An Interface Program for
The EPC Disaster Database

Technical Report 93-2

Canadä

An Interface Program for
The EPC Disaster Database

Technical Report 93-2

by M. Provencher

Evaluation and Analysis

Emergency Preparedness Canada

June 1993

Foreword

This report has been written so that future users of the Disaster Database may easily access the database. It is intended for people with some computer knowledge.

Contents

An Interface Program for The EPC Disaster Database

Introduction

This document describes an interface program which was created to facilitate access to the database in various modes; for example, searching for entries (i.e. individual disasters), modifying them, adding new entries and printing them. The interface program created for EPC Disaster Database is not unlike the one created for the Transborder Agreements Database documented in [1].

For more information on the origin, the content, and the updating process of the Disaster Database, please see [2], [3], [4] and [5].

How to Use the Program

The interface program is called Disastertool. It is in fact the main program of a series of other programs all designed to make access to EPC Disaster Database easier. All programs are FoxBASE programs and are listed in the annex.

To be able to run the program Disastertool, you have to make sure that all the program files, database files, the memory file "tool.mem," the index files and the report file are in the same folder[1] . Then you double-click on Disastertool.prg and it starts! Alternatively, if the Relational Database Management System FoxBASE is already active, you go under the menu item **Program**, select **Do...** and the program Disastertool.prg (possibly through a set of folders/directories), which will start upon selection.

In designing the program it was decided that the standard mode would be one of only displaying the information, thus preventing the user from accidentally modifying the information already recorded. Disastertool will

---

[1] The files CanDisaster.dbf, chrondisas.idx, CanCom.dbf, CanCom.dbt, CanCom.idx, type.dbf, type.idx, origin.dbf, origin.idx, provinces.dbf, provinces.idx, sgldisaster.frx, sgldisaster.vue in addition to the files referred to on page A-1 need to be in the same folder.

position the user at its first record which refers to the first disaster as in Figure 1. All the information on any particular disaster is similarly shown.

Figure 1: Display of a disaster
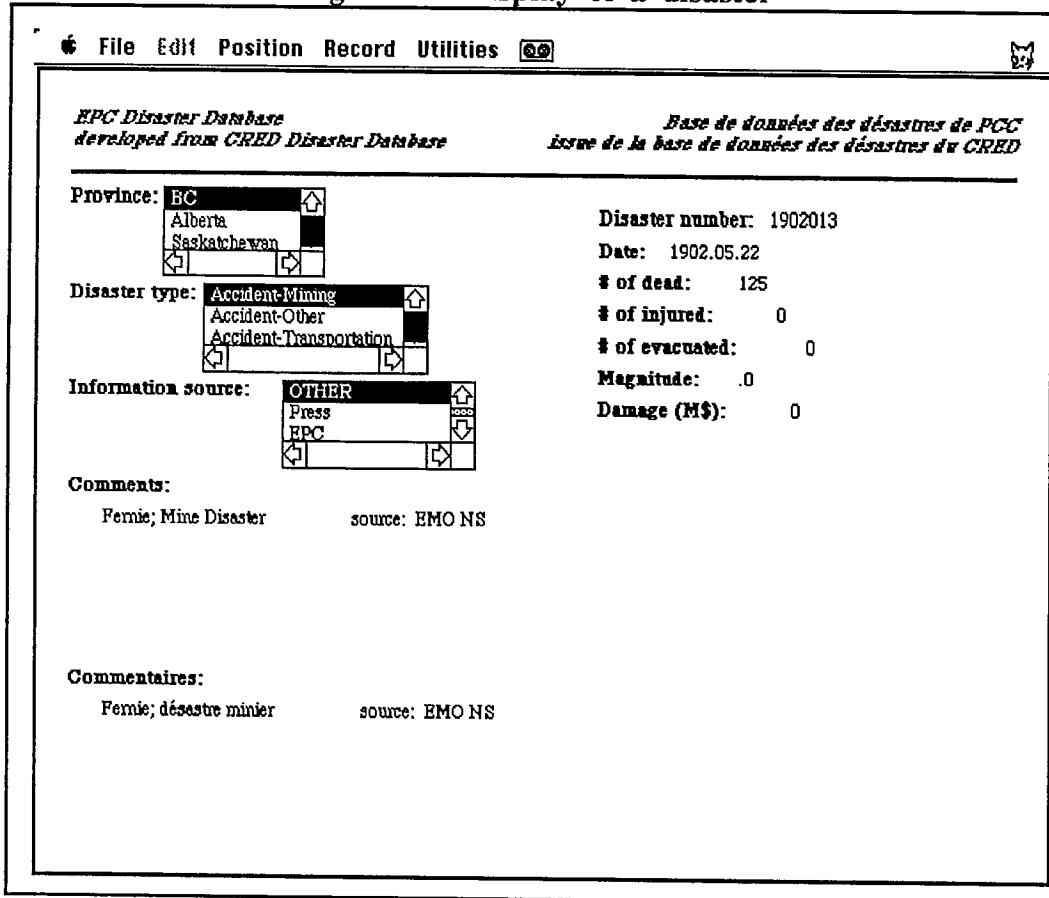
Notice the menu items "Position," "Record" and "Utilities." "Position" (Figure 2) allows the user to navigate through the records in a simple way. The user can go to the top or bottom of the file, and shift records to the next or to the previous, always displaying the information on the current record. This can be done through the menu or alternatively through the keys when applicable (e.g. Command-N, Command-P).

