clo** and press **Enter** key.



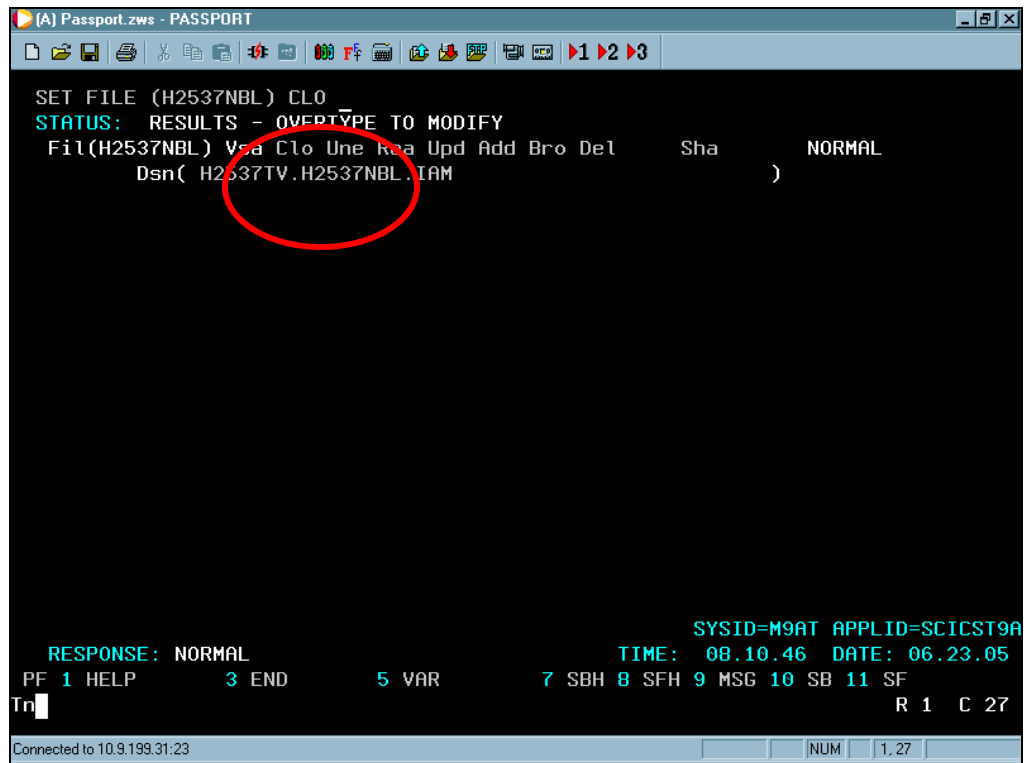
```

(A) Passport.zws - PASSPORT
SET FILE (H2537NBL) CLO
STATUS: RESULTS - OVERTYPE TO MODIFY
  Fil(H2537NBL) Vsa Ope Ena Rea Upd Add Bro Del      Sha      NORMAL
      Dsn( H2537TV.H2537NBL.IAM                      )

RESPONSE: NORMAL
                                SYSID=M9AT APPLID=SCICST9A
PF 1 HELP      3 END      5 VAR      7 SBH 8 SFH 9 MSG 10 SB 11 SF
Tn             ↑
                                TIME: 08.09.42 DATE: 06.23.05
                                R 1 C 26
Connected to 10.9.199.31:23 CAP NUM 1,26
  
```

Figure 12: Typing Command

Output is as shown below:



```

(A) Passport.zws - PASSPORT
SET FILE (H2537NBL) CLO
STATUS: RESULTS - OVERTYPE TO MODIFY
Fil(H2537NBL) Ysa Clo Une Rea Upd Add Bro Del Sha NORMAL
Dsn( H2537TV.H2537NBL.IAM )

RESPONSE: NORMAL
TIME: 08.10.46 DATE: 06.23.05
PF 1 HELP 3 END 5 VAR 7 SBH 8 SFH 9 MSG 10 SB 11 SF
Tr
Connected to 10.9.199.31:23 NUM 1.27
  
```

Figure 13: Output

As you can see, the file was earlier **Open (Ope)** and now after keying in the command '**Clo**' the status of the file has changed to **Close (Clo)**.

To open the file, issue command '**Ope**' in similar fashion.

End the transaction by pressing **PF3** key.

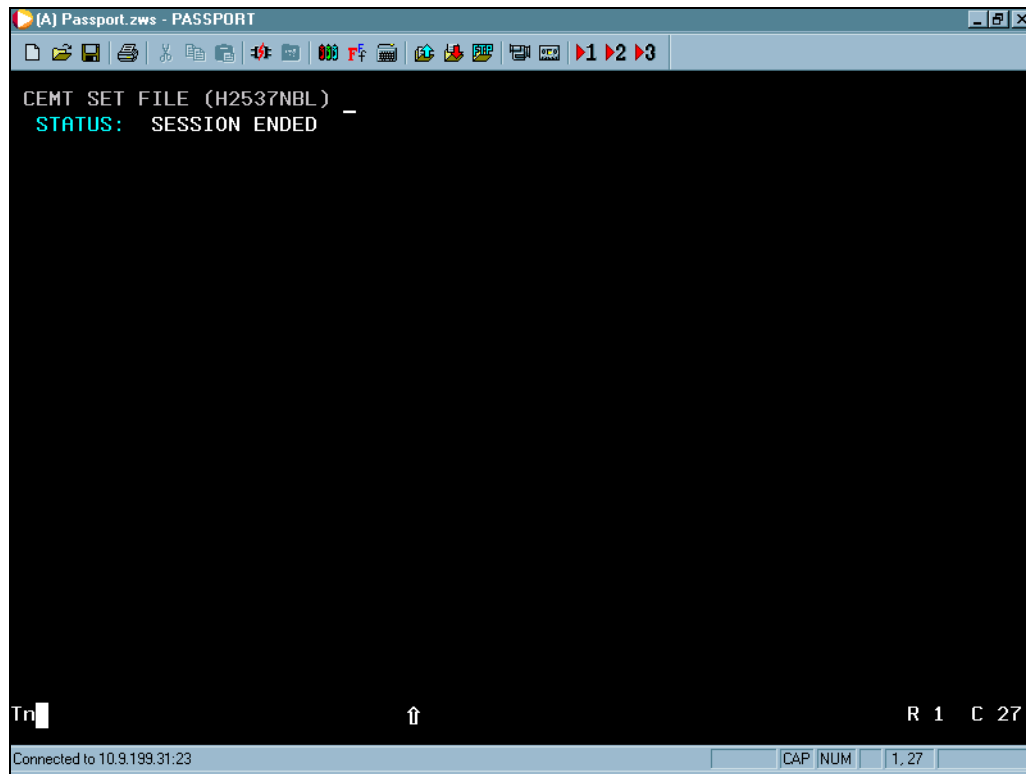


Figure 14: Session ended message

You can see that **Status: Session Ended**.

Note: The command is not case sensitive.

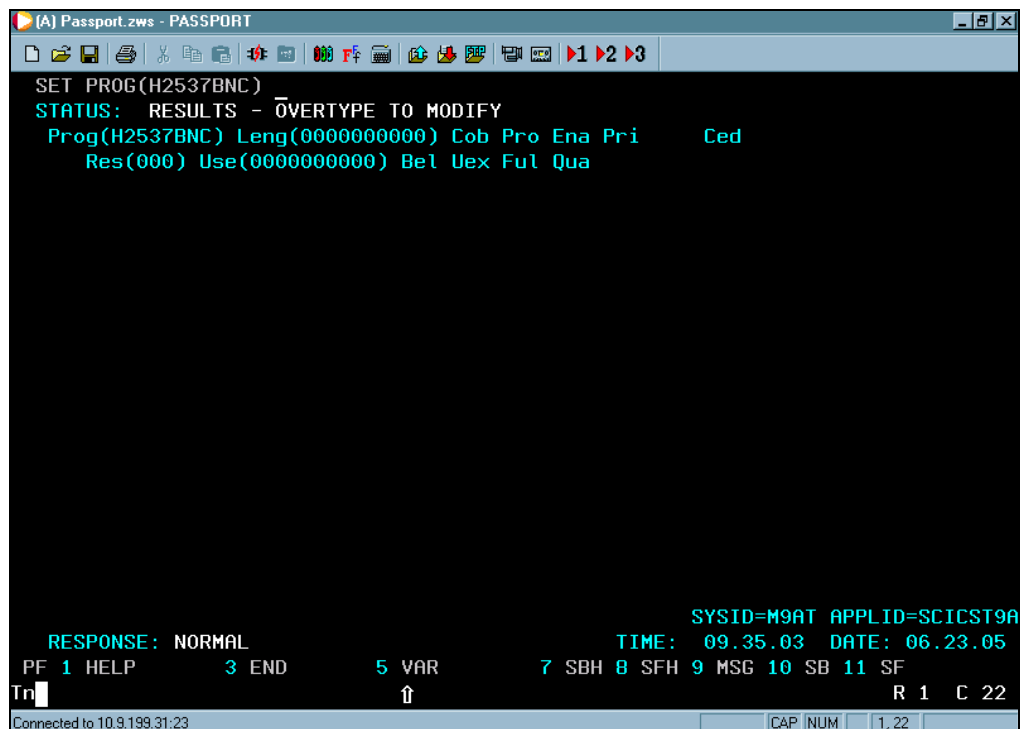
Using **CEMT** you can also NEWCOPY module.

Say you have made changes to your module but it has not reflected in the region. You need to do a new copy (this means overlay the previous version).



The screenshot shows a CICS terminal window titled '(A) Passport.zws - PASSPORT'. The command 'CENT SET PROG(H2537BNC)' has been entered. The status bar at the bottom indicates 'Connected to 10.9.199.31:23' and 'CAP NUM 1,24'.

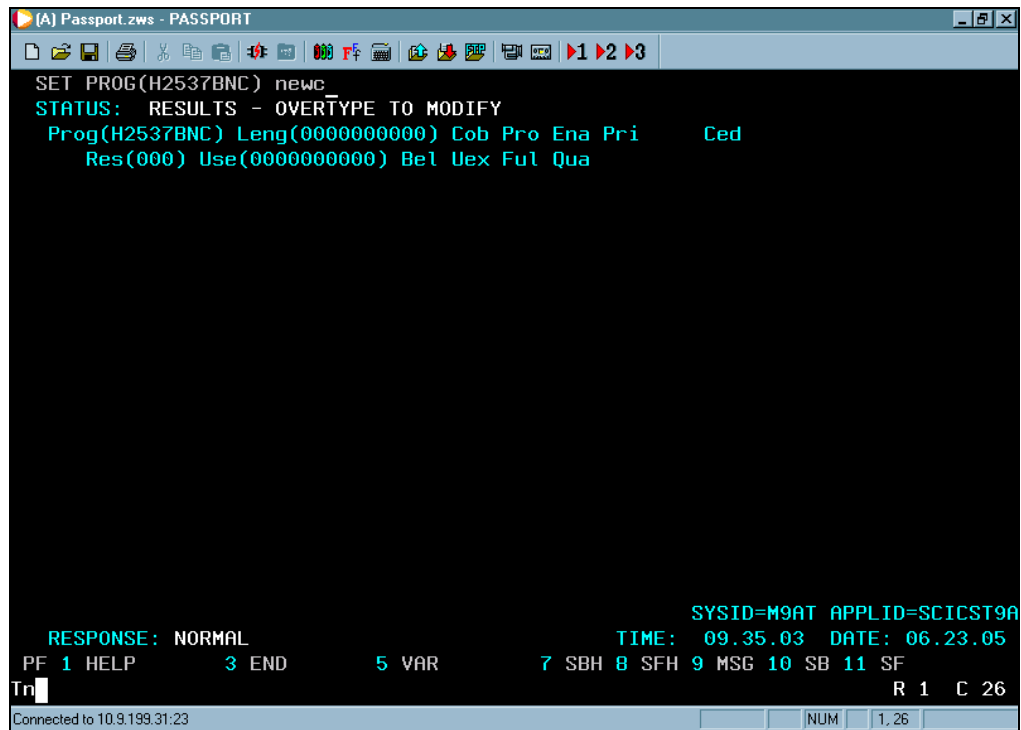
Figure 15: Typing command



The screenshot shows the output of the command 'CENT SET PROG(H2537BNC)'. The output includes the command itself, the status 'RESULTS - OVERTYPE TO MODIFY', and the program details: 'Prog(H2537BNC) Leng(0000000000) Cob Pro Ena Pri Ced' and 'Res(000) Use(0000000000) Bel Uex Ful Qua'. The status bar at the bottom indicates 'Connected to 10.9.199.31:23' and 'CAP NUM 1,22'.

Figure 16: Output

As you can see, the Length is (00000000). This means that the module is not present in the region.



