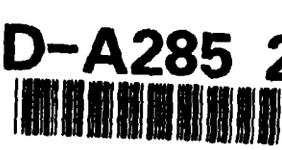


NAVAL HEALTH RESEARCH CENTER

HAZARDOUS MATERIAL LIFE-CYCLE COST MODEL

TECHNICAL MANUAL

VERSION 1.2

H. L. Ly

G. Pang

Technical Document 93-3C

Approved for public release: distribution unlimited.

NAVAL HEALTH RESEARCH CENTER
P. O. BOX 85122
SAN DIEGO, CALIFORNIA 92186 – 5122

NAVAL MEDICAL RESEARCH AND DEVELOPMENT COMMAND
BETHESDA, MARYLAND

Hazardous Material Life-Cycle Cost Model

Technical Manual

Version 1.2

Prepared by:

Hoa Le Ly
Gerald Pang

Naval Health Research Center
Medical Information Systems and
Operations Research Department
P.O. Box 85122
San Diego, CA 92186-5122

Technical Document 93-3C was supported by the Naval Medical Research and Development Command, Bethesda, MD, Department of the Navy, under a NAVSUP Reimbursable Work Unit. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Navy, Department of Defense, nor the U.S. Government.

SUMMARY

This technical manual contains the information on the program source code, data elements, and file structures needed to maintain the Hazardous Material Life-Cycle Cost Model. This documentation was created using the FoxDoc Version 2.5a program.

TABLE OF CONTENTS

Introduction. This document was created using the FoxDoc Version 2.5a program to generate the technical documentation for the Hazardous Material Life-Cycle Cost Model (HMLCCM) system. The documentation is separated into seven sections: (1) System Summary, (2) Menu Summary, (3) Screen Summary, (4) Data Dictionary, (5) Tree Diagram, (6) Procedure Summary, and (7) Source Code Program Listings.

Section I. System Summary. See Section V for the tree diagram of the programs, procedures, functions, and format files.

```
This system has:
 11971 lines of code
     1 program file
    40 procedure files
   155 procedures and functions
    63 databases
    11 structural index files
     1 menu file
    11 screen files
     1 other file
```

Section II. Menu Summary. The following description lists the menu options used to drive the system. HMENU.MNX is the only menu template created.

OPTION NAME	KEYS\FUNCTION CALLED
System	ALT+S _MSM_SYSTM
Help...	F1 _MST_HELP
----------	_MST_SP100
Calculator	_MST_CALCU
----------	(Submenu)
HMLCCM	(Submenu HMLCCM)
Cost Analysis	(Submenu COSTANALYS)
Build Hazmat Scenario	do hmsc
OSHA Z-Table	do hmztab
Reference Material	do hmmsds
System Maintenance	(Submenu SYSTEMMAIN)
Back-Up (floppy)	(Procedure)
Up-Load Data	(Procedure)
Set Parameters	(Submenu SETPARAMET)
Materials	CTRL+M do hmat.spr
Life Cycle Phase	CTRL+L do hmlc.spr
Process	CTRL+W do hmwp.spr
EXposure Type	CTRL+X do hmet.spr
Cost Factors	CTRL+F do hmcf.spr
Cost Factor Elements	CTRL+E do hmcfe.spr
Cost Factor Element Items	CTRL+I do hmcfei.spr
Build Hazmat Table	CTRL+B do hmtab.spr
File	ALT+F _MSM_FILE
Printer Setup...	_MFI_SETUP
Print...	_MFI_PRINT
----------	_MFI_SP300
Quit	do _quit in hminit

OPTION NAME	KEYS\FUNCTION CALLED	
Edit	ALT+E	_MSM_EDIT
Undo	CTRL+U	_MED_UNDO
Redo	CTRL+R	_MED_REDO
----------		_MED_SP100
Cut	CTRL+X	_MED_CUT
Copy	CTRL+C	_MED_COPY
Paste	CTRL+V	_MED_PASTE
Clear		_MED_CLEAR
----------		_MED_SP200
Select All	CTRL+A	_MED_SLCTA
----------		_MED_SP300
Goto Line...		_MED_GOTO
Find...	CTRL+F	_MED_FIND
Find Again	CTRL+G	_MED_FINDA
Replace And Find Again	CTRL+E	_MED_REPL
Replace All		_MED_REPLA
----------		_MED_SP400
Preferences...		_MED_PREF
Database	ALT+D	_MSM_DATA
Browse		_MDA_BROW
----------		_MDA_SP100
Sort...		_MDA_SORT
Total...		_MDA_TOTAL
----------		_MDA_SP200
Average...		_MDA_AVG
Count...		_MDA_COUNT
Sum...		_MDA_SUM
Calculate...		_MDA_CALC
Report...		_MDA_REPRT
Record	ALT+R	_MSM_RECRD
Goto...		_MRC_GOTO
Locate...		_MRC_LOCAT
Continue	CTRL+K	_MRC_CONT
Seek...		_MRC_SEEK
----------		_MRC_SP200
Replace...		_MRC_REPL
Delete...		_MRC_DELET
Recall...		_MRC_RECAL
Program	ALT+P	_MSM_PROG
Cancel		_MPR_CANCL
Resume	CTRL+M	_MPR_RESUM
Window	ALT+W	_MSM_WINDO
Hide		_MWI_HIDE
Hide All		_MWI_HIDEA
Show All		_MWI_SHOWA
Clear		_MWI_CLEAR
----------		_MWI_SP100
Move	CTRL+F7	_MWI_MOVE
Size	CTRL+F8	_MWI_SIZE
Zoom ^X	CTRL+F10	_MWI_ZOOM
Zoom ^Y	CTRL+F9	_MWI_MIN
Cycle	CTRL+F1	_MWI_ROTAT
Color...		_MWI_COLOR
----------		_MWI_SP200
Command	CTRL+F2	_MWI_CMD
Debug		_MWI_DEBUG
Trace		_MWI_TRACE
View		_MWI_VIEW

Section III. Screen Summary. Ten (10) screen files were created as input templates for the HMLCCM system: HMAT.SCX, HMLC.SCX, HMWP.SCX, HMET.SCX, HMCF.SCX, HMCFE.SCX, HMCFEI.SCX, HMTAB.SCX, HMCOMP.SCX, and W_PRINT.SCX.

HMAT.SCX **Last updated: 09/15/93 at 13:34**

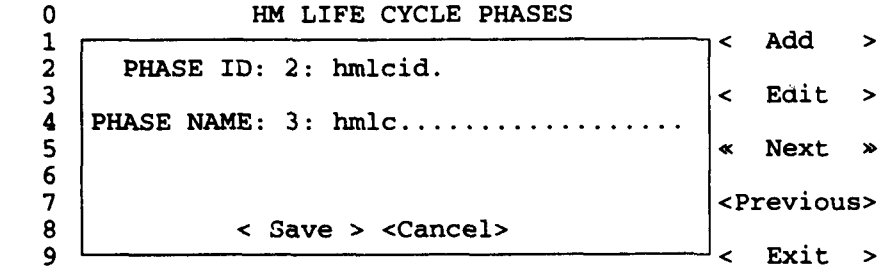
HAZARDOUS MATERIALS

```
 0   ID 1: hm    Materials Name:   2: hmatname.....................................
 1  Manufactory: 3: mfg.....................................................
 2   NIIN #: 4: niin...   Common Name:  5: hmcom.....................................
 3
 46:msds.....................................................................^^
 5.....................................................................
 6.....................................................................
 7.....................................................................
 8.....................................................................
 9.....................................  .....................................
10.....................................  .....................................
11..................................... .....................................
12.....................................  .....................................
13.....................................  .....................................
14.....................................  .....................................
15.....................................  .....................................
16.....................................  .....................................^_
17
18       < Add   > < Next  > <Previous> < Exit  >  < Save > <Cancel>
```

Window name: Hmat
Coordinates: FROM 0,0 TO 0,79
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: M.hmatid	Field	
2: M.hmatname	Field	"@!"
3: m.mfg	Field	
4: m.niin	Field	
5: m.hmcom	Field	
6: m.msds	Field	
7: m.Action	Push button	"@*HN \<Add;\<Next;\<Previous; \?E\<xit"
8: m.Save	Push button	"@*HN \<Save;\<Cancel"

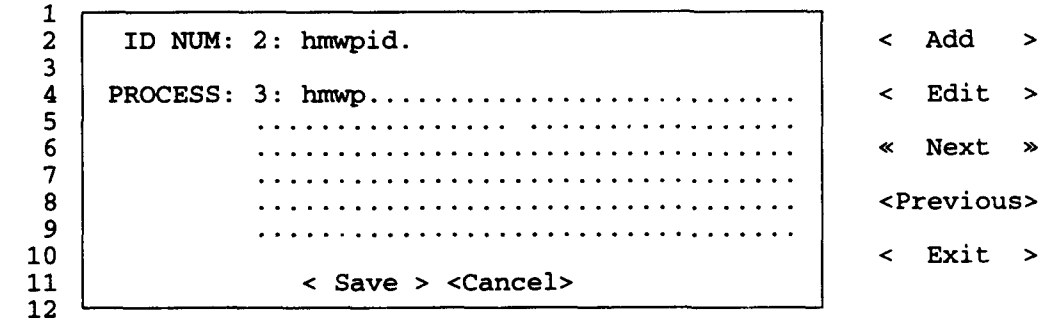
HMLC.SCX **Last updated: 09/15/93 at 13:34**



Window name: Hmlc
Coordinates: FROM 0,0 TO 0,51
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\!\<Next;\<Previous; \?E\<xit"
2: m.hmlcid	Field	
3: m.hmlc	Field	"@!"
4: m.Save	Push button	"@*HN \<Save;\<Cancel"

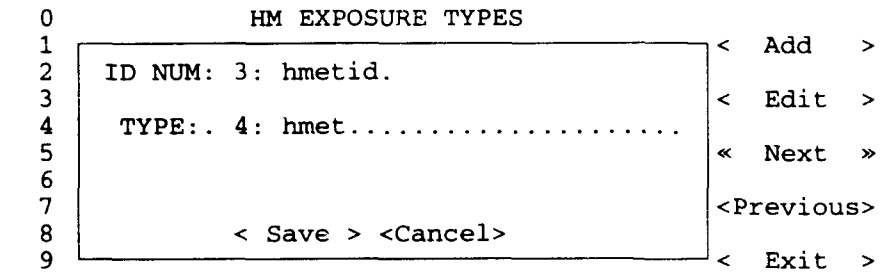
HMWP.SCX **Last updated: 09/15/93 at 13:34**



Window name: Hmwp
Coordinates: FROM 0,0 TO 0,63
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\!\<Next;\<Previous; \?E\<xit"
2: m.hmwpid	Field	
3: m.hmwp	Field	"@!"
4: m.Save	Push button	"@*HN \<Save;\<Cancel"

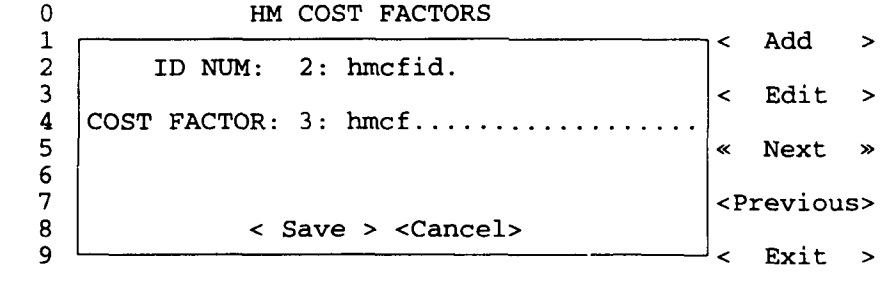
HMET.SCX **Last updated: 09/15/93 at 13:34**



Window name: Hmet
Coordinates: FROM 0,0 TO 0,51
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\!\<Next;\<Previous; \?E\<xit"
2: m.Save	Push button	"@*HN \<Save;\<Cancel"
3: m.hmetid	Field	
4: m.hmet	Field	"@!"

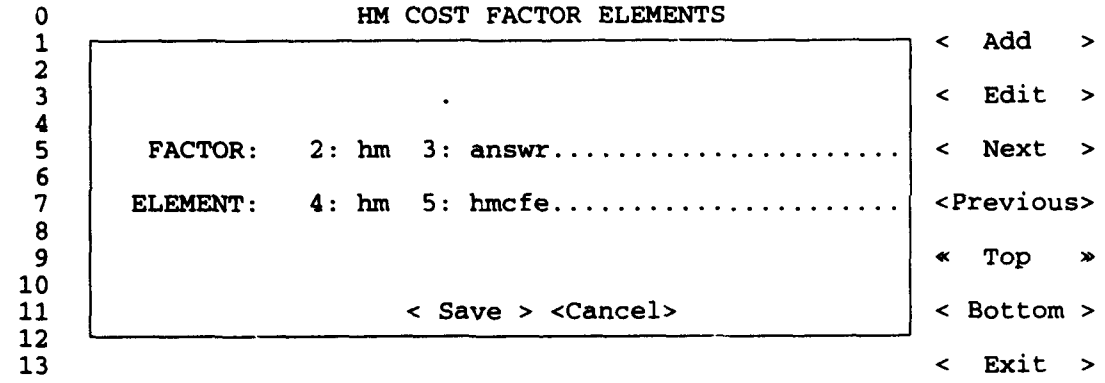
HMCF.SCX **Last updated: 09/15/93 at 13:34**



Window name: Hmcf
Coordinates: FROM 0,0 TO 0,52
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\!\<Next;\<Previous; \?E\<xit"
2: m.hmcfid	Field	
3: m.hmcf	Field	"@!"
4: m.Save	Push button	"@*HN \<Save;\<Cancel"

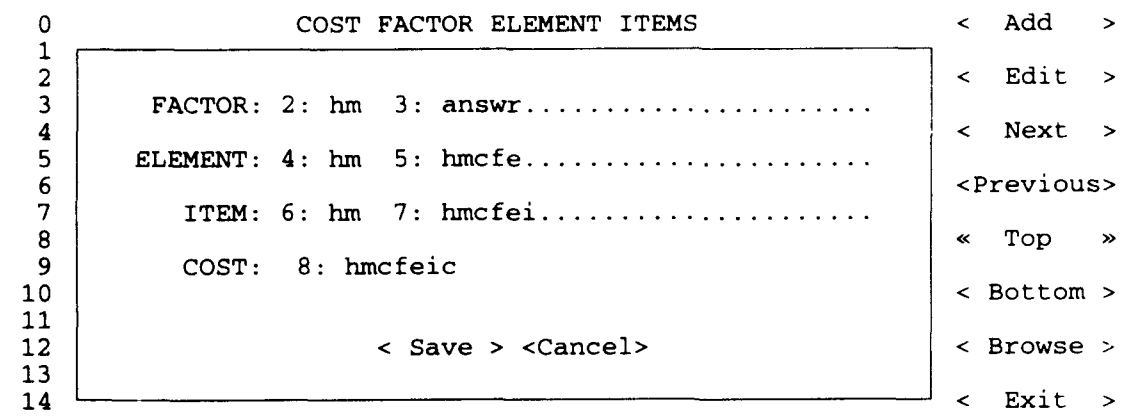
HMCFE.SCX **Last updated: 09/15/93 at 13:34**



Window name: Hmcfe
Coordinates: *FROM 0,0 TO 0,66*
Window options: FLOAT CLOSE MINIMIZE SHADOW

	Name	Type	Picture
1:	m.Action	Push button	"@*VN \<Add;\<Edit;\<Next;\<Previous; \!\<Top;\<Bottom;\?E\<xit"
2:	m.hmcfid	Field	
3:	m.answr	Field	
4:	m.hmcfeid	Field	
5:	m.hmcfe	Field	"@!"
6:	m.Save	Push button	"@*HN \<Save;\<Cancel"

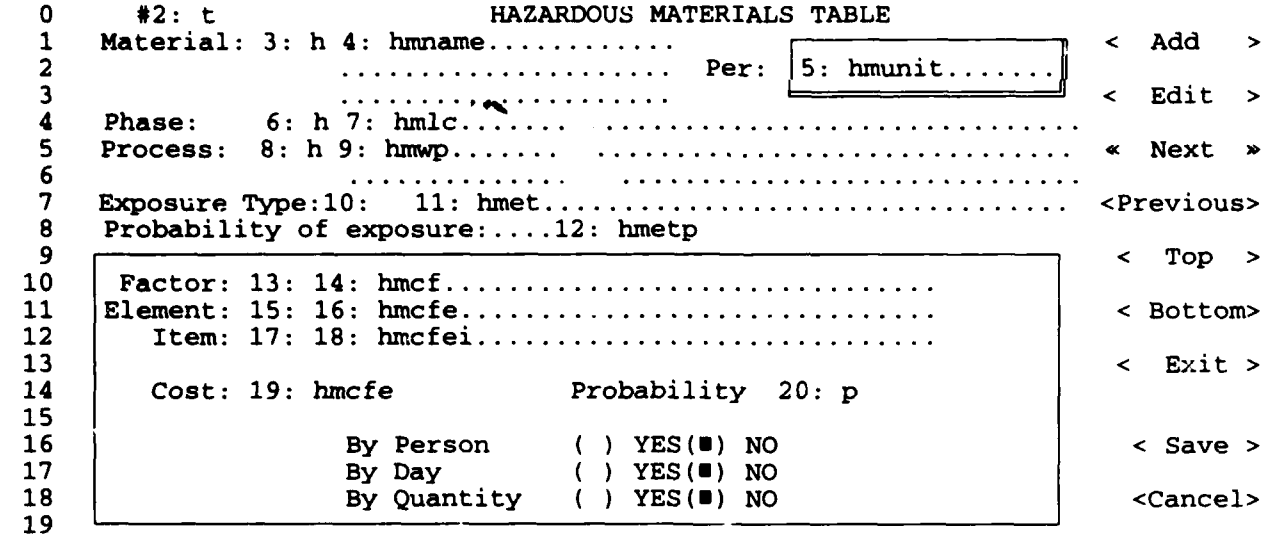
HMCFEI.SCX **Last updated: 09/15/93 at 13:34**



Window name: Hmcfei
Coordinates: FROM 0,0 TO 0,68
Window options: FLOAT CLOSE MINIMIZE SHADOW

	Name	Type	Picture
1:	m.Action	Push button	"@*VN \<Add;\<Edit;\<Next;\<Previous; \!\<Top;\<Bottom;B\<rowse;\?E\<xit"
2:	m.hmcfid	Field	
3:	m.answr	Field	
4:	m.hmcfeid	Field	
5:	m.hmcfe	Field	"@!"
6:	M.hmcfeiid	Field	
7:	M.hmcfei	Field	"@!"
8:	M.hmcfeicost	Field	"@$"
9:	m.Save	Push button	"@*HN \<Save;\<Cancel"

HMTAB.SCX Last updated: 09/15/93 at 13:34



Window name: Hmtab
Coordinates: FROM 0,0 TO 0,78
Window options: FLOAT CLOSE MINIMIZE SHADOW

	Name	Type	Picture
1:	m.Action	Push button	"@*VN \<Add;\<Edit;\.\<Next;\<Previous; \<Top;\<Bottom;\?E\<xit"
2:	m.tabid	Field	
3:	m.hmatid	Field	"@Z"
4:	m.hmname	Field	"@!"
5:	m.hmunit	Popup	"@^ "
6:	m.hmlcid	Field	"@Z"
7:	m.hmlc	Field	"@!"
8:	m.hmwpid	Field	"@Z"
9:	m.hmwp	Field	"@!"
10:	m.hmetid	Field	"@Z"
11:	m.hmet	Field	"@!"
12:	m.hmetprob	Field	
13:	m.hmcfid	Field	"@Z"
14:	m.hmcf	Field	
15:	m.hmcfeid	Field	"@Z"
16:	m.hmcfe	Field	"@!"
17:	m.hmcfeiid	Field	"@Z"
18:	m.hmcfei	Field	"@!"
19:	m.hmcfecost	Field	"@$$$,$$$.99"
20:	m.prob	Field	"9.99"
21:	m.perp	Radio button	"@*RHN YES;NO"
22:	m.perd	Radio button	"@*RHN YES;NO"
23:	m.perq	Radio button	"@*RHN YES;NO"
24:	m.Save	Push button	"@*VN \<Save;\<Cancel"

HMCOMP.SCX Last updated: 12/01/93 at 9:52

COMPUTE COST VALUE

```
 0  Estimated                                                    Estimated
 1   Cost                  Scenario:1:  2: chmscname...          Variance
 2
 3          for Product:  3: hmat.....................................
 4
 5
 6   [ ] . for Step:      4: step.....................................   [ ] .
 7
 8
 9   [ ] . for Factor:    7: fact.....................................   [ ] .
10
11
12   [ ] . for Factor:    10: wfact........ at step: 11: wstep....       [ ] .
13
14
15   [ ] . for Phase:     14: phase...................................   [ ] .
16
17   # of Iterations      Total of Scenario cost:   ( ) Yes   (■) No
18   18: sample..
19                          < Ok >   <Browse>   <Cancel>
```

Window name: W_hmcost
Coordinates: FROM 0,0 TO 0,75
Window options: FLOAT CLOSE MINIMIZE SHADOW

	Name	Type	Picture
1:	m.chmscid	Field	"@Z"
2:	m.chmscname	Field	"@!T"
3:	m.hmat	Popup	"@^ "
4:	m.step	Popup	"@^ "
5:	m.cstep	Check box	"@*C "
6:	m.bstep	Check box	"@*C "
7:	m.fact	Popup	"@^ "
8:	m.cfact	Check box	"@*C "
9:	m.bfact	Check box	"@*C "
10:	m.wfact	Popup	"@^ "
11:	m.wstep	Popup	"@^ "
12:	m.cwstep	Check box	"@*C "
13:	m.bwstep	Check box	"@*C "
14:	m.phase	Popup	"@^ "
15:	m.cphase	Check box	"@*C "
16:	m.bphase	Check box	"@*C "
17:	m.scost	Radio button	"@*RHN Yes ;No"
18:	m.sample	Field	"@Z"
19:	m.action	Push button	"@*HN \<Ok;\<Browse;\<Cancel"

W_PRINT.SCX **Last updated: 09/15/93 at 13:34**

Print Option

```
 0
 1  Print to:.       (■) File
 2                   ( ) Printer
 3
 4
 5
 6  Filename: 1: mprintfile.............
 7
 8
 9
10           <  OK  >   <Cancel>
```

Window name: W_prn
Coordinates: FROM 0,0 TO 0,42
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mprintfile	Field	
2: mbuttons	Push button	"@*HT OK;\?Cancel"
3: mchoice	Radio button	"@*RVN File;Printer"

Section IV. Data Dictionary. There are eighteen database files in the HMLCCM system:

```
HMAT.DBF    -- Hazardous Materials File
HMCOM.DBF   -- Hazaradous Material Common Name Table
HMLC.DBF    -- Hazardous Material Life Cycle Table
HMWP.DBF    -- Hazardous Material Work Process Table
HMET.DBF    -- Hazardous Material Exposure Type Table
HMCF.DBF    -- Hazardous Material Cost Factor Table
HMCFE.DBF   -- Hazardous Material Cost Factor Element Table
HMCFEI.DBF  -- Hazardous Material Cost Factor Element Item Table
HMUNIT.DBF  -- Hazardous Material Unit Table
HMTAB.DBF   -- Hazardous Material Scenario Table
HMSTEP.DBF  -- Hazardous Material Step Table
HMCOMP.DBF  -- Hazardous Material Cost Computation File
HMSCEN.DBF  -- Hazardous Material Scenario File
BOOT.DBF    -- Temporary file used for the bootstrap computations
HMTEMP.DBF  -- Temporary file used to store data from tables
TEMP.DBF    -- Temporary file used to store data from tables
TEST.DBF    -- Temporary file used to store data from tables
TLV         -- Threshhold Limit Values for OSHA Z-Table
```

A. Database Structure Summary.

Structure for database : **HMAT.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMATID	Numeric	4		1	4
2	HMATNAME	Character	45		5	49
3	MFG	Character	25		50	74
4	NIIN	Character	9		75	83
5	MILSPEC	Character	11		84	94
6	HMCOST	Numeric	10	2	95	104
7	HMUNIT	Character	10		105	114
8	HMDESCRIPT	Memo	10		115	124
9	CHEM_PHY	Memo	10		125	134
10	HEALTH_HAZ	Memo	10		135	144
11	MED_SURV	Memo	10		145	154
12	OCC_EXP	Memo	10		155	164
13	PPE_TREAT	Memo	10		165	174
14	SPEC_TESTS	Memo	10		175	184
15	TREATMENT	Memo	10		185	194
16	SYN_TRADE	Memo	10		195	204
17	COM_USES	Memo	10		205	214
18	EXP_LIMITS	Memo	10		215	224
19	MSDS	Memo	10		225	234
** Total **			235			

This database is associated with the memo file: HMAT.FPT

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMAT.CDX

Used by: HMAT.SPR
: HMLU.PRG
: GET_HMCOMID() (function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
: GET_HMNAME (procedure in C:\HAZMAT\GHM\WORK\HMLU.PRG)

Structure for database : **HMCOM.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMCOMID	Numeric	4		1	4
2	COMMON	Character	30		5	34
3	NIIN	Character	9		35	43
** Total **			44			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMCOM.CDX

Used by: COMNAME() (function in C:\HAZMAT\GHM\WORK\HMAT.SPR)
: SAVECOM (procedure in C:\HAZMAT\GHM\WORK\HMAT.SPR)
: GET_HMCOMID() (function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
: COMPUTSTEP() (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: COMPUTMAT() (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: COMPUTFACT() (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: COMPUTWFACT() (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)

```
       : COMPUTPHASE()            (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
       : BSTEP()                  (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
       : BHMAT()                  (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
       : BFACT()                  (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
       : BWSTEP()                 (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
       : BPHASE()                 (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)

Structure for database : HMLC.DBF

Field  Field name  Type        Width    Dec    Start      End
    1  HMLCID      Numeric         2              1        2
    2  HMLC        Character      30              3       32
** Total **                      33

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
       : C:\HAZMAT\GHM\WORK\HMLC.CDX

Used by: HMLC.SPR


Structure for database : HMWP.DBF

Field  Field name  Type        Width    Dec    Start      End
    1  HMWPID      Numeric         2              1        2
    2  HMWP        Character      80              3       82
    3  HMPLATFORM  Numeric         2             83       84
** Total **                      85

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
       : C:\HAZMAT\GHM\WORK\HMWP.CDX

Used by: HMWP.SPR


Structure for database : HMET.DBF

Field  Field name  Type        Width    Dec    Start      End
    1  HMETID      Numeric         2              1        2
    2  HMET        Character      30              3       32
** Total **                      33

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
       : C:\HAZMAT\GHM\WORK\HMET.CDX

Used by: HMET.SPR
```

```
Structure for database : HMCF.DBF        Alias: CFTMP

Field  Field name  Type         Width     Dec     Start       End
    1  HMCFID      Numeric          2               1         2
    2  HMCF        Character       30               3        32
** Total **                       33

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
       : C:\HAZMAT\GHM\WORK\HMCF.CDX

Used by: HMCF.SPR
       : REL()                    (function in C:\HAZMAT\GHM\WORK\HMCFE.SPR)
       : GET_HMCFE                (procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
       : GET_CFE                  (procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
       : GET_CFEID                (procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
       : BROWSEITEM               (procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
       : GET_CFAR1()              (function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
       : GET_CF1()                (function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
       : GET_EID()                (function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)

Structure for database : HMCFE.DBF       Alias: CFETMP

Field  Field name  Type         Width     Dec     Start       End
    1  HMCFEID     Numeric          2               1         2
    2  HMCFID      Numeric          2               3         4
    3  HMCFE       Character       30               5        34
** Total **                       35

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
       : C:\HAZMAT\GHM\WORK\HMCFE.CDX

Used by: HMCFE.SPR
       : REL()                    (function in C:\HAZMAT\GHM\WORK\HMCFE.SPR)
       : BROWSEITEM               (procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
       : GET_EI()                 (function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
       : GET_EIID()               (function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
```

Structure for database : **HMCFEI.DBF** Alias: CFEITMP

Field	Field name	Type	Width	Dec	Start	End
1	HMCFEIID	Numeric	4		1	4
2	HMCFEID	Numeric	2		5	6
3	HMCFID	Numeric	2		7	8
4	HMCFEI	Character	50		9	58
5	HMCFEINO	Character	15		59	73
6	HMCFEICOST	Numeric	10	2	74	83
7	HMCFEIUNIT	Character	10		84	93
** Total **			94			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMCFEI.CDX

Used by: HMCFEI.SPR
: REL() (function in C:\HAZMAT\GHM\WORK\HMCFE.SPR)
: BROWSEITEM (procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)

Structure for database : **HMUNIT.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMUNIT	Character	20		1	20
** Total **			21			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMUNIT.CDX

Used by: HMTAB.SPR

Structure for database : **HMTAB.DBF** Alias: HMTAB

Field	Field name	Type	Width	Dec	Start	End
1	TABID	Numeric	4		1	4
2	HMCOMID	Numeric	4		5	8
3	HMATID	Numeric	4		9	12
4	HMLCID	Numeric	2		13	14
5	HMWPID	Numeric	2		15	16
6	HMETID	Numeric	2		17	18
7	HMETPROB	Float	5		19	23
8	HMCFID	Numeric	2		24	25
9	HMCFEID	Numeric	2		26	27
10	HMCFEIID	Numeric	4		28	29
11	PERP	Numeric	1		30	30
12	PERD	Numeric	1		31	31
13	PERQ	Numeric	1		32	32
14	HMCFECOST	Numeric	5	2	33	37
15	HMUNIT	Character	10		38	47
16	WTAVERAGE	Float	5		48	52
17	PROB	Float	5		53	57
** Total **			58			

FoxDoc did not find any associated index files

FoxDoc did not find any associated multiple indexes

Used by: HMTAB.SPR
: CHANGE (procedure in C:\HAZMAT\GHM\WORK\HMAT.SPR)

Structure for database : **HMSTEP.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMSCID	Numeric	4		1	4
2	HMSTEP	Numeric	4		5	8
3	HMATID	Numeric	4		9	12
4	HMLCID	Numeric	2		13	14
5	HMWPID	Numeric	2		15	16
6	PERNUM	Numeric	3		17	19
7	DURNUM	Numeric	3		20	22
8	QTYNUM	Numeric	3		23	25
9	WSTNUM	Numeric	3		26	28
10	UNIT	Character	20		29	48
** Total **			49			

Used by: GET_HMARRAY (procedure in HMCOMP.SPR)
: HMSTEP.PRG

Structure for database : **HMCOMP.DBF** Status: Temporary

Used by: HMCOMP.SPR
: COMPUTSTEP() (function in HMCOMP.SPR)
: COMPUTFACT() (function in HMCOMP.SPR)
: COMPUTWFACT() (function in HMCOMP.SPR)
: COMPUTPHASE() (function in HMCOMP.SPR)
: BSTEP() (function in HMCOMP.SPR)
: BFACT() (function in HMCOMP.SPR)
: BWSTEP() (function in HMCOMP.SPR)
: BPHASE() (function in HMCOMP.SPR)

Structure for database : **HMSCEN.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMSCID	Numeric	4		1	4
2	HMSCNAME	Character	40		5	44
** Total **			45			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMSCEN.CDX

Used by: HMSCEN.PRG
: SCSAVE (procedure in C:\HAZMAT\GHM\WORK\HMSTEP.PRG)

Structure for database: **BOOT.DBF** Status: Temporary File

Used by: BSTEP() (function in HMCOMP.SPR)
: BFACT() (function in HMCOMP.SPR)
: BWSTEP() (function in HMCOMP.SPR)
: BPHASE() (function in HMCOMP.SPR)

Structure for database : **HMTEMP.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMSCID	Numeric	4		1	4
2	HMSTEP	Numeric	4		5	8
3	HMATID	Numeric	4		9	12
4	HMLCID	Numeric	2		13	14
5	HMWPID	Numeric	2		15	16
6	PERNUM	Numeric	3		17	19
7	DURNUM	Numeric	3		20	22
8	QTYNUM	Numeric	3		23	25
9	WSTNUM	Numeric	3		26	28
10	UNIT	Character	20		29	48
** Total **			49			

Used by: SCSAVE (procedure in HMSTEP.PRG)

Structure for database : **TEMP.DBF** Status: Temporary File

Used by: GET_TABLE (procedure in HMCOMP.SPR)

Structure for database : **TEST.DBF** Status: Temporary File

Used by: CALCULATE() (function in HMCOMP.SPR)

Structure for database : **TLV.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	SUBSTANCE	Character	40		1	40
2	GENERIC	Character	35		41	75
3	CHEMICAL	Character	35		76	110
4	CAS_NO	Character	12		111	122
5	NAV_TWA	Character	6		123	128
6	NAV_STEL	Character	6		129	134
7	NAV_CEIL	Character	6		135	140
8	NAV_SOURCE	Character	5		141	145
9	TLV_TWA	Character	6		146	151
10	TLV_C	Character	6		152	157
11	SKIN	Logical	1		158	158
12	TLV_STEL	Character	6		159	164
13	APPENDIX	Character	2		165	166
14	PEL_TWA	Character	6		167	172
15	PEL_C	Character	6		173	178
16	PEL_STEL	Character	6		179	184
17	OSHA_SKIN	Logical	1		185	185
18	PTLV_TWA	Character	6		186	191
19	PTLV_STEL	Character	6		192	197
20	PTLV_C	Character	6		198	203
21	NOTIFY	Logical	1		204	204
22	BASIS	Character	35		205	239
23	REPRO	Character	10		240	249
24	S_BASIS	Character	10		250	259
** Total **			260			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\TLV.CDX