Figure 2: The Position Menu

⌘ File	Edit	**Position**	Record	Utilities	🔘

```
                  Top of File
 EPC Disaster     Bottom of File
 developed fro ·················· tabase         list

                  Next      ⌘N
 Province: BC    Previous    ⌘P
           Alberta              ■
           Saskatchewan
```
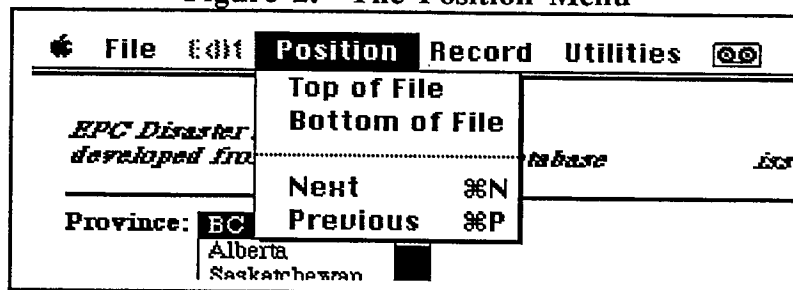
However, when one wants to see (a) specific record(s), the menu item "Record" has to be used (Figure 3). The "Locate" option calls the FoxBASE Expression Builder dialog. This is when the user has to determine the set of records wanted. If a matching record is found, it is displayed and the Continue option is enabled. Thus, the capability is there to display the whole set of records. If a matching record is not found, an ALERT is displayed.

Figure 3: The Record Menu

⌘ File	Edit	Position	**Record**	Utilities	🔘

```
                          Append Record    ⌘A
 EPC Disaster Database     Edit Record      ⌘E
 developed from CRED Di    Delete Record    ⌘D    Extra

 ....................................................

 Province: BC        ]     Locate           ⌘L
           Alberta         Continue         ⌘C
           Saskatchewan
```

The other options under the Record menu have to be used carefully since they can modify the content of the database. The Edit option, as well as the Append option, are very similar in their presentation of the record (Figure 4). The principal difference is that in the Append mode there is a blank in every field exce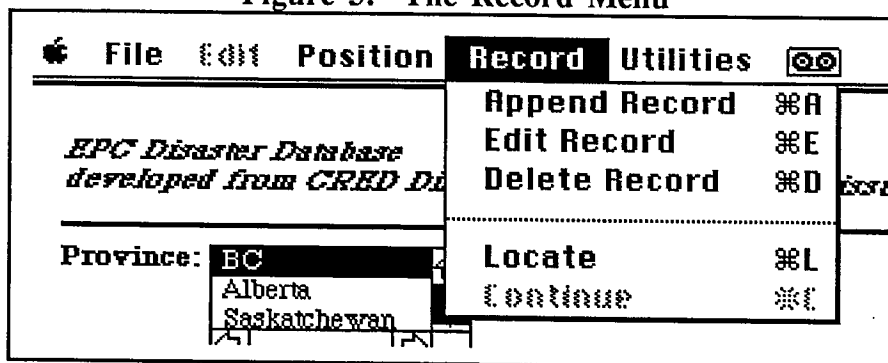pt for disno (disaster number), which is automatically generated. This Append option is, clearly, for adding a new record to the database. In either case disno cannot be changed. Please note the "Finished" button on the right. "Finished" allows the user to get out of Edit/Append mode at any time.