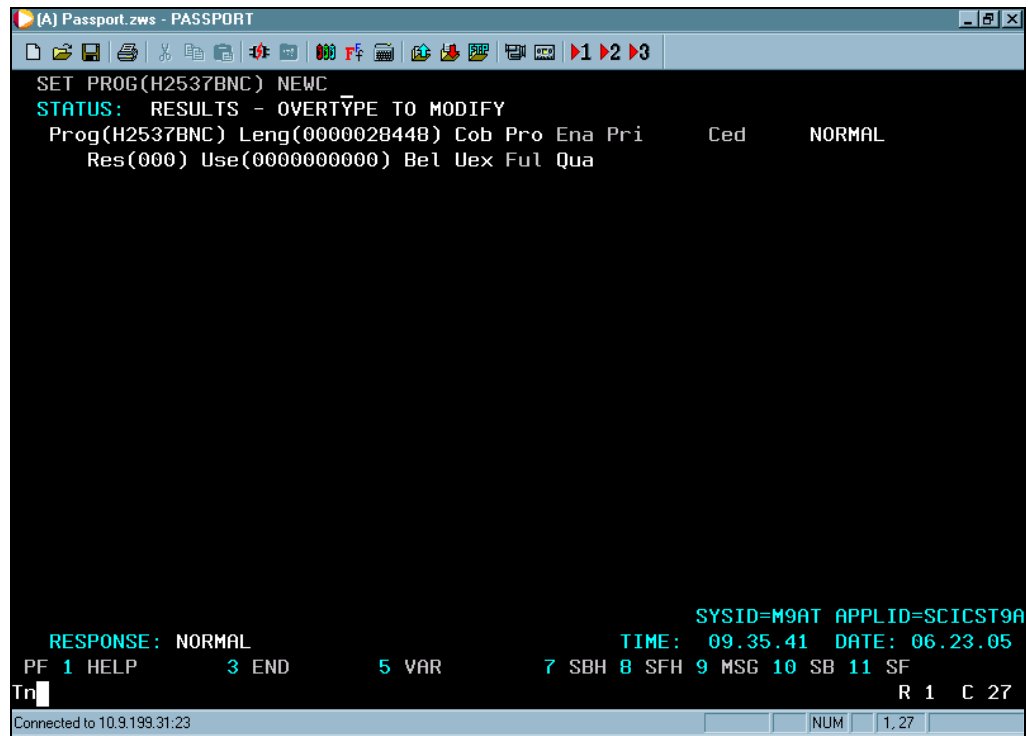
```

(A) Passport.zws - PASSPORT
SET PROG(H2537BNC) newc
STATUS: RESULTS - OVERTYPE TO MODIFY
  Prog(H2537BNC) Leng(0000000000) Cob Pro Ena Pri      Ced
    Res(000) Use(0000000000) Bel Uex Ful Qua

RESPONSE: NORMAL                                SYSID=M9AT APPLID=SCICST9A
TIME: 09.35.03 DATE: 06.23.05
PF 1 HELP      3 END      5 VAR      7 SBH 8 SFH 9 MSG 10 SB 11 SF
Tn
Connected to 10.9.199.31:23
NUM 1,26
  
```

Figure 17: Output

Suppose you type the command **NEWC**.



```

(A) Passport.zws - PASSPORT
SET PROG(H2537BNC) NEWC
STATUS: RESULTS - OVERTYPE TO MODIFY
  Prog(H2537BNC) Leng(0000028448) Cob Pro Ena Pri    Ced    NORMAL
    Res(000) Use(0000000000) Bel Uex Ful Qua

RESPONSE: NORMAL
TIME: 09.35.41 DATE: 06.23.05
PF 1 HELP 3 END 5 VAR 7 SBH 8 SFH 9 MSG 10 SB 11 SF
Tn R 1 C 27
Connected to 10.9.199.31:23 NUM 1,27

```

Figure 18: Output

The Length changes indicating that the module is now present in the region.

Appendix A-4: CEDF

CEDF is the Execution Diagnostic Facility.

Step 1: Type **CEDF** on the screen, and press **Enter** key.

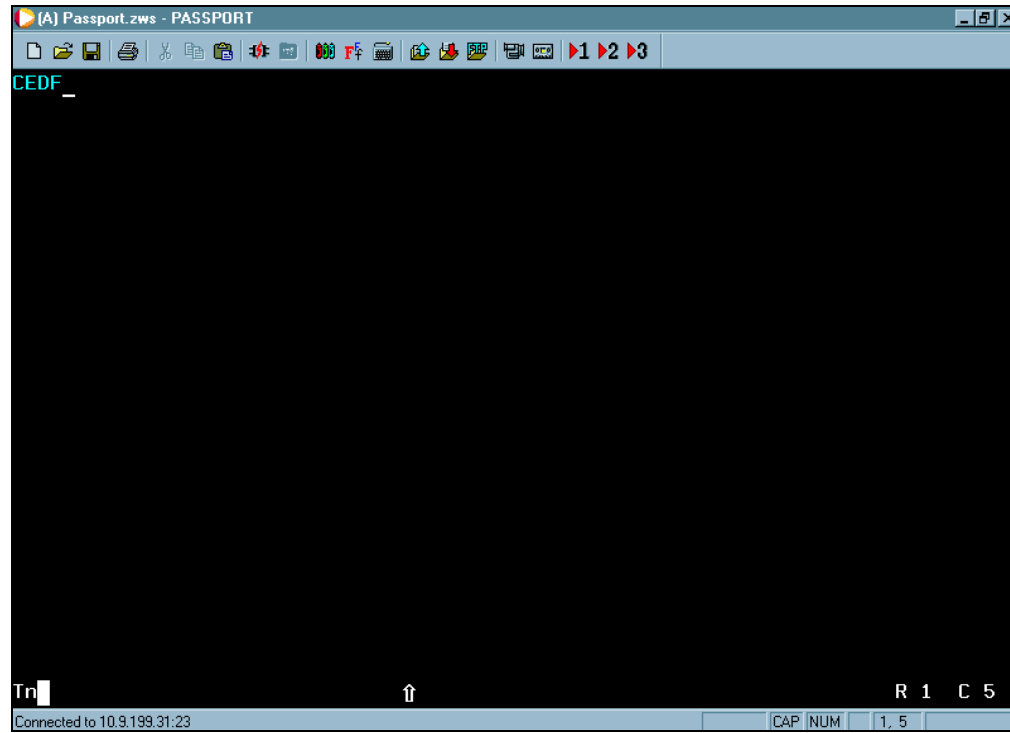


Figure 19: Typing the CEDF command

Output will be as shown below.

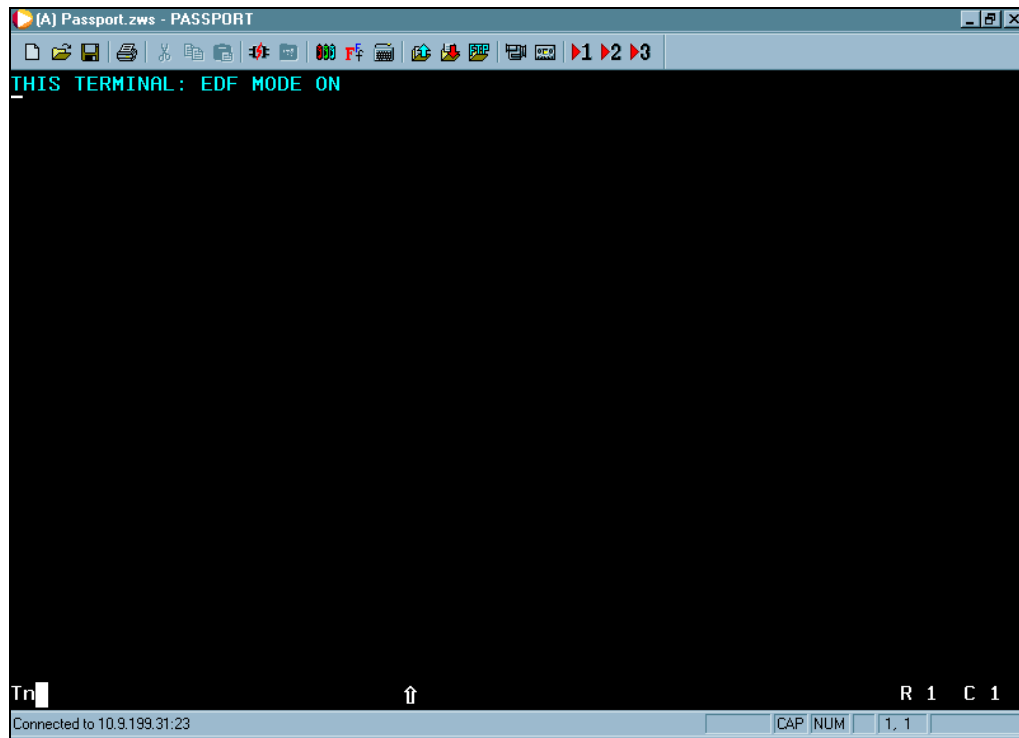


Figure 20: Output

Step 2: Type the transaction to spawn, and press **Enter** key.

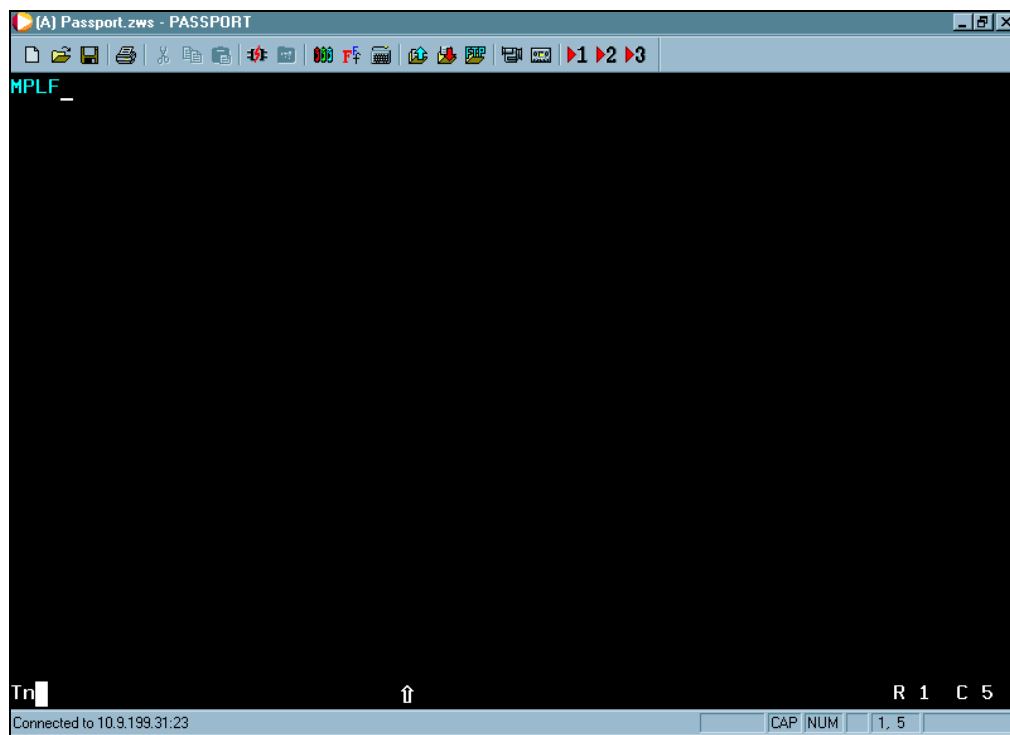


Figure 21: Typing the command

Output will be as shown below:

```

(A) Passport.zws - PASSPORT
TRANSACTION: MPLF PROGRAM: H2550I01 TASK: 0003966 APPLID: SCICST9A DISPLAY: 00
STATUS: PROGRAM INITIATION

EIBTIME      = 94324
EIBDATE      = 0105174
EIBTRNID     = 'MPLF'
EIBTASKN     = 3966
EIBTRMID     = 'a002'

EIBCPOSN     = 4
EIBCALEN     = 0
EIBRID       = X'7D'
EIBFN        = X'0000'
EIBRCODE     = X'000000000000'
EIBDS        = '.....'
+ EIBREQID    = '.....'

ENTER: CONTINUE
PF1 : UNDEFINED      PF2 : SWITCH HEX/CHAR    PF3 : END EDF SESSION
PF4 : SUPPRESS DISPLAYS PF5 : WORKING STORAGE    PF6 : USER DISPLAY
PF7 : SCROLL BACK    PF8 : SCROLL FORWARD    PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY PF11: EIB DISPLAY        PF12: UNDEFINED

Tn
↑
R 1 C 1
Connected to 10.9.199.31:23 CAP NUM 1, 1

```

Figure 22: Output

Create Maps using CA-Realia Simulator

Goals

- Learn to create, edit and compile Maps

Lab Setup

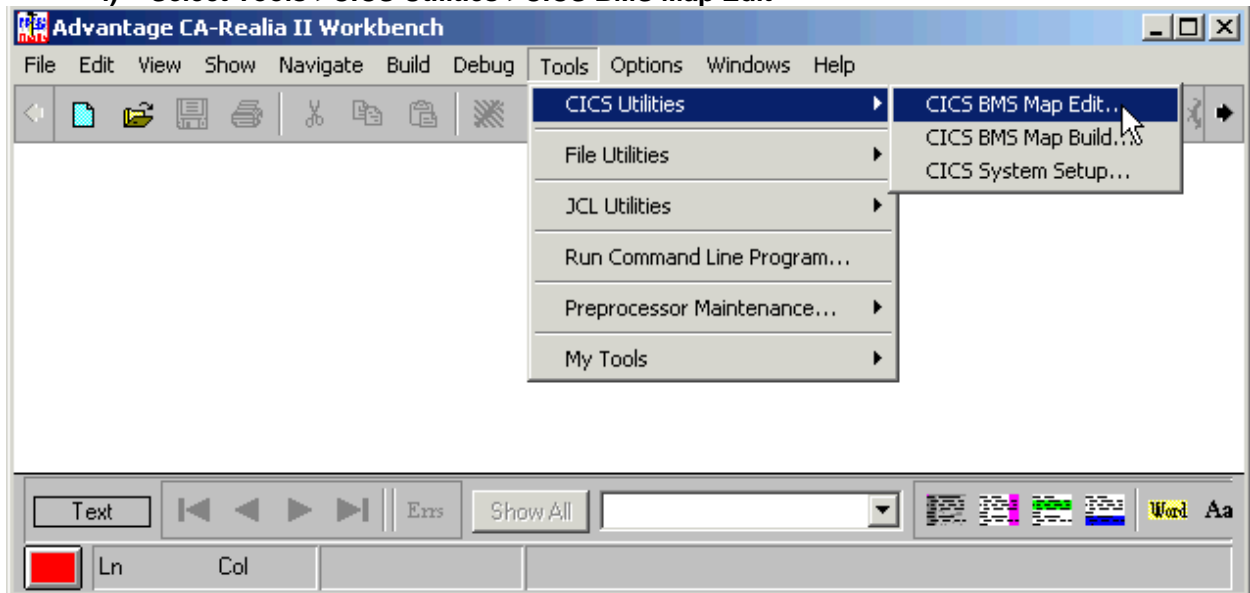
Successful installation of Advantage CA-Realia WorkBench 3.2 IDE

I. Start the WorkBench IDE

- i) Select Start->Programs->Computer Associates->Advantage->CA-Realia->WorkBench ID