Used by: HMZTAB.PRG

B. Database Field Summary

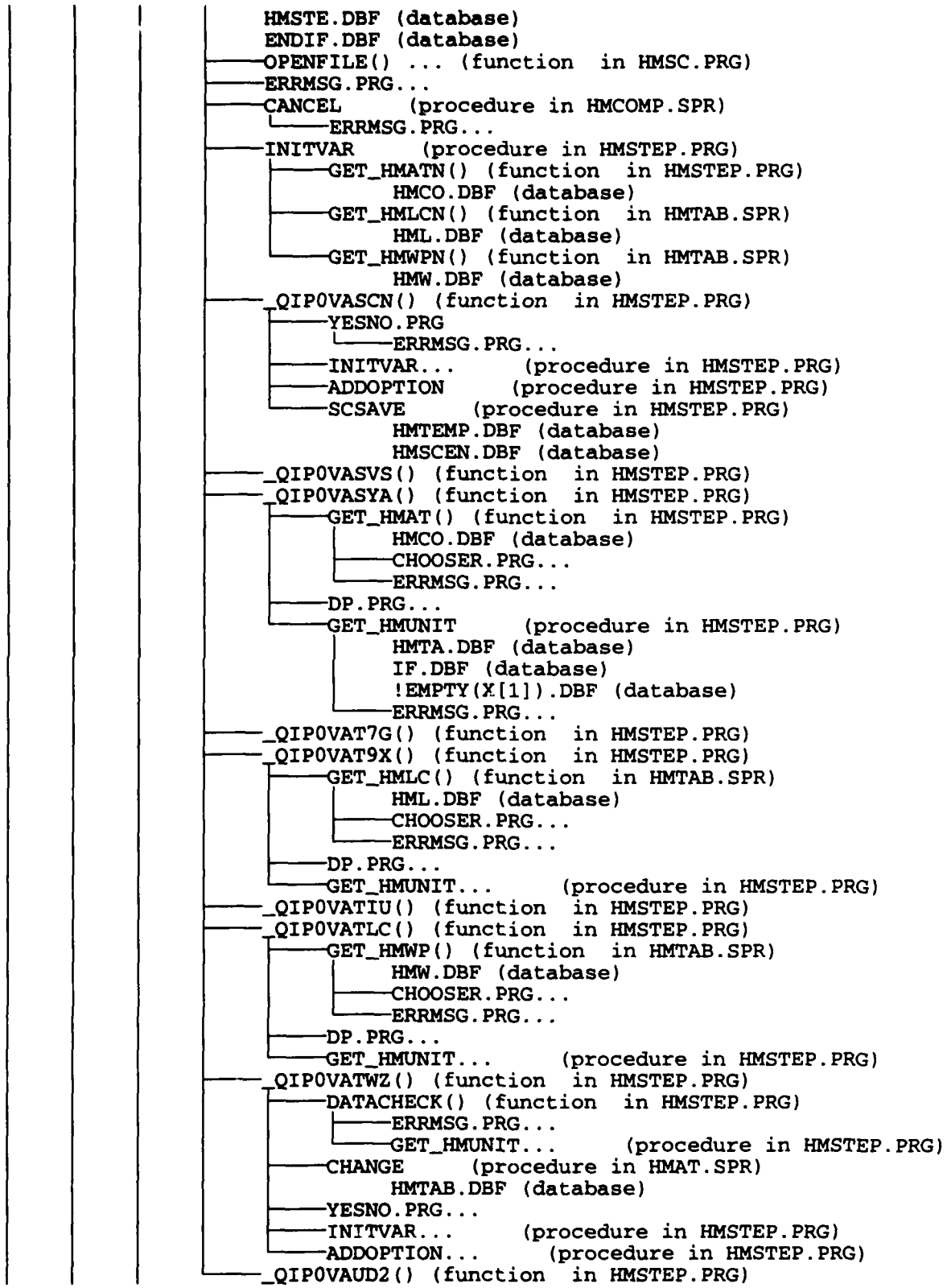
Field Name	Type	Len	Dec	Database
APPENDIX	C	2	0	TLV.DBF
BASIS	C	35	0	TLV.DBF
CAS_NO	C	12	0	TLV.DBF
CHEMICAL	C	35	0	TLV.DBF
CHEM_PHY	M	10	0	HMAT.DBF
COMMON	C	30	0	HMCOM.DBF
COM_USES	M	10	0	HMAT.DBF
EXP_LIMITS	M	10	0	HMAT.DBF

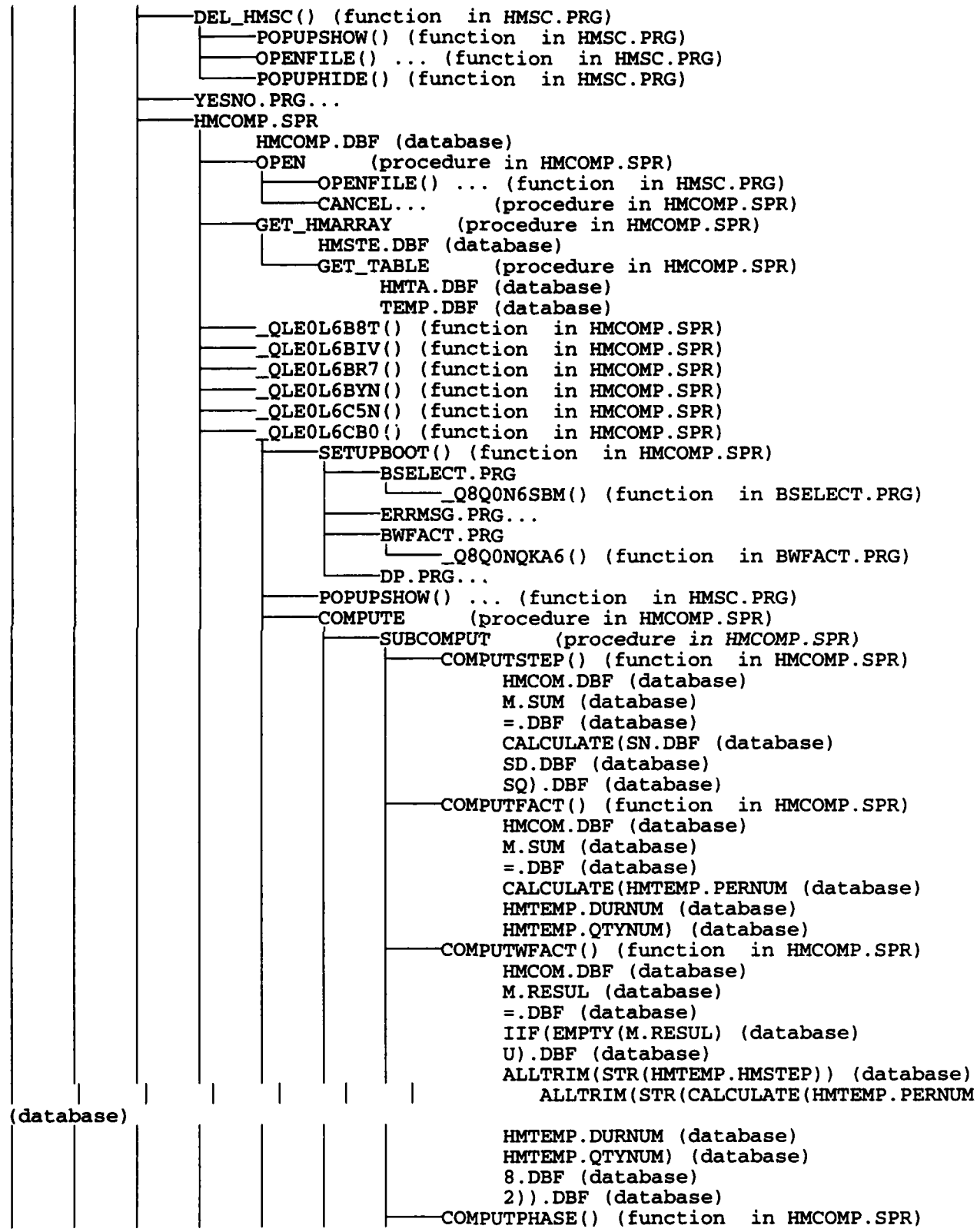
Field Name	Type	Len	Dec	Database
GENERIC	C	35	0	TLV.DBF
HEALTH_HAZ	M	10	0	HMAT.DBF
HMATID	N	4	0	HMAT.DBF
				HMTAB.DBF
HMATNAME	C	45	0	HMAT.DBF
HMCF	C	30	0	HMCF.DBF
HMCFE	C	30	0	HMCFE.DBF
HMCFECOST	N	5	2	HMTAB.DBF
HMCFEI	C	50	0	HMCFEI.DBF
HMCFEICOST	N	10	2	HMCFEI.DBF
HMCFEID	N	2	0	HMCFEI.DBF
				HMTAB.DBF
				HMCFE.DBF
HMCFEIID	N	4	0	HMCFEI.DBF
HMCFEIID	N	4	0	HMTAB.DBF
HMCFEINO	C	15	0	HMCFEI.DBF
HMCFEIUNIT	C	10	0	HMCFEI.DBF
HMCFID	N	2	0	HMCFEI.DBF
				HMTAB.DBF
				HMCFE.DBF
				HMCF.DBF
HMCOMID	N	4	0	HMTAB.DBF
				HMCOM.DBF
HMCOST	N	10	2	HMAT.DBF
HMDESCRIPT	M	10	0	HMAT.DBF
HMET	C	30	0	HMET.DBF
HMETID	N	2	0	HMTAB.DBF
				HMET.DBF
HMETPROB	F	5	3	HMTAB.DBF
HMLC	C	30	0	HMLC.DBF
HMLCID	N	2	0	HMLC.DBF
				HMTAB.DBF
HMPLATFORM	N	2	0	HMWP.DBF
HMSCID	N	4	0	HMSCEN.DBF
HMSCNAME	C	40	0	HMSCEN.DBF
HMUNIT	C	10	0	HMAT.DBF
				HMTAB.DBF
HMUNIT	C	20	0	HMUNIT.DBF
HMWP	C	80	0	HMWP.DBF
HMWPID	N	2	0	HMTAB.DBF
				HMWP.DBF
MED_SURV	M	10	0	HMAT.DBF
MFG	C	25	0	HMAT.DBF
MILSPEC	C	11	0	HMAT.DBF

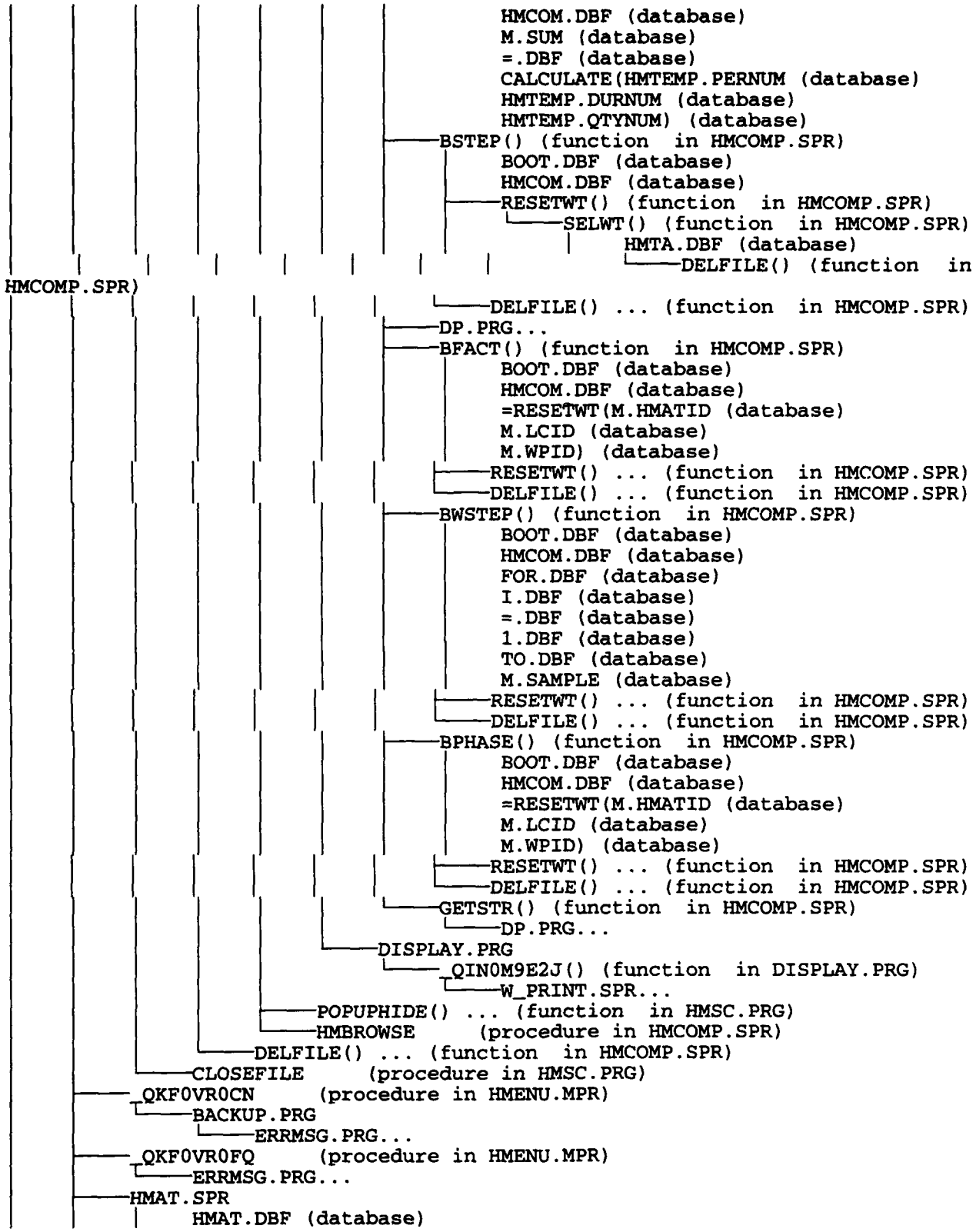
Field Name	Type	Len	Dec	Database
MSDS	M	10	0	HMAT.DBF
NAV_CEIL	C	6	0	TLV.DBF
NAV_SOURCE	C	5	0	TLV.DBF
NAV_STEL	C	6	0	TLV.DBF
NAV_TWA	C	6	0	TLV.DBF
NIIN	C	9	0	HMCOM.DBF
				HMAT.DBF
NOTIFY	L	1	0	TLV.DBF
OCC_EXP	M	10	0	HMAT.DBF
OSHA_SKIN	L	1	0	TLV.DBF
PEL_C	C	6	0	TLV.DBF
PEL_STEL	C	6	0	TLV.DBF
PEL_TWA	C	6	0	TLV.DBF
PERD	N	1	0	HMTAB.DBF
PERP	N	1	0	HMTAB.DBF
PERQ	N	1	0	HMTAB.DBF
PPE_TREAT	M	10	0	HMAT.DBF
PROB	F	5	2	HMTAB.DBF
PTLV_C	C	6	0	TLV.DBF
PTLV_STEL	C	6	0	TLV.DBF
PTLV_TWA	C	6	0	TLV.DBF
REPRO	C	10	0	TLV.DBF
SKIN	L	1	0	TLV.DBF
SPEC_TESTS	M	10	0	HMAT.DBF
SUBSTANCE	C	40	0	TLV.DBF
SYN_TRADE	M	10	0	HMAT.DBF
S_BASIS	C	10	0	TLV.DBF
TABID	N	4	0	HMTAB.DBF
TLV_C	C	6	0	TLV.DBF
TLV_STEL	C	6	0	TLV.DBF
TLV_TWA	C	6	0	TLV.DBF
TREATMENT	M	10	0	HMAT.DBF
WTAVERAGE	F	5	2	HMTAB.DBF

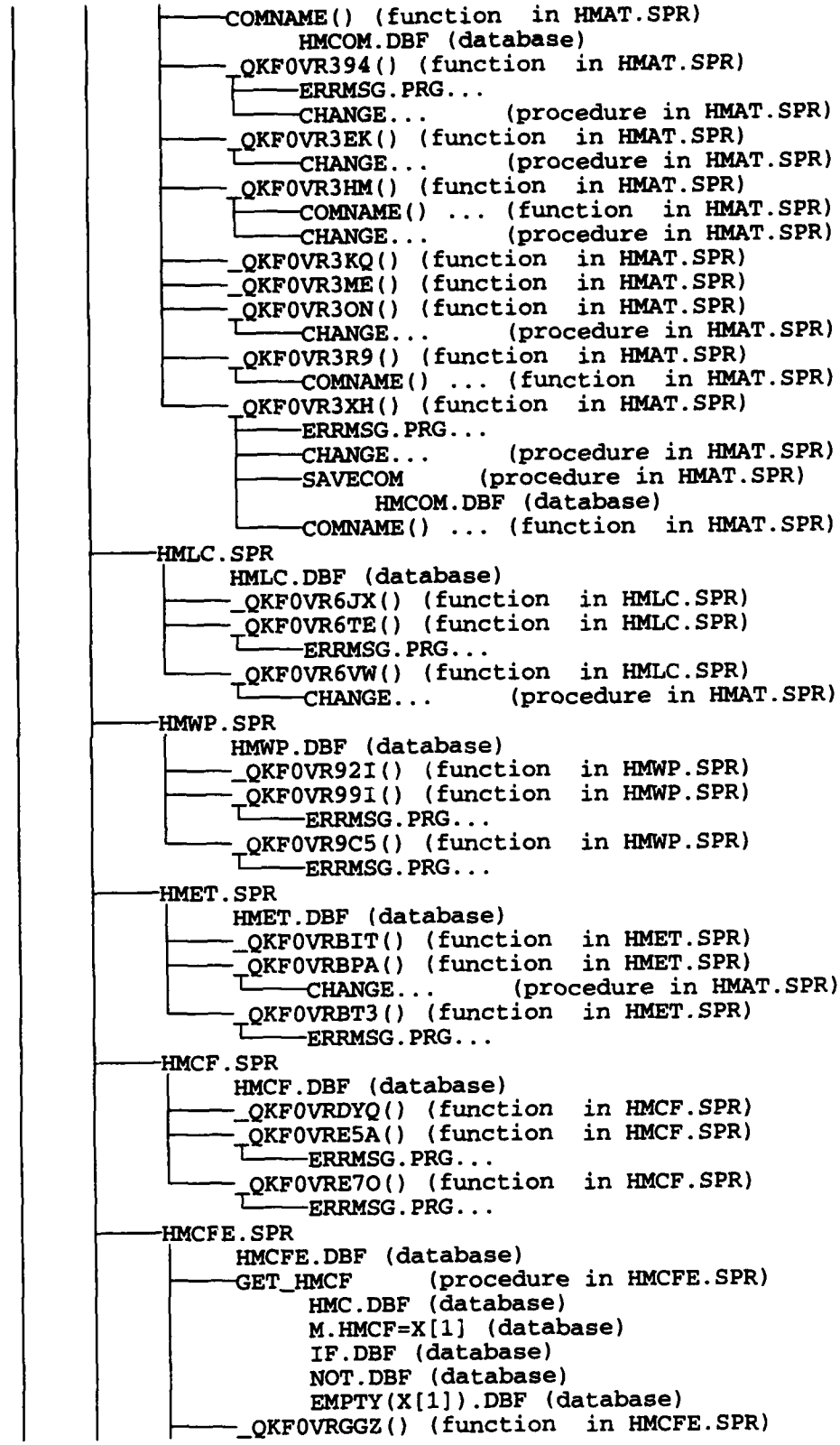
Section V. Tree Diagram. The tree diagram lists each program that is called by the order in which it is used. Under each program there is a list of all functions called and the where the procedure/function is stored.

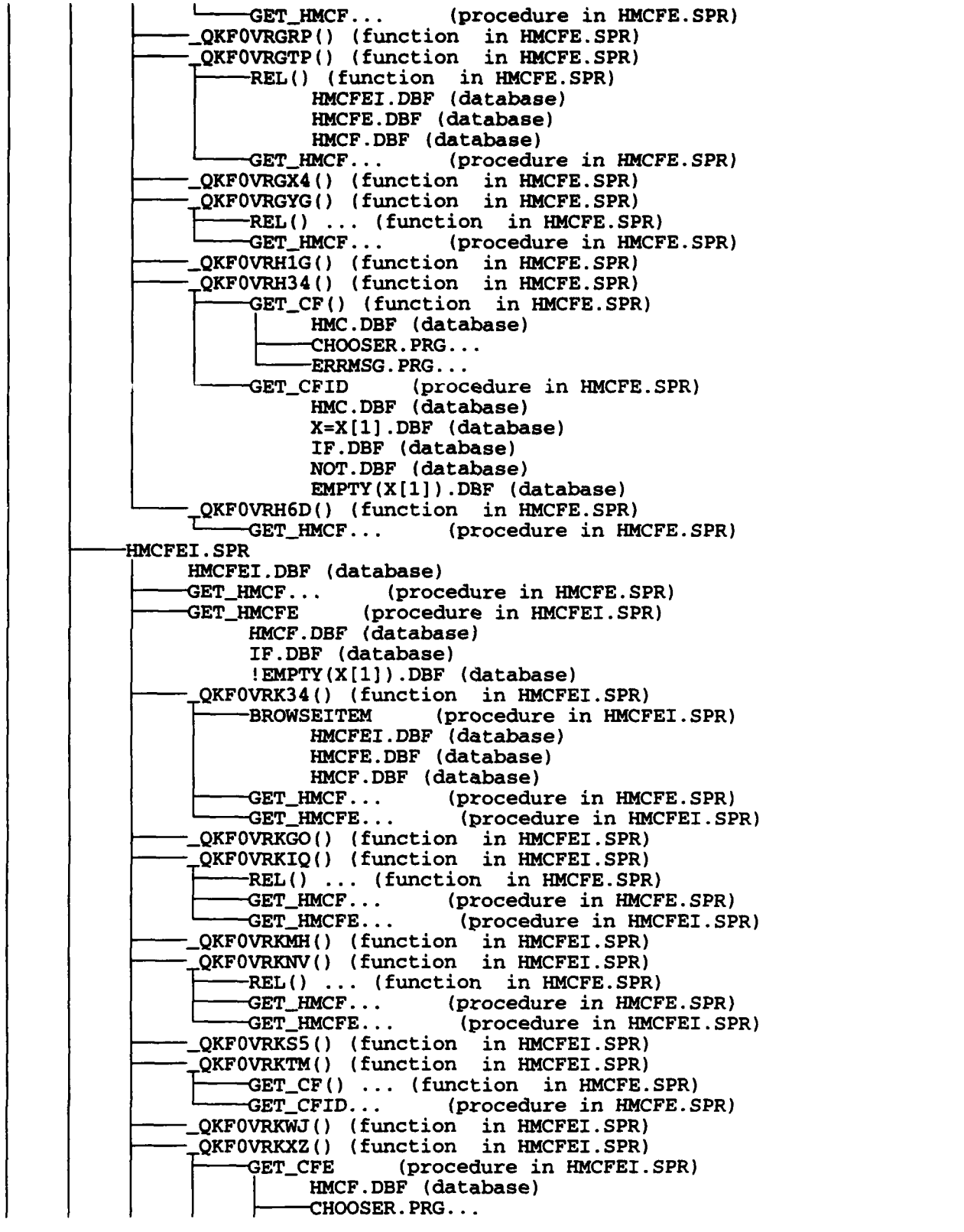
```
HMINIT.PRG
├──HMGETPWD.PRG
│  └──ERRMSG.PRG
├──BIGCHARS.PRG
├──HMENU.MPR
│  ├──HMZTAB.PRG
│  │     TLV.DBF (database)
│  ├──HMMSDS.PRG
│  │  ├──OPENFILE() (function  in HMSC.PRG)
│  │  │  └──FILEFIND() (function  in HMSC.PRG)
│  │  │     └──DP.PRG
│  │  ├──DP.PRG...
│  │  ├──HMLU.PRG
│  │  │     HMAT.DBF (database)
│  │  │  ├──_Q7D0O1EI8() (function  in HMLU.PRG)
│  │  │  │  ├──GET_HMNAME      (procedure in HMLU.PRG)
│  │  │  │  │     HMA.DBF (database)
│  │  │  │  │     ELSE.DBF (database)
│  │  │  │  │     ENDIF.DBF (database)
│  │  │  │  │  ├──CHOOSER.PRG
│  │  │  │  │  │  ├──_Q120IDFTF() (function  in CHOOSER.PRG)
│  │  │  │  │  │  ├──_Q120IDG17() (function  in CHOOSER.PRG)
│  │  │  │  │  │  └──_Q120IDG86() (function  in CHOOSER.PRG)
│  │  │  │  │  └──ERRMSG.PRG...
│  │  │  │  └──DP.PRG...
│  │  │  └──_Q7D0O1EUB() (function  in HMLU.PRG)
│  │  └──HMVIEW() (function  in HMREF.PRG)
│  │     ├──NEWLINE() (function  in HMAT.SPR)
│  │     └──MEMOWIN.PRG
│  │        ├──_QIG0IXOXD() (function  in MEMOWIN.PRG)
│  │        │  └──W_PRINT.SPR
│  │        │     ├──_QKF0VS4NC() (function  in W_PRINT.SPR)
│  │        │     ├──_QKF0VS4PH() (function  in W_PRINT.SPR)
│  │        │     │  └──PRINTFILE() (function  in W_PRINT.SPR)
│  │        │     └──_QKF0VS4RM() (function  in W_PRINT.SPR)
│  │        └──_QIG0IXPJO() (function  in MEMOWIN.PRG)
│  │           └──W_PRINT.SPR...
│  └──HMSC.PRG
│     ├──HMSCEN.PRG
│     │     HMSCEN.DBF (database)
│     │  ├──_Q8K0INGJH() (function  in HMSCEN.PRG)
│     │  │  ├──GET_HMSC() (function  in HMSCEN.PRG)
│     │  │  │     HMSCE.DBF (database)
│     │  │  │  └──CHOOSER.PRG...
│     │  │  ├──MEVENT      (procedure in HMSCEN.PRG)
│     │  │  └──DP.PRG...
│     │  ├──_Q8K0INGTV() (function  in HMSCEN.PRG)
│     │  │  ├──ERRMSG.PRG...
│     │  │  ├──GET_HMSC() ... (function  in HMSCEN.PRG)
│     │  │  ├──MEVENT...      (procedure in HMSCEN.PRG)
│     │  │  └──DP.PRG...
│     │  └──_Q8K0INHBV() (function  in HMSCEN.PRG)
│     │     └──MEVENT...      (procedure in HMSCEN.PRG)
│     ├──DP.PRG...
│     └──HMSTEP.PRG
```


HMSTE.DBF (database)
ENDIF.DBF (database)
OPENFILE() ... (function in HMSC.PRG)
ERRMSG.PRG...
CANCEL (procedure in HMCOMP.SPR)
ERRMSG.PRG...
INITVAR (procedure in HMSTEP.PRG)
GET_HMATN() (function in HMSTEP.PRG)
HMCO.DBF (database)
GET_HMLCN() (function in HMTAB.SPR)
HML.DBF (database)
GET_HMWPN() (function in HMTAB.SPR)
HMW.DBF (database)
_QIP0VASCN() (function in HMSTEP.PRG)
YESNO.PRG
ERRMSG.PRG...
INITVAR... (procedure in HMSTEP.PRG)
ADDOPTION (procedure in HMSTEP.PRG)
SCSAVE (procedure in HMSTEP.PRG)
HMTEMP.DBF (database)
HMSCEN.DBF (database)
_QIP0VASVS() (function in HMSTEP.PRG)
_QIP0VASYA() (function in HMSTEP.PRG)
GET_HMAT() (function in HMSTEP.PRG)
HMCO.DBF (database)
CHOOSER.PRG...
ERRMSG.PRG...
DP.PRG...
GET_HMUNIT (procedure in HMSTEP.PRG)
HMTA.DBF (database)
IF.DBF (database)
!EMPTY(X[1]).DBF (database)
ERRMSG.PRG...
_QIP0VAT7G() (function in HMSTEP.PRG)
_QIP0VAT9X() (function in HMSTEP.PRG)
GET_HMLC() (function in HMTAB.SPR)
HML.DBF (database)
CHOOSER.PRG...
ERRMSG.PRG...
DP.PRG...
GET_HMUNIT... (procedure in HMSTEP.PRG)
_QIP0VATIU() (function in HMSTEP.PRG)
_QIP0VATLC() (function in HMSTEP.PRG)
GET_HMWP() (function in HMTAB.SPR)
HMW.DBF (database)
CHOOSER.PRG...
ERRMSG.PRG...
DP.PRG...
GET_HMUNIT... (procedure in HMSTEP.PRG)
_QIP0VATWZ() (function in HMSTEP.PRG)
DATACHECK() (function in HMSTEP.PRG)
ERRMSG.PRG...
GET_HMUNIT... (procedure in HMSTEP.PRG)
CHANGE (procedure in HMAT.SPR)
HMTAB.DBF (database)
YESNO.PRG...
INITVAR... (procedure in HMSTEP.PRG)
ADDOPTION... (procedure in HMSTEP.PRG)
_QIP0VAUD2() (function in HMSTEP.PRG)

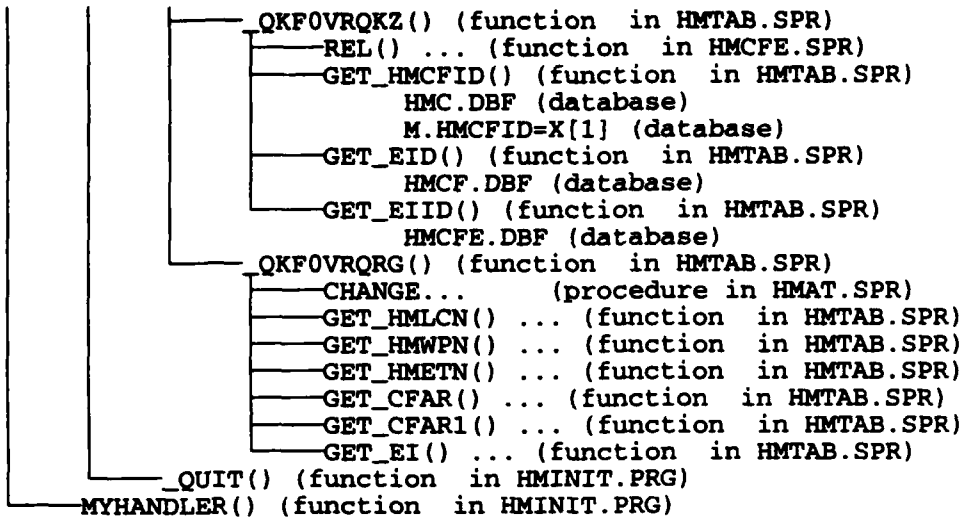

DEL_HMSC() (function in HMSC.PRG)
POPUPSHOW() (function in HMSC.PRG)
OPENFILE() ... (function in HMSC.PRG)
POPUPHIDE() (function in HMSC.PRG)
YESNO.PRG...
HMCOMP.SPR
HMCOMP.DBF (database)
OPEN (procedure in HMCOMP.SPR)
OPENFILE() ... (function in HMSC.PRG)
CANCEL... (procedure in HMCOMP.SPR)
GET_HMARRAY (procedure in HMCOMP.SPR)
HMSTE.DBF (database)
GET_TABLE (procedure in HMCOMP.SPR)
HMTA.DBF (database)
TEMP.DBF (database)
_QLE0L6B8T() (function in HMCOMP.SPR)
_QLE0L6BIV() (function in HMCOMP.SPR)
_QLE0L6BR7() (function in HMCOMP.SPR)
_QLE0L6BYN() (function in HMCOMP.SPR)
_QLE0L6C5N() (function in HMCOMP.SPR)
_QLE0L6CB0() (function in HMCOMP.SPR)
SETUPBOOT() (function in HMCOMP.SPR)
BSELECT.PRG
_Q8Q0N6SBM() (function in BSELECT.PRG)
ERRMSG.PRG...
BWFACT.PRG
_Q8Q0NQKA6() (function in BWFACT.PRG)
DP.PRG...
POPUPSHOW() ... (function in HMSC.PRG)
COMPUTE (procedure in HMCOMP.SPR)
SUBCOMPUT (procedure in HMCOMP.SPR)
COMPUTSTEP() (function in HMCOMP.SPR)
HMCOM.DBF (database)
M.SUM (database)
=.DBF (database)
CALCULATE(SN.DBF (database)
SD.DBF (database)
SQ).DBF (database)
COMPUTFACT() (function in HMCOMP.SPR)
HMCOM.DBF (database)
M.SUM (database)
=.DBF (database)
CALCULATE(HMTEMP.PERNUM (database)
HMTEMP.DURNUM (database)
HMTEMP.QTYNUM) (database)
COMPUTWFACT() (function in HMCOMP.SPR)
HMCOM.DBF (database)
M.RESUL (database)
=.DBF (database)
IIF(EMPTY(M.RESUL) (database)
U).DBF (database)
ALLTRIM(STR(HMTEMP.HMSTEP)) (database)
ALLTRIM(STR(CALCULATE(HMTEMP.PERNUM
(database)
HMTEMP.DURNUM (database)
HMTEMP.QTYNUM) (database)
8.DBF (database)
2)).DBF (database)
COMPUTPHASE() (function in HMCOMP.SPR)


HMCOM.DBF (database)
M.SUM (database)
=.DBF (database)
CALCULATE(HMTEMP.PERNUM (database)
HMTEMP.DURNUM (database)
HMTEMP.QTYNUM) (database)
BSTEP() (function in HMCOMP.SPR)
BOOT.DBF (database)
HMCOM.DBF (database)
RESETWT() (function in HMCOMP.SPR)
SELWT() (function in HMCOMP.SPR)
HMTA.DBF (database)
DELFILE() (function in HMCOMP.SPR)
DELFILE() ... (function in HMCOMP.SPR)
DP.PRG...
BFACT() (function in HMCOMP.SPR)
BOOT.DBF (database)
HMCOM.DBF (database)
=RESETWT(M.HMATID (database)
M.LCID (database)
M.WPID) (database)
RESETWT() ... (function in HMCOMP.SPR)
DELFILE() ... (function in HMCOMP.SPR)
BWSTEP() (function in HMCOMP.SPR)
BOOT.DBF (database)
HMCOM.DBF (database)
FOR.DBF (database)
I.DBF (database)
=.DBF (database)
1.DBF (database)
TO.DBF (database)
M.SAMPLE (database)
RESETWT() ... (function in HMCOMP.SPR)
DELFILE() ... (function in HMCOMP.SPR)
BPHASE() (function in HMCOMP.SPR)
BOOT.DBF (database)
HMCOM.DBF (database)
=RESETWT(M.HMATID (database)
M.LCID (database)
M.WPID) (database)
RESETWT() ... (function in HMCOMP.SPR)
DELFILE() ... (function in HMCOMP.SPR)
GETSTR() (function in HMCOMP.SPR)
DP.PRG...
DISPLAY.PRG
_QIN0M9E2J() (function in DISPLAY.PRG)
W_PRINT.SPR...
POPUPHIDE() ... (function in HMSC.PRG)
HMBROWSE (procedure in HMCOMP.SPR)
DELFILE() ... (function in HMCOMP.SPR)
CLOSEFILE (procedure in HMSC.PRG)
_QKF0VR0CN (procedure in HMENU.MPR)
BACKUP.PRG
ERRMSG.PRG...
_QKF0VR0FQ (procedure in HMENU.MPR)
ERRMSG.PRG...
HMAT.SPR
HMAT.DBF (database)