The Delete option is straight forward. It just deletes the current record.

Figure 4:   Disaster in Edit mode

Figure 4:   Disaster in Edit mode

The Utilities menu (Figure 5) has two basic options. The first allows the user to print the current entry (i.e. disaster) through the "Report" option. "Quit Application" may be used when all the desired operations on the database are completed.

Figure 5:   The Utilities Menu

Figure 5:   The Utilities Menu

Conclusion

The purpose of this paper was to explain an interface program recently created to facilitate access to the EPC Disaster Database. Screens are presented to assist users in a pictorial manner, in their understanding of the procedures. Program listings are also given which would allow for future adjustments.

Bibliography

1. *An Interface Program for the Canada/United States Transborder Agreements Database* by M. Provencher, EPC Technical Report 93-1, February 1993.

2. *Disaster Events Database. Programme User's Manual*, Centre for Research on the Epidemiology of Disasters (CRED).

3. *A Canadian Disaster Database* by M. Provencher, EPC Technical Report 91-2, December 1991.

4. *EPC Disaster Database and Its Characteristics* by M. Provencher, EPC Technical Report 92-1, May 1992.

5. *The Development of a Database on Disasters* by D.G. Sapir and C. Misson, Disasters Vol.16, No.1.

Annex

Program Listings

```
* Program...: Add_Rec
*------------------------------------------------------------
* ---Program to add a record
*------------------------------------------------------------
* ---Disable all user-defined menus in the menu bar.
*       This prevents user from making another selection while in add mode.
MENU OFF 1
MENU OFF 2
MENU OFF 3
*
CLEAR
ACCEPT "What is the year of this new disaster (format 9999) ? " TO year1
SELECT A
GOTO TOP
STORE "000" TO str1
STORE STUFF(year1,5,0,str1) TO year1
yearm=year1
finished=0
DO WHILE finished=0
    IF SUBSTR(CanDisaster->disno,1,4)=SUBSTR(year1,1,4)
        finished=1
        DO WHILE SUBSTR(CanDisaster->disno,1,4)=SUBSTR(year1,1,4)
            yearm=STR(MAX(VAL(disno),VAL(yearm)),7)
            SKIP
        ENDDO
    ELSE
        SKIP
        IF EOF()
            finished=1
        ENDIF
    ENDIF
ENDDO
APPEND BLANK
REPLACE CanDisaster->disno WITH STR((VAL(yearm)+1),7)
SELECT B
APPEND BLANK
REPLACE CanCom->disno WITH STR((VAL(yearm)+1),7)
SELECT A
finished=0
temp1=13
temp2=17
temp3=12
DO WHILE finished=0
    CLEAR
    DO SAYS
    @ PIXELS 81,452 GET Candisaste->disno STYLE 0 FONT "Times",12 PICTURE "9999999" SIZE 12,88 COLOR 0,0,0,-1,-1,-1
    CLEAR GETs
    @ PIXELS 61,75 GET temp1 FROM tprovinces STYLE 0 FONT "Times",12 SIZE 50,100 COLOR 0,0,0,-1,-1,-1
    @ PIXELS 115,100 GET temp2 FROM ttypes STYLE 0 FONT "Times",12 SIZE 50,140 COLOR 0,0,0,-1,-1,-1
    @ PIXELS 169,150 GET temp3 FROM torigins STYLE 0 FONT "Times",12 SIZE 50,120 COLOR 0,0,0,-1,-1,-1
    DO GET2s
    @ PIXELS 207,490 GET finished STYLE 0 FONT "Chicago",10 PICTURE "@*V Finished" SIZE 2068,63 COLOR 0,0,0,-1,-1,-1
    READ
    REPLACE province WITH tprovince(temp1)
    REPLACE type WITH ttype(temp2)
```

```
    REPLACE origin WITH torigin(temp3)
ENDDO
* ---Enable menu system again, now that record is added
MENU ON 1
MENU ON 2
MENU ON 3
RETURN
* EOF : Add_Rec.prg
```

```
*-----------------------------------------------------------
*
* Program...: Disastertool.prg
* Date......: February 1993
* By........: Michèle Provencher
*
*-----------------------------------------------------------
* --- SET Environment
*-----------------------------------------------------------
*
SET SAFETY OFF              && overwrite files w/out confirmation
SET TALK OFF                && suppress interactive talk
SET BELL OFF                && bell will not sound on full field
ON ERROR DO Err_Proc        && do this when an error occurs
CLOSE ALL                   && close all data files
*
*-----------------------------------------------------------
* --- Open Files
*-----------------------------------------------------------
*
USE "CanDisaster.dbf"
SET INDEX TO "chrondisas.idx"
SELECT B
USE "CanCom.dbf"
SET INDEX TO "CanCom.idx"
SELECT C
USE "type.dbf"
SET INDEX TO "type.idx"
SELECT D
USE "origin.dbf"
SET INDEX TO "origin.idx"
SELECT E
USE "provinces.dbf"
SET INDEX TO "provinces.idx"
SELECT A
SET RELATION TO CanDisaster->disno INTO CanCom ADDITIVE
SET RELATION TO CanDisaster->type INTO type ADDITIVE