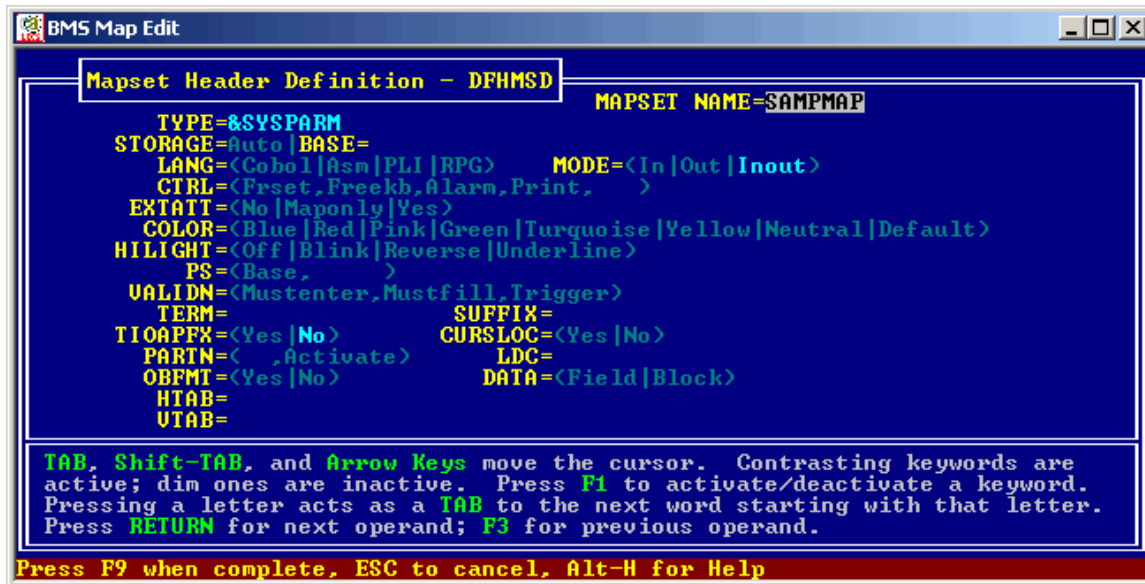
II. Create the Map

- i) Select Tools->CICS Utilities->CICS BMS Map Edit



- ii) Select F4 to Edit the Mapset Header definition

- (1) Navigate using tab keys
- (2) Select TYPE=&SYSPARM,CTRL->Freekb,MODE=Inout



```

Mapset Header Definition - DFHMSD
MAPSET NAME=SAMPMAP

TYPE=&SYSPARM
STORAGE=Auto |BASE=
LANG=<Cobol|Asm|PLI|RPG>    MODE=<In|Out|Inout>
CTRL=<Freset,Freekb,Alarm,Print,>
EXTATT=<No|Maponly|Yes>
COLOR=<Blue|Red|Pink|Green|Turquoise|Yellow|Neutral|Default>
HIGHLIGHT=<Off|Blink|Reverse|Underline>
PS=<Base,>
UALIDN=<Mustenter,Mustfill,Trigger>
TERM=
SUPFIX=
TIOAPFX=<Yes|No>    CURSLOC=<Yes|No>
PARTN=<,>Activate>    LDC=
OBFMT=<Yes|No>    DATA=<Field|Block>
HTAB=
UTAB=

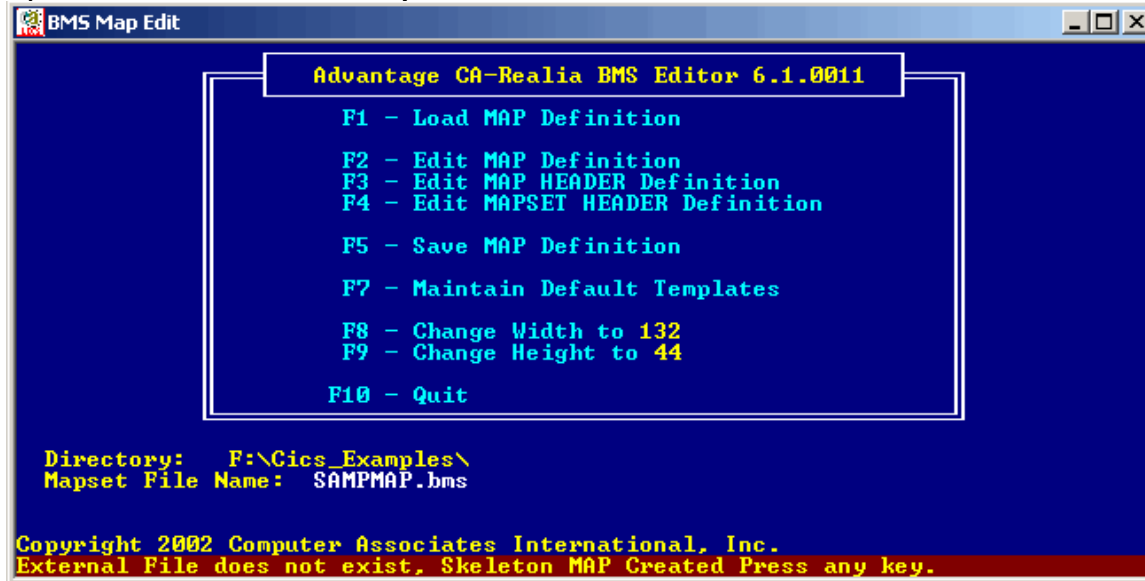
TAB, Shift-TAB, and Arrow Keys move the cursor. Contrasting keywords are
active; dim ones are inactive. Press F1 to activate/deactivate a keyword.
Pressing a letter acts as a TAB to the next word starting with that letter.
Press RETURN for next operand; F3 for previous operand.

Press F9 when complete, ESC to cancel, Alt-H for Help

```

(3) Select F9 to complete the definition of Map Header

iii) Select F3 to Edit the Map Header definition



```

Advantage CA-Realia BMS Editor 6.1.0011

F1 - Load MAP Definition
F2 - Edit MAP Definition
F3 - Edit MAP HEADER Definition
F4 - Edit MAPSET HEADER Definition

F5 - Save MAP Definition
F7 - Maintain Default Templates

F8 - Change Width to 132
F9 - Change Height to 44

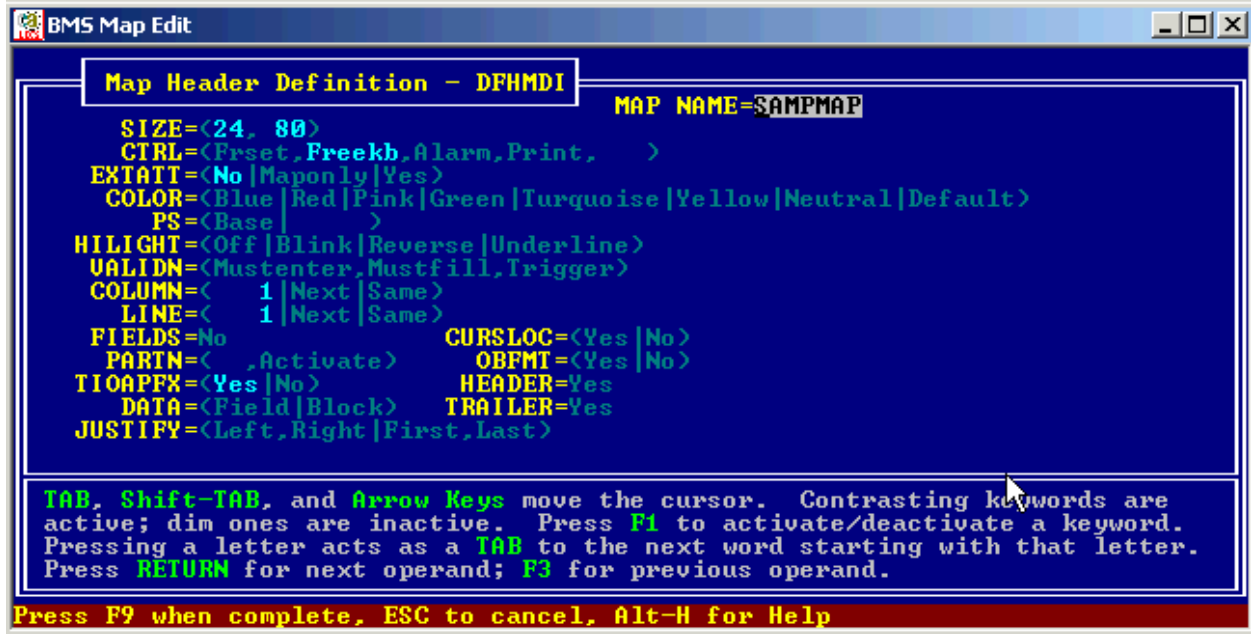
F10 - Quit

Directory: F:\Cics_Examples\
Mapset File Name: SAMPMAP.bms

Copyright 2002 Computer Associates International, Inc.
External File does not exist, Skeleton MAP Created Press any key.

```

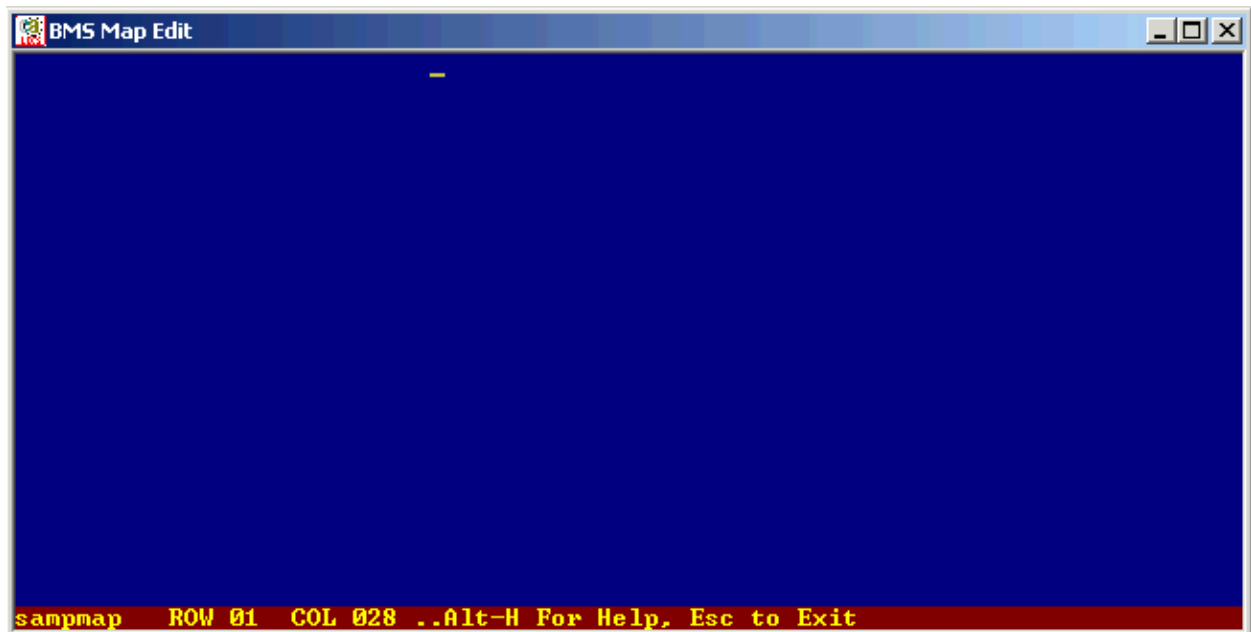
- (1) Navigate using tab keys
- (2) Select CTRL->Freekb,EXTATT->No,COLUMN->,LINE->1,TIOAPFX->Yes



(3) Select F9 to complete the definition of Mapset Header

iv) Select F2 to Edit the map definition

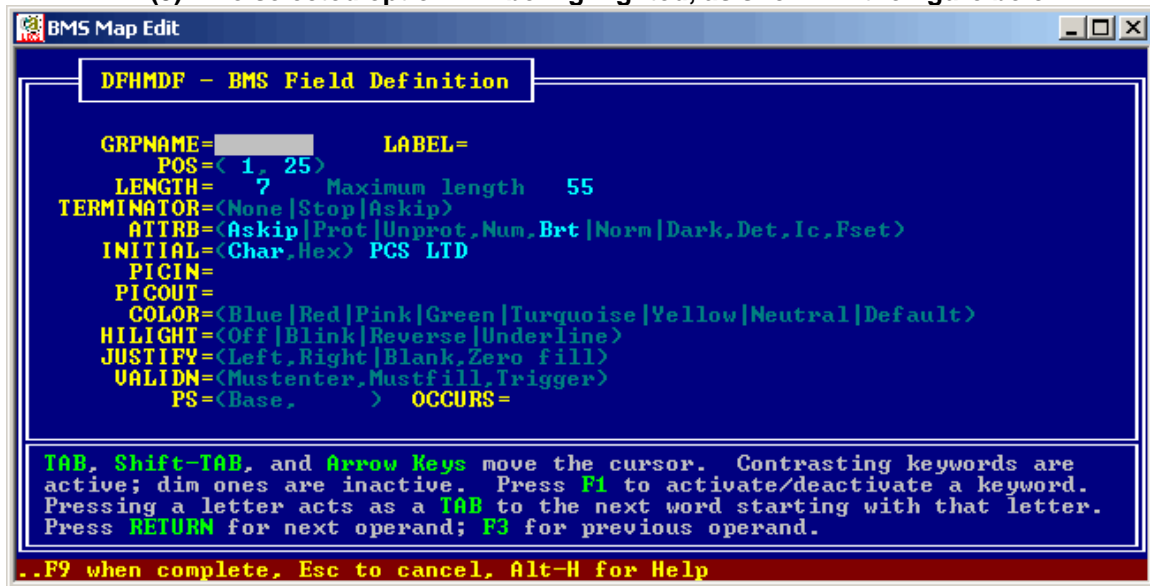
(1) Move the Arrow keys and position it at the point where the heading (constant) has to appear



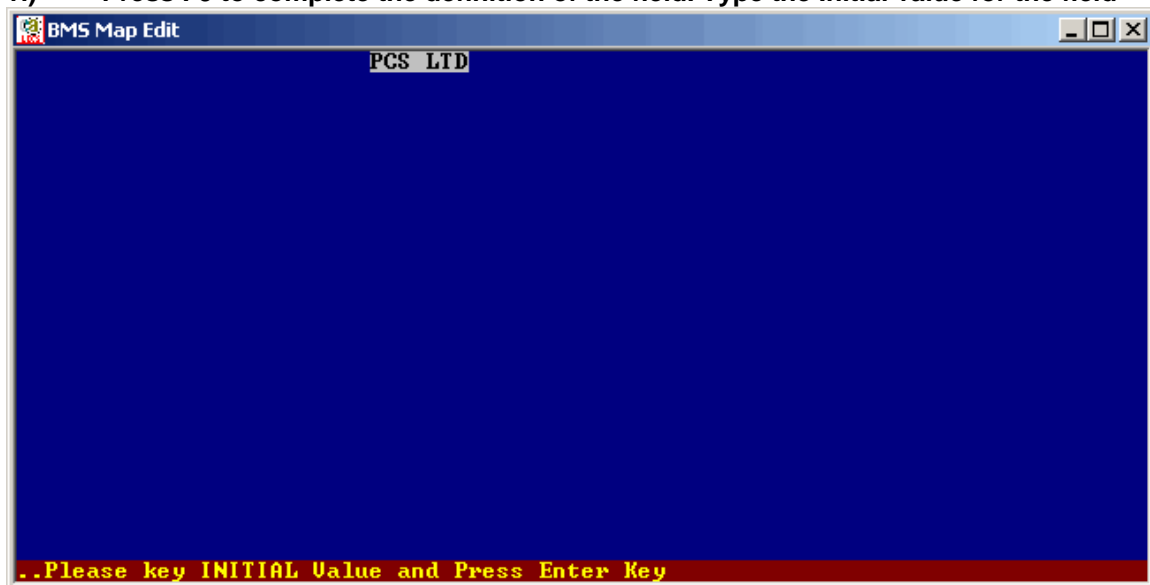
v) Press F2 to select the attributes of the field