COMNAME() (function in HMAT.SPR)
HMCOM.DBF (database)
_QKF0VR394() (function in HMAT.SPR)
ERRMSG.PRG...
CHANGE... (procedure in HMAT.SPR)
_QKF0VR3EK() (function in HMAT.SPR)
CHANGE... (procedure in HMAT.SPR)
_QKF0VR3HM() (function in HMAT.SPR)
COMNAME() ... (function in HMAT.SPR)
CHANGE... (procedure in HMAT.SPR)
_QKF0VR3KQ() (function in HMAT.SPR)
_QKF0VR3ME() (function in HMAT.SPR)
_QKF0VR3ON() (function in HMAT.SPR)
CHANGE... (procedure in HMAT.SPR)
_QKF0VR3R9() (function in HMAT.SPR)
COMNAME() ... (function in HMAT.SPR)
_QKF0VR3XH() (function in HMAT.SPR)
ERRMSG.PRG...
CHANGE... (procedure in HMAT.SPR)
SAVECOM (procedure in HMAT.SPR)
HMCOM.DBF (database)
COMNAME() ... (function in HMAT.SPR)
HMLC.SPR
HMLC.DBF (database)
_QKF0VR6JX() (function in HMLC.SPR)
_QKF0VR6TE() (function in HMLC.SPR)
ERRMSG.PRG...
_QKF0VR6VW() (function in HMLC.SPR)
CHANGE... (procedure in HMAT.SPR)
HMWP.SPR
HMWP.DBF (database)
_QKF0VR92I() (function in HMWP.SPR)
_QKF0VR99I() (function in HMWP.SPR)
ERRMSG.PRG...
_QKF0VR9C5() (function in HMWP.SPR)
ERRMSG.PRG...
HMET.SPR
HMET.DBF (database)
_QKF0VRBIT() (function in HMET.SPR)
_QKF0VRBPA() (function in HMET.SPR)
CHANGE... (procedure in HMAT.SPR)
_QKF0VRBT3() (function in HMET.SPR)
ERRMSG.PRG...
HMCF.SPR
HMCF.DBF (database)
_QKF0VRDYQ() (function in HMCF.SPR)
_QKF0VRE5A() (function in HMCF.SPR)
ERRMSG.PRG...
_QKF0VRE7O() (function in HMCF.SPR)
ERRMSG.PRG...
HMCFE.SPR
HMCFE.DBF (database)
GET_HMCF (procedure in HMCFE.SPR)
HMC.DBF (database)
M.HMCF=X[1] (database)
IF.DBF (database)
NOT.DBF (database)
EMPTY(X[1]).DBF (database)
_QKF0VRGGZ() (function in HMCFE.SPR)


GET_HMCF... (procedure in HMCFE.SPR)
_QKF0VRGRP() (function in HMCFE.SPR)
_QKF0VRGTP() (function in HMCFE.SPR)
REL() (function in HMCFE.SPR)
HMCFEI.DBF (database)
HMCFE.DBF (database)
HMCF.DBF (database)
GET_HMCF... (procedure in HMCFE.SPR)
_QKF0VRGX4() (function in HMCFE.SPR)
_QKF0VRGYG() (function in HMCFE.SPR)
REL() ... (function in HMCFE.SPR)
GET_HMCF... (procedure in HMCFE.SPR)
_QKF0VRH1G() (function in HMCFE.SPR)
_QKF0VRH34() (function in HMCFE.SPR)
GET_CF() (function in HMCFE.SPR)
HMC.DBF (database)
CHOOSER.PRG...
ERRMSG.PRG...
GET_CFID (procedure in HMCFE.SPR)
HMC.DBF (database)
X=X[1].DBF (database)
IF.DBF (database)
NOT.DBF (database)
EMPTY(X[1]).DBF (database)
_QKF0VRH6D() (function in HMCFE.SPR)
GET_HMCF... (procedure in HMCFE.SPR)
HMCFEI.SPR
HMCFEI.DBF (database)
GET_HMCF... (procedure in HMCFE.SPR)
GET_HMCFE (procedure in HMCFEI.SPR)
HMCF.DBF (database)
IF.DBF (database)
!EMPTY(X[1]).DBF (database)
_QKF0VRK34() (function in HMCFEI.SPR)
BROWSEITEM (procedure in HMCFEI.SPR)
HMCFEI.DBF (database)
HMCFE.DBF (database)
HMCF.DBF (database)
GET_HMCF... (procedure in HMCFE.SPR)
GET_HMCFE... (procedure in HMCFEI.SPR)
_QKF0VRKGO() (function in HMCFEI.SPR)
_QKF0VRKIQ() (function in HMCFEI.SPR)
REL() ... (function in HMCFE.SPR)
GET_HMCF... (procedure in HMCFE.SPR)
GET_HMCFE... (procedure in HMCFEI.SPR)
_QKF0VRKMH() (function in HMCFEI.SPR)
_QKF0VRKNV() (function in HMCFEI.SPR)
REL() ... (function in HMCFE.SPR)
GET_HMCF... (procedure in HMCFE.SPR)
GET_HMCFE... (procedure in HMCFEI.SPR)
_QKF0VRKS5() (function in HMCFEI.SPR)
_QKF0VRKTM() (function in HMCFEI.SPR)
GET_CF() ... (function in HMCFE.SPR)
GET_CFID... (procedure in HMCFE.SPR)
_QKF0VRKWJ() (function in HMCFEI.SPR)
_QKF0VRKXZ() (function in HMCFEI.SPR)
GET_CFE (procedure in HMCFEI.SPR)
HMCF.DBF (database)
CHOOSER.PRG...

```
                    └──ERRMSG.PRG...
               └──GET_CFEID      (procedure in HMCFEI.SPR)
                         HMCF.DBF (database)
                         IF.DBF (database)
                         !EMPTY(X[1]).DBF (database)
          └──_QKF0VRL1J() (function  in HMCFEI.SPR)
               ├──CHANGE...      (procedure in HMAT.SPR)
               ├──GET_HMCF...      (procedure in HMCFE.SPR)
               └──GET_HMCFE...      (procedure in HMCFEI.SPR)
     ├──HMTAB.SPR
     │         HMUNIT.DBF (database)
     │         HMTAB.DBF (database)
     │    ├──HMLU.PRG...
     │    ├──DP.PRG...
     │    ├──ERRMSG.PRG...
     │    ├──GET_HMCOMID() (function  in HMTAB.SPR)
     │    │         HMA.DBF (database)
     │    │         IF.DBF (database)
     │    │         !=.DBF (database)
     │    │         HMCOM.DBF (database)
     │    ├──GET_HMLCN() ... (function  in HMTAB.SPR)
     │    ├──GET_HMWPN() ... (function  in HMTAB.SPR)
     │    ├──GET_HMETN() (function  in HMTAB.SPR)
     │    │         HME.DBF (database)
     │    ├──GET_CFAR() (function  in HMTAB.SPR)
     │    │         HMC.DBF (database)
     │    ├──GET_CFAR1() (function  in HMTAB.SPR)
     │    │         HMCF.DBF (database)
     │    ├──GET_EI() (function  in HMTAB.SPR)
     │    │         HMCFE.DBF (database)
     │    ├──_QKF0VRPCS() (function  in HMTAB.SPR)
     │    │    ├──GET_HMLCN() ... (function  in HMTAB.SPR)
     │    │    ├──GET_HMWPN() ... (function  in HMTAB.SPR)
     │    │    ├──GET_HMETN() ... (function  in HMTAB.SPR)
     │    │    ├──GET_CFAR() ... (function  in HMTAB.SPR)
     │    │    ├──GET_CFAR1() ... (function  in HMTAB.SPR)
     │    │    └──GET_EI() ... (function  in HMTAB.SPR)
     │    ├──_QKF0VRPYS() (function  in HMTAB.SPR)
     │    ├──_QKF0VRQ1C() (function  in HMTAB.SPR)
     │    │    │    HML.DBF (database)
     │    │    │    M.HMLCID=X[1] (database)
     │    │    └──GET_HMLC() ... (function  in HMTAB.SPR)
     │    ├──_QKF0VRQ4T() (function  in HMTAB.SPR)
     │    ├──_QKF0VRQ72() (function  in HMTAB.SPR)
     │    │    │    HMW.DBF (database)
     │    │    │    M.HMWPID=X[1] (database)
     │    │    └──GET_HMWP() ... (function  in HMTAB.SPR)
     │    ├──_QKF0VRQAL() (function  in HMTAB.SPR)
     │    ├──_QKF0VRQC2() (function  in HMTAB.SPR)
     │    │    │    HME.DBF (database)
     │    │    │    M.HMETID=X[1] (database)
     │    │    └──GET_HMET() (function  in HMTAB.SPR)
     │    │         │    HME.DBF (database)
     │    │         ├──CHOOSER.PRG...
     │    │         └──ERRMSG.PRG...
     │    ├──_QKF0VRQFR() (function  in HMTAB.SPR)
     │    ├──_QKF0VRQH7() (function  in HMTAB.SPR)
     │    │    │    HMC.DBF (database)
     │    │    └──GET_CF() ... (function  in HMCFE.SPR)
```


_QKF0VRQKZ() (function in HMTAB.SPR)
REL() ... (function in HMCFE.SPR)
GET_HMCFID() (function in HMTAB.SPR)
HMC.DBF (database)
M.HMCFID=X[1] (database)
GET_EID() (function in HMTAB.SPR)
HMCF.DBF (database)
GET_EIID() (function in HMTAB.SPR)
HMCFE.DBF (database)
_QKF0VRQRG() (function in HMTAB.SPR)
CHANGE... (procedure in HMAT.SPR)
GET_HMLCN() ... (function in HMTAB.SPR)
GET_HMWPN() ... (function in HMTAB.SPR)
GET_HMETN() ... (function in HMTAB.SPR)
GET_CFAR() ... (function in HMTAB.SPR)
GET_CFAR1() ... (function in HMTAB.SPR)
GET_EI() ... (function in HMTAB.SPR)
_QUIT() (function in HMINIT.PRG)
MYHANDLER() (function in HMINIT.PRG)

Section VI. Procedure and Function Summary. There are 26 files containing procedures in the HMLCCM system: HMINIT.PRG, HMAT.SPR, HMLC.SPR, HMWP.SPR, HMET.SPR, HMCF.SPR, HMCFE.SPR, HMCFEI.SPR, HMTAB.SPR, BACKUP.PRG, HMCOMP.SPR, CHOOSER.PRG, HMLU.PRG, W_PRINT.SPR, HMSC.PRG, HMSCEN.PRG, HMSTEP.PRG, MEMOEDIT.PRG, DISPLAY.PRG, BWFACT.PRG, BSELECT.PRG, HMREF.PRG, HMMSDS.PRG, MEMOWIN.PRG, HMENU.MPR, and &OLDPROC.

---

1. **HMINIT.PRG**

```
Contains: MYHANDLER()                    (Params: none)
   Called by: HMINIT.PRG
   Called by: HMINIT.PRG
Contains: _QUIT()                        (Params: none)
   Called by: HMENU.MPR
```

2. **HMAT.SPR**

```
Contains: CHANGE                         (Params: none)
   Called by: _QKF0VR394()               (function  in HMAT.SPR)
   Called by: _QKF0VR3EK()               (function  in HMAT.SPR)
   Called by: _QKF0VR3HM()               (function  in HMAT.SPR)
   Called by: _QKF0VR3ON()               (function  in HMAT.SPR)
   Called by: _QKF0VR3XH()               (function  in HMAT.SPR)
   Called by: _QKF0VR6VW()               (function  in HMLC.SPR)
   Called by: _QKF0VRBPA()               (function  in HMET.SPR)
   Called by: _QKF0VRL1J()               (function  in HMCFEI.SPR)
   Called by: _QKF0VRQRG()               (function  in HMTAB.SPR)
   Called by: _QIP0VATWZ()               (function  in HMSTEP.PRG)
Contains: NEWLINE()                      (Params: TEXT)
   Called by: HMVIEW()                   (function  in HMREF.PRG)
Contains: COMNAME()                      (Params: MKEY)
   Called by: HMAT.SPR
   Called by: _QKF0VR3HM()               (function  in HMAT.SPR)
   Called by: _QKF0VR3R9()               (function  in HMAT.SPR)
   Called by: _QKF0VR3XH()               (function  in HMAT.SPR)
Contains: SAVECOM                        (Params: MKEY, MNAME)
   Called by: _QKF0VR3XH()               (function  in HMAT.SPR)
Contains: _QKF0VR394()                   (Params: none)
   Called by: HMAT.SPR
       Calls: ERRMSG.PRG
       Calls: CHANGE                     (procedure in HMAT.SPR)
Contains: _QKF0VR3EK()                   (Params: none)
   Called by: HMAT.SPR
       Calls: CHANGE                     (procedure in HMAT.SPR)
Contains: _QKF0VR3HM()                   (Params: none)
   Called by: HMAT.SPR
       Calls: COMNAME()                  (function  in HMAT.SPR)
       Calls: CHANGE                     (procedure in HMAT.SPR)
Contains: _QKF0VR3KQ()                   (Params: none)
   Called by: HMAT.SPR
Contains: _QKF0VR3ME()                   (Params: none)
   Called by: HMAT.SPR
Contains: _QKF0VR3ON()                   (Params: none)
   Called by: HMAT.SPR
       Calls: CHANGE                     (procedure in HMAT.SPR)
Contains: _QKF0VR3R9()                   (Params: none)
   Called by: HMAT.SPR
       Calls: COMNAME()                  (function  in HMAT.SPR)
```

```
   Contains: _QKF0VR3XH()                    (Params: none)
      Called by: HMAT.SPR
          Calls: ERRMSG.PRG
          Calls: CHANGE                     (procedure in HMAT.SPR)
          Calls: SAVECOM                    (procedure in HMAT.SPR)
          Calls: COMNAME()                  (function  in HMAT.SPR)
```

3. **HMLC.SPR**

```
   Contains: CHANGE                         (Params: none)
      Called by: _QKF0VR394()               (function  in HMAT.SPR)
      Called by: _QKF0VR3EK()               (function  in HMAT.SPR)
      Called by: _QKF0VR3HM()               (function  in HMAT.SPR)
      Called by: _QKF0VR3ON()               (function  in HMAT.SPR)
      Called by: _QKF0VR3XH()               (function  in HMAT.SPR)
      Called by: _QKF0VR6VW()               (function  in HMLC.SPR)
      Called by: _QKF0VRBPA()               (function  in HMET.SPR)
      Called by: _QKF0VRL1J()               (function  in HMCFEI.SPR)
      Called by: _QKF0VRQRG()               (function  in HMTAB.SPR)
      Called by: _QIP0VATWZ()               (function  in HMSTEP.PRG)
   Contains: _QKF0VR6JX()                   (Params: none)
      Called by: HMLC.SPR
   Contains: _QKF0VR6TE()                   (Params: none)
      Called by: HMLC.SPR
          Calls: ERRMSG.PRG
   Contains: _QKF0VR6VW()                   (Params: none)
      Called by: HMLC.SPR
          Calls: CHANGE                     (procedure in HMAT.SPR)
```

4. **HMWP.SPR**

```
   Contains: CHANGE                         (Params: none)
      Called by: _QKF0VR394()               (function  in HMAT.SPR)
      Called by: _QKF0VR3EK()               (function  in HMAT.SPR)
      Called by: _QKF0VR3HM()               (function  in HMAT.SPR)
      Called by: _QKF0VR3ON()               (function  in HMAT.SPR)
      Called by: _QKF0VR3XH()               (function  in HMAT.SPR)
      Called by: _QKF0VR6VW()               (function  in HMLC.SPR)
      Called by: _QKF0VRBPA()               (function  in HMET.SPR)
      Called by: _QKF0VRL1J()               (function  in HMCFEI.SPR)
      Called by: _QKF0VRQRG()               (function  in HMTAB.SPR)
      Called by: _QIP0VATWZ()               (function  in HMSTEP.PRG)
   Contains: _QKF0VR92I()                   (Params: none)
      Called by: HMWP.SPR
   Contains: _QKF0VR99I()                   (Params: none)
      Called by: HMWP.SPR
          Calls: ERRMSG.PRG
   Contains: _QKF0VR9C5()                   (Params: none)
      Called by: HMWP.SPR
          Calls: ERRMSG.PRG
```

5. **HMET.SPR**

```
   Contains: CHANGE                         (Params: none)
      Called by: _QKF0VR394()               (function  in HMAT.SPR)
      Called by: _QKF0VR3EK()               (function  in HMAT.SPR)
      Called by: _QKF0VR3HM()               (function  in HMAT.SPR)
      Called by: _QKF0VR3ON()               (function  in HMAT.SPR)
      Called by: _QKF0VR3XH()               (function  in HMAT.SPR)
```

```
      Called by: _QKF0VR6VW()              (function  in HMLC.SPR)
      Called by: _QKF0VRBPA()              (function  in HMET.SPR)
      Called by: _QKF0VRL1J()              (function  in HMCFEI.SPR)
      Called by: _QKF0VRQRG()              (function  in HMTAB.SPR)
      Called by: _QIP0VATWZ()              (function  in HMSTEP.PRG)
   Contains: _QKF0VRBIT()                  (Params: none)
      Called by: HMET.SPR
   Contains: _QKF0VRBPA()                  (Params: none)
      Called by: HMET.SPR
          Calls: CHANGE                    (procedure in HMAT.SPR)
   Contains: _QKF0VRBT3()                  (Params: none)
      Called by: HMET.SPR
          Calls: ERRMSG.PRG
```

---

6. **HMCF.SPR**

```
   Contains: CHANGE                        (Params: none)
      Called by: _QKF0VR394()              (function  in HMAT.SPR)
      Called by: _QKF0VR3EK()              (function  in HMAT.SPR)
      Called by: _QKF0VR3HM()              (function  in HMAT.SPR)
      Called by: _QKF0VR3ON()              (function  in HMAT.SPR)
      Called by: _QKF0VR3XH()              (function  in HMAT.SPR)
      Called by: _QKF0VR6VW()              (function  in HMLC.SPR)
      Called by: _QKF0VRBPA()              (function  in HMET.SPR)
      Called by: _QKF0VRL1J()              (function  in HMCFEI.SPR)
      Called by: _QKF0VRQRG()              (function  in HMTAB.SPR)
      Called by: _QIP0VATWZ()              (function  in HMSTEP.PRG)
   Contains: _QKF0VRDYQ()                  (Params: none)
      Called by: HMCF.SPR
   Contains: _QKF0VRE5A()                  (Params: none)
      Called by: HMCF.SPR
          Calls: ERRMSG.PRG
   Contains: _QKF0VRE7O()                  (Params: none)
      Called by: HMCF.SPR
          Calls: ERRMSG.PRG
```

7. **HMCFE.SPR**

```
   Contains: ESCPRESSED                    (Params: none)
   Contains: CHANGE                        (Params: none)
      Called by: _QKF0VR394()              (function  in HMAT.SPR)
      Called by: _QKF0VR3EK()              (function  in HMAT.SPR)
      Called by: _QKF0VR3HM()              (function  in HMAT.SPR)
      Called by: _QKF0VR3ON()              (function  in HMAT.SPR)
      Called by: _QKF0VR3XH()              (function  in HMAT.SPR)
      Called by: _QKF0VR6VW()              (function  in HMLC.SPR)
      Called by: _QKF0VRBPA()              (function  in HMET.SPR)
      Called by: _QKF0VRL1J()              (function  in HMCFEI.SPR)
      Called by: _QKF0VRQRG()              (function  in HMTAB.SPR)
      Called by: _QIP0VATWZ()              (function  in HMSTEP.PRG)
   Contains: GET_HMCF                      (Params: M.HMCFID)
      Called by: HMCFE.SPR
      Called by: HMCFEI.SPR
      Called by: _QKF0VRGGZ()              (function  in HMCFE.SPR)
      Called by: _QKF0VRGTP()              (function  in HMCFE.SPR)
      Called by: _QKF0VRGYG()              (function  in HMCFE.SPR)
      Called by: _QKF0VRH6D()              (function  in HMCFE.SPR)
```

```
      Called by: _QKF0VRK34()          (function  in HMCFEI.SPR)
      Called by: _QKF0VRKIQ()          (function  in HMCFEI.SPR)
      Called by: _QKF0VRKNV()          (function  in HMCFEI.SPR)
      Called by: _QKF0VRL1J()          (function  in HMCFEI.SPR)
   Contains: GET_CF()                  (Params: M.HMCF)
      Called by: _QKF0VRH34()          (function  in HMCFE.SPR)
      Called by: _QKF0VRKTM()          (function  in HMCFEI.SPR)
      Called by: _QKF0VRQH7()          (function  in HMTAB.SPR)
          Calls: CHOOSER.PRG
          Calls: ERRMSG.PRG
   Contains: GET_CFID                  (Params: M.ANSWR)
      Called by: _QKF0VRH34()          (function  in HMCFE.SPR)
      Called by: _QKF0VRKTM()          (function  in HMCFEI.SPR)
   Contains: REL()                     (Params: M.ID)
      Called by: _QKF0VRGTP()          (function  in HMCFE.SPR)
      Called by: _QKF0VRGYG()          (function  in HMCFE.SPR)
      Called by: _QKF0VRKIQ()          (function  in HMCFEI.SPR)
      Called by: _QKF0VRKNV()          (function  in HMCFEI.SPR)
      Called by: _QKF0VRQKZ()          (function  in HMTAB.SPR)
   Contains: _QKF0VRGGZ()              (Params: none)
      Called by: HMCFE.SPR
          Calls: GET_HMCF              (procedure in HMCFE.SPR)
   Contains: _QKF0VRGRP()              (Params: none)
      Called by: HMCFE.SPR
   Contains: _QKF0VRGTP()              (Params: none)
      Called by: HMCFE.SPR
          Calls: REL()                 (function  in HMCFE.SPR)
          Calls: GET_HMCF              (procedure in HMCFE.SPR)
   Contains: _QKF0VRGX4()              (Params: none)
      Called by: HMCFE.SPR
   Contains: _QKF0VRGYG()              (Params: none)
      Called by: HMCFE.SPR
          Calls: REL()                 (function  in HMCFE.SPR)
          Calls: GET_HMCF              (procedure in HMCFE.SPR)
   Contains: _QKF0VRH1G()              (Params: none)
      Called by: HMCFE.SPR
   Contains: _QKF0VRH34()              (Params: none)
      Called by: HMCFE.SPR
          Calls: GET_CF()              (function  in HMCFE.SPR)
          Calls: GET_CFID              (procedure in HMCFE.SPR)
   Contains: _QKF0VRH6D()              (Params: none)
      Called by: HMCFE.SPR
          Calls: GET_HMCF              (procedure in HMCFE.SPR)
```

8. **HMCFEI.SPR**

```
   Contains: ESCPRESSED                (Params: none)
   Contains: CHANGE                    (Params: none)
      Called by: _QKF0VR394()          (function  in HMAT.SPR)
      Called by: _QKF0VR3EK()          (function  in HMAT.SPR)
      Called by: _QKF0VR3HM()          (function  in HMAT.SPR)
      Called by: _QKF0VR3ON()          (function  in HMAT.SPR)
      Called by: _QKF0VR3XH()          (function  in HMAT.SPR)
      Called by: _QKF0VR6VW()          (function  in HMLC.SPR)
      Called by: _QKF0VRBPA()          (function  in HMET.SPR)
      Called by: _QKF0VRL1J()          (function  in HMCFEI.SPR)
      Called by: _QKF0VRQRG()          (function  in HMTAB.SPR)
      Called by: _QIP0VATWZ()          (function  in HMSTEP.PRG)
   Contains: GET_HMCF                  (Params: M.HMCFID)
```

```
   Called by: HMCFE.SPR
   Called by: HMCFEI.SPR
   Called by: _QKF0VRGGZ()              (function  in HMCFE.SPR)
   Called by: _QKF0VRGTP()              (function  in HMCFE.SPR)
   Called by: _QKF0VRGYG()              (function  in HMCFE.SPR)
   Called by: _QKF0VRH6D()              (function  in HMCFE.SPR)
   Called by: _QKF0VRK34()              (function  in HMCFEI.SPR)
   Called by: _QKF0VRKIQ()              (function  in HMCFEI.SPR)
   Called by: _QKF0VRKNV()              (function  in HMCFEI.SPR)
   Called by: _QKF0VRL1J()              (function  in HMCFEI.SPR)
Contains: GET_CF()                      (Params: M.HMCF)
   Called by: _QKF0VRH34()              (function  in HMCFE.SPR)
   Called by: _QKF0VRKTM()              (function  in HMCFEI.SPR)
   Called by: _QKF0VRQH7()              (function  in HMTAB.SPR)
       Calls: CHOOSER.PRG
       Calls: ERRMSG.PRG
Contains: GET_CFID                      (Params: M.ANSWR)
   Called by: _QKF0VRH34()              (function  in HMCFE.SPR)
   Called by: _QKF0VRKTM()              (function  in HMCFEI.SPR)
Contains: GET_HMCFE                     (Params: M.HMCFEID)
   Called by: HMCFEI.SPR
   Called by: _QKF0VRK34()              (function  in HMCFEI.SPR)
   Called by: _QKF0VRKIQ()              (function  in HMCFEI.SPR)
   Called by: _QKF0VRKNV()              (function  in HMCFEI.SPR)
   Called by: _QKF0VRL1J()              (function  in HMCFEI.SPR)
Contains: GET_CFE                       (Params: M.HMCFE)
   Called by: _QKF0VRKXZ()              (function  in HMCFEI.SPR)
       Calls: CHOOSER.PRG
       Calls: ERRMSG.PRG
Contains: GET_CFEID                     (Params: M.HMCFE)
   Called by: _QKF0VRKXZ()              (function  in HMCFEI.SPR)
Contains: REL()                         (Params: M.ID)
   Called by: _QKF0VRGTP()              (function  in HMCFE.SPR)
   Called by: _QKF0VRGYG()              (function  in HMCFE.SPR)
   Called by: _QKF0VRKIQ()              (function  in HMCFEI.SPR)
   Called by: _QKF0VRKNV()              (function  in HMCFEI.SPR)
   Called by: _QKF0VRQKZ()              (function  in HMTAB.SPR)
Contains: BROWSEITEM                    (Params: M.ID)
   Called by: _QKF0VRK34()              (function  in HMCFEI.SPR)
Contains: _QKF0VRK34()                  (Params: none)
   Called by: HMCFEI.SPR
       Calls: BROWSEITEM                (procedure in HMCFEI.SPR)
       Calls: GET_HMCF                  (procedure in HMCFE.SPR)
       Calls: GET_HMCFE                 (procedure in HMCFEI.SPR)
Contains: _QKF0VRKGO()                  (Params: none)
   Called by: HMCFEI.SPR
Contains: _QKF0VRKIQ()                  (Params: none)
   Called by: HMCFEI.SPR
       Calls: REL()                     (function  in HMCFE.SPR)
       Calls: GET_HMCF                  (procedure in HMCFE.SPR)
       Calls: GET_HMCFE                 (procedure in HMCFEI.SPR)
Contains: _QKF0VRKMH()                  (Params: none)
   Called by: HMCFEI.SPR
Contains: _QKF0VRKNV()                  (Params: none)
   Called by: HMCFEI.SPR
       Calls: REL()                     (function  in HMCFE.SPR)
       Calls: GET_HMCF                  (procedure in HMCFE.SPR)
       Calls: GET_HMCFE                 (procedure in HMCFEI.SPR)
Contains: _QKF0VRKS5()                  (Params: none)
```

```
      Called by: HMCFEI.SPR
   Contains: _QKF0VRKTM()                  (Params: none)
      Called by: HMCFEI.SPR
          Calls: GET_CF()                  (function  in HMCFE.SPR)
          Calls: GET_CFID                  (procedure in HMCFE.SPR)
   Contains: _QKF0VRKWJ()                  (Params: none)
      Called by: HMCFEI.SPR
   Contains: _QKF0VRKXZ()                  (Params: none)
      Called by: HMCFEI.SPR
          Calls: GET_CFE                   (procedure in HMCFEI.SPR)
          Calls: GET_CFEID                 (procedure in HMCFEI.SPR)
   Contains: _QKF0VRL1J()                  (Params: none)
      Called by: HMCFEI.SPR
          Calls: CHANGE                    (procedure in HMAT.SPR)
          Calls: GET_HMCF                  (procedure in HMCFE.SPR)
          Calls: GET_HMCFE                 (procedure in HMCFEI.SPR)
```

9. **HMTAB.SPR**

```
   Contains: ESCPRESSED                    (Params: none)
   Contains: CHANGE                        (Params: none)
      Called by: _QKF0VR394()              (function  in HMAT.SPR)
      Called by: _QKF0VR3EK()              (function  in HMAT.SPR)
      Called by: _QKF0VR3HM()              (function  in HMAT.SPR)
      Called by: _QKF0VR3ON()              (function  in HMAT.SPR)
      Called by: _QKF0VR3XH()              (function  in HMAT.SPR)
      Called by: _QKF0VR6VW()              (function  in HMLC.SPR)
      Called by: _QKF0VRBPA()              (function  in HMET.SPR)
      Called by: _QKF0VRL1J()              (function  in HMCFEI.SPR)
      Called by: _QKF0VRQRG()              (function  in HMTAB.SPR)
      Called by: _QIP0VATWZ()              (function  in HMSTEP.PRG)
   Contains: GET_HMCOMID()                 (Params: M.HMATID)
      Called by: HMTAB.SPR
   Contains: GET_HMLC()                    (Params: M.HMLC)
      Called by: _QKF0VRQ1C()              (function  in HMTAB.SPR)
      Called by: _QIP0VAT9X()              (function  in HMSTEP.PRG)
          Calls: CHOOSER.PRG
          Calls: ERRMSG.PRG
   Contains: GET_HMLCN()                   (Params: M.ID)
      Called by: HMTAB.SPR
      Called by: _QKF0VRPCS()              (function  in HMTAB.SPR)
      Called by: _QKF0VRQRG()              (function  in HMTAB.SPR)
      Called by: INITVAR                   (procedure in HMSTEP.PRG)
   Contains: GET_HMWP()                    (Params: M.MATCH)
      Called by: _QKF0VRQ72()              (function  in HMTAB.SPR)
      Called by: _QIP0VATLC()              (function  in HMSTEP.PRG)
          Calls: CHOOSER.PRG
          Calls: ERRMSG.PRG
   Contains: GET_HMWPN()                   (Params: M.ID)
      Called by: HMTAB.SPR
      Called by: _QKF0VRPCS()              (function  in HMTAB.SPR)
      Called by: _QKF0VRQRG()              (function  in HMTAB.SPR)
      Called by: INITVAR                   (procedure in HMSTEP.PRG)
   Contains: GET_HMET()                    (Params: M.HMET)
      Called by: _QKF0VRQC2()              (function  in HMTAB.SPR)
          Calls: CHOOSER.PRG
          Calls: ERRMSG.PRG
   Contains: GET_HMETID()                  (Params: M.HMET)
   Contains: GET_HMETN()                   (Params: M.HMETID)
```

```
   Called by: HMTAB.SPR                    (function  in HMTAB.SPR)
   Called by: _QKF0VRPCS()                 (function  in HMTAB.SPR)
   Called by: _QKF0VRQRG()                 (Params: M.HMCFID)
Contains: GET_CFAR()
   Called by: HMTAB.SPR                    (function  in HMTAB.SPR)
   Called by: _QKF0VRPCS()                 (function  in HMTAB.SPR)
   Called by: _QKF0VRQRG()                 (Params: M.HMCFEID)
Contains: GET_CFAR1()
   Called by: HMTAB.SPR                    (function  in HMTAB.SPR)
   Called by: _QKF0VRPCS()                 (function  in HMTAB.SPR)
   Called by: _QKF0VRQRG()                 (Params: M.HMCF)
Contains: GET_CF()                         (function  in HMCFE.SPR)
   Called by: _QKF0VRH34()                 (function  in HMCFEI.SPR)
   Called by: _QKF0VRKTM()                 (function  in HMTAB.SPR)
   Called by: _QKF0VRQH7()
       Calls: CHOOSER.PRG
       Calls: ERRMSG.PRG
Contains: GET_HMCFID()                     (Params: M.HMCF)
   Called by: _QKF0VRQKZ()                 (function  in HMTAB.SPR)
Contains: GET_CF1()                        (Params: M.HMCFE)
       Calls: CHOOSER.PRG
       Calls: ERRMSG.PRG
Contains: GET_EID()                        (Params: M.HMCFE)
   Called by: _QKF0VRQKZ()                 (function  in HMTAB.SPR)
Contains: GET_EI()                         (Params: M.HMCFEIID)
   Called by: HMTAB.SPR                    (function  in HMTAB.SPR)
   Called by: _QKF0VRPCS()                 (function  in HMTAB.SPR)
   Called by: _QKF0VRQRG()                 (Params: M.HMCFEI)
Contains: GET_EIID()                       (function  in HMTAB.SPR)
   Called by: _QKF0VRQKZ()                 (Params: M.ID)
Contains: REL()                            (function  in HMCFE.SPR)
   Called by: _QKF0VRGTP()                 (function  in HMCFE.SPR)
   Called by: _QKF0VRGYG()                 (function  in HMCFEI.SPR)
   Called by: _QKF0VRKIQ()                 (function  in HMCFEI.SPR)
   Called by: _QKF0VRKNV()                 (function  in HMTAB.SPR)
   Called by: _QKF0VRQKZ()                 (Params: none)
Contains: _QKF0VRPCS()
   Called by: HMTAB.SPR                    (function  in HMTAB.SPR)
       Calls: GET_HMLCN()                  (function  in HMTAB.SPR)
       Calls: GET_HMWPN()                  (function  in HMTAB.SPR)
       Calls: GET_HMETN()                  (function  in HMTAB.SPR)
       Calls: GET_CFAR()                   (function  in HMTAB.SPR)
       Calls: GET_CFAR1()                  (function  in HMTAB.SPR)
       Calls: GET_EI()                     (Params: none)
Contains: _QKF0VRPYS()
   Called by: HMTAB.SPR                    (Params: none)
Contains: _QKF0VRQ1C()
   Called by: HMTAB.SPR                    (function  in HMTAB.SPR)
       Calls: GET_HMLC()                   (Params: none)
Contains: _QKF0VRQ4T()
   Called by: HMTAB.SPR                    (Params: none)
Contains: _QKF0VRQ72()
   Called by: HMTAB.SPR                    (function  in HMTAB.SPR)
       Calls: GET_HMWP()                   (Params: none)
Contains: _QKF0VRQAL()
   Called by: HMTAB.SPR                    (Params: none)
Contains: _QKF0VRQC2()
   Called by: HMTAB.SPR                    (function  in HMTAB.SPR)
       Calls: GET_HMET()
```

```
   Contains: _QKF0VRQFR()                    (Params: none)
      Called by: HMTAB.SPR
   Contains: _QKF0VRQH7()                    (Params: none)
      Called by: HMTAB.SPR
          Calls: GET_CF()                    (function  in HMCFE.SPR)
   Contains: _QKF0VRQKZ()                    (Params: none)
      Called by: HMTAB.SPR
          Calls: REL()                       (function  in HMCFE.SPR)
          Calls: GET_HMCFID()                (function  in HMTAB.SPR)
          Calls: GET_EID()                   (function  in HMTAB.SPR)
          Calls: GET_EIID()                  (function  in HMTAB.SPR)
   Contains: _QKF0VRQRG()                    (Params: none)
      Called by: HMTAB.SPR
          Calls: CHANGE                      (procedure in HMAT.SPR)
          Calls: GET_HMLCN()                 (function  in HMTAB.SPR)
          Calls: GET_HMWPN()                 (function  in HMTAB.SPR)
          Calls: GET_HMETN()                 (function  in HMTAB.SPR)
          Calls: GET_CFAR()                  (function  in HMTAB.SPR)
          Calls: GET_CFAR1()                 (function  in HMTAB.SPR)
          Calls: GET_EI()                    (function  in HMTAB.SPR)
```

