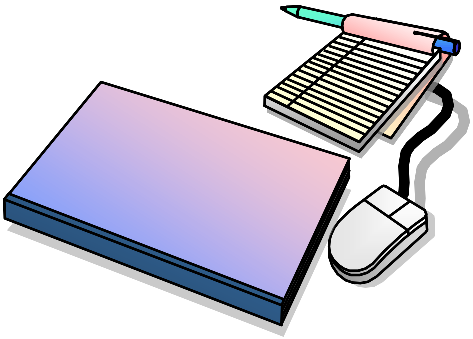
Exercises



Contents

Exercise 1 – A simple program

In this exercise you will write a simple program, just to familiarize yourself with CICS-commands.

Send

- The program will have the name `EDx1PGM` and the transaction code `EDx1`. Replace x with the number you have been given by your instructor.
- The program should only write a message to the terminal like the message you see in the box, and then terminate.

```
This is Exercise 1


Best Regards xxxx
```

- The first message is placed on row 10 and the second message on row 15.
- Compile your program.
- Activate EDF, execute your program and check the result.
- When you find the program working, run again without EDF.

End of exercise.

Exercise 2 – A conversation

In this exercise, you will write a pseudo-conversational program.

Part 1 – Conversation

- In this exercise you will wright a pseudo-conversational program.
- The program will have the name `EDx2PGM` and the transaction code `EDx2`. Replace the x with the number you have been given by your instructor.
- When the program starts, it should write information similar to the information in the box below, and set itself in a pseudo-conversational state.

```
This is a pseudoconversational program, entered n times
Press ENTER to continue, press CLEAR to terminate
```

- Messages should be placed on rows 15 and 16.
- You should add a counter to the program, that should be increased by one every time the program is entered.
- To terminate the conversation, the user should press the CLEAR-key. When this key is pressed, you should terminate the program and return to CICS.
- If any other key is pressed, the program should remain in pseudo-conversational mode.
- If an invalid key is pressed, the Invalid key message should appear on row 18, otherwise not.

```
This is a pseudoconversational program, entered n times
Press ENTER to continue, press CLEAR to terminate

*** Invalid key pressed ***
```

- Before this, the screen should be cleared and the keyboard should be unlocked.

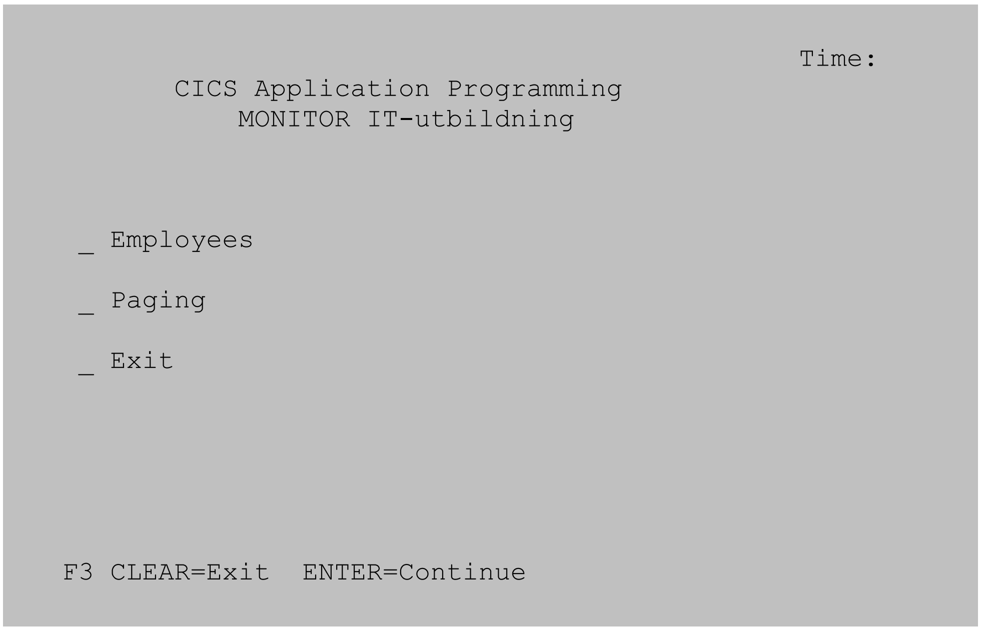
- Compile the program.
- Activate CEDF, execute the program and check the result.
- When the result seems OK, run again without CEDF.

Exercise 3 – Menu program

In this exercise you will write a program that handles a Meny-map.