- (1) Navigate using tab keys
- (2) To select individual attributes such as Brt, Askip press F1

(3) The selected option will be highlighted, as shown in the figure below

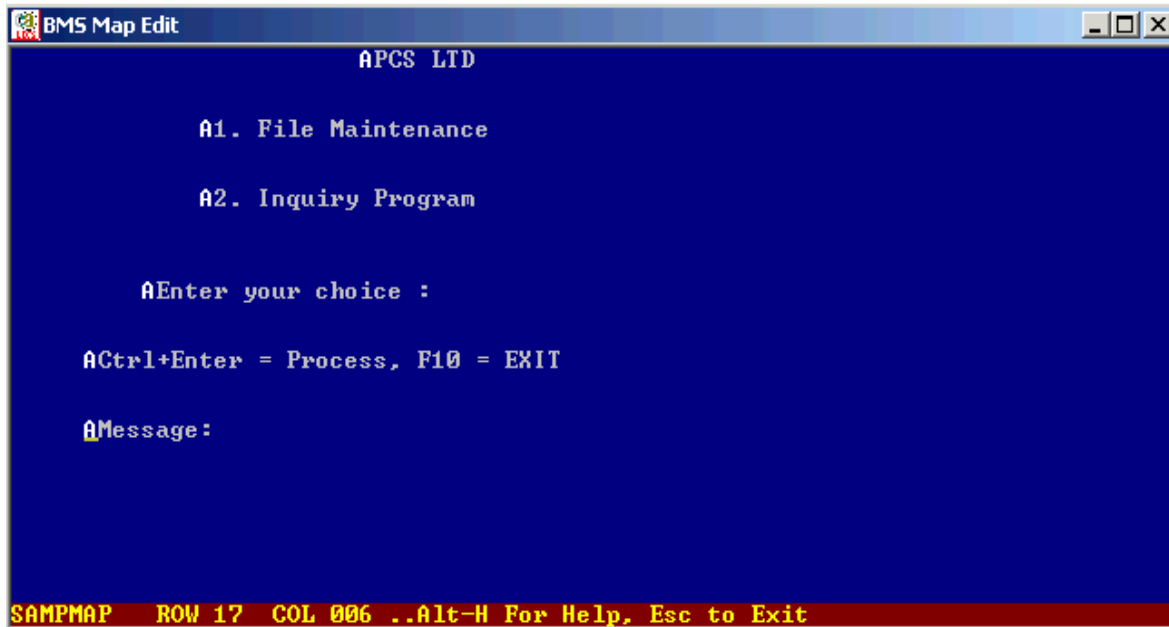


vi) Press F9 to complete the definition of the field. Type the Initial value for the field



<<to do>>

Similarly, create a Map with the following fields (constants)



```

BMS Map Edit
APCS LTD

A1. File Maintenance

A2. Inquiry Program

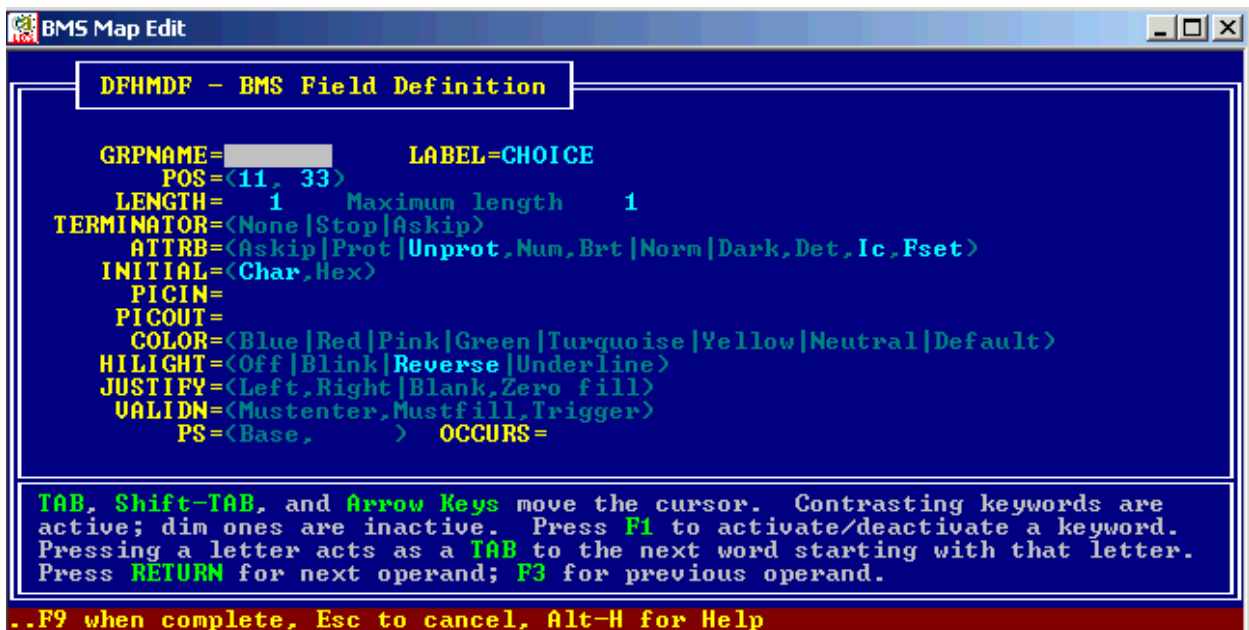
AEnter your choice :

ACtrl+Enter = Process, F10 = EXIT

AMessage:

SAMPMP ROW 17 COL 006 ..Alt-H For Help, Esc to Exit
  
```

(1) Create a Data entry field named choice with the following attributes



```

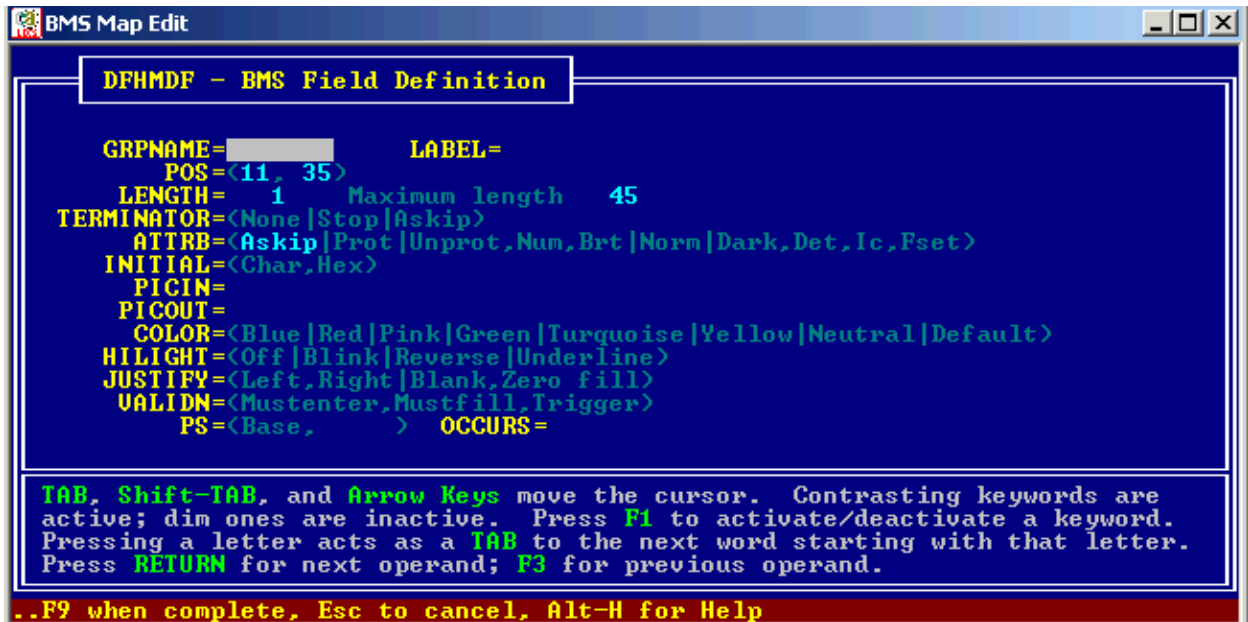
BMS Map Edit
DFHMDF - BMS Field Definition

GRPNAME=          LABEL=CHOICE
  POS=(11, 33)
  LENGTH= 1      Maximum length 1
  TERMINATOR=(None|Stop|Askip)
  ATTRB=(Askip|Prot|Unprot,Num,Brt|Norm|Dark,Det,Ic,Fset)
  INITIAL=(Char,Hex)
  PICIN=
  PICOUT=
  COLOR=(Blue|Red|Pink|Green|Turquoise|Yellow|Neutral|Default)
  HILIGHT=(Off|Blink|Reverse|Underline)
  JUSTIFY=(Left,Right|Blank,Zero fill)
  VALIDN=(Mustenter,Mustfill,Trigger)
  PS=(Base,      ) OCCURS=

TAB, Shift-TAB, and Arrow Keys move the cursor. Contrasting keywords are
active; dim ones are inactive. Press F1 to activate/deactivate a keyword.
Pressing a letter acts as a TAB to the next word starting with that letter.
Press RETURN for next operand; F3 for previous operand.

..F9 when complete, Esc to cancel, Alt-H for Help
  
```

(2) Create a Stopper field for the choice field, with the following attributes



```

BMS Map Edit

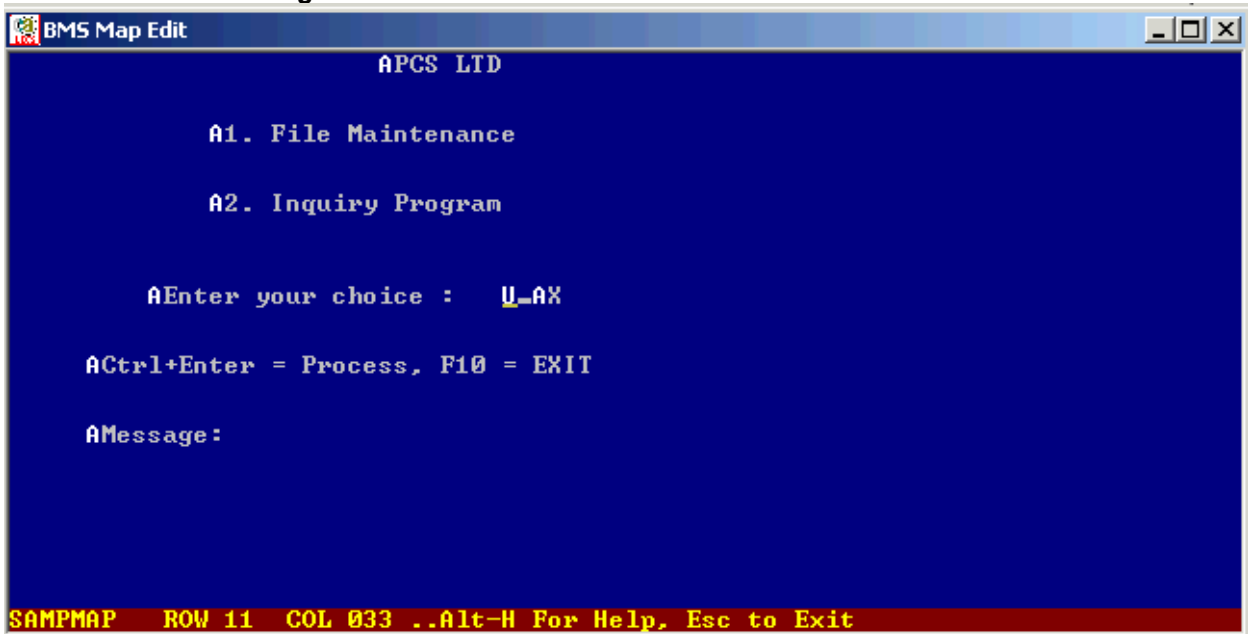
DFHMDF - BMS Field Definition

GRPNAME=      LABEL=
  POS=<11, 35>
  LENGTH= 1    Maximum length 45
TERMINATOR=<None|Stop|Askip>
ATTRB=<Askip|Prot|Unprot,Num,Brt|Norm|Dark,Det,Ic,Fset>
INITIAL=<Char,Hex>
PICIN=
PICOUT=
  COLOR=<Blue|Red|Pink|Green|Turquoise|Yellow|Neutral|Default>
HIGHLIGHT=<Off|Blink|Reverse|Underline>
JUSTIFY=<Left,Right|Blank,Zero fill>
VALIDN=<Mustenter,Mustfill,Trigger>
  PS=<Base,      > OCCURS=

TAB, Shift-TAB, and Arrow Keys move the cursor. Contrasting keywords are
active; dim ones are inactive. Press F1 to activate/deactivate a keyword.
Pressing a letter acts as a TAB to the next word starting with that letter.
Press RETURN for next operand; F3 for previous operand.

..F9 when complete, Esc to cancel, Alt-H for Help
  