10 **BACKUP.PRG**

```
   Contains: ERRHAND                         (Params: none)
          Calls: ERRMSG.PRG
   Contains: STOP                            (Params: none)
          Calls: ERRMSG.PRG
```

11. **HMCOMP.SPR**

```
   Contains: OPEN                            (Params: none)
      Called by: HMCOMP.SPR
          Calls: OPENFILE()                  (function  in HMSC.PRG)
          Calls: CANCEL                      (procedure in HMCOMP.SPR)
   Contains: DELFILE()                       (Params: FILE)
      Called by: HMCOMP.SPR
      Called by: CALCULATE()                 (function  in HMCOMP.SPR)
      Called by: SELWT()                     (function  in HMCOMP.SPR)
      Called by: BSTEP()                     (function  in HMCOMP.SPR)
      Called by: BFACT()                     (function  in HMCOMP.SPR)
      Called by: BWSTEP()                    (function  in HMCOMP.SPR)
      Called by: BPHASE()                    (function  in HMCOMP.SPR)
   Contains: CANCEL                          (Params: SUCCESS)
      Called by: HMSTEP.PRG
      Called by: OPEN                        (procedure in HMCOMP.SPR)
          Calls: ERRMSG.PRG
   Contains: GET_HMARRAY                     (Params: none)
      Called by: HMCOMP.SPR
          Calls: GET_TABLE                   (procedure in HMCOMP.SPR)
   Contains: GET_TABLE                       (Params: COMID, LCID, WPID)
      Called by: GET_HMARRAY                 (procedure in HMCOMP.SPR)
   Contains: COMPUTE                         (Params: none)
      Called by: _QLE0L6CB0()                (function  in HMCOMP.SPR)
          Calls: SUBCOMPUT                   (procedure in HMCOMP.SPR)
          Calls: DISPLAY.PRG
   Contains: SUBCOMPUT                       (Params: M.IHMATNAME, M.IHMATID)
      Called by: COMPUTE                     (procedure in HMCOMP.SPR)
          Calls: COMPUTSTEP()                (function  in HMCOMP.SPR)
```

```
          Calls: COMPUTFACT()              (function  in HMCOMP.SPR)
          Calls: COMPUTWFACT()             (function  in HMCOMP.SPR)
          Calls: COMPUTPHASE()             (function  in HMCOMP.SPR)
          Calls: BSTEP()                   (function  in HMCOMP.SPR)
          Calls: DP.PRG
          Calls: BFACT()                   (function  in HMCOMP.SPR)
          Calls: BWSTEP()                  (function  in HMCOMP.SPR)
          Calls: BPHASE()                  (function  in HMCOMP.SPR)
          Calls: GETSTR()                  (function  in HMCOMP.SPR)
   Contains: HMBROWSE                      (Params: none)
      Called by: _QLE0L6CB0()              (function  in HMCOMP.SPR)
   Contains: COMPUTSTEP()                  (Params: COM, HMAT, LCPHASE, WP, SN,
SD, SQ)
      Called by: SUBCOMPUT                 (procedure in HMCOMP.SPR)
   Contains: FACTNAME()                    (Params: MFACTID)
   Contains: COMPUTFACT()                  (Params: M.FACTID, M.HMATID)
      Called by: SUBCOMPUT                 (procedure in HMCOMP.SPR)
   Contains: COMPUTWFACT()                 (Params: M.FACTID, M.HMATID)
      Called by: SUBCOMPUT                 (procedure in HMCOMP.SPR)
   Contains: COMPUTPHASE()                 (Params: M.PHASEID, M.HMATID)
      Called by: SUBCOMPUT                 (procedure in HMCOMP.SPR)
   Contains: CALCULATE()                   (Params: SN, SD, SQ)
          Calls: DELFILE()                 (function  in HMCOMP.SPR)
   Contains: RESETWT()                     (Params: MATID, LCPHASE, WP)
      Called by: BSTEP()                   (function  in HMCOMP.SPR)
      Called by: BFACT()                   (function  in HMCOMP.SPR)
      Called by: BWSTEP()                  (function  in HMCOMP.SPR)
      Called by: BPHASE()                  (function  in HMCOMP.SPR)
          Calls: SELWT()                   (function  in HMCOMP.SPR)
   Contains: SELWT()                        (Params: MATID, LCPHASE, WP, METID,
CFID, CFEID)
      Called by: RESETWT()                 (function  in HMCOMP.SPR)
          Calls: DELFILE()                 (function  in HMCOMP.SPR)
   Contains: SETUPBOOT()                   (Params: none)
      Called by: _QLE0L6CB0()              (function  in HMCOMP.SPR)
          Calls: BSELECT.PRG
          Calls: ERRMSG.PRG
          Calls: BWFACT.PRG
          Calls: DP.PRG
   Contains: BSTEP()                     (Params: MAT, LCPHASE, WP, SN, SD, SQ)
      Called by: SUBCOMPUT                 (procedure in HMCOMP.SPR)
          Calls: RESETWT()                 (function  in HMCOMP.SPR)
          Calls: DELFILE()                 (function  in HMCOMP.SPR)
   Contains: BFACT()                       (Params: M.FACTID, M.HMATID)
      Called by: SUBCOMPUT                 (procedure in HMCOMP.SPR)
          Calls: RESETWT()                 (function  in HMCOMP.SPR)
          Calls: DELFILE()                 (function  in HMCOMP.SPR)
   Contains: BWSTEP()                    (Params: M.FACTID, M.STEPID, M.HMATID)
      Called by: SUBCOMPUT                 (procedure in HMCOMP.SPR)
          Calls: RESETWT()                 (function  in HMCOMP.SPR)
          Calls: DELFILE()                 (function  in HMCOMP.SPR)
   Contains: BPHASE()                      (Params: M.PHASEID, M.HMATID)
      Called by: SUBCOMPUT                 (procedure in HMCOMP.SPR)
          Calls: RESETWT()                 (function  in HMCOMP.SPR)
          Calls: DELFILE()                 (function  in HMCOMP.SPR)
   Contains: GETSTR()                            (Params: HMATNAME, SCENTOTAL,
STEPTOTAL, FACTOTAL, WFACTOTAL, PHASETOTAL, BOOTSTRAP)
      Called by: SUBCOMPUT                 (procedure in HMCOMP.SPR)
          Calls: DP.PRG
```

```
   Contains: _QLE0L6B8T()                (Params: none)
      Called by: HMCOMP.SPR
   Contains: _QLE0L6BIV()                (Params: none)
      Called by: HMCOMP.SPR
   Contains: _QLE0L6BR7()                (Params: none)
      Called by: HMCOMP.SPR
   Contains: _QLE0L6BYN()                (Params: none)
      Called by: HMCOMP.SPR
   Contains: _QLE0L6C5N()                (Params: none)
      Called by: HMCOMP.SPR
   Contains: _QLE0L6CB0()                (Params: none)
      Called by: HMCOMP.SPR
          Calls: SETUPBOOT()             (function  in HMCOMP.SPR)
          Calls: POPUPSHOW()             (function  in HMSC.PRG)
          Calls: COMPUTE                 (procedure in HMCOMP.SPR)
          Calls: POPUPHIDE()             (function  in HMSC.PRG)
          Calls: HMBROWSE                (procedure in HMCOMP.SPR)
```

12. **CHOOSER.PRG**

```
   Contains: _Q120IDFTF()                (Params: none)
      Called by: CHOOSER.PRG
   Contains: _Q120IDG17()                (Params: none)
      Called by: CHOOSER.PRG
   Contains: _Q120IDG86()                (Params: none)
      Called by: CHOOSER.PRG
```

13. **HMLU.PRG**

```
   Contains: GET_HMNAME                  (Params: M.NAME)
      Called by: _Q7D0O1EI8()            (function  in HMLU.PRG)
          Calls: CHOOSER.PRG
          Calls: ERRMSG.PRG
   Contains: _Q7D0O1EI8()                (Params: none)
      Called by: HMLU.PRG
          Calls: GET_HMNAME              (procedure in HMLU.PRG)
          Calls: DP.PRG
   Contains: _Q7D0O1EUB()                (Params: none)
      Called by: HMLU.PRG
```

14. **W_PRINT.SPR**

```
   Contains: PRINTFILE()                 (Params: none)
      Called by: _QKF0VS4PH()            (function  in W_PRINT.SPR)
   Contains: _QKF0VS4NC()                (Params: none)
      Called by: W_PRINT.SPR
   Contains: _QKF0VS4PH()                (Params: none)
      Called by: W_PRINT.SPR
          Calls: PRINTFILE()             (function  in W_PRINT.SPR)
   Contains: _QKF0VS4RM()                (Params: none)
      Called by: W_PRINT.SPR
```

15. **HMSC.PRG**

```
   Contains: CLOSEFILE                   (Params: none)
      Called by: HMSC.PRG
   Contains: DEL_HMSC()                  (Params: ID)
      Called by: HMSC.PRG
          Calls: POPUPSHOW()             (function  in HMSC.PRG)
```

```
        Calls: OPENFILE()                (function  in HMSC.PRG)
        Calls: POPUPHIDE()               (function  in HMSC.PRG)
  Contains: FILEFIND()                   (Params: MFILENAME)
     Called by: OPENFILE()               (function  in HMSC.PRG)
         Calls: DP.PRG
  Contains: OPENFILE()                   (Params: FILE)
     Called by: HMSTEP.PRG
     Called by: HMREF.PRG
     Called by: HMMSDS.PRG
     Called by: OPEN                     (procedure in HMCOMP.SPR)
     Called by: DEL_HMSC()               (function  in HMSC.PRG)
         Calls: FILEFIND()               (function  in HMSC.PRG)
  Contains: POPUPSHOW()                  (Params: ERRSTR)
     Called by: _QLE0L6CB0()             (function  in HMCOMP.SPR)
     Called by: DEL_HMSC()               (function  in HMSC.PRG)
  Contains: POPUPHIDE()                  (Params: W)
     Called by: _QLE0L6CB0()             (function  in HMCOMP.SPR)
     Called by: DEL_HMSC()               (function  in HMSC.PRG)
```

16. **HMSCEN.PRG**

```
  Contains: GET_HMSC()                   (Params: NAME)
     Called by: _Q8K0INGJH()             (function  in HMSCEN.PRG)
     Called by: _Q8K0INGTV()             (function  in HMSCEN.PRG)
         Calls: CHOOSER.PRG
  Contains: MEVENT                       (Params: MNEW)
     Called by: _Q8K0INGJH()             (function  in HMSCEN.PRG)
     Called by: _Q8K0INGTV()             (function  in HMSCEN.PRG)
     Called by: _Q8K0INHBV()             (function  in HMSCEN.PRG)
  Contains: _Q8K0INGJH()                 (Params: none)
     Called by: HMSCEN.PRG
         Calls: GET_HMSC()               (function  in HMSCEN.PRG)
         Calls: MEVENT                   (procedure in HMSCEN.PRG)
         Calls: DP.PRG
  Contains: _Q8K0INGTV()                 (Params: none)
     Called by: HMSCEN.PRG
         Calls: ERRMSG.PRG
         Calls: GET_HMSC()               (function  in HMSCEN.PRG)
         Calls: MEVENT                   (procedure in HMSCEN.PRG)
         Calls: DP.PRG
  Contains: _Q8K0INHBV()                 (Params: none)
     Called by: HMSCEN.PRG
         Calls: MEVENT                   (procedure in HMSCEN.PRG)
```

17. **HMSTEP.PRG**

```
  Contains: ESCPRESSED                   (Params: none)
  Contains: INITVAR                      (Params: none)
     Called by: HMSTEP.PRG
     Called by: _QIP0VASCN()             (function  in HMSTEP.PRG)
     Called by: _QIP0VATWZ()             (function  in HMSTEP.PRG)
         Calls: GET_HMATN()              (function  in HMSTEP.PRG)
         Calls: GET_HMLCN()              (function  in HMTAB.SPR)
         Calls: GET_HMWPN()              (function  in HMTAB.SPR)
  Contains: ADDOPTION                    (Params: none)
     Called by: _QIP0VASCN()             (function  in HMSTEP.PRG)
     Called by: _QIP0VATWZ()             (function  in HMSTEP.PRG)
  Contains: CHANGE                       (Params: none)
     Called by: _QKF0VR394()             (function  in HMAT.SPR)
```

```
    Called by: _QKF0VR3EK()          (function  in HMAT.SPR)
    Called by: _QKF0VR3HM()          (function  in HMAT.SPR)
    Called by: _QKF0VR3ON()          (function  in HMAT.SPR)
    Called by: _QKF0VR3XH()          (function  in HMAT.SPR)
    Called by: _QKF0VR6VW()          (function  in HMLC.SPR)
    Called by: _QKF0VRBPA()          (function  in HMET.SPR)
    Called by: _QKF0VRL1J()          (function  in HMCFEI.SPR)
    Called by: _QKF0VRQRG()          (function  in HMTAB.SPR)
    Called by: _QIP0VATWZ()          (function  in HMSTEP.PRG)
Contains: GET_HMAT()                 (Params: NAME)
    Called by: _QIP0VASYA()          (function  in HMSTEP.PRG)
        Calls: CHOOSER.PRG
        Calls: ERRMSG.PRG
Contains: GET_HMATN()                (Params: M.ID)
    Called by: INITVAR               (procedure in HMSTEP.PRG)
Contains: GET_HMLC()                 (Params: M.HMLC)
    Called by: _QKF0VRQ1C()          (function  in HMTAB.SPR)
    Called by: _QIP0VAT9X()          (function  in HMSTEP.PRG)
        Calls: CHOOSER.PRG
        Calls: ERRMSG.PRG
Contains: GET_HMLCN()                (Params: M.ID)
    Called by: HMTAB.SPR
    Called by: _QKF0VRPCS()          (function  in HMTAB.SPR)
    Called by: _QKF0VRQRG()          (function  in HMTAB.SPR)
    Called by: INITVAR               (procedure in HMSTEP.PRG)
Contains: GET_HMWP()                 (Params: M.MATCH)
    Called by: _QKF0VRQ72()          (function  in HMTAB.SPR)
    Called by: _QIP0VATLC()          (function  in HMSTEP.PRG)
        Calls: CHOOSER.PRG
        Calls: ERRMSG.PRG
Contains: GET_HMWPN()                (Params: M.ID)
    Called by: HMTAB.SPR
    Called by: _QKF0VRPCS()          (function  in HMTAB.SPR)
    Called by: _QKF0VRQRG()          (function  in HMTAB.SPR)
    Called by: INITVAR               (procedure in HMSTEP.PRG)
Contains: GET_HMUNIT                 (Params: none)
    Called by: _QIP0VASYA()          (function  in HMSTEP.PRG)
    Called by: _QIP0VAT9X()          (function  in HMSTEP.PRG)
    Called by: _QIP0VATLC()          (function  in HMSTEP.PRG)
    Called by: DATACHECK()           (function  in HMSTEP.PRG)
        Calls: ERRMSG.PRG
Contains: SCSAVE                     (Params: none)
    Called by: _QIP0VASCN()          (function  in HMSTEP.PRG)
Contains: DATACHECK()                (Params: none)
    Called by: _QIP0VATWZ()          (function  in HMSTEP.PRG)
        Calls: ERRMSG.PRG
        Calls: GET_HMUNIT            (procedure in HMSTEP.PRG)
Contains: _QIP0VASCN()               (Params: none)
    Called by: HMSTEP.PRG
        Calls: YESNO.PRG
        Calls: INITVAR               (procedure in HMSTEP.PRG)
        Calls: ADDOPTION             (procedure in HMSTEP.PRG)
        Calls: SCSAVE                (procedure in HMSTEP.PRG)
Contains: _QIP0VASVS()               (Params: none)
    Called by: HMSTEP.PRG
Contains: _QIP0VASYA()               (Params: none)
    Called by: HMSTEP.PRG
        Calls: GET_HMAT()            (function  in HMSTEP.PRG)
        Calls: DP.PRG
```

```
        Calls: GET_HMUNIT                (procedure in HMSTEP.PRG)
  Contains: _QIP0VAT7G()                 (Params: none)
     Called by: HMSTEP.PRG
  Contains: _QIP0VAT9X()                 (Params: none)
     Called by: HMSTEP.PRG
          Calls: GET_HMLC()              (function  in HMTAB.SPR)
          Calls: DP.PRG
          Calls: GET_HMUNIT              (procedure in HMSTEP.PRG)
  Contains: _QIP0VATIU()                 (Params: none)
     Called by: HMSTEP.PRG
  Contains: _QIP0VATLC()                 (Params: none)
     Called by: HMSTEP.PRG
          Calls: GET_HMWP()              (function  in HMTAB.SPR)
          Calls: DP.PRG
          Calls: GET_HMUNIT              (procedure in HMSTEP.PRG)
  Contains: _QIP0VATWZ()                 (Params: none)
     Called by: HMSTEP.PRG
          Calls: DATACHECK()             (function  in HMSTEP.PRG)
          Calls: CHANGE                  (procedure in HMAT.SPR)
          Calls: YESNO.PRG
          Calls: INITVAR                 (procedure in HMSTEP.PRG)
          Calls: ADDOPTION               (procedure in HMSTEP.PRG)
  Contains: _QIP0VAUD2()                 (Params: none)
     Called by: HMSTEP.PRG
```

18. **MEMOEDIT.PRG**

```
  Contains: _Q8L0LJ6F0()                 (Params: none)
     Called by: MEMOEDIT.PRG
```

19. **DISPLAY.PRG**

```
  Contains: _QIN0M9E2J()                 (Params: none)
     Called by: DISPLAY.PRG
          Calls: W_PRINT.SPR
```

20. **BWFACT.PRG**

```
  Contains: _Q8Q0NQKA6()                 (Params: none)
     Called by: BWFACT.PRG
```

21. **BSELECT.PRG**

```
  Contains: _Q8Q0N6SBM()                 (Params: none)
     Called by: BSELECT.PRG
```

22. **HMREF.PRG**

```
  Contains: HMVIEW()                     (Params: NAME)
     Called by: HMREF.PRG
     Called by: HMMSDS.PRG
          Calls: NEWLINE()               (function  in HMAT.SPR)
          Calls: MEMOWIN.PRG
  Contains: NEWLINE()                    (Params: TEXT)
     Called by: HMVIEW()                 (function  in HMREF.PRG)
```

23. **HMMSDS.PRG**

```
  Contains: HMVIEW()                     (Params: NAME)
```

```
      Called by: HMREF.PRG
      Called by: HMMSDS.PRG
          Calls: NEWLINE()                    (function  in HMAT.SPR)
          Calls: MEMOWIN.PRG
```

24. **MEMOWIN.PRG**

```
   Contains: _QIG0IXOXD()                     (Params: none)
      Called by: MEMOWIN.PRG
          Calls: W_PRINT.SPR
   Contains: _QIG0IXPJO()                     (Params: none)
      Called by: MEMOWIN.PRG
          Calls: W_PRINT.SPR
```

25. **HMENU.MPR**

```
   Contains: _QKF0VR0CN                       (Params: none)
      Called by: HMENU.MPR
          Calls: BACKUP.PRG
   Contains: _QKF0VR0FQ                       (Params: none)
      Called by: HMENU.MPR
          Calls: ERRMSG.PRG
```