SET RELATION TO CanDisaster->origin INTO origin ADDITIVE
SET RELATION TO CanDisaster->province INTO provinces ADDITIVE
*
*-----------------------------------------------------------
* --- RESTORE Memory Variables
*-----------------------------------------------------------
*
RESTORE FROM tool
STORE .F. to done
*
*-----------------------------------------------------------
* --- Menu System
*-----------------------------------------------------------
MENU BAR "Position;Record;Utilities"
*
```

```
* ---Define Position menu options
* ---Note /N & /P keyboard shortcuts and divider line
* ---Feel free to add /T & /B shortcuts to Top and Bottom options
MENU 1,"Top of File;Bottom of File;(-;Next/N;Previous/P"
*
* ---Define Record menu options
* ---Note /A, /E, /D, /L, /C keyboard shortcuts and divider line
MENU 2,"Append Record/A;Edit Record/E;Delete Record/D;(-;Locate/L;(Continue/C"
*
* --- Define Utilities menu options
* --- Note /R & /Q keyboard shortcuts and divider line
MENU 3,"Report/R;(-;Quit Application/Q"
*
* ---The following command activates the entire menu system.
*       When the user makes a selection from any one of the user-
*       defined menus, the DO MenuProc command will be executed.
*       When MenuProc is called, MENU(0) and MENU(1) pass the numbers
*       of the menu and option that was selected.
*
ON MENU DO MenuProc WITH MENU(0),MENU(1)
*
*------------------------------------------------------------
* --- Main
*------------------------------------------------------------
*
* ---We will then stay within the following loop until done is changed to True.
*       This change will only happen when you Quit the application.
PUBLIC temp1,temp2,temp3
DO WHILE .NOT. done
    CLEAR
    DO SAYs
    @ PIXELS 81,452 GET Candisaste->disno STYLE 0 FONT "Times",12 PICTURE "9999999" SIZE 12,88 COLOR 0,0,0,-1,-1,-1
    DO GET2s
    CLEAR GETS
    DO GET1s
    READ
ENDDO
*
*------------------------------------------------------------
* --- Closing Operations
*------------------------------------------------------------
*
ON ERROR
SET BELL ON
SET TALK ON
SET SAFETY ON
* ---Return to calling program
CLOSE ALL
RETURN
* END : Disastertool.prg
```

```
* Program...: Edit_Rec
*------------------------------------------------------------
* ---Program to edit a record
*------------------------------------------------------------
* ---Disable all user-defined menus in the menu bar.
*        This prevents user from making another selection while in edit mode.
MENU OFF 1
MENU OFF 2
MENU OFF 3
*
finished=0
DO WHILE finished=0
    CLEAR
    DO SAYS
    @ PIXELS 81,452 GET Candisaste->disno STYLE 0 FONT "Times",12 PICTURE "9999999" SIZE 12,88 COLOR 0,0,0,-1,-1,-1
    CLEAR GETs
    DO GET1s
    DO GET2s
    @ PIXELS 207,530 GET finished STYLE 0 FONT "Chicago",10 PICTURE "@*V Finished" SIZE 2068,63 COLOR 0,0,0,-2120,-1,23624
    READ
    REPLACE province WITH tprovince(temp1)
    REPLACE type WITH ttype(temp2)
    REPLACE origin WITH torigin(temp3)
ENDDO
* ---Enable menu system again, since editing is complete
MENU ON 1
MENU ON 2
MENU ON 3
RETURN
* EOF : Edit_Rec.prg
```

```
* Program...: Err_Proc
*------------------------------------------------------
* ---Program to handle errors
*------------------------------------------------------
* ---Display an alert that contains the error message
ALERT STOP 2 MESSAGE()
RETURN
* EOF : Err_Proc.prg
```

```
* Program...: Find_Rec
*------------------------------------------------------
* ---Program to find a record
*------------------------------------------------------
* ---Store current record number to lastrec variable
STORE RECNO() TO lastrec
* ---Display the Expression Builder dialog with the message LOCATE FOR.
*      Store user input to lookkey variable.
GETEXPR "LOCATE FOR:" TO lookkey
* ---If the expression builder returned an expression
IF .NOT. (lookkey == "")
   * ---LOCATE for user-specified lookkey, note use of macro substitution.
   LOCATE FOR &lookkey
   * ---If a matching record was found
   IF FOUND()
      * ---Update the menu system (enables the CONTINUE option)
      DO UpdtMenu
   ELSE      && a matching record was not found
      * ---Display a message -CHR(13) causes line break-
      ALERT NOTE 1 "No record found for:",CHR(13),lookkey
      * ---Return to previous record
      GOTO lastrec
   ENDIF
ENDIF
RETURN
* EOF : Find_Rec.prg
```

```
* Program...: GET1s