```

- (3) The map with the choice field and the stopper field should resemble the following screen



```

BMS Map Edit

APCS LTD

A1. File Maintenance

A2. Inquiry Program

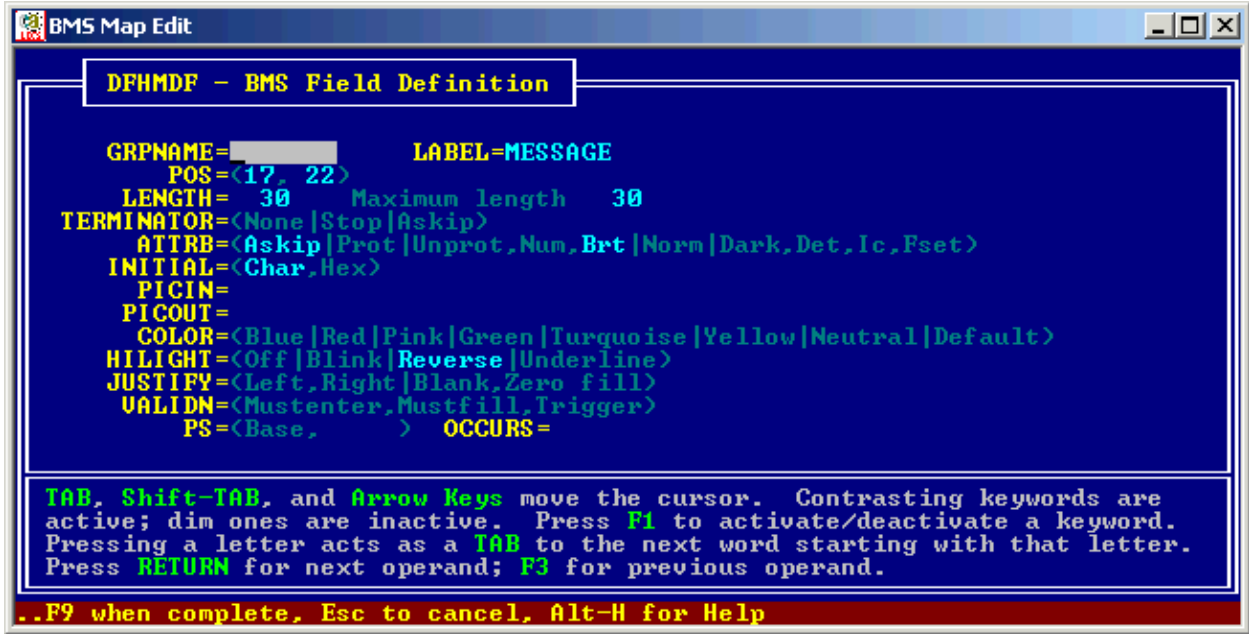
AEnter your choice :  U=AX

ACtrl+Enter = Process, F10 = EXIT

AMessage:

SAMPMAP  ROW 11  COL 033 ..Alt-H For Help, Esc to Exit
  
```

- (4) Create the Message field (Display-only) with the following attributes



```

DFHMDf - BMS Field Definition


GRPNAME=          LABEL=MESSAGE
  POS=(17, 22)
  LENGTH= 30      Maximum length  30
  TERMINATOR=(None|Stop|Askip)
  ATTRB=(Askip|Prot|Unprot,Num,Brt|Norm|Dark,Det,Ic,Fset)
  INITIAL=(Char,Hex)
  PICIN=
  PICOUT=
  COLOR=(Blue|Red|Pink|Green|Turquoise|Yellow|Neutral|Default)
  HILIGHT=(Off|Blink|Reverse|Underline)
  JUSTIFY=(Left,Right|Blank,Zero fill)
  VALIDM=(Mustenter,Mustfill,Trigger)
  PS=(Base,      )  OCCURS=

TAB, Shift-TAB, and Arrow Keys move the cursor. Contrasting keywords are
active; dim ones are inactive. Press F1 to activate/deactivate a keyword.
Pressing a letter acts as a TAB to the next word starting with that letter.
Press RETURN for next operand; F3 for previous operand.

..F9 when complete, Esc to cancel, Alt-H for Help
  
```

- (5) Press F9 to complete the definition of the Message field
- (6) Define a stopper field for the Message field

- (7) Define a Dummy field with the following attributes



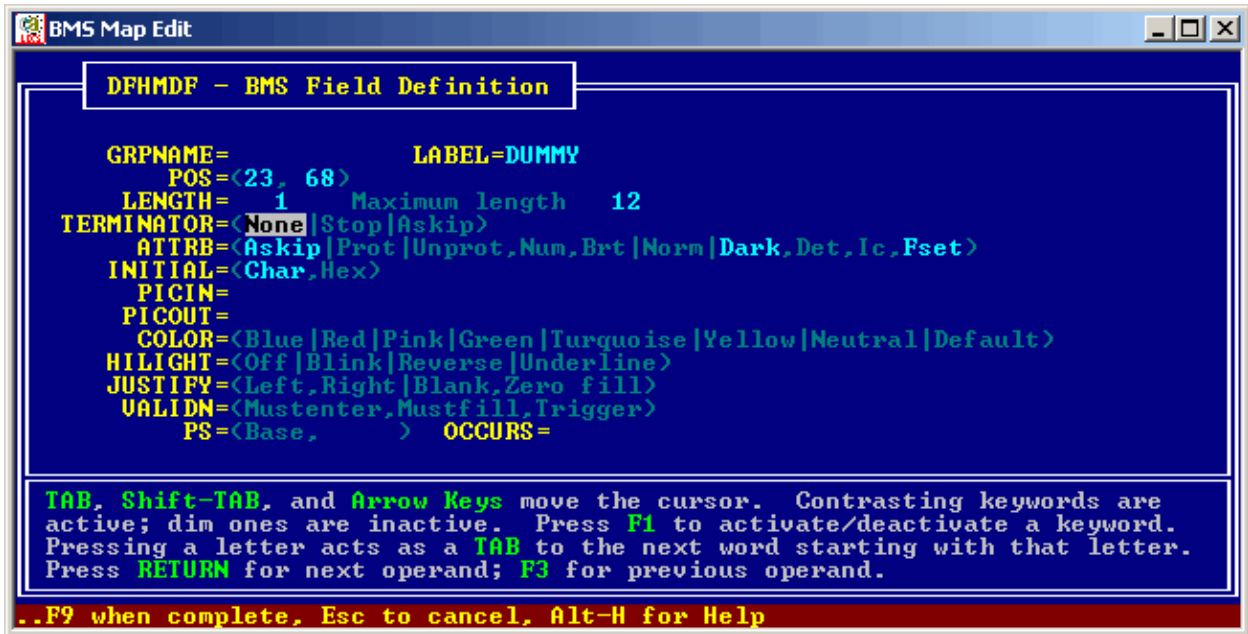
```

DFHMDf - BMS Field Definition

GRPNAME=          LABEL=
  POS=(23, 68)
  LENGTH= 1      Maximum length  12
  TERMINATOR=(None|Stop|Askip)
  ATTRB=(Askip|Prot|Unprot,Num,Brt|Norm|Dark,Det,Ic,Fset)
  INITIAL=(Char,Hex)
  PICIN=
  PICOUT=
  COLOR=(Blue|Red|Pink|Green|Turquoise|Yellow|Neutral|Default)
  HILIGHT=(Off|Blink|Reverse|Underline)
  JUSTIFY=(Left,Right|Blank,Zero fill)
  VALIDM=(Mustenter,Mustfill,Trigger)
  PS=(Base,      )  OCCURS=

TAB, Shift-TAB, and Arrow Keys move the cursor. Contrasting keywords are
active; dim ones are inactive. Press F1 to activate/deactivate a keyword.
Pressing a letter acts as a TAB to the next word starting with that letter.
Press RETURN for next operand; F3 for previous operand.

..F9 when complete, Esc to cancel, Alt-H for Help
  
```



```

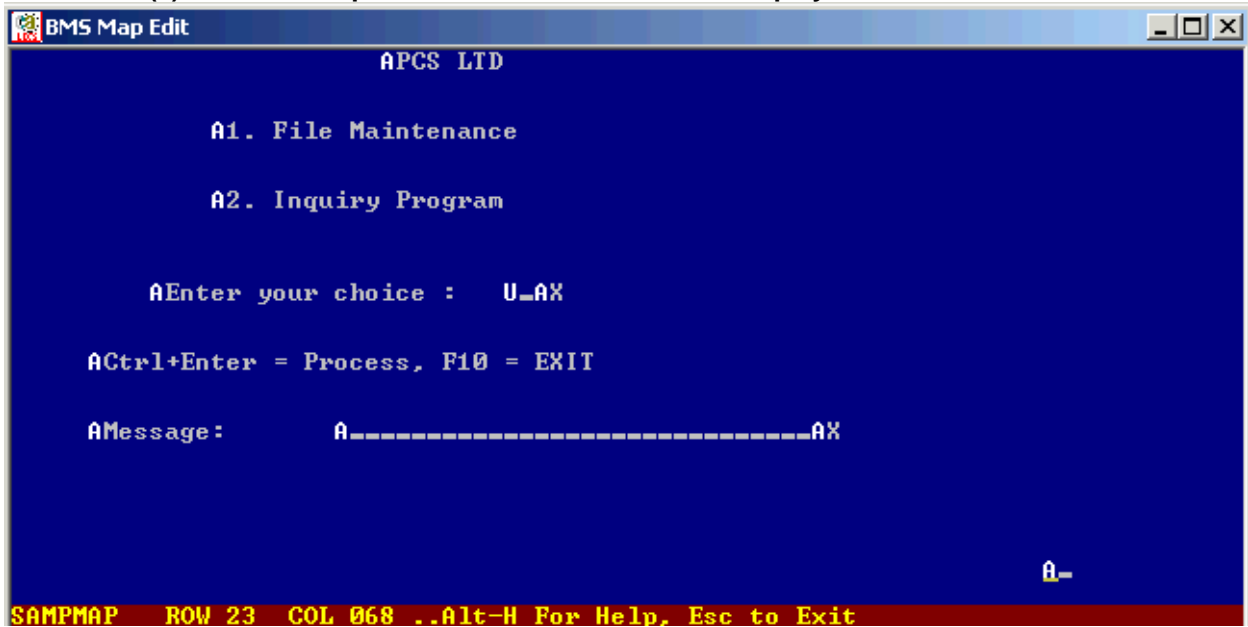
DFHMDf - BMS Field Definition

GRPNAME=          LABEL=DUMMY
  POS=(23, 68)
  LENGTH= 1      Maximum length  12
TERMINATOR=(None|Stop|Askip)
ATTRB=(Askip|Prot|Unprot,Num,Brt|Norm|Dark,Det,Ic,Fset)
INITIAL=(Char,Hex)
PICIN=
PICOUT=
  COLOR=(Blue|Red|Pink|Green|Turquoise|Yellow|Neutral|Default)
HILIGHT=(Off|Blink|Reverse|Underline)
JUSTIFY=(Left,Right|Blank,Zero fill)
VALIDM=(Mustenter,Mustfill,Trigger)
  PS=(Base,      )  OCCURS=

TAB, Shift-TAB, and Arrow Keys move the cursor.  Contrasting keywords are
active; dim ones are inactive.  Press F1 to activate/deactivate a keyword.
Pressing a letter acts as a TAB to the next word starting with that letter.
Press RETURN for next operand; F3 for previous operand.

..F9 when complete, Esc to cancel, Alt-H for Help
  
```

(8) The final map should resemble the screen displayed below



```

BMS Map Edit

      APCS LTD

      A1. File Maintenance

      A2. Inquiry Program

      AEnter your choice :    U_AX

      ACtrl+Enter = Process, F10 = EXIT

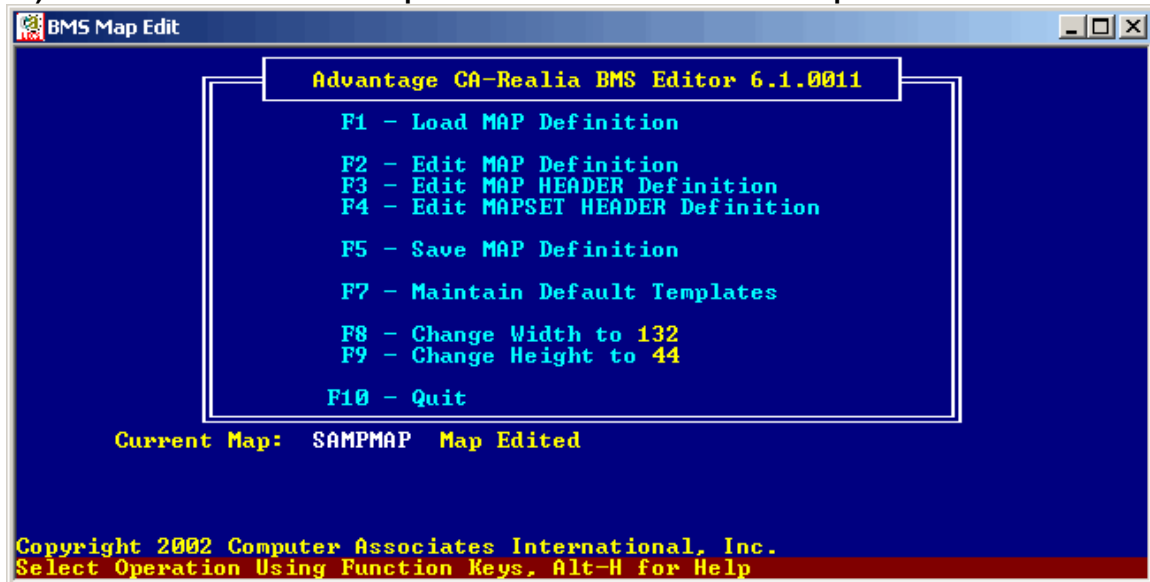
      AMessage:      A-----AX

                                     A_

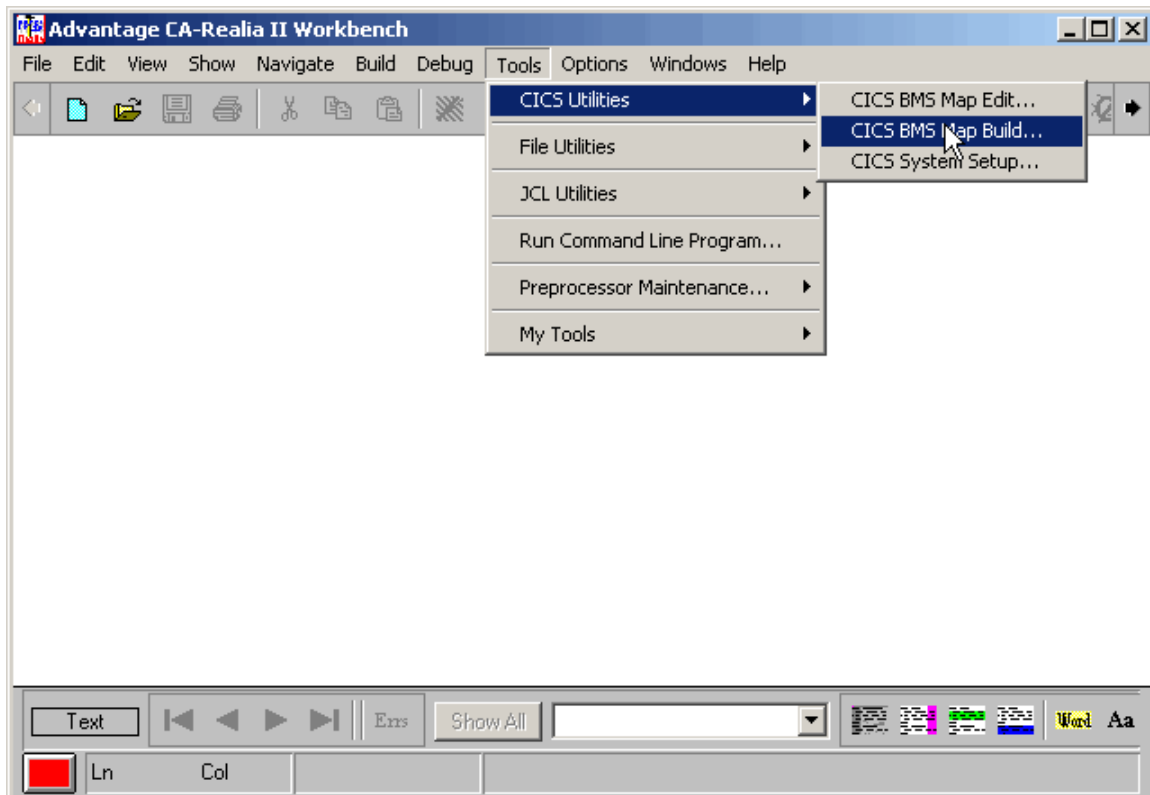
SAMP MAP   ROW 23  COL 068 ..Alt-H For Help, Esc to Exit
  