Send Map

- The program will have the name `EDx3PGM` and the transaction code `EDx3`. Replace the x with the number you have been given by your instructor.
- The program should send and receive the map with the name `EDMAP1` which looks like this:



```
                                                    Time:
          CICS Application Programming
              MONITOR IT-utbildning



  _ Employees

  _ Paging

  _ Exit






 F3 CLEAR=Exit  ENTER=Continue
```

- The COBOL-copy / structure for the map can be found in `teacher.DC16.COPYLIB` or on the website Cobolskolan.se/dc15.
- Take a look at it using Browse, and you will find names given to variables.
- The program should be a pseudo-conversational program and the task is either to send or receive the map.
- Check the users selection by checking the cursor location.
- If the program encounters an error, send a message to the user. There is a location in the map reserved for message.
- When the user chooses `Employees` control should be given to a program with the name `EDx4PGM`. If this program is not available, send a message to the terminal.
- When the user chooses `Paging` control should be given to a program with the name `EDx8PGM`. If this program is not available, send a message to the terminal.
- When the user chooses to Exit, the program should clear the screen and return normally to CICS.

- Compile the program.
- Activate CEDF, execute the program end check the result.

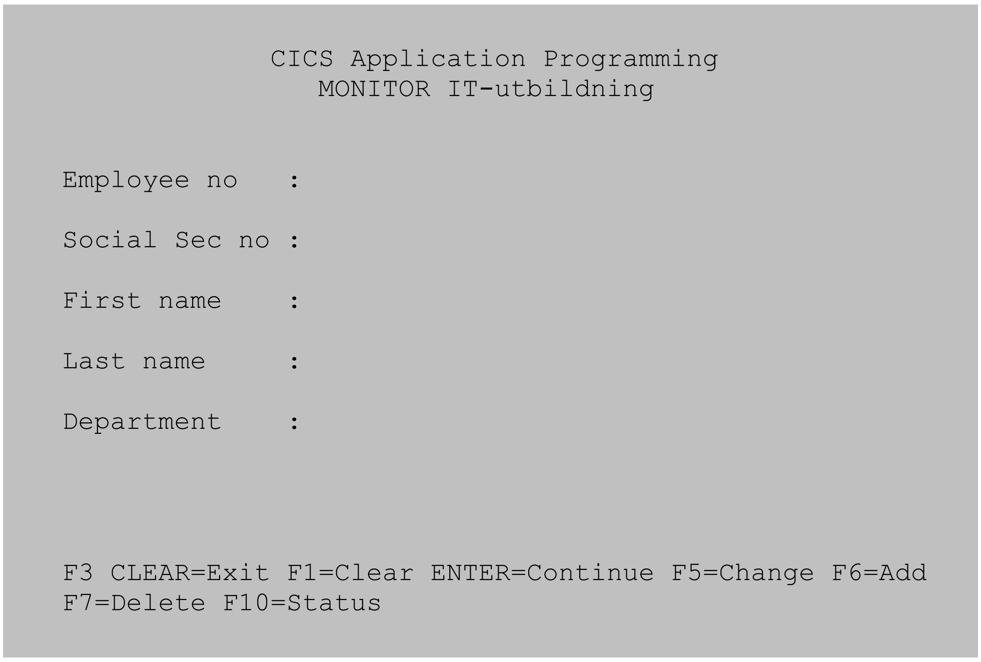
End of exercise.

Exercise 4 – Employees

In this exercise you will write a program that read employee-records and presents them in a map.