26. **&OLDPROC**
   Temporary Variable

Section VII. Program Source Code

```
  1  *:*********************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HMLCCM\WORK\BACKUP.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 12/01/93     14:51
  9  *:
 10  *:  Procs & Fncts: ERRHAND
 11  *:               : STOP
 12  *:
 13  *:         Set by: _QLEOVCQDC         (procedure in HMENU.MPR)
 14  *:
 15  *:          Calls: ERRMSG.PRG
 16  *:
 17  *:      Documented 12/02/93 at 10:50               FoxDoc  version 2.10f
 18  *:*********************************************************************
=> ******
 19      set talk off
 20      =errmsg("Insert diskette in drive A and hi: RETURN")
 21      =errmsg("Please wait.",1)
 22
 23      ! pkzip a:\hmbackup.zip *.dbf
 24      ! pkzip -u a:\hmbackup.zip *.cdx
 25      ! pkzip -u a:\hmbackup.zip *.fpt
 26
 27      =errmsg("DONE.",1)
 28
 29      clear
 30      return
 31
 32      ****************************************************
 40  *!*********************************************************************
=> ******
 41  *!
 42  *!      Procedure: ERRHAND
 43  *!
 44  *!          Calls: ERRMSG.PRG
 45  *!
 46  *!*********************************************************************
=> ******
 40      procedure errhand
 41      =errmsg("BACKUP CRASHED CHECK PARAMETERS AND TRY AGAIN.")
 42      return
 43      quit
 44
 54  *!*********************************************************************
=> ******
 55  *!
 56  *!      Procedure: STOP
 57  *!
 58  *!          Calls: EMPTY()            (function  in ?)
 59  *!               : ERROR()            (function  in ?)
 60  *!               : ERRMSG.PRG
 61  *!
 62  *!*********************************************************************
=> ******
 54      procedure stop
 55      if !empty(error())
 56         =errmsg("Please check drive A")
 57         retry
 58      endif
 59      return
 60      *: EOF: BACKUP.ACT
```

```
  1  *:*************************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\BIGCHARS.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 09/13/93      8:43
  9  *:
 10  *:         Set by: HMINIT.PRG
 11  *:              : HMINIT.PRG
 12  *:
 13  *:      Documented 12/01/93 at 11:31               FoxDoc  version 2.10f
 14  *:*************************************************************************
=> ******
 15      ************************************************************************
 16      *
 17      *    FUNCTION: BIGCHARS
 18      *    PURPOSE: PRINT HEADER
 19      *    RETURN: None.
 20      *    PARAMETERS:
 21      *             startx          x-coordinate to start   (sugg: 6)
 22      *             starty          y-coordianate to start  (sugg: 17)
 23      *             instring        title to be blown up
 24      *             o_t                        waiting time  (sugg: 6)
 25      *    REFERENCE:      C:\FOXPRO2\GOODIES\DEMO\PRGS\BIGCHARS.PRG
 26      *    SIDE EFFECTS: Not known.
 27      *    CREATED BY:  Anh Le             18 JUN 92
 28      *    MODIFIED:       AL              23 JUN 92
 29      *
 30      ************************************************************************
 31      PARAMETER startx,starty,instring,o_t
 32
 33      IF PARAMETERS() = 3
 34         o_t = 6
 35      ENDIF
 36
 37      x_coor = startx
 38      y_coor = starty
 39      instring = UPPER(instring)
 40      len_str = LEN(instring)
 41
 42      DO CASE
 43      CASE (len_str = 5)
 44          starty = starty
 45      CASE (len_str = 3)
 46          starty = starty + 9
 47      CASE (len_str < 5)
 48          starty = starty + 3
 49      CASE (len_str = 6)
 50          starty = starty - 3
 51      CASE (len_str = 7)
 52          starty = starty - 6
 53      OTHERWISE
 54          starty = starty -starty+4
 55      ENDCASE
 56
 57      SET BLINK OFF
 58      CLEAR
 59
 60      IF startx > 20
 61  <-------RETURN
 62      ENDIF
 63
 64      curenty = starty
 65      curlet = 1
 66
 67      IF "MONO" $ UPPER(SYS(2006))
 68         colorvar = "n/n"
 69      ELSE
 70         colorvar = "n/b"
 71      ENDIF
 72      CLEAR
 73
 74      DO WHILE curlet <= LEN(instring) AND curenty < 89
 75         DO CASE
 76         CASE SUBSTR(instring,curlet,1) = " "
 77            DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
 78                FROM startx,curenty TO startx + 3,curenty + 5 ;
 79                NONE NOSHADOW
 80            ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
 81            curenty = curenty + 6
 82         CASE SUBSTR(instring,curlet,1) = "t"
 83            DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
 84                FROM startx,curenty TO startx + 5,curenty + 5 ;
 85                NONE NOSHADOW
 86            ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
 87            curenty = curenty + 6
 88            @ 0,0 SAY " ▄▀"
 89            @ 0,3 SAY "▄" COLOR (colorvar)
 90
 91            @ 1,0 SAY "▀▀▀"
 92            @ 1,3 SAY "▀" COLOR W+/N
 93            @ 1,4 SAY "▄" COLOR (colorvar)
 94
 95            @ 2,1 SAY "▀▄ " COLOR W+/N
 96
 97            @ 3,0 SAY " ▀▄"
 98            @ 3,3 SAY "▄" COLOR W+/N
 99
100            @ 4,0 SAY "   ▀▀" COLOR (colorvar)
101         CASE SUBSTR(instring,curlet,1) = "s"
102            DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
103                FROM startx,curenty TO startx + 5,curenty + 5 ;
104                NONE NOSHADOW
105            ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
106            curenty = curenty + 7
107            @ 1,0 SAY "▄▀"
108            @ 1,2 SAY "▀▀" COLOR W+/N
109            @ 1,3 SAY "▀▄"
110
111            @ 2,0 SAY " ▀▄"
112            @ 2,2 SAY "▄▀" COLOR W+/N
113            @ 2,5 SAY "▀▀" COLOR (colorvar)
114
115            @ 3,0 SAY "▀▄▄▀"
116            @ 3,4 SAY "▀" COLOR W+/N
117
118            @ 4,0 SAY "  ▀▀▀" COLOR (colorvar)
119         CASE SUBSTR(instring,curlet,1) = "e"
120            DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
121                FROM startx,curenty TO startx + 5,curenty + 5 ;
122                NONE NOSHADOW
123            ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
124            curenty = curenty + 7
125            @ 1,0 SAY "▄▀"
126            @ 1,2 SAY "▀▄" COLOR W+/N
127            @ 1,4 SAY "▄"
128
129            @ 2,0 SAY "▀▀▀▀ " COLOR W+/N
130
```

```
131     @ 3,0 SAY "▄▄"
132     @ 3,2 SAY "▄▀▀" COLOR W+/N
133     @ 3,5 SAY "▄" COLOR (colorvar)
134
135     @ 4,0 SAY " ▀▀▀" COLOR (colorvar)
136   CASE SUBSTR(instring,curlet,1) = "r"
137     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
138         FROM startx,curenty TO startx + 5,curenty + 5 ;
139         NONE NOSHADOW
140     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
141     curenty = curenty + 6
142     @ 1,0 SAY "▄▄"
143     @ 1,3 SAY "▀▀" COLOR W+/N
144     @ 1,4 SAY "▄" COLOR (colorvar)
145
146     @ 2,0 SAY "▀▀ " COLOR W+/N
147     @ 2,3 SAY "▀▀" COLOR (colorvar)
148
149     @ 3,0 SAY "▀▀ " COLOR W+/N
150
151     @ 4,0 SAY " ▀▀" COLOR (colorvar)
152   CASE SUBSTR(instring,curlet,1) = "o"
153     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
154         FROM startx,curenty TO startx + 5,curenty + 5 ;
155         NONE NOSHADOW
156     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
157     curenty = curenty + 7
158     @ 1,0 SAY "▄▄"
159     @ 1,2 SAY "▀▀" COLOR W+/N
160     @ 1,4 SAY "▄"
161
162     @ 2,0 SAY "▀▀ ▀▀ " COLOR W+/N
163
164     @ 3,0 SAY "▄▄"
165     @ 3,2 SAY "▀▀ " COLOR W+/N
166
167     @ 4,0 SAY " ▀▀▀" COLOR (colorvar)
168   CASE SUBSTR(instring,curlet,1) = "a"
169     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
170         FROM startx,curenty TO startx + 5,curenty + 6 ;
171         NONE NOSHADOW
172     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
173     curenty = curenty + 8
174     @ 1,0 SAY "▄▄"
175     @ 1,2 SAY "▀▀" COLOR W+/N
176     @ 1,4 SAY "▄"
177
178     @ 2,0 SAY "▄▄"
179     @ 2,2 SAY "▀▀▀ " COLOR W+/N
180
181     @ 3,0 SAY "▀▀"
182     @ 3,2 SAY "▄▄▄" COLOR W+/N
183
184     @ 4,0 SAY " ▀▀ ▀▀" COLOR (colorvar)
185   CASE SUBSTR(instring,curlet,1) = "i"
186     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
187         FROM startx,curenty TO startx + 5,curenty + 3 ;
188         NONE NOSHADOW
189     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
190     curenty = curenty + 4
191     @ 0,0 SAY "▀▀"
192     @ 0,1 SAY "▀▀" COLOR W+/N
193     @ 0,2 SAY "▄" COLOR (colorvar)
194
195     @ 1,0 SAY "▀▀"
196     @ 1,2 SAY "▄" COLOR (colorvar)
197
198     @ 2,0 SAY "▀▀ " COLOR W+/N
199
200     @ 3,0 SAY "▀▀ " COLOR W+/N
201
202     @ 4,0 SAY " ▀▀" COLOR (colorvar)
203   CASE SUBSTR(instring,curlet,1) = "n"
204     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
205         FROM startx,curenty TO startx + 5,curenty + 5 ;
206         NONE NOSHADOW
207     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
208     curenty = curenty + 7
209     @ 1,0 SAY "▄▄"
210     @ 1,2 SAY "▀▀" COLOR W+/N
211     @ 1,4 SAY "▄"
212
213     @ 2,0 SAY "▀▀ ▀▀ " COLOR W+/N
214
215     @ 3,0 SAY "▀▀ ▀▀ " COLOR W+/N
216
217     @ 4,0 SAY " ▀▀ ▀▀" COLOR (colorvar)
218   CASE SUBSTR(instring,curlet,1) = "m"
219     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
220         FROM startx,curenty TO startx + 5,curenty + 8 ;
221         NONE NOSHADOW
222     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
223     curenty = curenty + 10
224     @ 1,0 SAY "▄▄"
225     @ 1,2 SAY "▀▀" COLOR W+/N
226     @ 1,4 SAY "▄"
227     @ 1,5 SAY "▀▀" COLOR W+/N
228     @ 1,7 SAY "▄"
229
230     @ 2,0 SAY "▀▀ ▀▀ ▀▀ " COLOR W+/N
231
232     @ 3,0 SAY "▀▀ ▀▀ ▀▀ " COLOR W+/N
233
234     @ 4,0 SAY " ▀▀ ▀▀ ▀▀" COLOR (colorvar)
235   CASE SUBSTR(instring,curlet,1) = "f"
236     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
237         FROM startx,curenty TO startx + 5,curenty + 6 ;
238         NONE NOSHADOW
239     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
240     curenty = curenty + 7
241     @ 0,0 SAY " ▄▀▀"
242     @ 0,3 SAY "▀▀▀" COLOR W+/N
243     @ 0,5 SAY "▄" COLOR (colorvar)
244
245     @ 1,0 SAY "▀▀▀"
246     @ 1,3 SAY "▀▀" COLOR W+/N
247     @ 1,4 SAY "▄" COLOR (colorvar)
248
249     @ 2,1 SAY "▀▀ " COLOR W+/N
250
251     @ 3,1 SAY "▀▀ " COLOR W+/N
252
253     @ 4,0 SAY " ▀▀" COLOR (colorvar)
254   CASE SUBSTR(instring,curlet,1) = "l"
255     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
256         FROM startx,curenty TO startx + 5,curenty + 3 ;
257         NONE NOSHADOW
258     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
259     curenty = curenty + 4
260     @ 0,0 SAY "▀▀"
261     @ 0,2 SAY "▄" COLOR (colorvar)
262
```

```
263         @ 1,0 SAY "▀▀ " COLOR W+/N
264
265         @ 2,0 SAY "▀▀ " COLOR W+/N
266
267         @ 3,0 SAY "▀▀ " COLOR W+/N
268
269         @ 4,0 SAY "  ▀▀" COLOR (colorvar)
270      CASE SUBSTR(instring,curlet,1) = "k"
271         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
272             FROM startx,curenty TO startx + 5,curenty + 7 ;
273             NONE NOSHADOW
274         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
275         curenty = curenty + 8
276         @ 0,0 SAY "▀▀"
277         @ 0,2 SAY "▄" COLOR (colorvar)
278
279         @ 1,0 SAY "▀▀▄▀" COLOR W+/N
280         @ 1,5 SAY "▄" COLOR (colorvar)
281
282         @ 2,0 SAY "▀▀▄▄" COLOR W+/N
283         @ 2,4 SAY "▀▀" COLOR (colorvar)
284
285         @ 3,0 SAY "▀▄ " COLOR W+/N
286         @ 3,3 SAY "▀▄"
287
288         @ 4,0 SAY "  ▀▀  ▀▀" COLOR (colorvar)
289      CASE SUBSTR(instring,curlet,1) = "u"
290         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
291             FROM startx,curenty TO startx + 5,curenty + 6 ;
292             NONE NOSHADOW
293         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
294         curenty = curenty + 7
295         @ 1,0 SAY "▀▀"
296         @ 1,2 SAY "▄" COLOR (colorvar)
297         @ 1,3 SAY "▀▀"
298         @ 1,5 SAY "▄" COLOR (colorvar)
299
300         @ 2,0 SAY "▀▀ ▀▀ " COLOR W+/N
301
302         @ 3,0 SAY "▀▀"
303         @ 3,2 SAY "▄" COLOR W+/N
304         @ 3,3 SAY "▀▀"
305         @ 3,4 SAY "▀ " COLOR W+/N
306
307         @ 4,0 SAY "  ▀▀ ▀▀" COLOR (colorvar)
308      CASE SUBSTR(instring,curlet,1) = "b"
309         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
310             FROM startx,curenty TO startx + 5,curenty + 6 ;
311             NONE NOSHADOW
312         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
313         curenty = curenty + 7
314         @ 0,0 SAY "▀▀"
315         @ 0,2 SAY "▄" COLOR (colorvar)
316
317         @ 1,0 SAY "▀▀▄▀" COLOR W+/N
318         @ 1,4 SAY "▄"
319
320         @ 2,0 SAY "▀▀ " COLOR W+/N
321         @ 2,3 SAY "▀▀"
322         @ 2,5 SAY "▄" COLOR (colorvar)
323
324         @ 3,0 SAY "▀▀▄▄ " COLOR W+/N
325
326         @ 4,0 SAY "  ▀▀▀▀" COLOR (colorvar)
327      CASE SUBSTR(instring,curlet,1) = "c"
328         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
```



```
329             FROM startx,curenty TO startx + 5,curenty + 5 ;
330             NONE NOSHADOW
331         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
332         curenty = curenty + 6
333         @ 1,0 SAY "▄▀"
334         @ 1,2 SAY "▀▄" COLOR W+/N
335         @ 1,4 SAY "▄" COLOR (colorvar)
336
337         @ 2,0 SAY "▀▄ " COLOR W+/N
338         @ 2,3 SAY "▀▄" COLOR (colorvar)
339
340         @ 3,0 SAY "▀▄"
341         @ 3,2 SAY "▄▀" COLOR W+/N
342         @ 3,4 SAY "▄" COLOR (colorvar)
343
344         @ 4,0 SAY "  ▀▀▀" COLOR (colorvar)
345      CASE SUBSTR(instring,curlet,1) = "d"
346         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
347             FROM startx,curenty TO startx + 5,curenty + 5 ;
348             NONE NOSHADOW
349         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
350         curenty = curenty + 7
351         @ 0,0 SAY "   ▀▀"
352         @ 0,5 SAY "▄" COLOR (colorvar)
353
354         @ 1,0 SAY "▄▀"
355         @ 1,2 SAY "▀▄ " COLOR W+/N
356
357         @ 2,0 SAY "▀▄ " COLOR W+/N
358         @ 2,2 SAY "▀▀" COLOR (colorvar)
359         @ 2,3 SAY "▀▀ " COLOR W+/N
360
361         @ 3,0 SAY "▀▄▀▀"
362         @ 3,5 SAY "▄" COLOR (colorvar)
363
364         @ 4,0 SAY "  ▀▀▀▀" COLOR (colorvar)
365      CASE SUBSTR(instring,curlet,1) = "q"
366         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
367             FROM startx,curenty TO startx + 5,curenty + 5 ;
368             NONE NOSHADOW
369         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
370         curenty = curenty + 7
371         @ 1,0 SAY "▄▀"
372         @ 1,2 SAY "▀▀" COLOR W+/N
373         @ 1,3 SAY "▄▀"
374         @ 1,5 SAY "▄" COLOR (colorvar)
375
376         @ 2,0 SAY "▀▄ " COLOR W+/N
377         @ 2,2 SAY "▀▀" COLOR (colorvar)
378         @ 2,3 SAY "▀▀ " COLOR W+/N
379
380         @ 3,0 SAY "▀▄▀▀"
381         @ 3,5 SAY "▀" COLOR (colorvar)
382
383         @ 4,0 SAY "  ▀▀" COLOR (colorvar)
384         @ 4,3 SAY "▀▀ " COLOR W+/N
385
386         @ 5,0 SAY "    ▀▀" COLOR (colorvar)
387
388      CASE SUBSTR(instring,curlet,1) = "g"
389         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
390             FROM startx,curenty TO startx + 5,curenty + 6 ;
391             NONE NOSHADOW
392         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
393         curenty = curenty + 7
394         @ 1,0 SAY "▄▀"
```

```
395         @ 1,2 SAY "▀▀" COLOR W+/N
396         @ 1,3 SAY "▄▀"
397         @ 1,5 SAY "▄" COLOR (colorvar)
398
399         @ 2,0 SAY "▀ " COLOR W+/N
400         @ 2,2 SAY "▀▀" COLOR (colorvar)
401         @ 2,3 SAY "▀▀ " COLOR W+/N
402
403         @ 3,0 SAY "▀▄▄▀"
404         @ 3,5 SAY "▀" COLOR (colorvar)
405
406         @ 4,0 SAY " ▄"
407         @ 4,2 SAY "▄" COLOR W+/N
408         @ 4,3 SAY "▀▀"
409         @ 4,4 SAY "▀▀ " COLOR W+/N
410
411         @ 5,0 SAY "  ▀▀▀" COLOR (colorvar)
412      CASE SUBSTR(instring,curlet,1) = "h"
413         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
414             FROM startx,curenty TO startx + 5,curenty + 5 ;
415             NONE NOSHADOW
416         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
417         curenty = curenty + 7
418         @ 0,0 SAY "▄▀" COLOR W+/N
419         @ 0,2 SAY "▄" COLOR (colorvar)
420
421         @ 1,0 SAY "▀▄▄▀" COLOR W+/N
422         @ 1,4 SAY "▄"
423
424         @ 2,0 SAY "▀▄ ▄▀ " COLOR W+/N
425
426         @ 3,0 SAY "▀▄ ▄▀ " COLOR W+/N
427
428         @ 4,0 SAY " ▀▀ ▀▀" COLOR (colorvar)
429      CASE SUBSTR(instring,curlet,1) = "j"
430         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
431             FROM startx,curenty TO startx + 5,curenty + 3 ;
432             NONE NOSHADOW
433         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
434         curenty = curenty + 5
435         @ 0,0 SAY " ▀▀"
436         @ 0,2 SAY "▀▀" COLOR W+/N
437         @ 0,3 SAY "▄" COLOR (colorvar)
438
439         @ 1,0 SAY " ▀▀"
440         @ 1,3 SAY "▄" COLOR (colorvar)
441
442         @ 2,1 SAY "▀▀ " COLOR W+/N
443
444         @ 3,1 SAY "▀▀ " COLOR W+/N
445
446         @ 4,0 SAY "▀▀"
447         @ 4,1 SAY "▀▀ " COLOR W+/N
448         @ 4,3 SAY "▀▀" COLOR (colorvar)
449      CASE SUBSTR(instring,curlet,1) = "p"
450         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
451             FROM startx,curenty TO startx + 5,curenty + 6 ;
452             NONE NOSHADOW
453         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
454         curenty = curenty + 7
455         @ 1,0 SAY "▄▄▄▄"
456
457         @ 2,0 SAY "▀▄ " COLOR W+/N
458         @ 2,3 SAY "▀▀"
459         @ 2,5 SAY "▀" COLOR (colorvar)
460
461         @ 3,0 SAY "▀▄▄▀ " COLOR W+/N
462
463         @ 4,0 SAY "▀▄ " COLOR W+/N
464         @ 4,3 SAY "▀▀" COLOR (colorvar)
465
466         @ 5,0 SAY " ▀▀" COLOR (colorvar)
467      CASE SUBSTR(instring,curlet,1) = "w"
468         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
469             FROM startx,curenty TO startx + 5,curenty + 8 ;
470             NONE NOSHADOW
471         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
472         curenty = curenty + 9
473         @ 1,0 SAY "▀▀"
474         @ 1,2 SAY "▄" COLOR (colorvar)
475         @ 1,3 SAY "▀▀"
476         @ 1,4 SAY "▄" COLOR (colorvar)
477         @ 1,5 SAY "▀▀"
478         @ 1,7 SAY "▄" COLOR (colorvar)
479
480         @ 2,0 SAY "▀▀"
481         @ 2,2 SAY "▀▄▄▀ " COLOR W+/N
482
483         @ 3,1 SAY "▀▄ ▄▀ " COLOR W+/N
484
485         @ 4,0 SAY " ▀▀ ▀▀" COLOR (colorvar)
486      CASE SUBSTR(instring,curlet,1) = "x"
487         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
488             FROM startx,curenty TO startx + 5,curenty + 7 ;
489             NONE NOSHADOW
490         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
491         curenty = curenty + 8
492         @ 1,0 SAY "▀▄▄▀"
493         @ 1,5 SAY "▀▀" COLOR W+/N
494         @ 1,6 SAY "▄" COLOR (colorvar)
495
496         @ 2,0 SAY " ▄▀▀"
497         @ 2,4 SAY "▄" COLOR W+/N
498         @ 2,5 SAY "▀▀" COLOR (colorvar)
499
500         @ 3,0 SAY "▀▀ " COLOR W+/N
501         @ 3,3 SAY "▀▀" COLOR (colorvar)
502         @ 3,4 SAY "▀▀" COLOR W+/N
503         @ 3,6 SAY "▄" COLOR (colorvar)
504
505         @ 4,0 SAY " ▀▀  ▀▀" COLOR (colorvar)
506      CASE SUBSTR(instring,curlet,1) = "y"
507         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
508             FROM startx,curenty TO startx + 5,curenty + 6 ;
509             NONE NOSHADOW
510         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
511         curenty = curenty + 7
512         @ 1,0 SAY "▀▀"
513         @ 1,2 SAY "▄" COLOR (colorvar)
514         @ 1,3 SAY "▀▀"
515         @ 1,5 SAY "▄" COLOR (colorvar)
516
517         @ 2,0 SAY "▀▄ ▄▀ " COLOR W+/N
518
519         @ 3,0 SAY " ▀▀"
520         @ 3,4 SAY "▀▀ " COLOR W+/N
521
522         @ 4,0 SAY " ▄▀"
523         @ 4,3 SAY "▀▀" COLOR W+/N
524
525         @ 5,0 SAY "  ▀▀" COLOR (colorvar)
526      CASE SUBSTR(instring,curlet,1) = "v"
```

```
527         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
528             FROM startx,curenty TO startx + 5,curenty + 6 ;
529             NONE NOSHADOW
530         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
531         curenty = curenty + 7
532         @ 1,0 SAY "[redacted]"
533         @ 1,2 SAY "[redacted]" COLOR (colorvar)
534         @ 1,3 SAY "[redacted]"
535         @ 1,5 SAY "[redacted]" COLOR (colorvar)
536
537         @ 2,0 SAY "[redacted]"
538         @ 2,1 SAY "[redacted]" COLOR W+/N
539
540         @ 3,0 SAY "[redacted]"
541         @ 3,3 SAY "[redacted]" COLOR W+/N
542
543         @ 4,0 SAY "[redacted]" COLOR (colorvar)
544     CASE SUBSTR(instring,curlet,1) = "/"
545         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
546             FROM startx,curenty TO startx + 5,curenty + 4 ;
547             NONE NOSHADOW
548         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
549         curenty = curenty + 6
550         @ 0,0 SAY "[redacted]"
551         @ 0,4 SAY "[redacted]" COLOR (colorvar)
552
553         @ 1,0 SAY "[redacted]"
554         @ 1,3 SAY "[redacted]" COLOR (colorvar)
555
556         @ 2,0 SAY "[redacted]"
557         @ 2,2 SAY "[redacted]" COLOR (colorvar)
558
559         @ 3,0 SAY "[redacted]"
560         @ 3,1 SAY "[redacted]" COLOR (colorvar)
561
562         @ 4,0 SAY "[redacted]" COLOR (colorvar)
563     CASE SUBSTR(instring,curlet,1) = "z"
564         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
565             FROM startx,curenty TO startx + 5,curenty + 6 ;
566             NONE NOSHADOW
567         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
568         curenty = curenty + 7
569         @ 1,0 SAY "[redacted]"
570         @ 1,1 SAY "[redacted]" COLOR W+/N
571         @ 1,5 SAY "[redacted]" COLOR (colorvar)
572
573         @ 2,0 SAY "[redacted]"
574         @ 2,3 SAY "[redacted]" COLOR W+/N
575         @ 2,5 SAY "[redacted]" COLOR (colorvar)
576
577         @ 3,0 SAY "[redacted]" COLOR W+/N
578         @ 3,4 SAY "[redacted]"
579
580         @ 4,0 SAY "[redacted]" COLOR (colorvar)
581     CASE SUBSTR(instring,curlet,1) = "|"
582         startx = startx + 6
583         curenty = starty
584     CASE SUBSTR(instring,curlet,1) = "A"
585         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
586             FROM startx,curenty TO startx + 5,curenty + 7 ;
587             NONE NOSHADOW
588         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
589         curenty = curenty + 9
590         @ 0,2 SAY "[redacted]" COLOR W+/N
591         @ 0,5 SAY "[redacted]" COLOR (colorvar)
592
```



```
593         @ 1,1 SAY "[redacted]" COLOR W+/N
594         @ 1,6 SAY "[redacted]" COLOR (colorvar)
595
596         @ 2,0 SAY "[redacted]"
597         @ 2,1 SAY "[redacted]" COLOR W+/N
598
599         @ 3,0 SAY "[redacted]" COLOR W+/N
600         @ 3,3 SAY "[redacted]" COLOR (colorvar)
601         @ 3,5 SAY "[redacted]" COLOR W+/N
602
603         @ 4,0 SAY "[redacted]" COLOR (colorvar)
604     CASE SUBSTR(instring,curlet,1) = "B"
605         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
606             FROM startx,curenty TO startx + 5,curenty + 6 ;
607             NONE NOSHADOW
608         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)),)
609         curenty = curenty + 8
610         @ 0,0 SAY "[redacted]" COLOR W+/N
611         @ 0,4 SAY "[redacted]"
612
613         @ 1,0 SAY "[redacted]" COLOR W+/N
614         @ 1,3 SAY "[redacted]"
615         @ 1,5 SAY "[redacted]" COLOR W+/N
616
617         @ 2,0 SAY "[redacted]" COLOR W+/N
618         @ 2,3 SAY "[redacted]" COLOR (colorvar)
619         @ 2,4 SAY "[redacted]" COLOR W+/N
620         @ 2,6 SAY "[redacted]" COLOR (colorvar)
621
622         @ 3,0 SAY "[redacted]" COLOR W+/N
623         @ 3,3 SAY "[redacted]"
624         @ 3,5 SAY "[redacted]" COLOR W+/N
625
626         @ 4,0 SAY "[redacted]" COLOR (colorvar)
627     CASE SUBSTR(instring,curlet,1) = "C"
628         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
629             FROM startx,curenty TO startx + 5,curenty + 8 ;
630             NONE NOSHADOW
631         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
632         curenty = curenty + 9
633         @ 0,0 SAY "[redacted]"
634         @ 0,3 SAY "[redacted]" COLOR W+/N
635         @ 0,6 SAY "[redacted]"
636
637         @ 1,0 SAY "[redacted]" COLOR W+/N
638         @ 1,3 SAY "[redacted]" COLOR (colorvar)
639         @ 1,4 SAY "[redacted]"
640         @ 1,7 SAY "[redacted]" COLOR (colorvar)
641
642         @ 2,0 SAY "[redacted]" COLOR W+/N
643         @ 2,3 SAY "[redacted]"
644
645         @ 3,0 SAY "[redacted]"
646         @ 3,6 SAY "[redacted]" COLOR W+/N
647
648         @ 4,0 SAY "[redacted]" COLOR (colorvar)
649     CASE SUBSTR(instring,curlet,1) = "D"
650         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
651             FROM startx,curenty TO startx + 5,curenty + 6 ;
652             NONE NOSHADOW
653         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
654         curenty = curenty + 8
655         @ 0,0 SAY "[redacted]" COLOR W+/N
656         @ 0,4 SAY "[redacted]"
657
658         @ 1,0 SAY "[redacted]" COLOR W+/N
```

```
659     @ 1,3 SAY " "
660     @ 1,6 SAY "" COLOR (colorvar)
661
662     @ 2,0 SAY "" COLOR W+/N
663     @ 2,3 SAY ""
664     @ 2,6 SAY "" COLOR (colorvar)
665
666     @ 3,0 SAY "" COLOR W+/N
667     @ 3,6 SAY "" COLOR (colorvar)
668
669     @ 4,0 SAY " " COLOR (colorvar)
670  CASE SUBSTR(instring,curlet,1) = "E"
671     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
672         FROM startx,curenty TO startx + 5,curenty + 6 ;
673         NONE NOSHADOW
674     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
675     curenty = curenty + 7
676     @ 0,0 SAY "" COLOR W+/N
677     @ 0,5 SAY "" COLOR (colorvar)
678
679     @ 1,0 SAY "" COLOR W+/N
680     @ 1,3 SAY ""
681
682     @ 2,0 SAY "" COLOR W+/N
683     @ 2,3 SAY "" COLOR (colorvar)
684
685     @ 3,0 SAY "" COLOR W+/N
686     @ 3,3 SAY ""
687
688     @ 4,0 SAY " " COLOR (colorvar)
689  CASE SUBSTR(instring,curlet,1) = "F"
690     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
691         FROM startx,curenty TO startx + 5,curenty + 6 ;
692         NONE NOSHADOW
693     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
694     curenty = curenty + 7
695     @ 0,0 SAY "" COLOR W+/N
696     @ 0,5 SAY "" COLOR (colorvar)
697
698     @ 1,0 SAY "" COLOR W+/N
699     @ 1,3 SAY ""
700
701     @ 2,0 SAY "" COLOR W+/N
702     @ 2,3 SAY "" COLOR (colorvar)
703
704     @ 3,0 SAY "" COLOR W+/N
705
706     @ 4,0 SAY " " COLOR (colorvar)
707  CASE SUBSTR(instring,curlet,1) = "G"
708     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
709         FROM startx,curenty TO startx + 5,curenty + 8 ;
710         NONE NOSHADOW
711     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
712     curenty = curenty + 9
713     @ 0,0 SAY ""
714     @ 0,3 SAY "" COLOR W+/N
715     @ 0,6 SAY ""
716
717     @ 1,0 SAY "" COLOR W+/N
718     @ 1,3 SAY "" COLOR (colorvar)
719     @ 1,4 SAY "  "
720     @ 1,7 SAY "" COLOR (colorvar)
721
722     @ 2,0 SAY "" COLOR W+/N
723     @ 2,3 SAY ""
724     @ 2,7 SAY "" COLOR (colorvar)
```

```
72
726     @ 3,0 SAY "  "
727     @ 3,6 SAY "" COLOR W+/N
728
729     @ 4,0 SAY "  " COLOR (colorvar)
730  CASE SUBSTR(instring,curlet,1) = "H"
731     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
732         FROM startx,curenty TO startx + 5,curenty + 7 ;
733         NONE NOSHADOW
734     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
735     curenty = curenty + 8
736     @ 0,0 SAY ""
737     @ 0,2 SAY "" COLOR (colorvar)
738     @ 0,4 SAY ""
739     @ 0,6 SAY "" COLOR (colorvar)
740
741     @ 1,0 SAY "" COLOR W+/N
742     @ 1,3 SAY ""
743     @ 1,6 SAY "" COLOR (colorvar)
744
745     @ 2,0 SAY "" COLOR W+/N
746     @ 2,3 SAY "" COLOR (colorvar)
747     @ 2,4 SAY "" COLOR W+/N
748
749     @ 3,0 SAY "" COLOR W+/N
750     @ 3,3 SAY ""
751     @ 3,4 SAY "" COLOR W+/N
752
753     @ 4,0 SAY "  " COLOR (colorvar)
754  CASE SUBSTR(instring,curlet,1) = "I"
755     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
756         FROM startx,curenty TO startx + 5,curenty + 3 ;
757         NONE NOSHADOW
758     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
759     curenty = curenty + 4
760     @ 0,0 SAY ""
761     @ 0,2 SAY "" COLOR (colorvar)
762
763     @ 1,0 SAY "" COLOR W+/N
764
765     @ 2,0 SAY "" COLOR W+/N
766
767     @ 3,0 SAY "" COLOR W+/N
768
769     @ 4,0 SAY " " COLOR (colorvar)
770  CASE SUBSTR(instring,curlet,1) = "J"
771     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
772         FROM startx,curenty TO startx + 5,curenty + 6 ;
773         NONE NOSHADOW
774     ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
775     curenty = curenty + 7
776     @ 0,0 SAY "  "
777     @ 0,5 SAY "" COLOR (colorvar)
778
779     @ 1,3 SAY "" COLOR W+/N
780
781     @ 2,3 SAY "" COLOR W+/N
782
783     @ 3,0 SAY ""
784     @ 3,4 SAY "" COLOR W+/N
785
786     @ 4,0 SAY "  " COLOR (colorvar)
787  CASE SUBSTR(instring,curlet,1) = "K"
788     DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
789         FROM startx,curenty TO startx + 5,curenty + 7 ;
790         NONE NOSHADOW
```

```
791       ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
792       curenty = curenty + 9
793       @ 0,0 SAY "▄▄"
794       @ 0,2 SAY "▀ " COLOR (colorvar)
795       @ 0,3 SAY "▄▀"
796       @ 0,5 SAY "▀▀" COLOR W+/N
797       @ 0,6 SAY "▄" COLOR (colorvar)
798
799       @ 1,0 SAY "▀▀▀ " COLOR W+/N
800       @ 1,5 SAY "▀▀" COLOR (colorvar)
801
802       @ 2,0 SAY "▀▀▀" COLOR W+/N
803       @ 2,4 SAY "▄"
804
805       @ 3,0 SAY "▀▀ " COLOR W+/N
806       @ 3,3 SAY "▀▄"
807
808       @ 4,0 SAY " ▀▀   ▀▀" COLOR (colorvar)
809    CASE SUBSTR(instring,curlet,1) = "L"
810       DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
811           FROM startx,curenty TO startx + 5,curenty + 6 ;
812           NONE NOSHADOW
813       ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
814       curenty = curenty + 7
815       @ 0,0 SAY "▄▄"
816       @ 0,2 SAY "▄" COLOR (colorvar)
817
818       @ 1,0 SAY "▀▀ " COLOR W+/N
819
820       @ 2,0 SAY "▀▀ " COLOR W+/N
821
822       @ 3,0 SAY "▀▄" COLOR W+/N
823       @ 3,3 SAY "▄"
824
825       @ 4,0 SAY " ▀▀▀▀" COLOR (colorvar)
826    CASE SUBSTR(instring,curlet,1) = "M"
827       DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
828           FROM startx,curenty TO startx + 5,curenty + 10 ;
829           NONE NOSHADOW
830       ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
831       curenty = curenty + 11
832       @ 0,0 SAY "▄▀  ▀▄"
833       @ 0,9 SAY "▄" COLOR (colorvar)
834
835       @ 1,0 SAY "▀▄▄▀" COLOR W+/N
836       @ 1,4 SAY "▀▄▀"
837       @ 1,9 SAY "▀" COLOR (colorvar)
838
839
840       @ 2,0 SAY "▀▄ ▀▄ ▀ " COLOR W+/N
841
842       @ 3,0 SAY "▀▄ " COLOR W+/N
843       @ 3,3 SAY "▀"
844       @ 3,5 SAY "▀" COLOR (colorvar)
845       @ 3,7 SAY "▀ " COLOR W+/N
846
847       @ 4,0 SAY " ▀▀  ▀  ▀▀" COLOR (colorvar)
848    CASE SUBSTR(instring,curlet,1) = "N"
849       DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
850           FROM startx,curenty TO startx + 5,curenty + 8 ;
851           NONE NOSHADOW
852       ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
853       curenty = curenty + 9
854       @ 0,0 SAY "▀▄  ▄▀"
855       @ 0,7  SAY "▀" COLOR (colorvar)
856
```



```
857       @ 1,0 SAY "▀▄▄▄" COLOR W+/N
858       @ 1,4 SAY "▄▀"
859       @ 1,7 SAY "▀" COLOR (colorvar)
860
861       @ 2,0 SAY "▀▀ " COLOR W+/N
862       @ 2,3 SAY "▀▀"
863       @ 2,4 SAY "▄▀▀ " COLOR W+/N
864
865       @ 3,0 SAY "▀▄ " COLOR W+/N
866       @ 3,3 SAY "▄▀"
867       @ 3,7 SAY " " COLOR W+/N
868
869       @ 4,0 SAY " ▀▀   ▀▀" COLOR (colorvar)
870
871    CASE SUBSTR(instring,curlet,1) = "O"
872       DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
873           FROM startx,curenty TO startx + 7,curenty + 7 ;
874           NONE NOSHADOW
875       ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
876       curenty = curenty + 9
877       @ 0,0 SAY " ▄▀"
878       @ 0,3 SAY "▀▀" COLOR W+/N
879       @ 0,4 SAY "▄▀"
880
881       @ 1,0 SAY "▀▄ " COLOR W+/N
882       @ 1,3 SAY "▀▄" COLOR (colorvar)
883       @ 1,5 SAY "▀▄"
884       @ 1,7 SAY "▀" COLOR (colorvar)
885