```

(9) Press ESC to complete the definition of the map

vii) Press F5 to save the Map definition and F10 to exit the Map



III. Compile the Map



<<to do>>

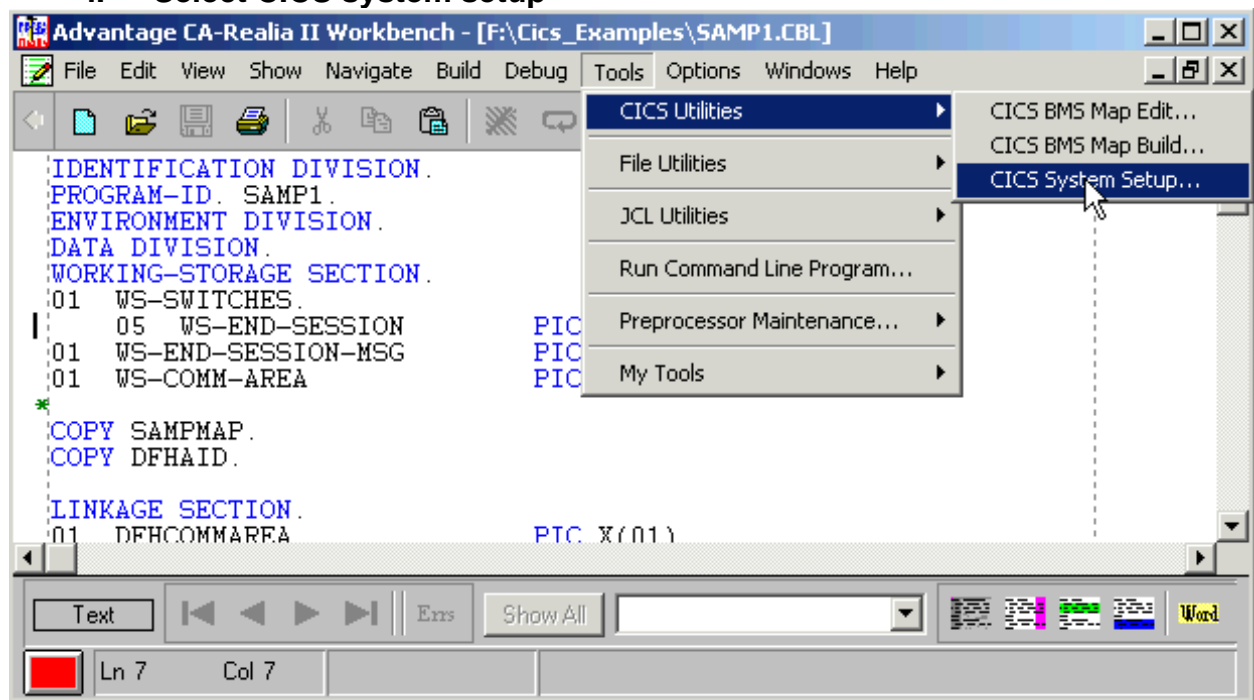
- 1) Select the Mapset from the appropriate directory.
- 2) View the additional files generated, after successful compilation of the mapset.
- 3) View the sampmap.bms,sampmap.cbl file in any editor

Translating and compiling COBOL programs using Ca-Realia

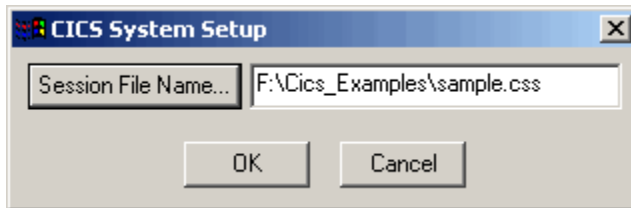
I. Code the following COBOL program in the Workbench IDE(Editor).

II. Add appropriate entries in the PPT and PCT for the COBOL Program and the Mapset.

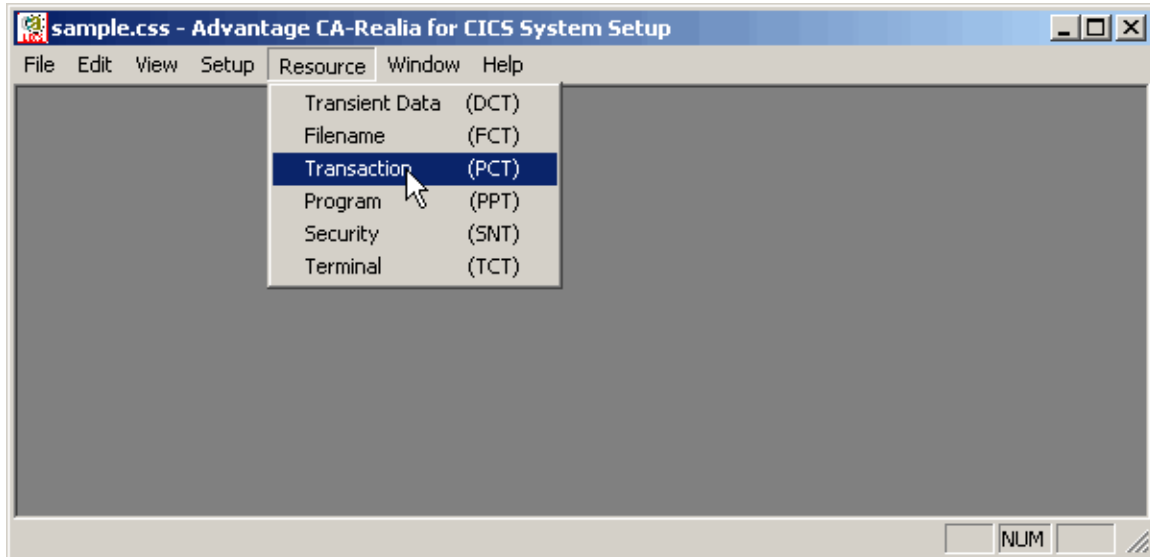
i. Select CICS system setup



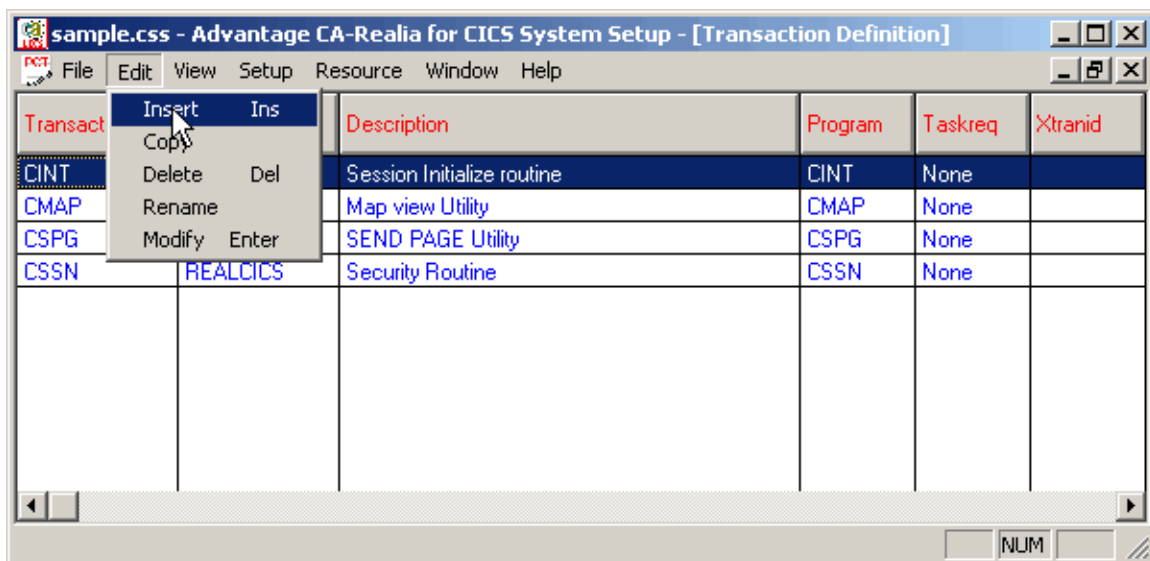
ii. Select the CSS file from the appropriate folder



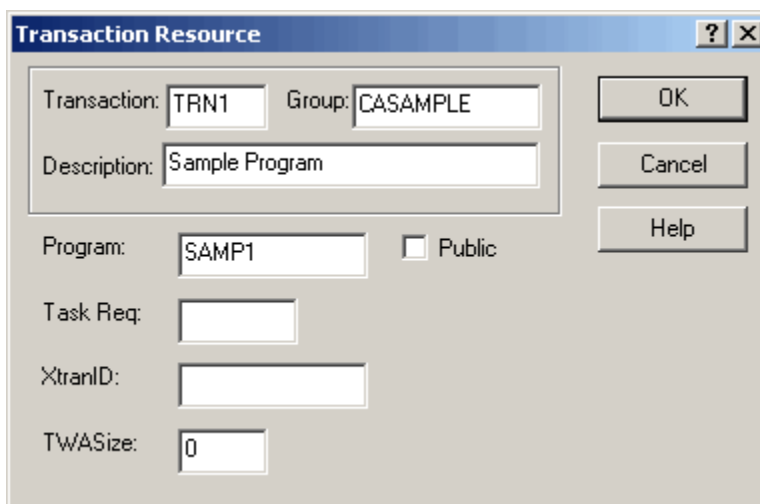
- ii. **Select Resource->Transaction (PCT), to add entries in the Program Control Table**



- iii. **Select Edit->Insert to add entries in the PCT**



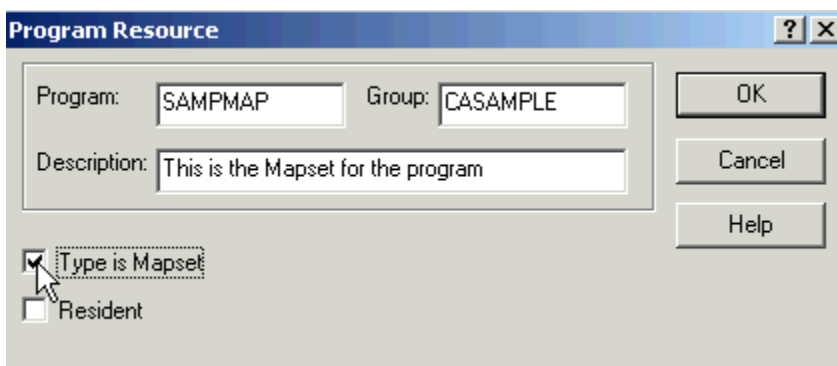
- iv. **Add the following entry to associate the Transaction with the program.**



The **Transaction Resource** dialog box contains the following fields and controls:

- Transaction:** TRN1
- Group:** CASAMPLE
- Description:** Sample Program
- Program:** SAMP1
- Public:** ☐
- Task Req:** (empty field)
- XtranID:** (empty field)
- TWASize:** 0
- Buttons:** OK, Cancel, Help

- v. **Select Resource->Program to add entries in the PPT**
vi. **Add the following entries to add the program and the Mapset in the PPT.**

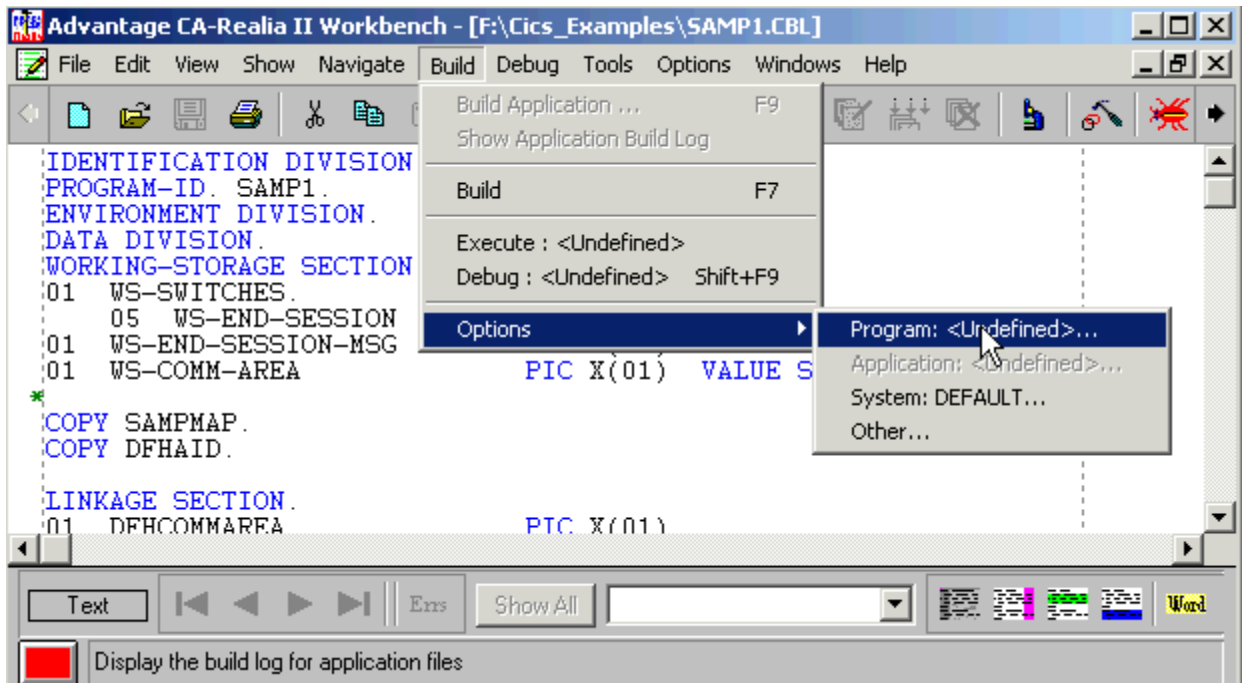


The **Program Resource** dialog box contains the following fields and controls:

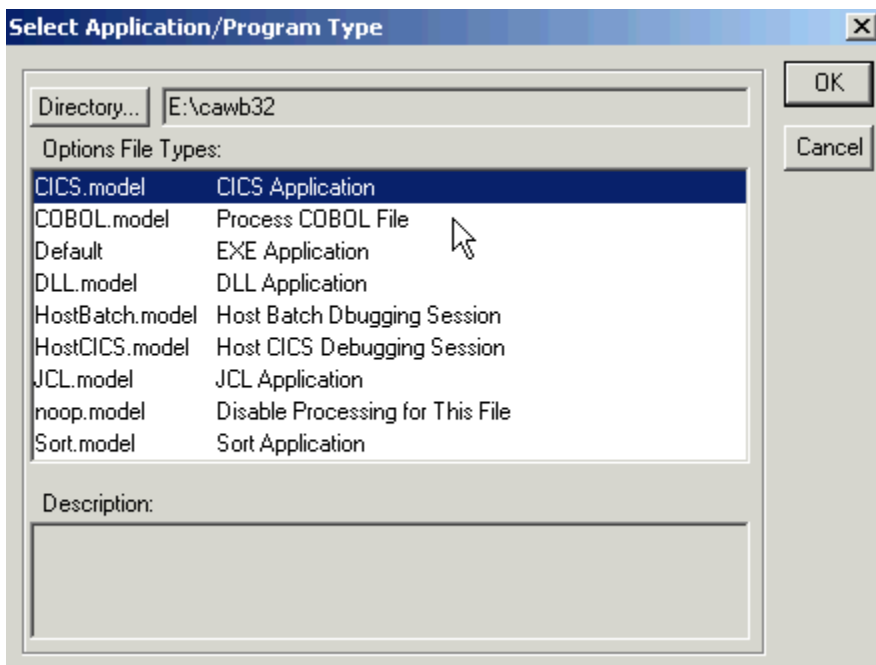
- Program:** SAMPMAP
- Group:** CASAMPLE
- Description:** This is the Mapset for the program
- Type is Mapset:** ☒ (indicated by a mouse cursor)
- Resident:** ☐
- Buttons:** OK, Cancel, Help

III. Build the Cobol Program.

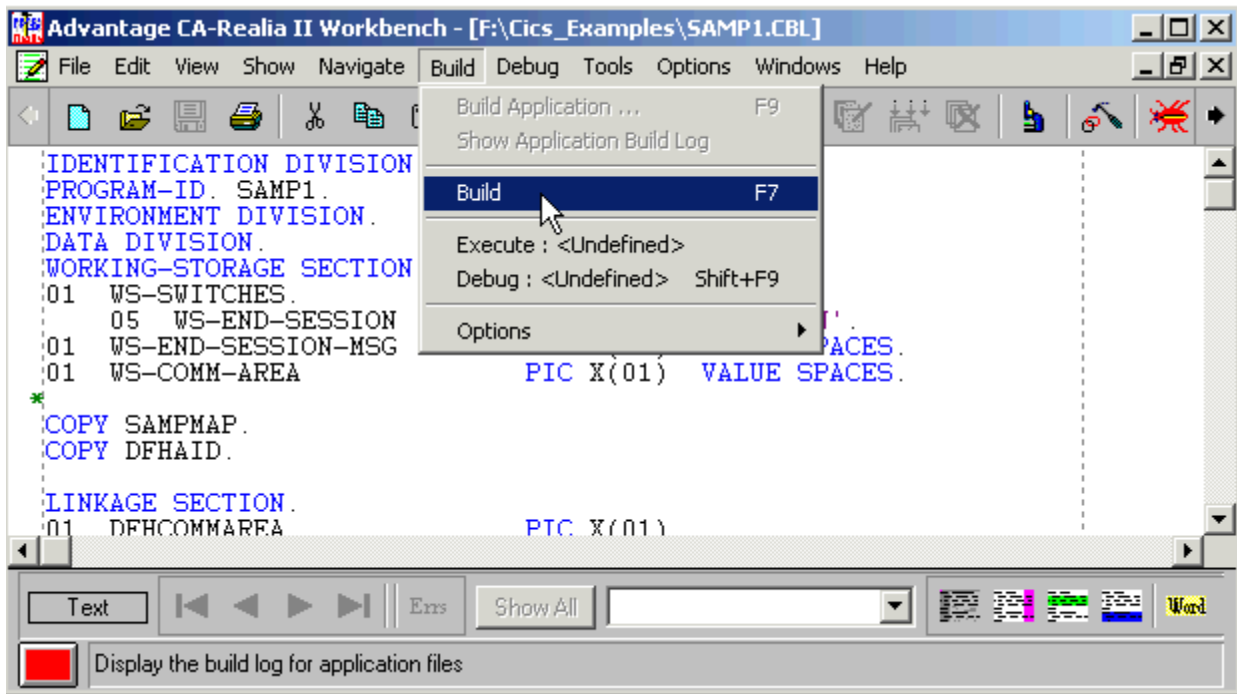
1. **Select Build->Options->Program:<Undefined>...**



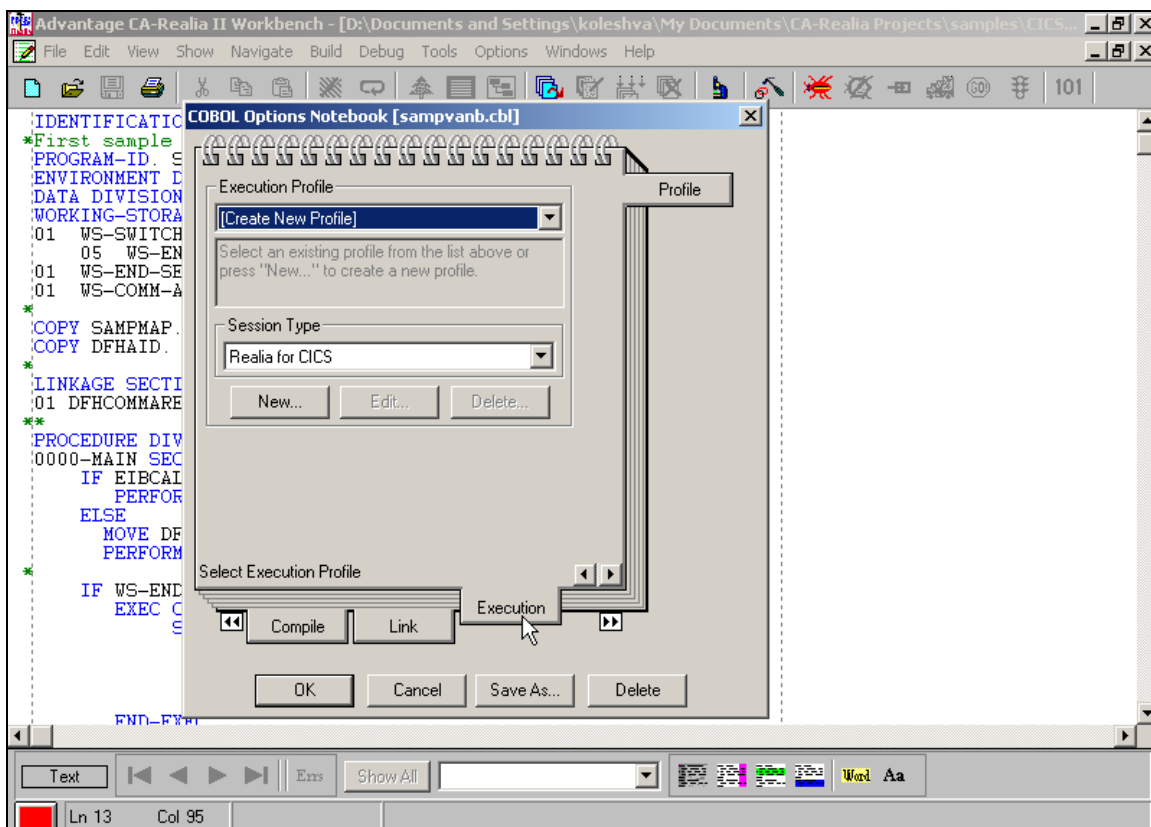
2. Select CICS Application



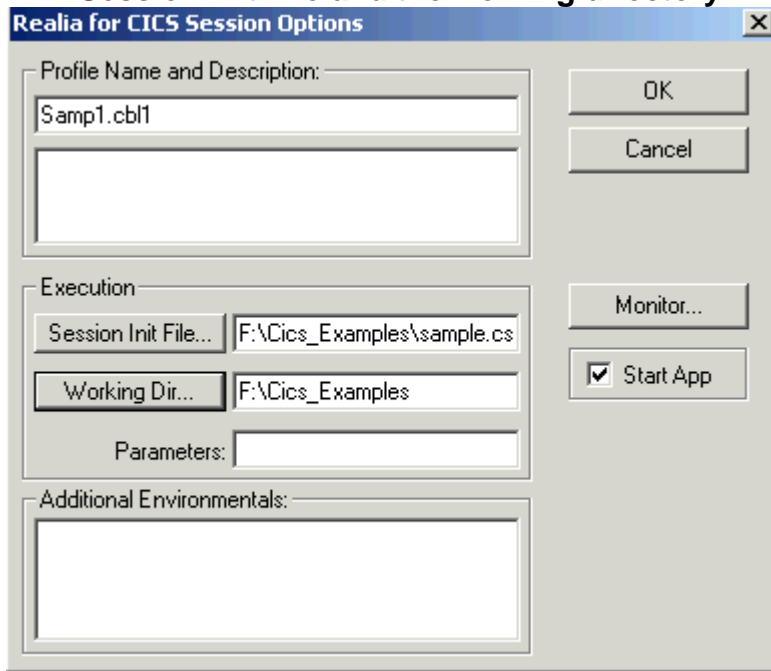
3. Build the COBOL program



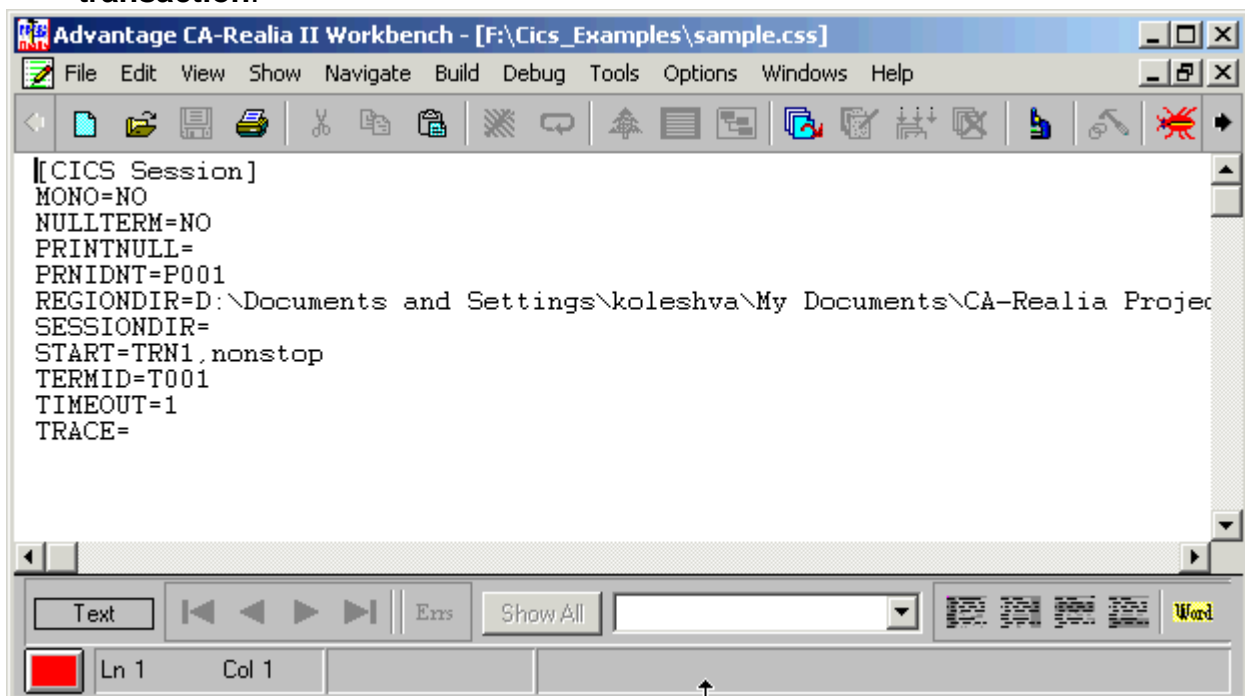
4. Select Build->Options->Program:<Undefined>...