Show Employee-info

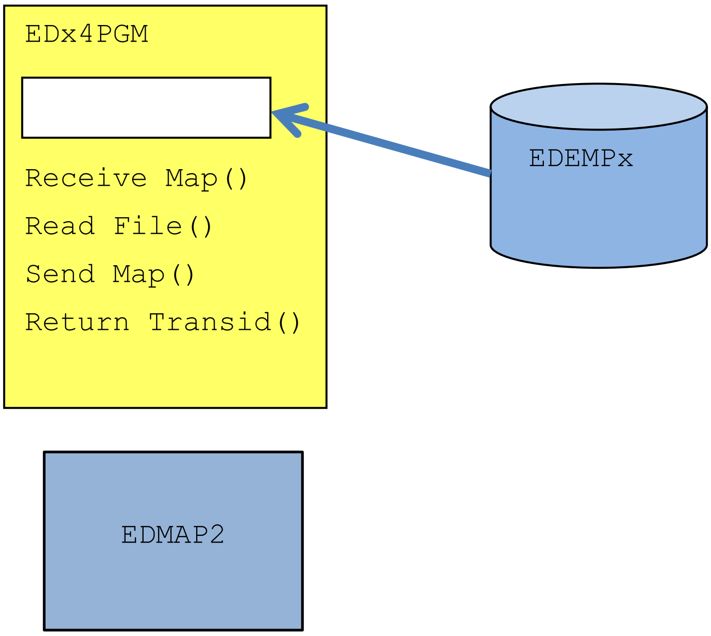
- The program will have the name `EDx4PGM` and the transaction code `EDx4`. Replace the x with the number you have been given by your instructor.
- The program gets control from the menu-program and should either send or receive the map `EDMAP2` with or without any data.
- The name of the employee-file is `EDEMPx`. The physical dataset name is not necessary to know, but can be found by entering `CEMT I FILE(EDEMPx)`.
- The mapset and the map has the name `EDMAP2` and looks as follows:



```
                CICS Application Programming
                   MONITOR IT-utbildning


   Employee no    :

   Social Sec no  :

   First name     :

   Last name      :

   Department     :



   F3 CLEAR=Exit F1=Clear ENTER=Continue F5=Change F6=Add
   F7=Delete F10=Status
```

- The COBOL-copy / structure for the map can be found in *`teacher`*`.DC16.COPYLIB`
- To check what the map looks like, enter `CECI SEND MAP(EDMAP2) ERASE`.
- When the user presses the Enter-key, the program should read the file `EDEMPx` using the `Employee no` as the key-value when reading the file.
- This version of the program should implement support for `F1=Clear` , `F3 CLEAR=Exit` and `Enter=Continue`. Other selections should be considered invalid.
- Exceptions should be reported in the map field reserved for message.
- The file `EDEMPx` has the following structure:

```
Employee number      5  char
Social sec number    10 char
Given name           20 char
Family name          20 char
Department           5  char
Filler               20 char

Total length         80 char
```



- Compile the program.
- Activate CEDF, execute the program end check the result.

End of exercise.

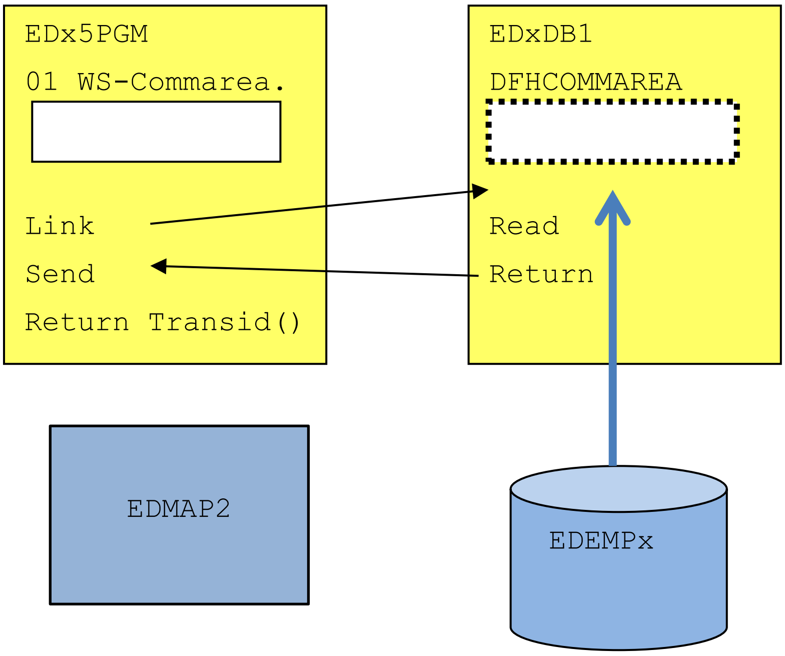
Exercise 5 – Tiers

In this exercise you will modify the program from the previous exercise and also write a new program that handles all data access.

Presentation and Data Access

- Use program `EDx4PGM` as a template for the new program that will handle the presentation. The new program name should be `EDx5PGM,` called the Presentation-program.
- The program should still retain its original name at compilation-time (use parameters `PGMNAMN=ED05PGM` and `PGMLOAD=ED04PGM`).

- The program will now only contain logic to present data on the screen.
- All functions for any kind of data access will be handled by a new program named `EDxDB1,` called the DB-program. The first version of the DB-program will use CICS File Control, but later versions can/may use DB2/SQL.
- The DB-program should be called by the Presentation program using `LINK`, and communication between the two should use `COMMAREA.`
- It is up to you to design a protocol to be used between the two programs. There must be a way to request service from the DB-program, and also check the outcome of the request as well as receive some data.
- Discuss your solution with the instructor before you start programming.