886       @ 2,0 SAY "▀▄ " COLOR W+/N
887       @ 2,3 SAY " ▀▄"
888       @ 2,7 SAY "▀" COLOR (colorvar)
889
890       @ 3,0 SAY " ▀▄▀"
891       @ 3,5 SAY "▀▄ " COLOR W+/N
892       @ 3,7 SAY "▀▄" COLOR (colorvar)
893
894       @ 4,0 SAY "   ▀▀▄" COLOR (colorvar)
895    CASE SUBSTR(instring,curlet,1) = "P"
896       DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
897           FROM startx,curenty TO startx + 5,curenty + 6 ;
898           NONE NOSHADOW
899       ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
900       curenty = curenty + 8
901       @ 0,0 SAY "▀▀" COLOR W+/N
902       @ 0,3 SAY "▀▄"
903
904       @ 1,0 SAY "▀▄ " COLOR W+/N
905       @ 1,3 SAY "▀▄"
906       @ 1,6 SAY "▀" COLOR (colorvar)
907
908       @ 2,0 SAY "▀▀▀▀ " COLOR W+/N
909       @ 2,6 SAY "▀▀" COLOR (colorvar)
910
911       @ 3,0 SAY "▀▀ " COLOR W+/N
912
913       @ 4,0 SAY " ▀▀" COLOR (colorvar)
914    CASE SUBSTR(instring,curlet,1) = "Q"
915       DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
916           FROM startx,curenty TO startx + 5,curenty + 8 ;
917           NONE NOSHADOW
918       ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
919       curenty = curenty + 9
920       @ 0,0 SAY " ▄▀"
921       @ 0,3 SAY "▀▀" COLOR W+/N
922       @ 0,4 SAY "▄▀"
```

```
923
924         @ 1,0 SAY "▀▀ " COLOR W+/N
925         @ 1,3 SAY "▀▀" COLOR (colorvar)
926         @ 1,5 SAY "▀▀"
927         @ 1,7 SAY "▄" COLOR (colorvar)
928
929         @ 2,0 SAY "▀▀ " COLOR W+/N
930         @ 2,3 SAY "▀▀"
931         @ 2,7 SAY "▀" COLOR (colorvar)
932
933         @ 3,0 SAY " ▀▀▀▄"
934         @ 3,6 SAY "▀▀" COLOR (colorvar)
935
936         @ 4,0 SAY "  ▀▀▀" COLOR (colorvar)
937         @ 4,6 SAY "▀▀"
938         @ 4,7 SAY "▄" COLOR (colorvar)
939      CASE SUBSTR(instring,curlet,1) = "R"
940         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
941             FROM startx,curenty TO startx + 5,curenty + 7 ;
942             NONE NOSHADOW
943         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
944         curenty = curenty + 8
945         @ 0,0 SAY "▀▀▀" COLOR W+/N
946         @ 0,3 SAY "▀▄"
947
948         @ 1,0 SAY "▀▄ " COLOR W+/N
949         @ 1,3 SAY "▀▀"
950         @ 1,6 SAY "▀" COLOR (colorvar)
951
952         @ 2,0 SAY "▀▀▀▄" COLOR W+/N
953         @ 2,6 SAY "▀▀" COLOR (colorvar)
954
955         @ 3,0 SAY "▀▀ " COLOR W+/N
956         @ 3,3 SAY "▀▀"
957         @ 3,6 SAY "▀" COLOR (colorvar)
958
959         @ 4,0 SAY "  ▀▀  ▀▀" COLOR (colorvar)
960      CASE SUBSTR(instring,curlet,1) = "S"
961         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
962             FROM startx,curenty TO startx + 5,curenty + 7 ;
963             NONE NOSHADOW
964         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
965         curenty = curenty + 8
966         @ 0,0 SAY "▄▀"
967         @ 0,2 SAY "▀▄▀" COLOR W+/N
968         @ 0,5 SAY "▄"
969
970         @ 1,0 SAY "▀▀▄ "
971         @ 1,5 SAY "▀▀" COLOR (colorvar)
972
973         @ 2,0 SAY "  ▀▀"
974         @ 2,6 SAY "▀" COLOR (colorvar)
975
976         @ 3,0 SAY "▀▄▄▀"
977         @ 3,5 SAY "▀▀ " COLOR W+/N
978
979         @ 4,0 SAY "  ▀▀▀▀" COLOR (colorvar)
980      CASE SUBSTR(instring,curlet,1) = "T"
981         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
982             FROM startx,curenty TO startx + 5,curenty + 7 ;
983             NONE NOSHADOW
984         ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
985         curenty = curenty + 8
986         @ 0,0 SAY "▀▀▀"
987         @ 0,1 SAY "▄▀▀▀▀" COLOR W+/N
988         @ 0,6 SAY "▄" COLOR (colorvar)
```

```
989
990         @ 1,2 SAY "▀▀ " COLOR W+/N
991
992         @ 2,2 SAY "▀▀ " COLOR W+/N
993
994         @ 3,2 SAY "▀▀ " COLOR W+/N
995
996         @ 4,0 SAY "   ▀▀▀" COLOR (colorvar)
997      CASE SUBSTR(instring,curlet,1) = "U"
998         DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
999             FROM startx,curenty TO startx + 5,curenty + 6 ;
1000            NONE NOSHADOW
1001        ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1002        curenty = curenty + 8
1003        @ 0,0 SAY "▀▀▀"
1004        @ 0,2 SAY "▀▄▀" COLOR (colorvar)
1005        @ 0,4 SAY "▀▀▀"
1006        @ 0,6 SAY "▄" COLOR (colorvar)
1007
1008        @ 1,0 SAY "▀▀ " COLOR W+/N
1009        @ 1,3 SAY "▀▀"
1010        @ 1,6 SAY "▀" COLOR (colorvar)
1011
1012        @ 2,0 SAY "▀▀ " COLOR W+/N
1013        @ 2,3 SAY "▀"
1014        @ 2,4 SAY "▀▀ " COLOR W+/N
1015
1016        @ 3,0 SAY "▀▀"
1017        @ 3,2 SAY "▄▀" COLOR W+/N
1018        @ 3,3 SAY "▀"
1019        @ 3,4 SAY "▀▀ " COLOR W+/N
1020
1021        @ 4,0 SAY "  ▀▀▀▀" COLOR (colorvar)
1022     CASE SUBSTR(instring,curlet,1) = "V"
1023        DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1024            FROM startx,curenty TO startx + 5,curenty + 7 ;
1025            NONE NOSHADOW
1026        ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1027        curenty = curenty + 9
1028        @ 0,0 SAY "▀▀▀"
1029        @ 0,2 SAY "▀▄▀" COLOR (colorvar)
1030        @ 0,3 SAY "  ▀▀"
1031        @ 0,7 SAY "▄▀" COLOR (colorvar)
1032
1033        @ 1,0 SAY " ▀▀"
1034        @ 1,3 SAY "▀▄" COLOR (colorvar)
1035        @ 1,4 SAY "▀▀ " COLOR W+/N
1036        @ 1,7 SAY "▀▀" COLOR (colorvar)
1037
1038        @ 2,0 SAY " ▄▀"
1039        @ 2,3 SAY "▄▀ " COLOR W+/N
1040
1041        @ 3,0 SAY "  ▀▀"
1042        @ 3,4 SAY "▀▀ " COLOR W+/N
1043
1044        @ 4,0 SAY "    ▀▀" COLOR (colorvar)
1045     CASE SUBSTR(instring,curlet,1) = "W"
1046        DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1047            FROM startx,curenty TO startx + 5,curenty + 10 ;
1048            NONE NOSHADOW
1049        ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1050        curenty = curenty + 12
1051        @ 0,0 SAY "▀▀▀"
1052        @ 0,2 SAY "▀▄▀" COLOR (colorvar)
1053        @ 0,4 SAY "▀▀▀"
1054        @ 0,6 SAY "▄▀ " COLOR (colorvar)
```

```
1055        @ 0,8 SAY "▀▀"
1056        @ 0,10 SAY "▄" COLOR (colorvar)
1057
1058        @ 1,0 SAY "▀▀"
1059        @ 1,2 SAY "▄" COLOR W+/N
1060        @ 1,3 SAY "▄▀"
1061        @ 1,6 SAY "▀" COLOR (colorvar)
1062        @ 1,7 SAY "▄"
1063        @ 1,9 SAY "▀ " COLOR W+/N
1064
1065        @ 2,1 SAY "▄▄▄▄ " COLOR W+/N
1066
1067        @ 3,2 SAY "▀▀ " COLOR W+/N
1068        @ 3,6 SAY "▀ " COLOR W+/N
1069        @ 3,9 SAY "▀▀" COLOR (colorvar)
1070
1071        @ 4,0 SAY "  ▀▀  ▀▀" COLOR (colorvar)
1072     CASE SUBSTR(instring,curlet,1) = "X"
1073        DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1074            FROM startx,curenty TO startx + 5,curenty + 8 ;
1075            NONE NOSHADOW
1076        ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1077        curenty = curenty + 9
1078        @ 0,0 SAY "▀▀"
1079        @ 0,2 SAY "▄" COLOR (colorvar)
1080        @ 0,3 SAY " ▀▀"
1081        @ 0,7 SAY "▄" COLOR (colorvar)
1082
1083        @ 1,0 SAY " ▀▄▄"
1084        @ 1,5 SAY "▀▀" COLOR W+/N
1085        @ 1,6 SAY "▀▀" COLOR (colorvar)
1086
1087        @ 2,0 SAY "  ▄▀"
1088        @ 2,3 SAY "▀▄" COLOR W+/N
1089
1090        @ 3,0 SAY "▀▀ " COLOR W+/N
1091        @ 3,3 SAY "▀▀" COLOR (colorvar)
1092        @ 3,4 SAY " ▀▀"
1093        @ 3,7 SAY "▄" COLOR (colorvar)
1094
1095        @ 4,0 SAY " ▀▀   ▀▀" COLOR (colorvar)
1096     CASE SUBSTR(instring,curlet,1) = "Y"
1097        DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1098            FROM startx,curenty TO startx + 5,curenty + 9 ;
1099            NONE NOSHADOW
1100        ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1101        curenty = curenty + 10
1102        @ 0,0 SAY "▀▀"
1103        @ 0,2 SAY "▄" COLOR (colorvar)
1104        @ 0,3 SAY "  ▀▀"
1105        @ 0,8 SAY "▄" COLOR (colorvar)
1106
1107        @ 1,0 SAY " ▀▀"
1108        @ 1,3 SAY "▄" COLOR (colorvar)
1109        @ 1,4 SAY "▄▀"
1110        @ 1,7 SAY "▀" COLOR (colorvar)
1111
1112        @ 2,0 SAY "  ▀▀"
1113        @ 2,5 SAY "▀ " COLOR W+/N
1114        @ 2,7 SAY "▀▀" COLOR (colorvar)
1115
1116        @ 3,3 SAY "▀▀ " COLOR W+/N
1117
1118        @ 4,0 SAY "    ▀▀" COLOR (colorvar)
1119     CASE SUBSTR(instring,curlet,1) = "Z"
1120        DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1121            FROM startx,curenty TO startx + 5,curenty + 6 ;
1122            NONE NOSHADOW
1123        ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1124        curenty = curenty + 8
1125        @ 0,0 SAY "▀▀"
1126        @ 0,1 SAY "▀▀▀▀" COLOR W+/N
1127        @ 0,6 SAY "▄" COLOR (colorvar)
1128
1129        @ 1,0 SAY "  ▄▀"
1130        @ 1,4 SAY "▀ " COLOR W+/N
1131        @ 1,6 SAY "▀▀" COLOR (colorvar)
1132
1133        @ 2,0 SAY " ▄▀"
1134        @ 2,3 SAY "▀▀" COLOR W+/N
1135
1136        @ 3,0 SAY "▀▄▄" COLOR W+/N
1137        @ 3,4 SAY "▄▀"
1138
1139        @ 4,0 SAY " ▀▀▀▀▀ " COLOR (colorvar)
1140     ENDCASE
1141     curlet = curlet + 1
1142  ENDDO
1143
1144     DEFINE WINDOW ("WRN" ) ;
1145         FROM x_coor+startx,y_coor-12 TO x_coor+startx + 11,y_coor + 60 ;
1146         NONE NOSHADOW
1147     ACTIVATE WINDOW ("WRN")
1148
1149     @ 2,17 SAY "Version 1.2  Date: Sept. 30, 1993"
1150     @ 3,17 SAY "(c) Naval Health Research Center."
1151
1152     @ 6,0 SAY "#################################################################
 => #############"
1153     @ 7,0 SAY "UNAUTHORIZED ACCESS TO THIS UNITED STATES GOVERNMENT COMPU
 => TER SYSTEM"
1154     @ 8,0 SAY "AND/OR SOFTWARE IS PROHIBITED BY TITLE 18, UNITED STATES C
 => ODE, SECTION"
1155     @ 9,0 SAY "1030.  THIS SYSTEM MAY ONLY BE USED FOR UNCLASSIFIED OFFIC
 => IAL BUSINESS."
1156     @ 10,0 SAY "################################################################
 => ############# "
1157
1158     WAIT "" TIMEOUT o_t
1159     IF NOT EMPTY(instring)
1160        curlet = 1
1161        FOR i = 1 TO len_str
1162           DEACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1163           RELEASE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1164           curlet = curlet + 1
1165        NEXT
1166     ENDIF
1167     DEACTIVATE WINDOW "wrn"
1168     RELEASE WINDOW "wrn"
1169     CLEAR
1170     *: EOF: BIGCHARS.ACT
```

```
*
*       09/02/92            BSELECT.PRG                 10:49:05
*
*       Author's Name
*
*       Copyright (c) 1992 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*

*
*                  BSELECT Setup Code - SECTION 1
*
*

#REGION 1
PARAMETER arraylist, msg
EXTERNAL ARRAY arraylist
IF PARAMETER() < 2
    RETURN ""
ENDIF

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()

*
*                  Window definitions
*
*

IF NOT WEXIST("bselect")
    DEFINE WINDOW bselect ;
        FROM INT((SROW()-8)/2),INT((SCOL()-60)/2) ;
        TO INT((SROW()-8)/2)+7,INT((SCOL()-60)/2)+59 ;
        TITLE "Boot Strap" ;
        NOFLOAT ;
        NOCLOSE ;
        SHADOW ;
        DOUBLE ;
        COLOR SCHEME 5
ENDIF
```



```


*
*                  BSELECT Setup Code - SECTION 2
*
*

#REGION 1
u = "^"
m.text = ""

*
*                  BSELECT Screen Layout
*
*

#REGION 1
IF WVISIBLE("bselect")
    ACTIVATE WINDOW bselect SAME
ELSE
    ACTIVATE WINDOW bselect NOSHOW
ENDIF
@ 1,0 GET m.f1 ;
    PICTURE "@^" ;
    FROM arraylist ;
    SIZE 3,57 ;
    DEFAULT 1 ;
    COLOR SCHEME 5, 6
@ 5,21 GET m.action ;
    PICTURE "@*HT \<Ok;\<Cancel" ;
    SIZE 1,8,1 ;
    DEFAULT 1 ;
    VALID _q8q0n6sbm()
@ 0,0 SAY m.msg ;
    SIZE 1,37

IF NOT WVISIBLE("bselect")
    ACTIVATE WINDOW bselect
ENDIF

READ CYCLE MODAL

RELEASE WINDOW bselect
SELECT (m.currarea)

#REGION 0
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF

*
*                  BSELECT Cleanup Code
*
*
```

```
133
134     #REGION 1
135     RETURN m.text
136
137     *
138     *
139     *           _Q8Q0N6SBM            m.action VALID
140     *
141     *           Function Origin:
142     *
143     *           From Screen:          BSELECT,     Record Number:    3
144     *           Variable:             m.action
145     *           Called By:            VALID Clause
146     *           Object Type:          Push Button
147     *           Snippet Number:       1
148     *
149     *
150     *
151     FUNCTION _q8q0n6sbm     &&  m.action VALID
152     #REGION 1
153     IF m.action = 1
154        m.text = arraylist[m.f1,1]
155     ELSE
156        m.text = ""
157     ENDIF
158     *: EOF: BSELECT.ACT
```

```
  1  *:*********************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\CHOOSER.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 11/30/93     17:29
  9  *:
 10  *:  Procs & Fncts: _Q120IDFTF()
 11  *:                : _Q120IDG17()
 12  *:                : _Q120IDG86()
 13  *:
 14  *:         Set by: GET_CF()          (function  in HMCFE.SPR)
 15  *:                : GET_CFE          (procedure in HMCFEI.SPR)
 16  *:                : GET_HMLC()       (function  in HMTAB.SPR)
 17  *:                : GET_HMWP()       (function  in HMTAB.SPR)
 18  *:                : GET_HMET()       (function  in HMTAB.SPR)
 19  *:                : GET_CF1()        (function  in HMTAB.SPR)
 20  *:                : GET_HMNAME       (procedure in HMLU.PRG)
 21  *:                : GET_HMSC()       (function  in HMSCEN.PRG)
 22  *:                : GET_HMAT()       (function  in HMSTEP.PRG)
 23  *:
 24  *:          Calls: _Q120IDFTF()      (function  in CHOOSER.PRG)
 25  *:                : _Q120IDG17()     (function  in CHOOSER.PRG)
 26  *:                : _Q120IDG86()     (function  in CHOOSER.PRG)
 27  *:
 28  *:      Documented 12/01/93 at 11:32                FoxDoc  version 2.10f
 29  *:*********************************************************************
=> ******
 30      *       ************************************************************
 31      *       *                                                          *
 32      *       * 12/03/91             CHOOSER.PRG                08:34:18 *
 33      *       *                                                          *
 34      *       ************************************************************
 35      *       *                                                          *
 36      *       * Adam Green                                               *
 37      *       *                                                          *
 38      *       * Copyright (c) 1991 Adam Green Seminars                   *
 39      *       * One Faneuil Hall                                         *
 40      *       * Boston, MA  02174                                        *
 41      *       *                                                          *
 42      *       * Description:                                             *
 43      *       * This program was automatically generated by GENSCRN.     *
 44      *       *                                                          *
 45      *       ************************************************************
 46
 47
 48      *       ************************************************************
 49      *       *                                                          *
 50      *       *             CHOOSER Setup Code - SECTION 1               *
 51      *       *                                                          *
 52      *       ************************************************************
 53      *
 54
 55      #REGION 1
 56      PARAMETERS choicelist, MESSAGE
 57
 58      * Create a default message
 59     ┌IF PARAMETERS() = 1
 60     │   m.message = "Make a Selection"
 61     │
 62     ├ELSE
 63     │   * Truncate the message to fit in an @SAY
 64     │  ┌IF LEN( m.message ) > 60
 65     │  │   m.message = SUBSTR( m.message, 1,60 )
 66     │  └ENDIF
 67     └ENDIF
 68      m.message = PADC( m.message, 60 )
 69
 70
 71      #REGION 0
 72      REGIONAL m.currarea, m.talkstat, m.compstat
 73      SET TALK OFF
 74
 75     ┌IF SET("TALK") = "ON"
 76     │   SET TALK OFF
 77     │   m.talkstat = "ON"
 78     ├ELSE
 79     │   m.talkstat = "OFF"
 80     └ENDIF
 81      m.compstat = SET("COMPATIBLE")
 82      SET COMPATIBLE FOXPLUS
 83
 84      *       ************************************************************
 85      *       *                                                          *
 86      *       *                     Window definitions                   *
 87      *       *                                                          *
 88      *       ************************************************************
 89      *
 90
 91     ┌IF NOT WEXIST("chooser")
 92     │   DEFINE WINDOW chooser ;
 93     │       FROM INT((SROW()-19)/2),INT((SCOL()-77)/2) ;
 94     │       TO INT((SROW()-19)/2)+18,INT((SCOL()-77)/2)+76 ;
 95     │       FLOAT ;
 96     │       CLOSE ;
 97     │       SHADOW ;
 98     │       MINIMIZE ;
 99     │       SYSTEM ;
100     │       COLOR SCHEME 8
101     └ENDIF
102
103
104      *       ************************************************************
105      *       *                                                          *
106      *       *             CHOOSER Setup Code - SECTION 2               *
107      *       *                                                          *
108      *       ************************************************************
109      *
110
111      #REGION 1
112      PRIVATE ALL
113
114      * Initialize the push buttons to the Ok button
115      m.s1 = 1
116
117      * If an array is referenced without being declared
118      * in this file the Project Manager reports an error
119      * EXTERNAL tells the Project Manager not to worry about it
120      EXTERNAL ARRAY choicelist
121
122      * Start the list box variable on the first array element
123      * and initialize it as character type
124      m.choice = choicelist[ 1 ]
125
126      *       ************************************************************
127      *       *                                                          *
128      *       *                   CHOOSER Screen Layout                  *
129      *       *                                                          *
130      *       ************************************************************
```

```
131     *
132
133     #REGION 1
134     IF WVISIBLE("chooser")
135        ACTIVATE WINDOW chooser SAME
136     ELSE
137        ACTIVATE WINDOW chooser NOSHOW
138     ENDIF
139     @ 1,1 SAY m.message ;
140         SIZE 1,69
141     @ 3,2 GET m.choice ;
142         PICTURE "@&N" ;
143         FROM choicelist ;
144         SIZE 11,68 ;
145         DEFAULT 1 ;
146         VALID  _q120idftf() ;
147         COLOR SCHEME 9
148     @ 15,27 GET m.s1 ;
149         PICTURE "@*HT \!\<OK;\?\<Cancel" ;
150         SIZE 1,8,3 ;
151         DEFAULT 1 ;
152         VALID _q120idg17()
153
154     IF NOT WVISIBLE("chooser")
155        ACTIVATE WINDOW chooser
156     ENDIF
157
158     READ CYCLE ;
159         WHEN _q120idg86()
160
161     RELEASE WINDOW chooser
162
163     #REGION 0
164     IF m.talkstat = "ON"
165        SET TALK ON
166     ENDIF
167     IF m.compstat = "ON"
168        SET COMPATIBLE ON
169     ENDIF
170
171
172     *       ************************************************************
173     *       *                                                          *
174     *       *                  CHOOSER Cleanup Code                    *
175     *       *                                                          *
176     *       ************************************************************
177     *
178
179     #REGION 1
180     * Return the selected text if the user pressed OK
181     IF m.s1 = 1
182  <------RETURN m.choice
183     ELSE
184  <------RETURN ""
185     ENDIF
186
187     *       ************************************************************
188     *       *                                                          *
189     *       * _Q120IDFTF           m.Choice VALID                      *
190     *       *                                                          *
191     *       * Function Origin:                                         *
192     *       *                                                          *
193     *       * From Screen:         CHOOSER,     Record Number:    2   *
194     *       * Variable:            m.Choice                            *
195     *       * Called By:           VALID Clause                        *
196     *       * Object Type:         List                                *
```



```
197     *       * Snippet Number:      1                                   *
198     *       *                                                          *
199     *       ************************************************************
200     *
208  *!*************************************************************************
=> ******
209  *!
210  *!       Function: _Q120IDFTF()
211  *!
212  *!      Called by: CHOOSER.PRG
213  *!
214  *!*************************************************************************
=> ******
208     FUNCTION _q120idftf     &&  m.Choice VALID
209     #REGION 1
210     CLEAR READ
211
212     *       ************************************************************
213     *       *                                                          *
214     *       * _Q120IDG17           m.S1 VALID                          *
215     *       *                                                          *
216     *       * Function Origin:                                         *
217     *       *                                                          *
218     *       * From Screen:         CHOOSER,     Record Number:    4   *
219     *       * Variable:            m.S1                                *
220     *       * Called By:           VALID Clause                        *
221     *       * Object Type:         Push Button                         *
222     *       * Snippet Number:      2                                   *
223     *       *                                                          *
224     *       ************************************************************
225     *
233  *!*************************************************************************
=> ******
234  *!
235  *!       Function: _Q120IDG17()
236  *!
237  *!      Called by: CHOOSER.PRG
238  *!
239  *!*************************************************************************
=> ******
233     FUNCTION _q120idg17     &&  m.S1 VALID
234     #REGION 1
235     CLEAR READ
236
237     *       ************************************************************
238     *       *                                                          *
239     *       * _Q120IDG86           Read Level When                     *
240     *       *                                                          *
241     *       * Function Origin:                                         *
242     *       *                                                          *
243     *       * From Screen:         CHOOSER                             *
244     *       * Called By:           READ Statement                      *
245     *       * Snippet Number:      3                                   *
246     *       *                                                          *
247     *       ************************************************************
248     *
256  *!*************************************************************************
=> ******
257  *!
258  *!       Function: _Q120IDG86()
259  *!
260  *!      Called by: CHOOSER.PRG
261  *!
262  *!*************************************************************************
=> ******
256     FUNCTION _q120idg86     && Read Level When
```

```
257      *
258      * When Code from screen: CHOOSER
259      *
260      #REGION 1
261      * Start the READ on the list box
262       CUROBJ = OBJNUM( m.choice )
263      *: EOF: CHOOSER.ACT
```

```
*
*       09/02/92            BWFACT.PRG                  11:04:28
*
*       Author's Name
*
*       Copyright (c) 1992 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*

*
*                  BWFACT Setup Code - SECTION 1
*
*

#REGION 1
PARAMETER array1, array2
EXTERNAL ARRAY array1, array2
IF PARAMETER() < 2
    RETURN ""
ENDIF


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()


*
*                  Window definitions
*
*

IF NOT WEXIST("bwfact")
    DEFINE WINDOW bwfact ;
        FROM INT((SROW()-11)/2),INT((SCOL()-64)/2) ;
        TO INT((SROW()-11)/2)+10,INT((SCOL()-64)/2)+63 ;
        TITLE "Boot strap" ;
        NOFLOAT ;
        NOCLOSE ;
        SHADOW ;
        DOUBLE ;
        COLOR SCHEME 5
ENDIF
```



```


*
*                  BWFACT Setup Code - SECTION 2
*
*

#REGION 1
u = "^"
m.text = ""


*
*                  BWFACT Screen Layout
*
*

#REGION 1
IF WVISIBLE("bwfact")
    ACTIVATE WINDOW bwfact SAME
ELSE
    ACTIVATE WINDOW bwfact NOSHOW
ENDIF
@ 1,18 GET m.f1 ;
    PICTURE "@^" ;
    FROM array1 ;
    SIZE 3,41 ;
    DEFAULT 1 ;
    COLOR SCHEME 5, 6
@ 4,18 GET m.f2 ;
    PICTURE "@^" ;
    FROM array2 ;
    SIZE 3,41 ;
    DEFAULT 1 ;
    COLOR SCHEME 5, 6
@ 5,8 SAY "at step:"
@ 2,2 SAY "Select Factor:"
@ 8,24 GET m.action ;
    PICTURE "@*HT \<Ok;\<Cancel" ;
    SIZE 1,8,1 ;
    DEFAULT 1 ;
    VALID _q8q0nqka6()

IF NOT WVISIBLE("bwfact")
    ACTIVATE WINDOW bwfact
ENDIF

READ CYCLE MODAL

RELEASE WINDOW bwfact
SELECT (m.currarea)


#REGION 0
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF
```

```
133     *
134     *
135     *                    BWFACT Cleanup Code
136     *
137     *
138     *
139
140     #REGION 1
141
142     RETURN m.text
143
144     *
145     *
146     *          _Q8Q0NQKA6          m.action VALID
147     *
148     *          Function Origin:
149     *
150     *          From Screen:        BWFACT,     Record Number:    6
151     *          Variable:           m.action
152     *          Called By:          VALID Clause
153     *          Object Type:        Push Button
154     *          Snippet Number:     1
155     *
156     *
157     *
158     FUNCTION _q8q0nqka6     &&  m.action VALID
159     #REGION 1
160     IF m.action = 1
161        m.text = array1[m.f1,1] + u + array2[m.f2,1]
162     ELSE
163        m.text = ""
164     ENDIF
165     *: EOF: BWFACT.ACT
```

```
*
*       08/24/93            DISPLAY.PRG                 10:23:08
*
*       Author's Name
*
*       Copyright (c) 1993 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*

PARAMETERS string

*
*            DISPLAY/MS-DOS Setup Code - SECTION 1
*
*

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()


*
*            MS-DOS Window definitions
*
*

IF NOT WEXIST("w_hmresult") ;
      OR UPPER(WTITLE("W_HMRESULT")) == "W_HMRESULT.PJX" ;
      OR UPPER(WTITLE("W_HMRESULT")) == "W_HMRESULT.SCX" ;
      OR UPPER(WTITLE("W_HMRESULT")) == "W_HMRESULT.MNX" ;
      OR UPPER(WTITLE("W_HMRESULT")) == "W_HMRESULT.PRG" ;
      OR UPPER(WTITLE("W_HMRESULT")) == "W_HMRESULT.FRX" ;
      OR UPPER(WTITLE("W_HMRESULT")) == "W_HMRESULT.QPR"
   DEFINE WINDOW w_hmresult ;
      FROM INT((SROW()-20)/2),INT((SCOL()-77)/2) ;
      TO INT((SROW()-20)/2)+19,INT((SCOL()-77)/2)+76 ;
      TITLE "Hazard Material Life Cycle Cost" ;
      NOFLOAT ;
      NOCLOSE ;
      SHADOW ;
      nominimize ;
      DOUBLE ;
      COLOR SCHEME 1
ENDIF
```



```


*
*            DISPLAY/MS-DOS Setup Code - SECTION 2
*
*

#REGION 1
m.action = 1
u = ""
m.text = m.string

*
*            DISPLAY/MS-DOS Screen Layout
*
*

#REGION 1
IF WVISIBLE("w_hmresult")
   ACTIVATE WINDOW w_hmresult SAME
ELSE
   ACTIVATE WINDOW w_hmresult NOSHOW
ENDIF
@ 1,2 EDIT m.text ;
     SIZE 14,72,0 ;
     DEFAULT " " ;
     SCROLL ;
     NOMODIFY

@ 16,31 GET m.action ;
     PICTURE "@*HT \<Print;\<Quit" ;
     SIZE 1,7,3 ;
     DEFAULT 1 ;
     VALID _qin0m9e2j()

IF NOT WVISIBLE("w_hmresult")
   ACTIVATE WINDOW w_hmresult
ENDIF

READ CYCLE MODAL

RELEASE WINDOW w_hmresult
SELECT (m.currarea)


#REGION 0
IF m.talkstat = "ON"
   SET TALK ON
ENDIF
IF m.compstat = "ON"
   SET COMPATIBLE ON
ENDIF


*
*       _QIN0M9E2J          m.action VALID
*
*       Function Origin:
*
*       From Platform:      MS-DOS
*       From Screen:        DISPLAY,     Record Number:    3
```

```
133     *         | Variable:            m.action
134     *         | Called By:           VALID Clause
135     *         | Snippet Number:      1
136     *         |
137     *         |
138     *
139     FUNCTION _qin0m9e2j     &&  m.action VALID
140     #REGION 1
141     IF m.action = 1
142        DO w_print.spr WITH m.text
143     ENDIF
144     *: EOF: DISPLAY.ACT
```

```
  1  *:*****************************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\DP.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 09/10/93      8:12
  9  *:
 10  *:         Set by: HMTAB.SPR
 11  *:               : HMSC.PRG
 12  *:               : HMMSDS.PRG
 13  *:               : SUBCOMPUT          (procedure in HMCOMP.SPR)
 14  *:               : SETUPBOOT()        (function  in HMCOMP.SPR)
 15  *:               : GETSTR()           (function  in HMCOMP.SPR)
 16  *:               : _Q7D001E18()       (function  in HMLU.PRG)
 17  *:               : FILEFIND()         (function  in HMSC.PRG)
 18  *:               : _Q8K0INGJH()       (function  in HMSCEN.PRG)
 19  *:               : _Q8K0INGTV()       (function  in HMSCEN.PRG)
 20  *:               : _Q1P0VASYA()       (function  in HMSTEP.PRG)
 21  *:               : _Q1P0VAT9X()       (function  in HMSTEP.PRG)
 22  *:               : _Q1P0VATLC()       (function  in HMSTEP.PRG)
 23  *:
 24  *:      Documented 12/01/93 at 11:32                FoxDoc  version 2.10f
 25  *:*****************************************************************************
=> ******
 26      * ============================================================
=> ==================
 27      * Date: 08/04/92  10:01:15
 28      * Program Name: Question.prg
 29      * Author's Name: Hoa Le Ly
 30      *
 31      * Copyright (c) 1992 Company Name: NHRC
 32      * Departament: Code 22
 33      * San Diego, CA 92138 - 5122
 34      * Description: This program emulate $PIECE of MUMPS function.  Which
=> return the
 35      *    the portion of string which is bounded by the characters in deli
=> miter. If both
 36      *    expr and expr2 are present, the value returned includes all char
=> acters from
 37      *    the expr1-1th occurrence of delimiter, up to but not including
=> the expr2th
 38      *    occurence of delimiter.  If expr2 is not present. it is assumed
=> to have the same
 39      *    value as expr.  If expr1 is not present. Then its value is assu
=> me to be 1
 40      * SYNTAX: DP(string, delimiter[,expr[,expr2]])
 41      * PARAMETER: string: Character expression which character extract fro
=> m
 42      *                delimiter: Character to delimiter
 43      *                    expr: start of number occurrence of delimiter
 44      *                    expr2: number of occurrence delimiter
 45      * eg: string = "last, first^age^date of birth"
 46      *     ?dp(string, "^",1)           ===>  last, first
 47      *     ?dp(string,"^",1,2)          ===>  last, first^age
 48      * ============================================================
=> ==================
 49
 50      PARAMETER ms,mm,mp,mp2
 51      PRIVATE ALL
 52      DO CASE
 53      CASE PARAMETER() < 2
 54  <-------RETURN ""
 55      CASE PARAMETER() = 2
 56          mp = 1
 57          mp2 = -1
 58      CASE PARAMETER() = 3
 59          IF TYPE("mp") != "N"
 60  <-----------RETURN ""
 61          ELSE
 62              IF mp < 1
 63  <---------------RETURN ""
 64              ENDIF
 65          ENDIF
 66          mp2 = -1
 67      CASE PARAMETER() = 4
 68          IF TYPE("mp") != "N"
 69  <-----------RETURN ""
 70          ENDIF
 71          IF TYPE("mp2") = "N"
 72              mp2 = IIF(mp >= mp2, -1, mp2)
 73          ELSE
 74              mp2 = -1
 75          ENDIF
 76      ENDCASE
 77      IF TYPE("ms") != "C"
 78          ms = STR(ms)
 79      ENDIF
 80      moccurs = OCCURS(mm,ms)
 81      IF moccurs = 0                                          && Return ms if
=> mp=1&(ms'[mm)
 82          mstr = IIF(mp = 1, ms, "")                  && Return null if mp>1&(
=> ms'[mm)
 83  <-------RETURN mstr
 84      ENDIF
 85      mbegin = IIF(mp = 1, 1, (AT(mm,ms,(mp-1))+1))
 86      IF mp2 = -1
 87          mend = AT(mm,ms,mp) - 1
 88          IF mend < 0
 89              IF mbegin > 1
 90                  mend = LEN(ms)