5. On the above screen select New... and then select the Session Init File and the working directory.

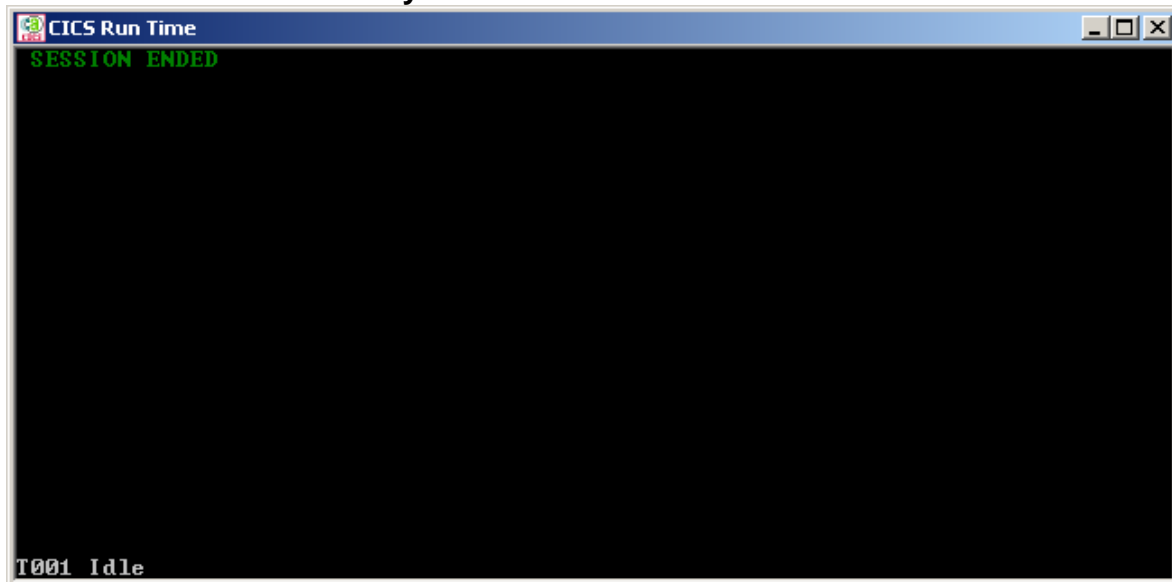


6. Open the file sample.css in any editor and enter TRN1 as the transaction.



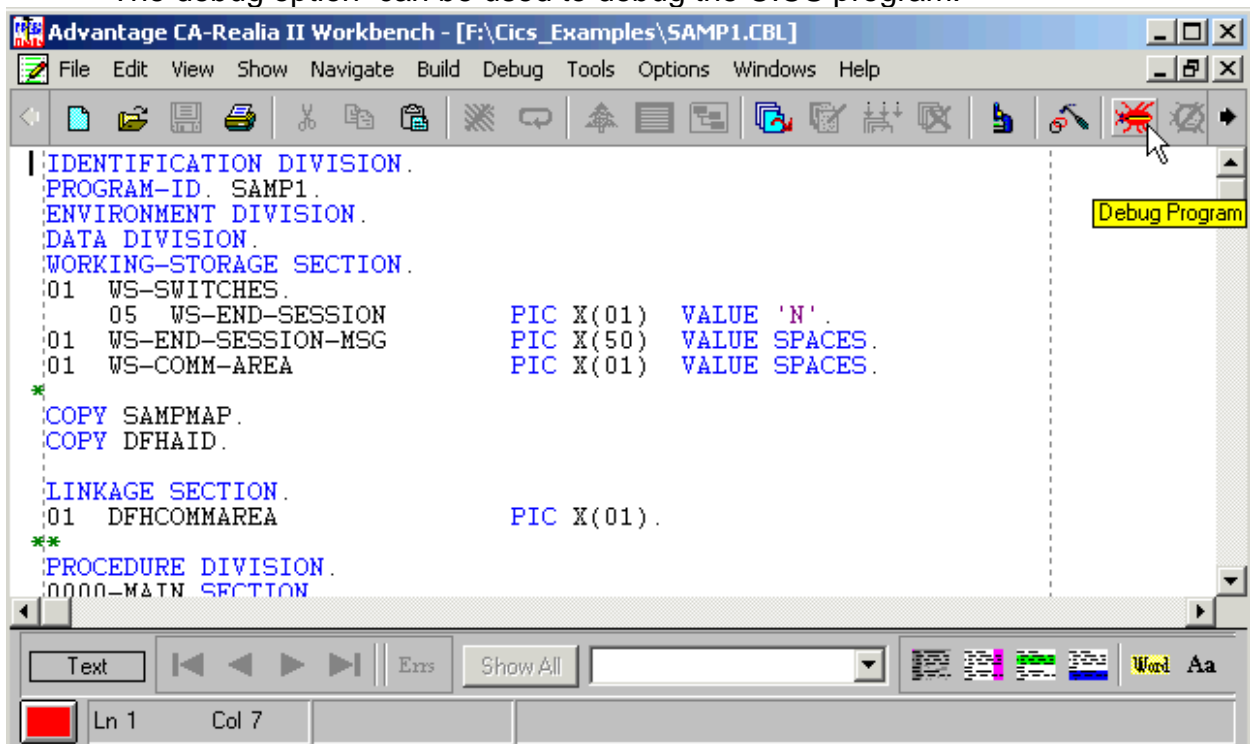
- 2) Press an invalid AID key and note the message

9. Press the F10 key to terminate the session



IV. Debug the CICS Program

The debug option can be used to debug the CICS program.



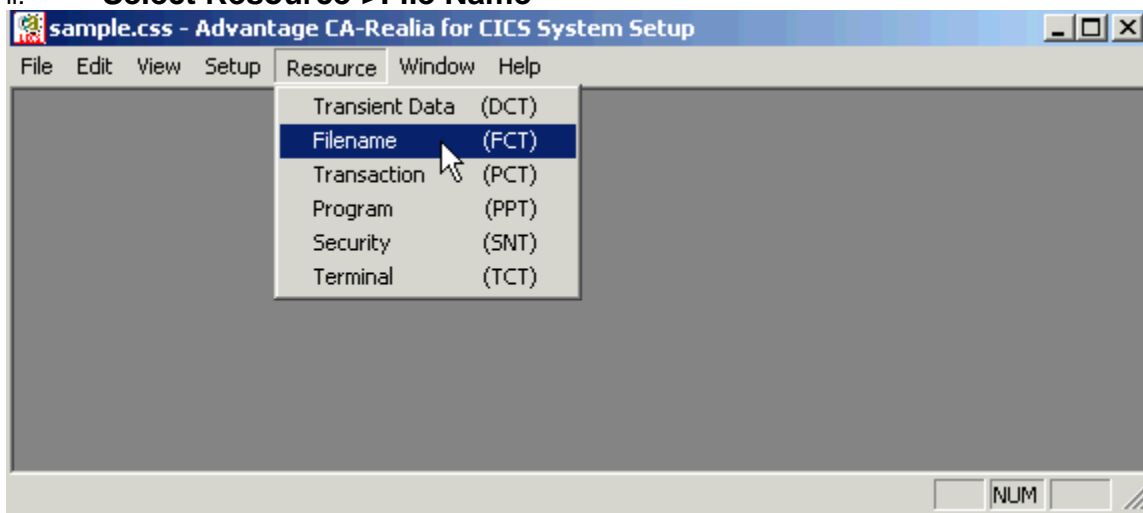
File Handling using CA-Realia

I. Add the dataset to the FCT.

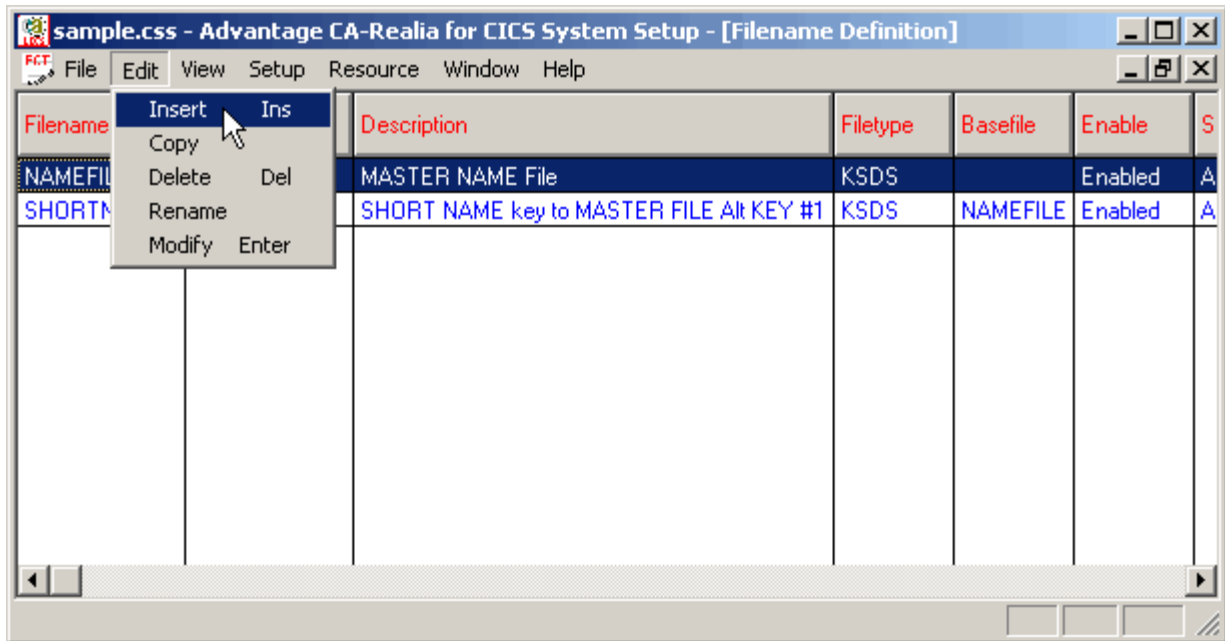
i. <<to do >>

1. **Select Tools->CICS utilities-> CICS system setup...**
2. **Select sample.css from the appropriate folder**

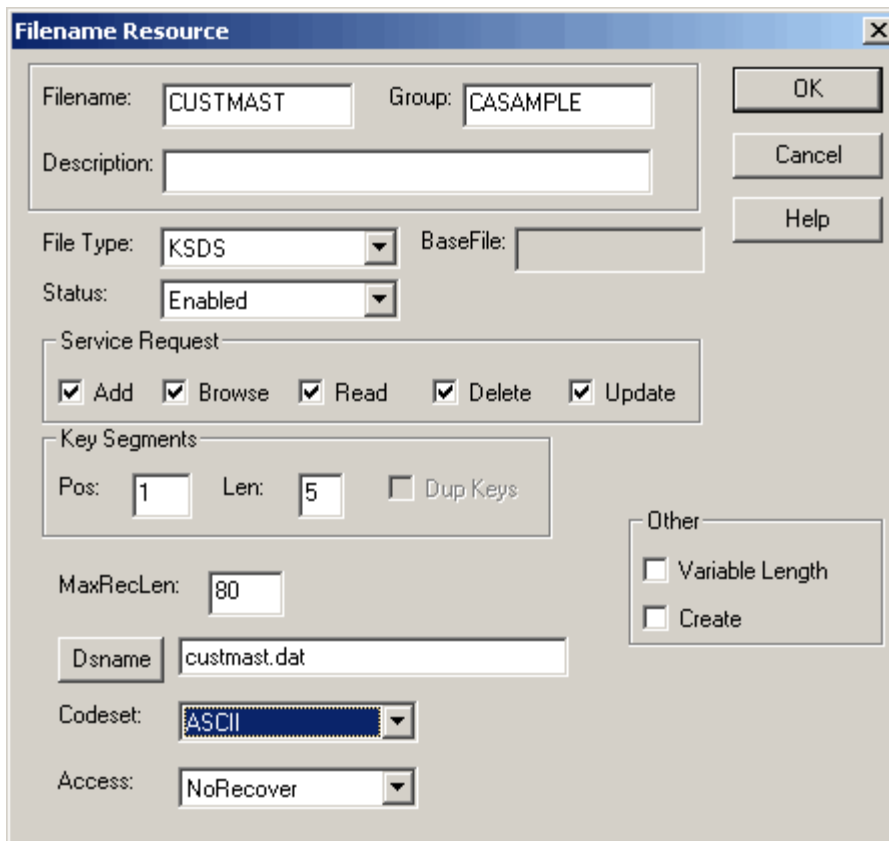
ii. **Select Resource->File Name**



iii. **Select Edit->Insert**



- iv. **Specify the file type, length of the key field, maximum record length, dsname**



Filename Resource

Filename: Group:

Description:

File Type: BaseFile:

Status:

Service Request

☒ Add ☒ Browse ☒ Read ☒ Delete ☒ Update

Key Segments

Pos: Len: ☐ Dup Keys

MaxRecLen:

Dsname:

Codeset:

Access:

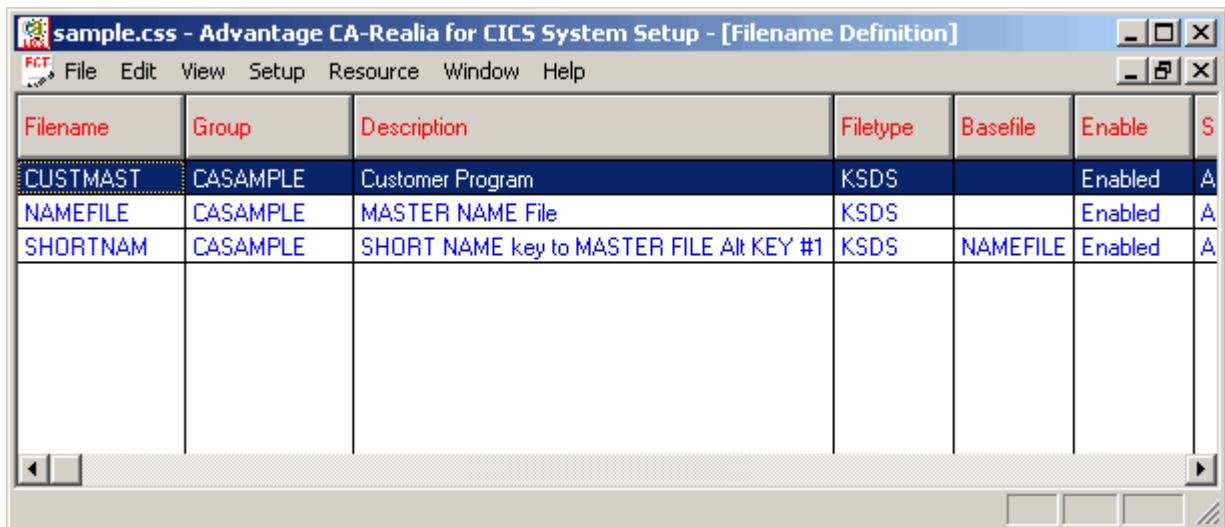
Other

☐ Variable Length

☐ Create

OK Cancel Help

v. **The Filename gets added in the FCT**



Filename	Group	Description	Filetype	Basefile	Enable	S
CUSTMAST	CASAMPLE	Customer Program	KSDS		Enabled	A
NAMEFILE	CASAMPLE	MASTER NAME File	KSDS		Enabled	A
SHORTNAM	CASAMPLE	SHORT NAME key to MASTER FILE Alt KEY #1	KSDS	NAMEFILE	Enabled	A

After adding the dataset to the FCT, the dataset can now be referred to in any program, which references the dataset.

Appendix B: Table of Figures

Figure 1: CEBR.....	34
Figure 2: CEBR.....	35
Figure 3: Browse ended message.....	36
Figure 4: CEBR.....	37
Figure 5: Typing the command.....	38
Figure 6: Output.....	39
Figure 7: Response: Normal.....	40
Figure 8: Response : INVREQ	41
Figure 9: Response: PGMIDERR.....	42
Figure 10: Typing the command.....	43
Figure 11: Output.....	44
Figure 12: Typing Command	45
Figure 13: Output.....	46
Figure 14: Session ended message	47
Figure 15: Typing command.....	48
Figure 16: Output.....	48
Figure 17: Output.....	49
Figure 18: Output.....	50
Figure 19: Typing the CEDF command.....	51
Figure 20: Output.....	52
Figure 21: Typing the command.....	52
Figure 22: Output.....	53