*------------------------------------------------
* ---Input Program
*------------------------------------------------
*
temp1=1
DO WHILE .NOT.(CanDisaster->province=tprovince(temp1))
    temp1=temp1+1
ENDDO
@ PIXELS 61,75 GET temp1 FROM tprovinces STYLE 0 FONT "Times",12 SIZE 50,100 COLOR 0,0,0,-1,-1,-1
temp2=1
DO WHILE .NOT.(CanDisaster->type=ttype(temp2))
    temp2=temp2+1
ENDDO
@ PIXELS 115,100 GET temp2 FROM ttypes STYLE 0 FONT "Times",12 SIZE 50,140 COLOR 0,0,0,-1,-1,-1
temp3=1
DO WHILE .NOT.(CanDisaster->origin=torigin(temp3))
    temp3=temp3+1
ENDDO
@ PIXELS 169,150 GET temp3 FROM torigins STYLE 0 FONT "Times",12 SIZE 50,120 COLOR 0,0,0,-1,-1,-1
*
RETURN
*
* EOF : GET1s.prg
```

```
* Program...: GET2s
*------------------------------------------------------
* ---Input Program
*------------------------------------------------------
*

@ PIXELS 99,390 GET Candisaste->d_strike STYLE 0 FONT "Times",12 PICTURE "9999.99.99" SIZE 12,88 COLOR 0,0,0,-1,-1,-1
@ PIXELS 117,412 GET Candisaste->dead STYLE 0 FONT "Times",12 PICTURE "9999999999" SIZE 12,106 COLOR 0,0,0,-1,-1,-1
@ PIXELS 135,430 GET Candisaste->injured STYLE 0 FONT "Times",12 PICTURE "9999999999" SIZE 12,106 COLOR 0,0,0,-1,-1,-1
@ PIXELS 153,448 GET Candisaste->evacuated STYLE 0 FONT "Times",12 PICTURE "9999999999" SIZE 12,106 COLOR 0,0,0,-1,-1,-1
@ PIXELS 171,430 GET Candisaste->magnitude STYLE 0 FONT "Times",12 PICTURE "9.9" SIZE 12,43 COLOR 0,0,0,-1,-1,-1
@ PIXELS 189,439 GET Candisaste->dollar_dam STYLE 0 FONT "Times",12 PICTURE "9999999999" SIZE 12,106 COLOR 0,0,0,-1,-1,-1
@ PIXELS 252,38 GET Cancom->commente STYLE 0 FONT "Times",12 SCROLL  SIZE 86,556 COLOR 0,0,0,-1,-1,-1
@ PIXELS 363,38 GET Cancom->commentf STYLE 0 FONT "Times",12 SCROLL  SIZE 86,556 COLOR 0,0,0,-1,-1,-1
*

RETURN
*

* EOF : GET2s.prg
```

```
* Program...: MenuProc
*------------------------------------------------------------
* ---Menu Processing Program
*------------------------------------------------------------
* ---Name the values passed from MENU(0), MENU(1)
PARAMETERS the_menu,the_item
* ---Process input
DO CASE

    * ---If the selection was made in the position menu
    CASE the_menu = 1
        DO CASE

            * ---If the first option was selected "Top of File"
            CASE the_item = 1
                * - Go to the top of file
                GOTO TOP
                RETURN

            * ---If the second option was selected "Bottom of File"
            CASE the_item = 2
                * - Go to the bottom of file
                GOTO BOTTOM
                RETURN

            * ---The third option was a divider line

            * ---If the fourth option was selected "Next Record"
            CASE the_item = 4
                * ---Move forward one record
                SKIP
                * ---If end-of-file encountered
                IF EOF()
                    * ---Error message
                    ALERT NOTE 2 "You are positioned at the last record in the database."
                    * ---Reset record pointer to last record
                    GOTO BOTTOM
                ENDIF
                RETURN

            * ---If the fifth option was selected "Previous Record"
            CASE the_item = 5
                * - Go backward
                SKIP -1
                * ---If beginning-of-file encountered
                IF BOF()
                    * ---Error message
                    ALERT NOTE 2 "You are positioned at the first record in the database."
                    * ---Reset record pointer to first record
                    GOTO TOP
```

```
                ENDIF
                RETURN
        ENDCASE

* ---If the selection was made in the record menu
CASE the_menu = 2
    DO CASE

        * ---If the first option was selected "Append a Record"
        CASE the_item = 1
            * ---Append a record
            DO Add_Rec
            RETURN

        * ---If the second option was selected "Edit Record"
        CASE the_item = 2
            * ---Edit the current record
            Do Edit_Rec
            RETURN

        * ---If the third option was selected "Delete Record"
        CASE the_item = 3
            * ---Delete the current record
            SELECT A
            STORE RECNO() TO delrec
            SKIP -1
            STORE RECNO() TO lastrec
            GOTO delrec
            DELETE
            PACK
            SELECT B
            GOTO delrec
            DELETE