 91              ELSE
 92  <---------------RETURN ""
 93              ENDIF
 94          ENDIF
 95      ELSE
 96          mend = AT(mm,ms,mp2) - 1
 97          mend = IIF(mend<0,LEN(ms),mend)
 98      ENDIF
 99      mstr = SUBSTR(ms, mbegin, (mend - mbegin +1))
100      RETURN mstr
101      *: EOF: DP.ACT
```

```
  1  *:**************************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\ERRMSG.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 09/10/93      8:12
  9  *:
 10  *:         Set by: HMTAB.SPR
 11  *:               : BACKUP.PRG
 12  *:               : YESNO.PRG
 13  *:               : HMSTEP.PRG
 14  *:               : HMGETPWD.PRG
 15  *:               : _QKFOVR394()       (function  in HMAT.SPR)
 16  *:               : _QKFOVR3XH()       (function  in HMAT.SPR)
 17  *:               : _QKFOVR6TE()       (function  in HMLC.SPR)
 18  *:               : _QKFOVR991()       (function  in HMWP.SPR)
 19  *:               : _QKFOVR9C5()       (function  in HMWP.SPR)
 20  *:               : _QKFOVR9T3()       (function  in HMET.SPR)
 21  *:               : _QKFOVRE5A()       (function  in HMCF.SPR)
 22  *:               : _QKFOVRE7O()       (function  in HMCF.SPR)
 23  *:               : GET_CF()           (function  in HMCFE.SPR)
 24  *:               : GET_CFE            (procedure in HMCFE1.SPR)
 25  *:               : GET_HMLC()         (function  in HMTAB.SPR)
 26  *:               : GET_HMWP()         (function  in HMTAB.SPR)
 27  *:               : GET_HMET()         (function  in HMTAB.SPR)
 28  *:               : GET_CF1()          (function  in HMTAB.SPR)
 29  *:               : ERRHAND            (procedure in BACKUP.PRG)
 30  *:               : STOP               (procedure in BACKUP.PRG)
 31  *:               : CANCEL             (procedure in HMCOMP.SPR)
 32  *:               : SETUPBOOT()        (function  in HMCOMP.SPR)
 33  *:               : GET_HMNAME         (procedure in HMLU.PRG)
 34  *:               : _Q8KOINGTV()       (function  in HMSCEN.PRG)
 35  *:               : GET_HMAT()         (function  in HMSTEP.PRG)
 36  *:               : GET_HMUNIT         (procedure in HMSTEP.PRG)
 37  *:               : DATACHECK()        (function  in HMSTEP.PRG)
 38  *:               : _QKFOVROFQ         (procedure in HMENU.MPR)
 39  *:
 40  *:      Documented 12/01/93 at 11:31                FoxDoc  version 2.10f
 41  *:**************************************************************************
=> ******
 42      * Open a message window
 43      *
 44      * ERRMSG( <expC>[, <expN>] )
 45      *
 46      PARAMETERS errmsg, timelimit
 47      PRIVATE ALL
 48
 49      * set talk off
 50      savetalk=SET("TALK")
 51      SET TALK OFF
 52
 53      * If no errmessage is sent then quit
 54     ┌IF PARAMETERS()=0
 55  <──┘  RETURN
 56     └ENDIF
 57
 58      * m.TimeLimit is used in a WAIT TIMEOUT command
 59      * to control the amount of time ERRMSG display's its message
 60      * If no time limit is received set time to 0 wait forever
 61
 62     ┌IF PARAMETERS()=1 OR EMPTY(m.timelimit)
 63     │   m.timelimit=0
 64     └ENDIF
 65
 66     ┌IF TYPE("m.TimeLimit")!="N"
 67     │   m.timelimit=IIF(TYPE("m.TimeLimit")="C",INT(VAL(m.timelimit)),0)
 68     └ENDIF
 69
 70      m.errmsg=IIF(TYPE("m.errmsg")!="C","",m.errmsg)
 71
 72     ┌IF PARAMETERS()=2 AND EMPTY(m.errmsg) AND m.timelimit>0
 73  <──┘  RETURN
 74     └ENDIF
 75
 76      * Get the length of the message to size window
 77      m.len=LEN(m.errmsg)
 78
 79      * Set miminum length for the Press any key
 80     ┌IF m.timelimit = 0
 81     │   m.len=IIF(m.len<28,28,m.len)
 82     └ENDIF
 83
 84     ┌IF m.len >70
 85     │   m.errmsg=SUBSTR(m.errmsg,1,70)
 86     │   m.len=70
 87     └ENDIF
 88
 89      *Find begining/ending of the window
 90      m.begin=40-(INT(m.len/2)+1)
 91      m.end=40+(INT(m.len/2)+1)
 92
 93      * Remember the current window status
 94      m.oldwindow =IIF( WOUTPUT() = errmsg, "" , WOUTPUT() )
 95
 96     ┌IF NOT WEXIST("ErrMsg")
 97     │   DEFINE WINDOW errmsg ;
 98     │       FROM 1,m.begin ;
 99     │       TO 4,m.end ;
100     │       SHADOW ;
101     │       DOUBLE ;
102     │       COLOR SCHEME 5
103     └ENDIF
104
105      ACTIVATE WINDOW errmsg
106
107      * Print the message centered in the window
108
109      CLEAR
110      @ 0, ( WCOL() - LEN( m.errmsg ) )/2 SAY m.errmsg
111
112     ┌IF m.timelimit = 0
113     │   m.presskey = "Press any key to continue"
114     │   @ 1, ( WCOL() - LEN( m.presskey ) )/2 SAY m.presskey
115     └ENDIF
116
117      * Wait for the number of seconds in m.Timeout
118      * A value of 0 will wait forever
119      WAIT "" TIMEOUT m.timelimit
120
121      * Close Window
122      RELEASE WINDOW errmsg
123
124      * If there was no output window originally
125     ┌IF EMPTY( m.oldwindow )
126     │
127     │   * Send future output back to the screen
128     │   ACTIVATE SCREEN
129     │
130     ├ELSE
```

```
131
132       * Return output to the original window
133       ACTIVATE WINDOW ( m.oldwindow )
134     ENDIF
135
136     SET TALK &savetalk   && Restore original TALK setting
137     SET COLOR SET TO && RESTORE OLD COLOR SET
138     RETURN
139     *: EOF: ERRMSG.ACT
```

```
  1  *:*************************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\HMGETPWD.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 09/10/93      8:12
  9  *:
 10  *:         Set by: HMINIT.PRG
 11  *:              : HMINIT.PRG
 12  *:
 13  *:          Calls: ERRMSG.PRG
 14  *:
 15  *:      Documented 11/24/93 at 07:55                FoxDoc  version 2.10f
 16  *:*************************************************************************
=> ******
 17      *-------------------------------------------------------------------
=> ---
 18      * Validation the password.  Is boolean returnable
 19      *-------------------------------------------------------------------
=>
 20      PARAMETER m.pass
 21      PRIVATE ALL
 22     IF SET('TALK') = 'ON'     && TALK handled as a special case.
 23        SET TALK OFF          && Turn TALK OFF
 24        savetalk = 'ON'       && TALK was ON, save the setting
 25     ELSE                      && TALK is OFF
 26        savetalk = 'OFF'      && TALK was OFF, save the setting
 27     ENDIF
 28
 29     IF SET('ESCAPE') = 'ON'  && ESCAPE handled as a special case.
 30        SET ESCAPE OFF        && Turn ESCAPE OFF
 31        savesc = 'ON'         && ESCAPE was ON, save the setting
 32     ELSE                      && ESCAPE is OFF
 33        savesc = 'OFF'        && ESCAPE was OFF, save the setting
 34     ENDIF
 35
 36     IF PARAMETER()=0
 37        m.pass=""
 38     ENDIF
 39
 40      mflag = .T.
 41      syspass="SYSHAZ"
 42      ************************************
 43      *    CREATE WINDOW  GET_PWD        *
 44      ************************************
 45
 46      DEFINE WINDOW get_pwd FROM 10,16 TO 16,63 DOUBLE FLOAT NOCLOSE GROW Z
=> OOM
 47      ACTIVATE WINDOW get_pwd
 48
 49      temp_pass = SPACE (8)
 50      @ 2,7 SAY "ENTER PASSWORD:"
 51      @ 2,33 GET temp_pass
 52      CLEAR GETS
 53      @ 2,32 SAY SPACE(1)
 54      tpasswrd=[]
 55      tcol = 33
 56      msuccess = .F.
 57      mtry = 0
 58     DO WHILE mflag
 59        key1 = 0
 60        DO WHILE key1 = 0
 61           key1 = INKEY()
 62        ENDDO
 63        IF ( (LASTKEY() >= -9 .AND. LASTKEY() <= -1) OR (LASTKEY() = 16) )
=>   && ignore F2 to F10 and DEL key
 64           key1 = 0
 65           LOOP
 66        ENDIF
 67        IF LASTKEY() = 28  && F1
 68           *HELP PASSWORD
 69           key1 = 0
 70           LOOP
 71        ENDIF
 72        IF ( (LASTKEY() = 19) OR (LASTKEY() = 127) )  && back arrow or bac
=> k space
 73           IF tcol = 33
 74              key1 = 0
 75              LOOP
 76           ELSE
 77              IF tcol > 33
 78                 tcol = tcol - 1
 79                 @ 2,tcol SAY [ ] COLOR SCHEME 10
 80                 @ 2,tcol SAY ""
 81              ELSE
 82                 @ 2,32 SAY SPACE(1)
 83              ENDIF
 84              tpasswrd = SUBSTR(tpasswrd,1,LEN(tpasswrd) - 1)
 85              LOOP
 86           ENDIF
 87        ENDIF
 88        IF ( (LASTKEY() = 27) OR (LASTKEY() = 5) )  && ESC
 89           mflag = .F.
 90           EXIT
 91        ENDIF
 92        IF (key1 = 13)
 93           IF UPPER(tpasswrd) == UPPER(m.pass) OR UPPER(tpasswrd)==syspass
 94              msuccess = .T.
 95              mflag = .F.
 96              F
 97                 uccess = .F.
 98                 ry = mtry + 1
 99                  mtry > 2
100                  mflag = .F.
101                  =errmsg("Invalid User, Try next time !!!!!!",1)
102               LSE
103                  =errmsg("Invalid Password, Try again !!",1)
104                  @ 2, 33 GET temp_pass
105                  CLEAR GETS
106                  @ 2, 32 SAY SPACE(1)
107                  tpasswrd = []
108                  tcol = 32
109              ENDIF
110           ENDIF
111        ELSE
112           tpasswrd = tpasswrd + CHR(key1)
113           @ 2,tcol SAY [*] COLOR SCHEME 10
114        ENDIF
115        tcol = tcol + 1
116     ENDDO
117     DEACTIVATE WINDOW get_pwd
118     RELEASE WINDOW get_pwd
119     SET ESCAPE &savesc && Restore original ESCAPE setting
120     SET TALK &savetalk   && Restore original TALK setting
121
122     RETURN tpasswrd
123     *: EOF: HMGETPWD.ACT
```

```
  1 *:*************************************************************************
=> ******
  2 *:
  3 *: Procedure file: C:\HAZMAT\GHM\WORK\HMINIT.PRG
  4 *:
  5 *:         System: Hazardous Material Life-Cycle Cost Model
  6 *:         Author: Naval Health Research Center
  7 *:      Copyright (c) 1993, Naval Health Research Center
  8 *:  Last modified: 09/10/93      8:11
  9 *:
 10 *:  Procs & Fncts: MYHANDLER()
 11 *:               : _QUIT()
 12 *:
 13 *:          Calls: HMGETPWD.PRG
 14 *:               : BIGCHARS.PRG
 15 *:               : HMENU.MPR
 16 *:               : MYHANDLER()        (function  in HMINIT.PRG)
 17 *:
 18 *:      Documented 12/01/93 at 11:31               FoxDoc  version 2.10f
 19 *:*************************************************************************
=> ******
 20      ************************************************************************
=> *
 21      *
 22      *    FUNCTION:HMINIT.PRG
 23      *    PURPOSE: DRIVER FOR HMENU
 24      *             NOTE: 3 CHANCES FOR ACCESING ALLOWED.
 25      *    PROCEDURES CALLED:
 26      *    SIDE EFFECTS: NONE KNOWN.
 27      *    REFERENCE: C:\EXAMPLE\PRGS\EX2.PRG
 28      *    CREATED: 06/12/92                AL
 29      *    MODIFIED: 06/14/92               AL
 30      *
 31      *
 32      ************************************************************************
=> *
 33
 34      ************************************************************************
=> *
 35      *
 36      *    Initialize system init_app()
 37      *
 38      ************************************************************************
=> *
 39      CLOSE ALL
 40      CLEAR WINDOWS ALL
 41      CLEAR ALL
 42      PUBLIC m.pass,savetalk,savesc        && Initialize public variables
 43      PUBLIC dropdead
 44
 45      IF SET('TALK') = 'ON'     && TALK handled as a special case.
 46         SET TALK OFF          && Turn TALK OFF
 47         savetalk = 'ON'       && TALK was ON, save the setting
 48      ELSE                      && TALK is OFF
 49         savetalk = 'OFF'      && TALK was OFF, save the setting
 50      ENDIF
 51
 52      IF SET('ESCAPE') = 'ON'  && ESCAPE handled as a special case.
 53         SET ESCAPE OFF       && Turn ESCAPE OFF
 54         savesc = 'ON'        && ESCAPE was ON, save the setting
 55      ELSE                     && ESCAPE is OFF
 56         savesc = 'OFF'       && ESCAPE was OFF, save the setting
 57      ENDIF
 58
 59      *HIDE WINDOW Command
 60
 61      *SET HELP TO HMHELP.DBF      && Change help database
 62      *SET HELP TO  && TO DEFAULT HELP
 63      SET SYSMENU TO DEFAULT       && DEFAULT SYSTEM MENU
 64
 65      && End of init_app()
 66
 67      ************************************************************************
=> *
 68      *
 69      *    Get password and print to screen*
 70      *
 71      ************************************************************************
=> *
 72      m.pass=UPPER(hmgetpwd("")) && Get password
 73
 74      IF m.pass !="SYSHAZ"
 75         m.pass="HAZMAT"
 76      ENDIF
 77
 78      m.appbr=IIF(m.pass="SYSHAZ","HMLCCM","HMLCCM") && Set title of applic
=> ation
 79
 80      =bigchars(6,17,m.appbr,6)  && TITLE OF APPLICATION
 81
 82      dropdead = .F.
 83      PUSH MENU _msysmenu                  && HLL
 84      DO hmenu.mpr && Launch application menu
 85      READ VALID myhandler()
 86      POP MENU _msysmenu           && HLL
 87
 88      ************************************************************************
=> *
 89      *     DONE
 90      ************************************************************************
=> *
 91      CLEAR WINDOWS ALL
 92      CLOSE DATABASES
 93      CLOSE ALL
 94      RETURN
 95
 96      ************************************************************************
=> *
 97      *
 98      *    Function: MYHANDLER()
 99      *
100      *    Called by: APPLIC.PRG
101      *
110  *!*************************************************************************
=> ******
111  *!
112  *!       Function: MYHANDLER()
113  *!
114  *!      Called by: HMINIT.PRG
115  *!               : HMINIT.PRG
116  *!
117  *!*************************************************************************
=> ******
110      FUNCTION myhandler
111      IF TYPE("dropdead") = "U"
112         dropdead = .F.
113      ENDIF
114      RETURN dropdead
115
123  *!*************************************************************************
=> ******
124  *!
```

```
125  *!       Function: _QUIT()
126  *!
127  *!      Called by: HMENU.MPR
128  *!
129  *!*************************************************************************
=> ******
123       FUNCTION _quit
124       dropdead = .T.
125       CLEAR READ ALL
126       RETURN
127       *: EOF: HMINIT.ACT
```

```
*
*       07/15/92              HMLU.SPR                  11:12:55
*
*       Author's Name D.M. PEARSALL
*
*       Copyright (c) 1992 Company Name NHRC
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()
IF USED("hmat")
    SELECT hmat
    SET ORDER TO 0
ELSE
    SELECT 0
    USE (LOCFILE("hmat.dbf","DBF","Where is hmat?"));
        AGAIN ALIAS hmat ;
        ORDER 0
ENDIF
u = ""

*
*           Window definitions
*

IF NOT WEXIST("hmlu")
    DEFINE WINDOW hmlu ;
        FROM INT((SROW()-9)/2),INT((SCOL()-66)/2) ;
        TO INT((SROW()-9)/2)+8,INT((SCOL()-66)/2)+65 ;
        FLOAT ;
        NOCLOSE ;
        SHADOW ;
        DOUBLE ;
        COLOR SCHEME 1
ENDIF

*
*           HMLu Setup Code - SECTION 2
*

#REGION 1
*#SECTION 1
*SET UP PARAMETERS
m.hmname=""
m.value = ""
*
*           hmlu Screen Layout
*

#REGION 1
IF WVISIBLE("hmlu")
    ACTIVATE WINDOW hmlu SAME
ELSE
    ACTIVATE WINDOW hmlu NOSHOW
ENDIF
@ 2,4 SAY "MATERIAL:"
@ 2,16 GET m.hmname ;
    SIZE 1,43 ;
    DEFAULT " " ;
    PICTURE "@!" ;
    VALID _q7d0o1ei8()
@ 3,17 SAY "(Enter NIIN, Brand Name or '?' for Help)"
@ 6,25 GET m.okcan ;
    PICTURE "@*HT \<Ok;\<Cancel" ;
    SIZE 1,8,1 ;
    DEFAULT 2 ;
    VALID _q7d0o1eub()

IF NOT WVISIBLE("hmlu")
    ACTIVATE WINDOW hmlu
ENDIF

READ CYCLE MODAL

RELEASE WINDOW hmlu
SELECT (m.currarea)

#REGION 0
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF

*
*           hmlu Cleanup Code
*

#REGION 1
*USE
RETURN m.value

*************************
PROCEDURE get_hmname
*************************
PARAMETER m.name
RELEASE MEMO LIKE hmn
```

```
133     DIMENSION hmn[1]
134
135     IF LEN(m.name) = 9 AND TYPE(m.name) = "N"
136        SELECT DISTINCT ALLTRIM(hmat.hmatname) + IIF(!EMPTY(hmat.mfg), " [
=> " + ALLTRIM(hmat.mfg) + "]", ""),;
137               hmat.hmatname, hmat.hmatid;
138               FROM hmat;
139               WHERE UPPER(hmat.niin) = UPPER(m.name);
140               INTO ARRAY hmn
141     ELSE
142        SELECT DISTINCT ALLTRIM(hmat.hmatname) + IIF(!EMPTY(hmat.mfg), " [
=> " + ALLTRIM(hmat.mfg) + "]", ""),;
143               hmat.hmatname, hmat.hmatid;
144               FROM hmat;
145               WHERE UPPER(hmat.hmatname) = UPPER(m.name);
146               INTO ARRAY hmn
147     ENDIF
148
149     m.resul=""
150     IF NOT EMPTY(hmn[1])
151        m.ans=chooser(@hmn,"Select a Material")
152        m.temp=ASUBSCRIPT(hmn,ASCAN(hmn,m.ans),1)
153        m.hmid = hmn[m.temp,3]
154        m.resul = ALLTRIM(hmn[m.temp,2]) + u + ALLTRIM(STR(m.hmid))
155     ELSE
156        =errmsg(m.name + " was not found",1)
157     ENDIF
158
159     *m.hmname=m.ans
160     RETURN m.resul
161
162
163     *
164     *
165     *        _Q7D001E18          m.hmname VALID
166     *
167     *        Function Origin:
168     *
169     *        From Screen:        hmlu,     Record Number:     2
170     *        Variable:           m.hmname
171     *        Called By:          VALID Clause
172     *        Object Type:        Field
173     *        Snippet Number:     1
174     *
175     *
176     *
177     FUNCTION _q7d0o1ei8     &&  m.hmname VALID
178     m.value =""
179     IF NOT EMPTY(m.hmname)
180        m.hmname=ALLTRIM(m.hmname)
181        m.hmname=IIF(m.hmname="?","",m.hmname)
182        m.value=get_hmname(m.hmname)
183        m.hmname = dp(m.value,u,1)
184     ENDIF
185
186     *
187     *
188     *        _Q7D001EU8          m.okcan VALID
189     *
190     *        Function Origin:
191     *
192     *        From Screen:        hmlu,     Record Number:     4
193     *        Variable:           m.okcan
194     *        Called By:          VALID Clause
195     *        Object Type:        Push Button
196     *        Snippet Number:     2
197     *
198     *
199     *
200     FUNCTION _q7d0o1eub     &&  m.okcan VALID
201     #REGION 1
202     DO CASE
203     CASE m.okcan=1
204        RETURN m.value
205
206     CASE m.okcan=2
207        m.value=""
208     ENDCASE
209     *: EOF: HMLU.ACT
```

```
*:*************************************************************************
****** 
*:
*: Procedure file: C:\HAZMAT\GHM\WORK\HMMSDS.PRG
*:
*:         System: Hazardous Material Life-Cycle Cost Model
*:         Author: Naval Health Research Center
*:      Copyright (c) 1993, Naval Health Research Center
*:  Last modified: 09/10/93      8:12
*:
*:         Set by: HMENU.MPR
*:
*:          Calls: OPENFILE()         (function  in HMSC.PRG)
*:               : DP.PRG
*:               : HMLU.PRG
*:               : HMVIEW()           (function  in HMREF.PRG)
*:
*:      Documented 12/01/93 at 11:32              FoxDoc  version 2.10f
*:*************************************************************************
****** 
    * hmmsds.prg gp 8/9/93
    * show msds for selected product

    IF SET("TALK") = "ON"
       SET TALK OFF
       m.talkstat = "ON"
    ELSE
       m.talkstat = "OFF"
    ENDIF
    m.oldproc = SET("PROC")
    m.oldescape = SET("ESCAPE")
    SET ESCAPE OFF
    SET PROC TO hmsc.prg
    mfind =openfile("hmat")
    IF !mfind
<-------RETURN
    ENDIF

    m.quit = .F.

    ****************************
    * SELECT HAZARDOUS MATERIAL
    ****************************
    DO WHILE !m.quit
       * now return hmname ^ hmid 9/7/93 gp
       m.hmname=dp(hmlu(),"^",1)
       IF !EMPTY(m.hmname)
          = hmview(m.hmname)
          m.quit = .F.
       ELSE
          m.quit = .T.
       ENDIF
    ENDDO

    SET ESCAPE &oldescape
    SET TALK &talkstat
    IF !EMPTY(m.oldproc)
       SET PROC TO &oldproc
    ELSE
       SET PROC TO
    ENDIF

    RETURN

    *-------------------------------------------------------
    *
    * Display the text to screen
    *
    *-------------------------------------------------------
*!*************************************************************************
****** 
*!
*!       Function: HMVIEW()
*!
*!      Called by: HMREF.PRG
*!               : HMMSDS.PRG
*!
*!          Calls: NEWLINE()          (functi   in HMAT.SPR)
*!               : MEMOWIN.PRG
*!
*!*************************************************************************
****** 
    FUNCTION hmview
    PARAMETER name

    IF PARAMETER() = 0
<-------RETURN
    ENDIF

    SELECT hmat
    SET ORDER TO TAG hmatname
    GO TOP
    SEEK name
    IF FOUND()
       SCATTER MEMVAR MEMO
       m.alltext = m.msds
       m.title = (m.niin) + " " + (UPPER(ALLTRIM(m.hmatname))) + " ";
          + (UPPER(ALLTRIM(m.mfg)))
       * m.title = UPPER(alltrim(name))
       = memowin(m.alltext, m.title)
    ENDIF
    RETURN
    *: EOF: HMMSDS.ACT
```

```
  1  *:*********************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\HMREF.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 09/10/93      8:12
  9  *:
 10  *:  Procs & Fncts: HMVIEW()
 11  *:
 12  *:          Calls: OPENFILE()         (function  in HMSC.PRG)
 13  *:               : HMLU.PRG
 14  *:               : HMVIEW()           (function  in HMREF.PRG)
 15  *:
 16  *:      Documented 12/01/93 at 11:32                FoxDoc  version 2.10f
 17  *:*********************************************************************
=> ******
 18
 19     IF SET("TALK") = "ON"
 20        SET TALK OFF
 21        m.talkstat = "ON"
 22     ELSE
 23        m.talkstat = "OFF"
 24     ENDIF
 25     m.oldproc = SET("PROC")
 26     m.oldescape = SET("ESCAPE")
 27     SET ESCAPE OFF
 28     SET PROC TO hmsc.prg
 29     mfind =openfile("hmat")
 30     IF !mfind
 31  <------RETURN
 32     ENDIF
 33
 34     m.quit = .F.
 35
 36     ******************************
 37     * SELECT HAZARDOUS MATERIAL
 38     ******************************
 39     DO WHILE !m.quit
 40        m.hmname=hmlu()
 41        IF !EMPTY(m.hmname)
 42           = hmview(m.hmname)
 43           m.quit = .F.
 44        ELSE
 45           m.quit = .T.
 46        ENDIF
 47     ENDDO
 48
 49     SET ESCAPE &oldescape
 50     SET TALK &talkstat
 51     IF !EMPTY(m.oldproc)
 52        SET PROC TO &oldproc
 53     ELSE
 54        SET PROC TO
 55     ENDIF
 56
 57     RETURN
 58
 59     *-----------------------------------------------------
 60     *
 61     * Display the text to screen
 62     *
 63     *-----------------------------------------------------
 75  *!*********************************************************************
```



```
=> ******
 76  *!
 77  *!       Function: HMVIEW()
 78  *!
 79  *!      Called by: HMREF.PRG
 80  *!              : HMMSDS.PRG
 81  *!
 82  *!          Calls: NEWLINE()          (function  in HMAT.SPR)
 83  *!               : MEMOWIN.PRG
 84  *!
 85  *!*********************************************************************
=> ******
 75     FUNCTION hmview
 76     PARAMETER name
 77
 78     IF PARAMETER() = 0
 79  <------RETURN
 80     ENDIF
 81
 82     SELECT hmat
 83     SET ORDER TO TAG hmatname
 84     GO TOP
 85     SEEK name
 86     IF FOUND()
 87        SCATTER MEMVAR MEMO
 88        nl = CHR(10) + CHR(13)
 89        m.alltext = ""
 90        m.alltext = newline(m.alltext);
 91           + IIF(!EMPTY(m.syn_trade),"SYNONYMS/TRADE NAMES";
 92           + nl + m.syn_trade,"")
 93        m.alltext = newline(m.alltext);
 94           + IIF(!EMPTY(m.com_uses),"COMMONS USES"+nl+m.com_uses, "")
 95        m.alltext = newline(m.alltext);
 96           +IIF(!EMPTY(m.hmdescript),"DESCRIPTIONS"+nl+m.hmdescript,"")
 97        m.alltext = newline(m.alltext);
 98           + IIF(!EMPTY(m.chem_phy),"CHEMICAL/PHYSICAL PROPERTIES";
 99           + nl + m.chem_phy,"")
100        m.alltext = newline(m.alltext);
101           + IIF(!EMPTY(m.occ_exp),"OCCUPATIONAL EXPOSURES";
102           + nl + m.occ_exp,"")
103        m.alltext = newline(m.alltext);
104           + IIF(!EMPTY(m.exp_limits),"EXPOSURE LIMITS";
105           + nl + m.exp_limits,"")
106        m.alltext = newline(m.alltext);
107           + IIF(!EMPTY(m.health_haz),"HEALTH HAZARDS"+nl+m.health_haz,""
=> )
108        m.alltext = newline(m.alltext);
109           + IIF(!EMPTY(m.med_surv),"MEDICAL SURVEILLANCE";
110           + nl + m.med_surv,"")
111        m.alltext = newline(m.alltext);
112           + IIF(!EMPTY(m.spec_tests),"SPECIAL TESTS"+nl+m.spec_tests,"")
113        m.alltext = newline(m.alltext);
114           + IIF(!EMPTY(m.ppe_treat),"PERSONAL PROTECTIVE EQUIPMENT";
115           + nl + m.ppe_treat,"")
116        m.alltext = newline(m.alltext);
117           + IIF(!EMPTY(m.treatment),"TREATMENTS"+nl+m.treatment,"")
118        m.title = UPPER(ALLTRIM(m.hmatname))
119        m.title = UPPER(ALLTRIM(name))
120        = memowin(m.alltext, m.title)
121     ENDIF
122     RETURN
123
124     ******************
132  *!*********************************************************************
=> ******
133  *!
```

```
134  *!       Function: NEWLINE()
135  *!
136  *!      Called by: HMVIEW()          (function  in HMREF.PRG)
137  *!
138  *!*************************************************************************
=> ******
132       FUNCTION newline
133       ********************
134       PARAMETER TEXT
135       nl = CHR(10) + CHR(13)
136     ┌─IF EMPTY(m.text)
137     │    m.text = ""
138     ├─ELSE
139     │    m.text = m.text + nl + nl
140     └─ENDIF
141       RETURN m.text
142       *: EOF: HMREF.ACT
```

```
  1  *:*************************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\HMSC.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 09/10/93      8:12
  9  *:
 10  *:  Procs & Fncts: OPENFILE()
 11  *:               : POPUPSHOW()
 12  *:               : POPUPHIDE()
 13  *:               : DEL_HMSC()
 14  *:               : CLOSEFILE
 15  *:               : FILEFIND()
 16  *:
 17  *:         Set by: HMENU.MPR
 18  *:
 19  *:          Calls: HMSCEN.PRG
 20  *:               : DP.PRG
 21  *:               : HMSTEP.PRG
 22  *:               : DEL_HMSC()         (function  in HMSC.PRG)
 23  *:               : YESNO.PRG
 24  *:               : HMCOMP.SPR
 25  *:               : CLOSEFILE          (procedure in HMSC.PRG)
 26  *:
 27  *:      Documented 12/01/93 at 11:32                FoxDoc  version 2.10f
 28  *:*************************************************************************
=> ******
 29      PUBLIC mtalk, mesc, u
 30     IF SET("TALK") = "ON"
 31        SET TALK OFF
 32        mtalk = "ON"
 33     ENDIF
 34     mesc = SET("ESCAPE")
 35     SET ESCAPE OFF
 36     u = "'"
 37     m.quitflag = .F.
 38     mdata = hmscen()
 39     mevent = dp(mdata,u,1)
 40     m.hmscid = INT(VAL(dp(mdata,u,2)))
 41     m.hmscname = dp(mdata,u,3)
 42
 43     DO CASE
 44     CASE mevent = "C" OR EMPTY(mevent)
 45  <------RETURN
 46     CASE mevent = "N"
 47        m.adding = .T.
 48        m.quitflag = hmstep(m.adding, m.hmscid, m.hmscname)
 49     CASE mevent = "R"
 50        m.adding = .F.
 51        DO hmstep WITH m.adding, m.hmscid, m.hmscname
 52        m.quitflag = .F.
 53     CASE mevent = "D"
 54        = del_hmsc(m.hmscid)
 55        m.quitflag = .T.
 56     OTHERWISE
 57  <------RETURN
 58     ENDCASE
 59     IF !m.quitflag
 60        m.compute = yesno("Do Computation Now?", "Yes", "No")
 61        IF m.compute
 62           DO hmcomp.spr WITH m.hmscid, m.hmscname
 63        ENDIF
 64     ENDIF
 65     DO closefile
 66
 67     RETURN
 68
 69     ****************************
 77  *!*************************************************************************
=> ******
 78  *!
 79  *!      Procedure: CLOSEFILE
 80  *!
 81  *!      Called by: HMSC.PRG
 82  *!
 83  *!*************************************************************************
=> ******
 77     PROCEDURE closefile
 78     ****************************
 79     CLOSE DATABASE
 80     IF !EMPTY(mesc)
 81        SET ESCAPE &mesc
 82     ENDIF
 83     IF !EMPTY(mtalk)
 84        SET TALK &mtalk
 85     ENDIF
 86     RETURN
 87
 88     ****************************
100  *!*************************************************************************
=> ******
101  *!
102  *!       Function: DEL_HMSC()
103  *!
104  *!      Called by: HMSC.PRG
105  *!
106  *!          Calls: POPUPSHOW()        (function  in HMSC.PRG)
107  *!               : OPENFILE()         (function  in HMSC.PRG)
108  *!               : POPUPHIDE()        (function  in HMSC.PRG)
109  *!
110  *!*************************************************************************
=> ******
100     FUNCTION del_hmsc
101     ****************************
102     PARAMETER id
103     wpop = popupshow("Deleting...")
104     filesuccess = openfile("hmstep")
105     SET ORDER TO TAG hmscid
106     IF filesuccess
107        DELETE FOR hmscid = m.id
108        PACK
109     ENDIF
110     =popuphide(wpop)
111     RETURN
112     *-----------------------------------------------------
113     *
114     * Find the location of the file and file path will update into new pa
=> th
115     * if the file path has not in the foxpro path yet.
116     *
117     *-----------------------------------------------------
118     ****************************
128  *!*************************************************************************
=> ******
129  *!
130  *!       Function: FILEFIND()
131  *!
132  *!      Called by: OPENFILE()         (function  in HMSC.PRG)
133  *!
```

```
*!          Calls: DP.PRG
*!
*!*************************************************************************
=> ******
     FUNCTION filefind
     ****************************
     PARAMETER mfilename
     PRIVATE mflag
     mflag = .T.
     IF PARAMETER() = 0
<-------RETURN .F.
     ENDIF
     ON ERROR RETURN .F.
     IF !FILE(mfilename)
        mfile = dp(mfilename,".",1)
        mext = dp(mfilename,".",2)
        =LOCFILE(mfile,mext,"Where is the " + mfilename + "?")
     ENDIF
     ON ERROR
     RETURN mflag

     ****************************
*!*************************************************************************
=> ******
*!
*!       Function: OPENFILE()
*!
*!      Called by: HMSTEP.PRG
*!               : HMREF.PRG
*!               : HMMSDS.PRG
*!               : OPEN              (procedure in HMCOMP.SPR)
*!               : DEL_HMSC()        (function  in HMSC.PRG)
*!
*!          Calls: FILEFIND()       (function  in HMSC.PRG)
*!
*!*************************************************************************
=> ******
     FUNCTION openfile
     ****************************
     PARAMETER FILE
     IF filefind(FILE + ".dbf")
        VALID = .T.
        IF !USED(FILE)
           SELECT 0
           USE (FILE)
        ELSE
           SELECT (FILE)
        ENDIF
     ELSE
        VALID = .F.
     ENDIF
     RETURN VALID

     ****************************
     * display popup notice
     ****************************

*!*************************************************************************
=> ******
*!
*!       Function: POPUPSHOW()
*!
*!      Called by: _QLEOL6CB0()      (function  in HMCOMP.SPR)
*!               : DEL_HMSC()        (function  in HMSC.PRG)
*!
*!*************************************************************************
=> ******
     FUNCTION popupshow                          && rwd07
     PARAMETERS errstr

     IF NOT WEXIST("w_popnote")
        DEFINE WINDOW w_popnote ;
            FROM INT((SROW()-8)/2),INT((SCOL()-36)/2) ;
            TO INT((SROW()-8)/2)+7,INT((SCOL()-36)/2)+35 ;
            TITLE "One moment" ;
            FLOAT ;
            CLOSE ;
            SHADOW ;
            DOUBLE ;
            COLOR SCHEME 5
     ENDIF
     IF WVISIBLE("w_popnote")
        ACTIVATE WINDOW w_popnote SAME
     ELSE
        ACTIVATE WINDOW w_popnote NOSHOW
     ENDIF
     @ 1,1 SAY errstr SIZE 3,31

     IF NOT WVISIBLE("w_popnote")
        ACTIVATE WINDOW w_popnote
     ENDIF
     RETURN ""

     ****************************
*!*************************************************************************
=> ******
*!
*!       Function: POPUPHIDE()
*!
*!      Called by: _QLEOL6CB0()      (function  in HMCOMP.SPR)
*!               : DEL_HMSC()        (function  in HMSC.PRG)
*!
*!*************************************************************************
=> ******
     FUNCTION popuphide
     ****************************
     PARAMETERS W
     RELEASE WINDOW w_popnote
     RETURN W
     *: EOF: HMSC.ACT
```

```
*
*       08/27/92              HMSCEN.PRG                  08:42:06
*