- Compile the program.
- Activate CEDF, execute the program end check the result.

- Write a new Database-Program named `EDxDB2` and use the `EDxDB1` as a template. Replace the CICS Read command with DB2/SQL.
- Compile the program.
- Activate CEDF, execute the program end check the result.

End of exercise.

Exercise 6 – Changes

In this exercise you will complete the Presentation- and DB-programs to handle all request-types.

Change Employee-record

- You can use the Presentation-program from Exercise 5 as a template. New name will be `EDx6PGM`. Update the existing Database-program, but use name `EDxDB25`. The programs should still retain their orginal names at compilation-time (use parameter `PGMNAMN=`*`cobolsourcename`* and `PGMLOAD=`*`loadmodulename`*).
- Update the programs to handle update requests.(`F5=Change`)
- Compile the program.
- Activate CEDF, execute the program end check the result.

.

More functions

- Update the programs to handle the rest of the request-types, if you have time available. Check with your instructor.
- Compile the program.
- Activate CEDF, execute the program end check the result.

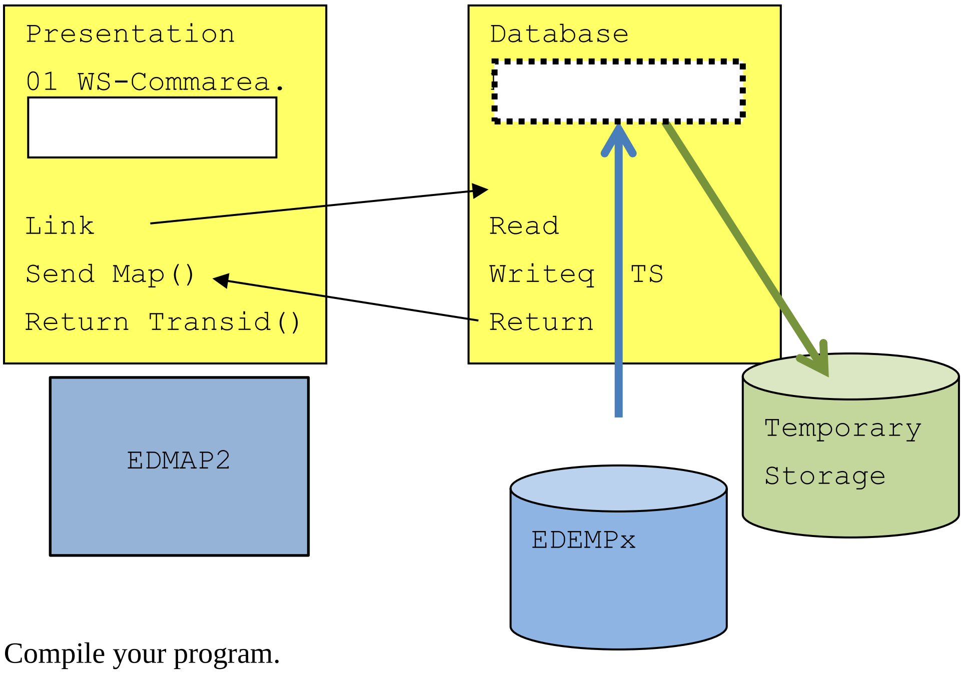
End of exercise.

Exercise 7 – Temporary Storage

In this exercise you will extend your program to check that the information on the screen hasn't been change in the file/database while you have been watching it, before making changes or deletions.

Check contents

- Add functions to the DB-program (`EDxDB2)` to save requested record on Temporary Storage before it is returned to the Presentation Program. To name the TSQ you can use the concatenation of your terminal id and the name of the transaction.
- Before the Database Program makes any change or delete, you should read the record from TS and read the record from the file/database and check that they are equal before making any change.
- If the two don't match, this means that someone else have made changes while you have been watching the information on the screen.
- When this situation occurs, you cannot allow changes and the presentation program should be informed.
- You can force this situation by running the transaction on two different screens.



- Compile your program.
- Activate CEDF, execute the program and check the result.