            PACK
            SELECT A
            GOTO lastrec
            RETURN

        * ---The fourth option was a divider line

        * ---If the fifth option was selected "Locate a Record"
        CASE the_item = 5
            * ---Locate a record using the Find_Rec routine
            DO Find_Rec
            RETURN

        * ---If the sixth option was selected "Continue"
        CASE the_item = 6
            * - Store the current record number to the lastrec variable
            *     If record can not be LOCATEd, program will return to this
```

```
*       record.
          STORE RECNO() TO lastrec
          * - Continue LOCATE
          CONTINUE
          * - If matching record was not found
          IF .NOT. FOUND()
              * - Display alert message
              ALERT STOP 2 "No more matching records!"
              * - Return to previous record location
              GOTO lastrec
              * - Do the menu update (disable CONTINUE option)
              DO UpdtMenu
          ENDIF
          RETURN
      ENDCASE

  * ---If the selection was made in the utility menu
  CASE the_menu = 3

      STORE RECNO() TO lastrec

      DO CASE

          * ---If the first option was selected "Report"
          CASE the_item = 1
              SET CONSOLE OFF
              REPORT FORM sgldisaster;
                  FOR RECNO() = lastrec TO PRINT PROMPT
              SET CONSOLE ON
              GOTO lastrec
              RETURN

          * ---The second option was a divider line

          * ---If the third option was selected "Quit Application"
          CASE the_item = 3
              * - Exit application
              CLOSE DATABASES
              SCREEN 1 OFF
              * ---Indicate that processing is done
              STORE .T. TO done
              * ---Return to calling procedure'
              RETURN
      ENDCASE

ENDCASE

* EOF : MenuProc.prg
```

```
* Program...: SAYs
*-------------------------------------------------
* ---Display Program
*-------------------------------------------------
*
SCREEN 1 TYPE 1 HEADING "Screen 1" AT 26,9 SIZE 447,623 PIXELS  FONT "Times",268 COLOR 0,0,0,-1,-31465,-30840 TOP
@ PIXELS 27,17 SAY "EPC Disaster Database" STYLE 65536 FONT "Times",780 COLOR 0,0,0,-1,-1,-1
@ PIXELS 39,17 SAY "developed from CRED Disaster Database" STYLE 65536 FONT "Times",780 COLOR 0,0,0,-1,-1,-1
@ PIXELS 27,384 SAY "Base de données des désastres de PCC" STYLE 65536 FONT "Times",780 COLOR 0,0,0,-1,-1,-1
@ PIXELS 39,312 SAY "issue de la base de données des désastres du CRED" STYLE 65536 FONT "Times",780 COLOR 0,0,0,-1,-1,-1
@ PIXELS 54,18 TO 54,604 STYLE 47 COLOR 0,0,0,0,0,0
@ PIXELS 71,17 SAY "Province:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 125,17 SAY "Disaster type:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 179,17 SAY "Information source:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 81,346 SAY "Disaster number:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 99,346 SAY "Date:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 117,346 SAY "# of dead:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 135,346 SAY "# of injured:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 153,346 SAY "# of evacuated:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 171,346 SAY "Magnitude:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 189,346 SAY "Damage (M$):" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 234,17 SAY "Comments:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
@ PIXELS 345,17 SAY "Commentaires:" STYLE 65536 FONT "Times",268 COLOR 0,0,0,-1,-1,-1
*
RETURN
*
* EOF : SAYs.prg
```

```
* Program...: UpdtMenu
*------------------------------------------------------------
* ---Program to update the menu system
*------------------------------------------------------------
* ---This routine is only called after a LOCATE or CONTINUE option has been
*      executed.  It properly enables or disables the CONTINUE option, based
*      on the results of the previous LOCATE or CONTINUE.
* ---If a record was successfully found
IF FOUND()
   * ---Enable the CONTINUE option
   MENU 2,"Append Record/A;Edit Record/E;Delete Record/D;(-;Locate/L;Continue/C"
ELSE     && a matching record was not found
   * ---Disable the CONTINUE option
   MENU 2,"Append Record/A;Edit Record/E;Delete Record/D;(-;Locate/L;(Continue/C"
ENDIF
RETURN
*
* EOF : UpdtMenu.prg
```