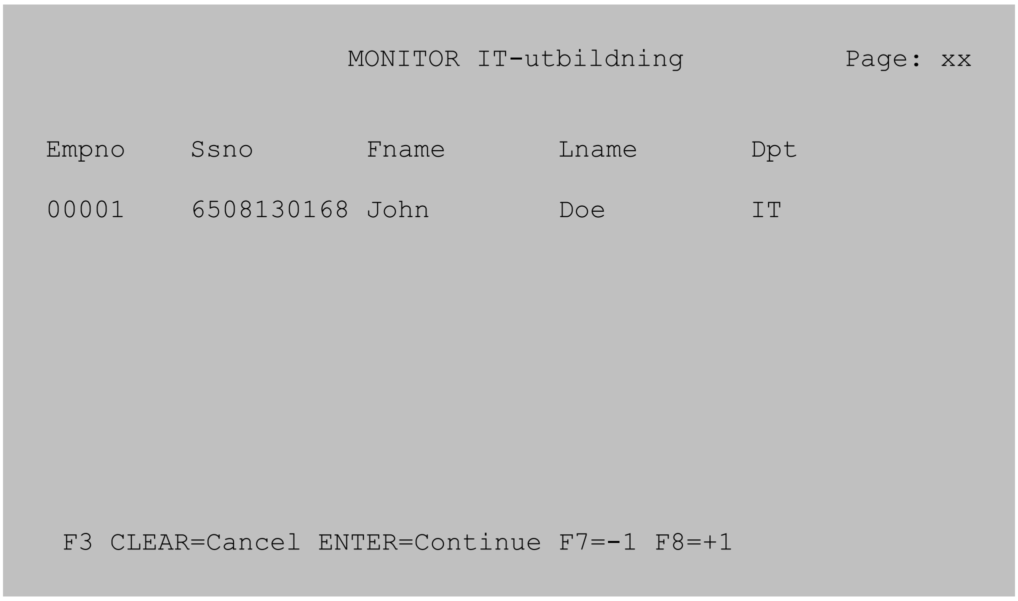
End of Exercise.

Exercise 8 – Paging

In this exercise you will write a program able to present many pages of information from the file/database.

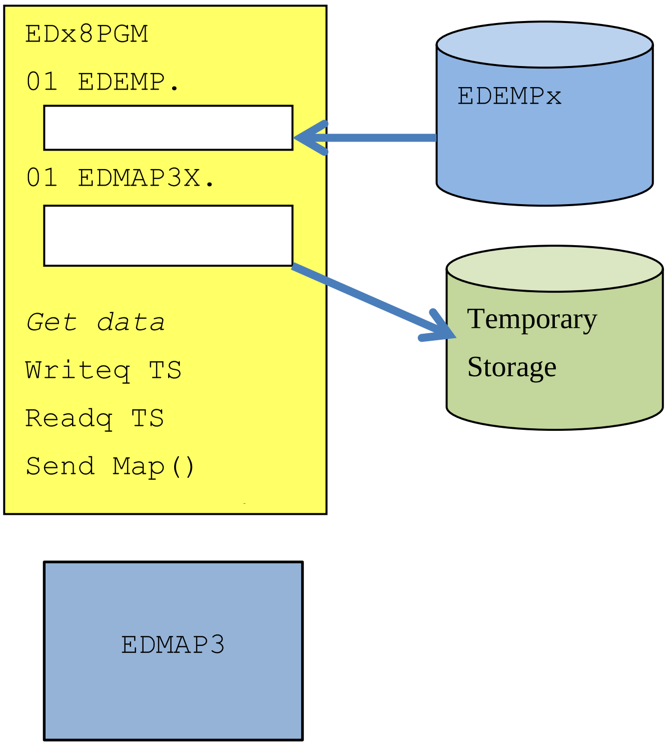
Paging

- The program `EDx8PGM`, transaction-code `EDx8`, will read information from the data-source `EDEMPx` and show on the screen when the user selects Paging from the map menu map `EDMAP1` .
- You may choose to use VSAM or DB2/SQL. Be aware of different compile JCLs for CICS (`COMPCC`) and CICS DB2 (`COMPCC2`).
- There is a map and a mapset with the name `EDMAP3` available for this.



```
                      MONITOR IT-utbildning          Page: xx


Empno    Ssno        Fname        Lname        Dpt

00001    6508130168  John         Doe          IT








 F3 CLEAR=Cancel ENTER=Continue F7=-1 F8=+1
```

- Use the transaction CECI to check what the map really looks like.
- The Application Structure for the map can be found on the library *`teacher`*`.DC16.COPYLIB, member name EDMAP3X`. It contains a Cobol table with space for nine (9) rows.
- The program will read all records/rows from the data-source `EDEMPx` and present it in the map.
- You will need to fill many maps for this. When one map has been filled, it should be saved on Temporary Storage using q-name `EDx8PGM.`
- Fill maps until you reach end of the data, and also save the last map on TS.
- When finished, read the first item from TS and present the map `EDMAP3` on the screen.
- Terminate your program and be prepared to show another page according to user selection, but just ignore invalid selections/PF.
- When `F3 CLEAR=Cancel` give control to the menu-program `ED3xPGM.`



- Compile your program.
- Activate CEDF, execute the program and check the result.