Content of tool.mem

torigin		torigins	
(1)	"01"	(1)	"OFDA"
(2)	"02"	(2)	"Insurance"
(3)	"03"	(3)	"CDC"
(4)	"04"	(4)	"Japan"
(5)	"05"	(5)	"OAU"
(6)	"06"	(6)	"WHO/PAHO"
(7)	"07"	(7)	"CRED SITREP"
(8)	"08"	(8)	"GEOCC SITREP"
(9)	"09"	(9)	"WHO/AFRO"
(10)	"10"	(10)	"Private"
(11)	"11"	(11)	"UNDRO"
(12)	"12"	(12)	"OTHER"
(13)	"13"	(13)	"Press"
(14)	"14"	(14)	"EPC"

tprovince		tprovinces	
(1)	"BC"	(1)	"BC"
(2)	"AB"	(2)	"Alberta"
(3)	"SK"	(3)	"Saskatchewan"
(4)	"MB"	(4)	"Manitoba"
(5)	"ON"	(5)	"Ontario"
(6)	"QB"	(6)	"Quebec"
(7)	"NB"	(7)	"N Brunswick"
(8)	"NS"	(8)	"Nova Scotia"
(9)	"PE"	(9)	"PEI"
(10)	"NF"	(10)	"NFLD"
(11)	"YU"	(11)	"Yukon"
(12)	"NW"	(12)	"NWT"
(13)	"SV"	(13)	"Several"
(14)	"IN"	(14)	"International"

ttype		ttypes	
(1)	"AM"	(1)	"Accident-Mining"
(2)	"AO"	(2)	"Accident-Other"
(3)	"AT"	(3)	"Accident-Transportation"
(4)	"AV"	(4)	"Avalanche"
(5)	"CU"	(5)	"Civil Unrest"
(6)	"CW"	(6)	"Cold Wave"
(7)	"DR"	(7)	"Drought"
(8)	"EP"	(8)	"Epidemic"
(9)	"EQ"	(9)	"Earthquake"
(10)	"FF"	(10)	"Forest Fire"
(11)	"FI"	(11)	"Fire-Other"
(12)	"FL"	(12)	"Flood"
(13)	"HC"	(13)	"Hazardous Chemicals"
(14)	"HU"	(14)	"Hurricane"
(15)	"HW"	(15)	"Heat Wave"
(16)	"LS"	(16)	"Landslide"
(17)	"OT"	(17)	"Other"
(18)	"ST"	(18)	"Storm"
(19)	"TO"	(19)	"Tornado"
(20)	"TS"	(20)	"Tsunami"