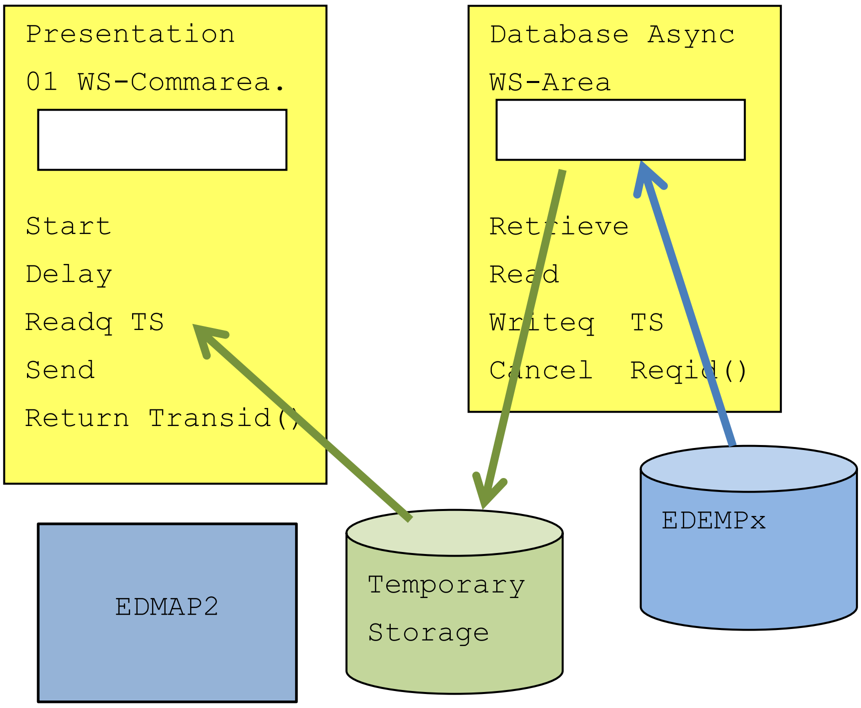
End of Exercise.

Exercise 9 – Asynchronous Transactions

This exercise will introduce the CICS commands START and DELAY.

Start background task

- Copy the old version of the presentation program, and name it EDx9PGM. This program currently calls the Database Program using a CICS LINK-command.
- Instead of using the command LINK, you will now use the command START. The started program needs the same information as before to complete. You can't use the Commarea option in the START-command, but you can use the from option in a similar way or you may use a Channel().
- The name of the asynchronous Database-program will be EDxDB2A. The transaction code will be EDxA
- The return-information, which previously was placed in the Commarea, will now be placed on a TS-queue. You will have to decide the Q-name.
- The Presentation Program will have to wait a maximum of 10 seconds using the DELAY-command and a Request-ID. When this time limit has expired, this will be considered an error situation.
- When the Database Program has finished its request, it has to cancel the Request-ID, which has to be handled over from the Presentation Program in the START–command.



- Compile your program.
- To test the solution you will need two (2) separate CICS-screens.
- Activate CEDF on the screen where you will run the presentation program
- Activate the transaction CEDX for transaction EDxA on the other screen.
- Start the Presentation program on screen one.
- Check the result.

End of Exercise.

Exercise 10 – Transient Data

This exercise will introduce the CICS commands Writeq TD

Error checking and logging

- When you read temporary storage to get hold of the data that the Database Program has created, you should check for QIDERR-condition.
- If this condition occurs after the DELAY has been completed, it might indicate that the Database Program didn't function correctly.
- When this situation occurs, write information to the Transient Data Destination named ERRA. This destination is an indirect destination pointing at CICS System Log, CSSL-destination.
- Abend the program using command EXEC CICS ABEND using abend-code ERRA. This will terminate your transaction.
- Use Google to find out how to use the command.
- Before you abend the transaction, you should clear the screen and write a message to the user.

Compile the program.

Test the situation.

End of Exercise.