*       Author's Name
*
*       Copyright (c) 1992 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()


*
*                    Window definitions
*
*


IF NOT WEXIST("w_hmscen")
   DEFINE WINDOW w_hmscen ;
        FROM INT((SROW()-7)/2),INT((SCOL()-66)/2) ;
        TO INT((SROW()-7)/2)+6,INT((SCOL()-66)/2)+65 ;
        TITLE "Cost Analysis" ;
        NOFLOAT ;
        NOCLOSE ;
        SHADOW ;
        DOUBLE ;
        COLOR SCHEME 1
ENDIF


*
*                 HMSCEN Setup Code - SECTION 2
*
*


#REGION 1
m.hmscen = ""
m.hmscenid = 0
m.driven = ""
mevent = 1
mstatus = 1
u = "'"
msel = SELECT()

IF !USED("hmscen")
   SELECT 0
   USE hmscen
ELSE
   SELECT hmscen
ENDIF


*
*                    HMSCEN Screen Layout
*
*


#REGION 1
IF WVISIBLE("w_hmscen")
   ACTIVATE WINDOW w_hmscen SAME
ELSE
   ACTIVATE WINDOW w_hmscen NOSHOW
ENDIF
@ 1,5 SAY "Scenario:"
@ 1,15 GET m.hmscen ;
    SIZE 1,42 ;
    DEFAULT " " ;
    PICTURE "@!T" ;
    VALID _q8k0ingjh()
@ 4,5 GET mevent ;
    PICTURE "@*HN \<New;\<Retrieve;\<Delete;\<Browse;\<Cancel" ;
    SIZE 1,10,1 ;
    DEFAULT 1 ;
    VALID _q8k0ingtv()

IF NOT WVISIBLE("w_hmscen")
   ACTIVATE WINDOW w_hmscen
ENDIF

READ CYCLE MODAL ;
    WHEN _q8k0inhbv()

RELEASE WINDOW w_hmscen
SELECT (m.currarea)


#REGION 0
IF m.talkstat = "ON"
   SET TALK ON
ENDIF
IF m.compstat = "ON"
   SET COMPATIBLE ON
ENDIF


*
*                    HMSCEN Cleanup Code
*
*


#REGION 1
SELECT hmscen
```

```
133     USE
134     SELECT (msel)
135     m.ans = m.driven + u + ALLTRIM(STR(m.hmscenid)) + u + m.hmscen
136     RETURN m.ans
137
138     *-------------------------------------------------------
139     * This procedure will return:
140     * Scenario name (Characters)`Scenario id (Numeric)
141     *-------------------------------------------------------
142     FUNCTION get_hmsc
143     PARAMETER name
144     RELEASE hmscn
145     DIMENSION hmscn[1]
146
147     name = ALLTRIM(name)
148     SELECT DISTINCT hmscen.hmscname, hmscen.hmscid;
149         FROM hmscen;
150         WHERE UPPER(hmscen.hmscname) IN (UPPER(name));
151         INTO ARRAY hmscn
152
153     m.ans=""
154     IF NOT EMPTY(hmscn[1])
155        m.ans=chooser(@hmscn,"Select a Material Scenario")
156        IF !EMPTY(m.ans)
157           m.index=ASCAN(hmscn,m.ans)
158           m.id = hmscn[m.index + 1]
159           m.ans = ALLTRIM(m.ans) + u + ALLTRIM(STR(m.id))
160        ENDIF
161     ENDIF
162     RETURN m.ans
163     *-------------------------------------------------------
164     *
165     * This procedure show the status of option.
166     * status 1: if new entry allow for add, browse, and cancel
167     * status 2: if entry already exist allow for retrieve, delete, browse
=>
168     *           and cancel.
169     *
170     *-------------------------------------------------------
171
172     PROCEDURE mevent
173     PARAMETER mnew
174     IF mnew
175        SHOW GET mevent,1 enabled
176        SHOW GET mevent,2 disabled
177        SHOW GET mevent,3 disabled
178     ELSE
179        SHOW GET mevent,1 disabled
180        SHOW GET mevent,2 enabled
181        SHOW GET mevent,3 enabled
182     ENDIF
183
184     RETURN
185
186     *
187     *
188     *        _Q8K0INGJH          m.hmscen VALID
189     *
190     *        Function Origin:
191     *
192     *        From Screen:        HMSCEN,     Record Number:    3
193     *        Variable:           m.hmscen
194     *        Called By:          VALID Clause
195     *        Object Type:        Field
196     *        Snippet Number:     1
197     *
198     *
199     *
200     FUNCTION _q8k0ingjh     && m.hmscen VALID
201     #REGION 1
202     m.hmsc = ""
203     IF NOT EMPTY(m.hmscen)
204        m.hmscen=ALLTRIM(m.hmscen)
205        m.hmscen=IIF(m.hmscen="?","",UPPER(m.hmscen))
206        m.hmsc=get_hmsc(m.hmscen)
207     ENDIF
208     IF EMPTY(m.hmsc)
209        DO mevent WITH .T.
210     ELSE
211        DO mevent WITH .F.
212        m.hmscen = dp(m.hmsc, u, 1)
213        m.hmscenid = VAL(dp(m.hmsc, u, 2))
214     ENDIF
215     SHOW GETS
216
217     *
218     *
219     *        _Q8K0INGTV          mevent VALID
220     *
221     *        Function Origin:
222     *
223     *        From Screen:        HMSCEN,     Record Number:    4
224     *        Variable:           mevent
225     *        Called By:          VALID Clause
226     *        Object Type:        Push Button
227     *        Snippet Number:     2
228     *
229     *
230     *
231     FUNCTION _q8k0ingtv     && mevent VALID
232     #REGION 1
233     IF EMPTY(m.hmscen)
234        IF mevent < 4
235           _CUROBJ = OBJNUM(m.hmscen)
236           =errmsg("Input required",2)
237           SHOW GETS
238           RETURN
239        ENDIF
240     ENDIF
241     DO CASE
242     CASE mevent = 1                                          && New
243        SET ORDER TO TAG hmscid
244        GO BOTTOM
245        m.hmscenid = hmscid + 1
246        m.driven = "N"
247        CLEAR READ
248     CASE mevent = 2                                          && Retri
=> eve
249        m.driven = "R"
250        CLEAR READ
251     CASE mevent = 3                                          && Delet
=> e
252        m.driven = "D"
253        SET ORDER TO TAG hmscid
254        GO TOP
255        SEEK m.hmscenid
256        DELETE
257        PACK
258        CLEAR READ
259     CASE mevent = 4                                          && Brows
=> e
260        m.hmsc = ""
```

```
261         m.hmsc = get_hmsc(m.hmsc)
262         IF !EMPTY(m.hmsc)
263            DO mevent WITH .F.
264            m.hmscen = dp(m.hmsc,u,1)
265            m.hmscenid = VAL(dp(m.hmsc,u,2))
266         ENDIF
267         SHOW GETS
268
269      CASE mevent = 5
270         m.hmscenid = 0
271         m.hmscen = ""
272         m.driven = "C"
273         CLEAR READ
274      ENDCASE
275      RETURN
276
277      *
278      *
279      *        _Q8KOINH8V            Read Level When
280      *
281      *        Function Origin:
282      *
283      *        From Screen:          HMSCEN
284      *        Called By:            READ Statement
285      *        Snippet Number:       3
286      *
287      *
288      *
289      FUNCTION _q8kOinh8v     && Read Level When
290      *
291      * When Code from screen: HMSCEN
292      *
293      #REGION 1
294      DO mevent WITH .T.
295      *: EOF: HMSCEN.ACT
```

```
*
*       08/26/93              HMSTEP.PRG                    14:36:09
*
*       Author's Name
*
*       Copyright (c) 1993 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*

PARAMETERS adding, scid, scname

*
*            HMSTEP/MS-DOS Setup Code - SECTION 1
*

#REGION 1

m.invalid = .F.
u = "~"
m.adding = IIF(PARAMETER() = 0, .T., m.adding)
m.ghmscid = IIF(PARAMETER() < 1, 0, m.scid)
m.ghmscname = IIF(PARAMETER() < 2, "", m.scname)
m.gchange = IIF(m.adding, .T., .F.)
m.change = .F.
m.quitflag = .F.


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()


*
*              MS-DOS Window definitions
*

IF NOT WEXIST("hmscen") ;
      OR UPPER(WTITLE("HMSCEN")) == "HMSCEN.PJX" ;
      OR UPPER(WTITLE("HMSCEN")) == "HMSCEN.SCX" ;
      OR UPPER(WTITLE("HMSCEN")) == "HMSCEN.MNX" ;
      OR UPPER(WTITLE("HMSCEN")) == "HMSCEN.PRG" ;
      OR UPPER(WTITLE("HMSCEN")) == "HMSCEN.FRX" ;
      OR UPPER(WTITLE("HMSCEN")) == "HMSCEN.QPR"
   DEFINE WINDOW hmscen ;
      FROM INT((SROW()-16)/2),INT((SCOL()-78)/2) ;
      TO INT((SROW()-16)/2)+15,INT((SCOL()-78)/2)+77 ;
      TITLE "HAZARDOUS MATERIAL SCENARIO" ;
      FLOAT ;
      NOCLOSE ;
      SHADOW ;
      nominimize ;
      DOUBLE ;
      COLOR SCHEME 1
ENDIF


*
*            HMSTEP/MS-DOS Setup Code - SECTION 2
*

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
m.oldescape = SET( "ESCAPE" )
m.oldtalk = SET("talk")
m.oldsafe = SET("SAFETY")
m.oldproc = SET("PROC")
SET ESCAPE OFF
SET SAFETY OFF
SET PROCEDURE TO hmsc.prg

m.action = 1
m.step = 0
STORE "" TO m.oldhmname, m.oldhmlc, m.oldhmwp
STORE "" TO m.hmname, m.hmlc, hmwp
STORE .F. TO m.validproc

filesuccess = openfile("hmstep")
IF !filesuccess
   =errmsg("File doesn's Exist !!!!", 2)
   DO CANCEL
ENDIF
IF !EMPTY(m.ghmscid)
   SELECT *;
      FROM hmstep;
      WHERE hmstep.hmscid = m.ghmscid;
      INTO TABLE hmtemp
ENDIF
SELECT hmtemp
GO TOP
IF m.adding
   SCATTER MEMVAR BLANK
   m.hmstep = 1
   m.hmscid = m.ghmscid
   m.oldrec = 0
ELSE
   SCAN
      m.step = m.step + 1
      IF hmstep != m.step
         REPLACE hmstep WITH m.step
      ENDIF
   ENDSCAN
   GO TOP
   SCATTER MEMVAR
   m.oldrec = RECNO()
```

```
ENDIF
DO initvar
SHOW GETS



*
*                    HMSTEP/MS-DOS Screen Layout
*
*
*
*

#REGION 1
IF WVISIBLE("hmscen")
   ACTIVATE WINDOW hmscen SAME
ELSE
   ACTIVATE WINDOW hmscen NOSHOW
ENDIF
@ 4,1 SAY "Phase:" ;
   SIZE 1,6, 0
@ 5,1 SAY "Process:" ;
   SIZE 1,8, 0
@ 8,5 SAY "Number of Employees:" ;
   SIZE 1,20, 0
@ 9,5 SAY "Quantity of material:" ;
   SIZE 1,21, 0
@ 2,0 TO 10,63
@ 9,33 SAY "Unit:" ;
   SIZE 1,5, 0
@ 0,2 SAY "Step #" ;
   SIZE 1,6, 0
@ 0,44 SAY "Scenario:" ;
   SIZE 1,9, 0
@ 8,33 SAY "Number of days:" ;
   SIZE 1,15, 0
@ 3,1 SAY "Material:" ;
   SIZE 1,9, 0
@ 0,9 GET m.hmstep ;
   SIZE 1,4 ;
   DEFAULT 0 ;
   DISABLE
@ 0,54 GET m.ghmscid ;
   SIZE 1,4 ;
   DEFAULT 0 ;
   PICTURE "@Z" ;
   DISABLE
@ 0,60 GET m.ghmscname ;
   SIZE 1,15 ;
   DEFAULT " " ;
   PICTURE "@!T" ;
   DISABLE
@ 2,65 GET m.action ;
   PICTURE "@*VN \<Add;\<Edit;\!\<Next;\<Previous;\?E\<xit" ;
   SIZE 1,10,1 ;
   DEFAULT 1 ;
   VALID _qip0vascn() ;
   DISABLE
@ 3,12 GET m.hmcomid ;
   SIZE 1,4 ;
   DEFAULT 0 ;
   PICTURE "@Z" ;
   DISABLE
@ 3,17 GET m.hmname ;
   SIZE 1,45 ;
   DEFAULT " " ;
   PICTURE "@!" ;
   WHEN _qip0vasvs() ;
   VALID _qip0vasya()
@ 4,12 GET m.hmlcid ;
   SIZE 1,4 ;
   DEFAULT 0 ;
   PICTURE "@Z" ;
   DISABLE
@ 4,17 GET m.hmlc ;
   SIZE 1,45 ;
   DEFAULT " " ;
   PICTURE "@!" ;
   WHEN _qip0vat7g() ;
   VALID _qip0vat9x()
@ 5,12 GET m.hmwpid ;
   SIZE 1,4 ;
   DEFAULT 0 ;
   PICTURE "@Z" ;
   DISABLE
@ 5,17 GET m.hmwp ;
   SIZE 2,45 ;
   DEFAULT " " ;
   PICTURE "@!" ;
   WHEN _qip0vatiu() ;
   VALID _qip0vatlc()
@ 8,27 GET m.pernum ;
   SIZE 1,3 ;
   DEFAULT 0
@ 8,49 GET m.durnum ;
   SIZE 1,9 ;
   DEFAULT 0
@ 9,27 GET m.qtynum ;
   SIZE 1,3 ;
   DEFAULT 0
@ 9,39 GET m.unit ;
   SIZE 1,19 ;
   DEFAULT " "
@ 12,26 GET m.save ;
   PICTURE "@*HN \<Save;\<Cancel" ;
   SIZE 1,8,1 ;
   DEFAULT 1 ;
   VALID _qip0vatwz()

IF NOT WVISIBLE("hmscen")
   ACTIVATE WINDOW hmscen
ENDIF

READ CYCLE ;
   WHEN _qip0vaud2()

RELEASE WINDOW hmscen
SELECT (m.currarea)


#REGION 0
IF m.talkstat = "ON"
   SET TALK ON
ENDIF
IF m.compstat = "ON"
   SET COMPATIBLE ON
ENDIF


*
*                    HMSTEP/MS-DOS Cleanup Code
*
```

```
*
*
*

#REGION 1
POP KEY ALL
SET ESCAPE &oldescape
SET SAFETY &oldsafe
RETURN m.quitflag
********* End of Main Body - Entry Cleanup


*
*
*       HMSTEP/MS-DOS Supporting Procedures and Functions
*
*
*

#REGION 1
PROCEDURE escpressed
RETURN

*******************
PROCEDURE initvar
*******************
IF m.hmcomid>0
   m.hmname=get_hmatn(m.hmcomid)
   m.oldhmname=m.hmname
ELSE
   m.hmname=SPACE(46)
   m.oldhmname=SPACE(46)
ENDIF

IF m.hmlcid>0
   m.hmlc=get_hmlcn(m.hmlcid)
   m.oldhmlc=m.hmlc
ELSE
   m.hmlc=SPACE(46)
   m.oldhmlc=SPACE(46)
ENDIF

IF m.hmwpid>0
   m.hmwp=get_hmwpn(m.hmwpid)
   m.oldhmwp=m.hmwp
ELSE
   m.hmwp=SPACE(80)
   m.oldhmwp=SPACE(80)
ENDIF
RETURN
*******************
PROCEDURE addoption
*******************
SHOW GET m.action disabled
SHOW GET m.save enabled
SHOW GET m.hmname enabled
SHOW GET m.hmlc enabled
SHOW GET m.hmwp enabled
SHOW GET m.pernum enabled
SHOW GET m.durnum enabled
SHOW GET m.qtynum enabled
SHOW GET m.unit enabled
RETURN

*******************
PROCEDURE CHANGE
*******************
m.oldexact = SET( "EXACT" )
SET EXACT ON
m.change =(hmtemp.hmcomid <> m.hmcomid;
    OR hmtemp.hmlcid <> m.hmlcid;
    OR hmtemp.hmwpid <> m.hmwpid;
    OR hmtemp.durnum <> m.durnum;
    OR hmtemp.qtynum <> m.qtynum;
    OR hmtemp.pernum <> m.pernum;
    OR UPPER(ALLTRIM(hmtemp.unit)) <> UPPER(ALLTRIM(m.unit)))

SET EXACT &oldexact
RETURN m.change

*************************
FUNCTION get_hmat
*************************
PARAMETER name
RELEASE hmn
DIMENSION hmn[1]
hmn[1]=""
m.oldfile=SELECT()

SELECT hmcom.common, hmcom.hmcomid;
    FROM hmcom;
    WHERE hmcom.common IN (ALLTRIM(m.name));
    INTO ARRAY hmn

m.ans=""
IF NOT EMPTY(hmn[1])
   m.ans=chooser(@hmn,"Select a Material")
   IF !EMPTY(m.ans)
      m.idex = ASCAN(hmn,m.ans)
      m.id = hmn[m.idex + 1]
      m.ans = m.ans + u + STR(m.id)
   ENDIF
ELSE
   =errmsg(m.name + " was not found",1)
   m.ans=""
ENDIF

SELECT (m.oldfile)
RETURN m.ans

************************
FUNCTION get_hmatn
************************
PARAMETER m.id

SELECT hmcom.common;
    FROM hmcom;
    WHERE hmcom.hmcomid = m.id;
    INTO ARRAY X

m.name=X[1]
RETURN m.name

************************
*GET LIFE CYCLE PHASE
FUNCTION get_hmlc
************************
PARAMETER m.hmlc

RELEASE hmn
DIMENSION hmn[1]
hmn[1]=""
```

```
397     m.oldfile=SELECT()
398
399     SELECT hmlc.hmlc, hmlc.hmlcid;
400         FROM hmlc;
401         WHERE hmlc.hmlc IN (ALLTRIM(m.hmlc));
402         ORDER BY hmlcid;
403         INTO ARRAY hmn
404
405     m.ans=""
406     IF NOT EMPTY(hmn[1])
407        m.ans=chooser(@hmn,"Select a Life Cycle Phase")
408        IF !EMPTY(m.ans)
409           m.idex = ASCAN(hmn,m.ans)
410           m.id = hmn[m.idex + 1]
411           m.ans = m.ans + u + STR(m.id)
412        ENDIF
413     ELSE
414        =errmsg(m.hmlc + " was not found",1)
415     ENDIF
416
417     SELECT (m.oldfile)
418     RETURN m.ans
419
420     ************************
421     FUNCTION get_hmlcn
422     ************************
423     PARAMETER m.id
424
425     SELECT hmlc.hmlc;
426         FROM hmlc;
427         WHERE hmlc.hmlcid = m.id;
428         INTO ARRAY X
429
430     m.name=X[1]
431     RETURN m.name
432
433     ************************
434     *Working Processes
435     FUNCTION get_hmwp
436     ************************
437     PARAMETER m.match
438     RELEASE hmn
439     DIMENSION hmn[1]
440     hmn[1]=""
441     m.oldfile=SELECT()
442     m.match = UPPER(ALLTRIM(m.match))
443     SELECT hmwp.hmwp, hmwp.hmwpid;
444         FROM hmwp;
445         WHERE UPPER(hmwp.hmwp) IN (ALLTRIM(m.match));
446         INTO ARRAY hmn
447
448     m.ans=""
449     IF NOT EMPTY(hmn[1])
450        m.ans=chooser(@hmn,"Select a Process")
451        IF !EMPTY(m.ans)
452           m.idex = ASCAN(hmn,m.ans)
453           m.id = hmn[m.idex + 1]
454           m.ans = m.ans + u + STR(m.id)
455        ENDIF
456     ELSE
457        =errmsg(m.hmwp + " was not found",1)
458     ENDIF
459
460     SELECT (m.oldfile)
461
462     RETURN m.ans
463
464     ************************
465     FUNCTION get_hmwpn
466     ************************
467     PARAMETER m.id
468
469     SELECT hmwp.hmwp;
470         FROM hmwp;
471         WHERE hmwp.hmwpid = m.id;
472         INTO ARRAY X
473
474     m.name=X[1]
475     RETURN m.name
476
477     ************************
478     PROCEDURE get_hmunit
479     ************************
480
481     PRIVATE m.unit
482     m.unit = ""
483     IF !EMPTY(m.hmcomid) AND !EMPTY(m.hmlcid) AND !EMPTY(m.hmwpid)
484        DIMENSION X[1]
485        X[1]=""
486        SELECT hmtab.tabid, hmtab.hmunit;
487            FROM hmtab;
488            WHERE hmtab.hmcomid = m.hmcomid;
489            AND hmtab.hmlcid = m.hmlcid;
490            AND hmtab.hmwpid = m.hmwpid;
491            INTO ARRAY X
492        IF !EMPTY(X[1])
493           FOR i = 1 TO ALEN(X) STEP 2
494              m.unit = X[i+ 1]
495              IF !EMPTY(m.unit)
496                 EXIT
497              ENDIF
498           ENDFOR
499           m.validproc = .T.
500           SHOW GETS
501        ELSE
502           =errmsg("This Material Processing isn't defined in the table")
503           m.validproc = .F.
504        ENDIF
505     ENDIF
506     RETURN m.unit
507
508     ************************
509     PROCEDURE scsave
510     ************************
511     msel = SELECT()
512     SELECT hmstep
513     IF gchange
514        SET ORDER TO TAG hmscid
515        DELETE FOR hmscid = m.ghmscid
516        PACK
517        APPEND FROM hmtemp
518        USE hmscen
519        SET ORDER TO TAG hmscid
520        IF !SEEK(m.ghmscid)
521           APPEND BLANK
522           REPLACE hmscid WITH m.ghmscid
523           REPLACE hmscname WITH m.ghmscname
524        ENDIF
525     ENDIF
526     ************************
527     FUNCTION datacheck
528     ************************
```

```
529     DO CASE
530     CASE EMPTY(m.hmcomid)
531         = errmsg("Missing Material",1)
532         _CUROBJ = OBJNUM(hmname)
533         check = .F.
534     CASE EMPTY(m.hmlcid)
535         = errmsg("Missing Life Cycle Phase",1)
536         _CUROBJ = OBJNUM(hmlc)
537         check = .F.
538     CASE EMPTY(m.hmwpid)
539         = errmsg("Missing Working Process",1)
540         _CUROBJ = OBJNUM(hmwp)
541         check = .F.
542     CASE EMPTY(m.pernum) AND EMPTY(m.durnum) AND EMPTY(m.qtynum)
543         = errmsg("Enter either number of People, Duration, or Quantity",1)
544         _CUROBJ = OBJNUM(pernum)
545         check = .F.
546     OTHERWISE
547         =get_hmunit()
548       IF !m.validproc
549           _CUROBJ = OBJNUM(hmname)
550           check = .F.
551       ELSE
552           check = .T.
553       ENDIF
554     ENDCASE
555     RETURN check
556
557     *
558     *
559     *        _QIPOVASCN            m.Action VALID
560     *
561     *        Function Origin:
562     *
563     *        From Platform:        MS-DOS
564     *        From Screen:          HMSTEP,     Record Number:   15
565     *        Variable:             m.Action
566     *        Called By:            VALID Clause
567     *        Snippet Number:       1
568     *
569     *
570     *
571     FUNCTION _qip0vascn     &&  m.Action VALID
572     #REGION 1
573     m.prevhmcomid = m.hmcomid
574     m.prevhmlcid = m.hmlcid
575     m.oldrec = IIF(RECCOUNT() = 0, 0, RECNO())
576     DO CASE
577     CASE m.action = 1
578         msamephase = yesno("Same Phase?","Yes","No")
579         SCATTER MEMVAR BLANK
580         m.hmstep = RECCOUNT() + 1
581       IF msamephase
582           m.hmcomid = m.prevhmcomid
583           m.hmlcid = m.prevhmlcid
584           _CUROBJ = OBJNUM(m.hmwp)
585       ELSE
586           _CUROBJ = OBJNUM(m.hmname)
587       ENDIF
588         DO initvar
589         SHOW GETS
590         DO addoption
591         m.adding = .T.
592
593     CASE m.action = 2
594         validproc = .T.
```



```
595         DO addoption
596
597     CASE m.action = 3
598         SKIP
599       IF EOF()
600           ?? CHR( 7 )
601           WAIT "Last step" WINDOW NOWAIT
602           GO BOTTOM
603       ENDIF
604
605     CASE m.action = 4
606         SKIP -1
607       IF BOF()
608           ?? CHR( 7 )
609           WAIT "First step" WINDOW NOWAIT
610           GO TOP
611       ENDIF
612
613     CASE m.action = 5
614       IF m.gchange
615           m.scsave = yesno("Save Scenario? ", "YES", "NO")
616         IF m.scsave
617             DO scsave
618         ENDIF
619       ENDIF
620       CLEAR READ
621     ENDCASE
622
623
624     IF m.action >2 AND m.action < 5
625         m.oldrec = RECNO()
626         SCATTER MEMVAR
627         DO initvar
628         SHOW GETS disabled
629         SHOW GET action enabled
630
631     ENDIF
632
633
634     *
635     *
636     *        _QIPOVASVS            m.hmname WHEN
637     *
638     *        Function Origin:
639     *
640     *        From Platform:        MS-DOS
641     *        From Screen:          HMSTEP,     Record Number:   17
642     *        Variable:             m.hmname
643     *        Called By:            WHEN Clause
644     *        Snippet Number:       2
645     *
646     *
647     *
648     FUNCTION _qip0vasvs     &&  m.hmname WHEN
649     #REGION 1
650     m.oldhmname = m.hmname
651
652     *
653     *
654     *        _QIPOVASYA            m.hmname VALID
655     *
656     *        Function Origin:
657     *
658     *        From Platform:        MS-DOS
659     *        From Screen:          HMSTEP,     Record Number:   17
660     *        Variable:             m.hmname
```

```
661     *           Called By:          VALID Clause
662     *           Snippet Number:     3
663     *
664     *
665     *
666     FUNCTION _qip0vasya     &&  m.hmname VALID
667     #REGION 1
668     IF m.oldhmname <> m.hmname
669       IF NOT EMPTY(m.hmname)
670         m.hmname=ALLTRIM(m.hmname)
671         m.hmname=IIF(m.hmname="?","",UPPER(m.hmname))
672         m.hmdata=ALLTRIM(get_hmat(m.hmname))
673         IF NOT EMPTY(m.hmdata)
674           m.hmname = dp(m.hmdata,u,1)
675           m.hmcomid = VAL(dp(m.hmdata,u,2))
676         ELSE
677           m.hmname = SPACE(46)
678         ENDIF
679         SHOW GETS
680         m.t1 = get_hmunit()
681         IF !EMPTY(m.t1)
682           m.unit = m.t1
683         ENDIF
684       ENDIF
685       SHOW GETS
686     ENDIF
687
688
689     *
690     *           _QIPOVAT7G          m.hmlc WHEN
691     *
692     *           Function Origin:
693     *
694     *           From Platform:      MS-DOS
695     *           From Screen:        HMSTEP,     Record Number:   19
696     *           Variable:           m.hmlc
697     *           Called By:          WHEN Clause
698     *           Snippet Number:     4
699     *
700     *
701     *
702     FUNCTION _qip0vat7g     &&  m.hmlc WHEN
703     #REGION 1
704     m.oldhmlc=m.hmlc
705
706     *
707     *           _QIPOVAT9X          m.hmlc VALID
708     *
709     *           Function Origin:
710     *
711     *           From Platform:      MS-DOS
712     *           From Screen:        HMSTEP,     Record Number:   19
713     *           Variable:           m.hmlc
714     *           Called By:          VALID Clause
715     *           Snippet Number:     5
716     *
717     *
718     *
719     *
720     FUNCTION _qip0vat9x     &&  m.hmlc VALID
721     #REGION 1
722     IF m.oldhmlc <> m.hmlc
723       IF NOT EMPTY(m.hmlc)
724         m.hmlc=ALLTRIM(m.hmlc)
725         m.hmlc=IIF(m.hmlc="?","",UPPER(m.hmlc))
726         m.hmdata=get_hmlc(m.hmlc)
727         IF NOT EMPTY(m.hmdata)
728           m.hmlc = dp(m.hmdata,u,1)
729           m.hmlcid = VAL(dp(m.hmdata,u,2))
730         ELSE
731           m.hmlc = SPACE(46)
732         ENDIF
733         SHOW GETS
734         m.t1 = get_hmunit()
735         IF !EMPTY(m.t1)
736           m.unit = m.t1
737         ENDIF
738       ENDIF
739       SHOW GETS
740     ENDIF
741
742     *
743     *           _QIPOVATIU          m.hmmp WHEN
744     *
745     *           Function Origin:
746     *
747     *           From Platform:      MS-DOS
748     *           From Screen:        HMSTEP,     Record Number:   21
749     *           Variable:           m.hmmp
750     *           Called By:          WHEN Clause
751     *           Snippet Number:     6
752     *
753     *
754     *
755     FUNCTION _qip0vatiu     &&  m.hmmp WHEN
756     #REGION 1
757     m.oldhmmp=m.hmmp
758
759     *
760     *           _QIPOVATLC          m.hmmp VALID
761     *
762     *           Function Origin:
763     *
764     *           From Platform:      MS-DOS
765     *           From Screen:        HMSTEP,     Record Number:   21
766     *           Variable:           m.hmmp
767     *           Called By:          VALID Clause
768     *           Snippet Number:     7
769     *
770     *
771     *
772     FUNCTION _qip0vatlc     &&  m.hmmp VALID
773     #REGION 1
774     IF m.oldhmmp <> m.hmmp
775       IF NOT EMPTY(m.hmmp)
776         m.hmmp=ALLTRIM(m.hmmp)
777         m.hmmp=IIF(m.hmmp="?","",UPPER(m.hmmp))
778         m.hmdata=ALLTRIM(get_hmmp(m.hmmp))
779         IF NOT EMPTY(m.hmdata)
780           m.hmmp = dp(m.hmdata,u,1)
781           m.hmmpid = VAL(dp(m.hmdata,u,2))
782         ELSE
783           m.hmmp = SPACE(80)
784         ENDIF
785         SHOW GETS
786         m.t1 = get_hmunit()
787         IF !EMPTY(m.t1)
788           m.unit = m.t1
789         ENDIF
790
791
792
```

```
793         ENDIF
794         SHOW GETS
795      ENDIF
796
797      *
798      *
799      *       _QIP0VATWZ            m.Save VALID
800      *
801      *       Function Origin:
802      *
803      *       From Platform:        MS-DOS
804      *       From Screen:          HMSTEP,     Record Number:   26
805      *       Variable:             m.Save
806      *       Called By:            VALID Clause
807      *       Snippet Number:       8
808      *
809      *
810      *
811      *
812      FUNCTION _qip0vatwz     &&  m.Save VALID
813      #REGION 1
814      DO CASE
815      CASE m.save = 1 && Selected Save Button
816         m.check = datacheck()
817         IF m.check
818            m.hmscid = m.ghmscid
819            IF m.adding && Adding a new record
820               APPEND BLANK
821               GATHER MEMVAR
822               m.gchange = .T.
823            ELSE
824               DO CHANGE
825               IF m.change && Changing an old record
826                  GATHER MEMVAR
827                  m.gchange = .T.
828               ENDIF
829            ENDIF
830            m.oldrec = RECNO()
831         ELSE
832            RETURN
833         ENDIF
834      CASE m.save = 2
835         IF RECCOUNT() = 0
836            QUIT = .F.
837         ELSE
838            IF EMPTY(m.hmname) AND EMPTY(m.hmlc) AND EMPTY(hmwp)
839               QUIT = yesno("Do you want to","Quit","Continue")
840            ENDIF
841            IF QUIT
842               m.quitflag = .T.
843               CLEAR READ
844            ELSE
845               SCATTER MEMVAR BLANK
846               DO initvar
847               SHOW GETS
848               DO addoption
849               m.adding = .T.
850               m.change = .F.
851               m.quitflag = .F.
852               RETURN
853            ENDIF
854         ENDIF
855      ENDCASE
856      IF !EMPTY(m.oldrec)
857         GO m.oldrec
858         SCATTER MEMVAR
859      ENDIF
860      DO initvar
861      SHOW GETS disabled
862      SHOW GET action enabled
863
864      m.adding=.F.
865      m.change=.F.
866
867
868      *
869      *
870      *       _QIP0VAUD2            Read Level When
871      *
872      *       Function Origin:
873      *
874      *
875      *       From Platform:        MS-DOS
876      *       From Screen:          HMSTEP
877      *       Called By:            READ Statement
878      *       Snippet Number:       9
879      *
880      *
881      *
882      FUNCTION _qip0vaud2     && Read Level When
883      *
884      * When Code from screen: HMSTEP
885      *
886      #REGION 1
887      IF !m.adding
888         m.action = 1
889         SHOW GETS DISABLE
890         SHOW GET action enabled
891      ENDIF
892      *: EOF: HMSTEP.ACT
```

```
  1  *:**************************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\HMZTAB.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 09/10/93      8:12
  9  *:
 10  *:         Set by: HMENU.MPR
 11  *:
 12  *:           Uses: TLV.DBF
 13  *:
 14  *:      CDX files: TLV.CDX
 15  *:
 16  *:      Documented 12/01/93 at 11:32               FoxDoc  version 2.10f
 17  *:**************************************************************************
=> ******
 18       * hmztab.prg gp 8/9/93
 19       * OSHA Z table look up from tlv table from Smallett
 20       USE tlv
 21       BROWSE FIELDS substance, cas_no:H='CAS Number',pel_twa:H="TWA (mg/m3)
=> ",;
 22           pel_stel:H="STEL (mg/m3)"  NOMODIFY NORMAL
 23       USE
 24       *: EOF: HMZTAB.ACT
```

```
*
*       08/28/92            MEMOEDIT.PRG                10:02:45
*
*
*       Noa L. Ly
*
*       Copyright (c) 1992 Naval Health Research Center
*       P..Box 85122
*       San Diego, CA  92186-5122
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*


*
*               MEMOEDIT Setup Code - SECTION 1
*
*

#REGION 1
PARAMETER TEXT, TITLE, editallow
PRIVATE ALL
DO CASE
CASE PARAMETER() = 0
    m.text = ""
    m.title = ""
    m.editallow = .T.
CASE PARAMETER() = 1
    m.title = ""
    m.editallow = .T.
CASE PARAMETER() = 2
    m.editallow = .T.
ENDCASE

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()


*
*               Window definitions
*
*

IF NOT WEXIST("memoedit")
    DEFINE WINDOW memoedit ;
        FROM INT((SROW()-20)/2),INT((SCOL()-77)/2) ;
        TO INT((SROW()-20)/2)+19,INT((SCOL()-77)/2)+76 ;
        NOFLOAT ;
        NOCLOSE ;
        SHADOW ;
        DOUBLE ;
        COLOR SCHEME 1
ENDIF


*
*               MEMOEDIT Setup Code - SECTION 2
*
*

#REGION 1
m.oldtext = m.text
IF editallow
    m.title = m.title + " [Edit]"
ELSE
    m.title = m.title + " [No Edit]"
ENDIF

*
*               MEMOEDIT Screen Layout
*
*

#REGION 1
IF WVISIBLE("memoedit")
    ACTIVATE WINDOW memoedit SAME
ELSE
    ACTIVATE WINDOW memoedit NOSHOW
ENDIF
@ 0,5 GET m.title ;
    SIZE 1,65 ;
    DEFAULT " " ;
    PICTURE "@I" ;
    DISABLE
IF editallow
    @ 2,1 EDIT m.text ;
        SIZE 14,74,0 ;
        DEFAULT " " ;
        TAB ;
        SCROLL
ELSE
    @ 2,1 EDIT m.text ;
        SIZE 14,74,0 ;
        DEFAULT " " ;
        TAB ;
        NOMODIFY;
        SCROLL
ENDIF
@ 17,28 GET m.save ;
    PICTURE "@*HT \<Save;\<Cancel" ;
    SIZE 1,9,1 ;
    DEFAULT 1 ;
    VALID _q8l0lj6f0()

IF NOT WVISIBLE("memoedit")
    ACTIVATE WINDOW memoedit
ENDIF

READ CYCLE MODAL
```

```
133
134
135     RELEASE WINDOW memoedit
136     SELECT (m.currarea)
137
138
139     #REGION 0
140     IF m.talkstat = "ON"
141        SET TALK ON
142     ENDIF
143     IF m.compstat = "ON"
144        SET COMPATIBLE ON
145     ENDIF
146
147
148     *
149     *
150     *                    MEMOEDIT Cleanup Code
151     *
152     *
153     *
154
155     #REGION 1
156     RETURN m.text
157
158     *
159     *
160     *        _Q8L0LJ6F0            m.save VALID
161     *
162     *        Function Origin:
163     *
164     *        From Screen:          MEMOEDIT,     Record Number:    4
165     *        Variable:             m.save
166     *        Called By:            VALID Clause
167     *        Object Type:          Push Button
168     *        Snippet Number:       1
169     *
170     *
171     *
172     FUNCTION _q8l0lj6f0     &&  m.save VALID
173     #REGION 1
174     IF m.save = 2
175        m.text = m.oldtext
176     ENDIF
177
178     *: EOF: MEMOEDIT.ACT
```

```
  1  *:*************************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\MEMOWIN.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 08/17/93      8:54
  9  *:
 10  *:  Procs & Fncts: _QIGOIXOXD()
 11  *:               : _QIGOIXPJO()
 12  *:
 13  *:         Set by: HMVIEW()           (function  in HMREF.PRG)
 14  *:
 15  *:          Calls: _QIGOIXOXD()       (function  in MEMOWIN.PRG)
 16  *:               : _QIGOIXPJO()       (function  in MEMOWIN.PRG)
 17  *:
 18  *:      Documented 12/01/93 at 11:32               FoxDoc  version 2.10f
 19  *:*************************************************************************
=> ******
 20      *       ************************************************************
 21      *       *
 22      *       * 08/17/93             MEMOWIN.SPR                08:50:04
 23      *       *
 24      *       ************************************************************
 25      *       *
 26      *       * G. Pang
 27      *       *
 28      *       * Copyright (c) 1993 Company Name
 29      *       * Address
 30      *       * City,     Zip
 31      *       *
 32      *       * Description:
 33      *       * This program was automatically generated by GENSCRN.
 34      *       *
 35      *       ************************************************************
 36
 37      PARAMETERS TEXT, TITLE
 38     ┌DO CASE
 39     ├CASE _dos
 40     │
 41     │
 42     │    *       ********************************************************
 43     │    *       *
 44     │    *       *          MEMOWIN/MS-DOS Setup Code - SECTION 1
 45     │    *       *
 46     │    *       ********************************************************
 47     │    *
 48     │
 49     │    #REGION 1
 50     │    * PRIVATE all
 51     │   ┌DO CASE
 52     │   │   CLEAR
 53     │   ├CASE PARAMETER() = 0
 54     │   │   m.title = ""
 55     │   │   m.text = ""
 56     │   ├CASE PARAMETER() = 1
 57     │   │   m.title = ""
 58     │   └ENDCASE
 59     │
 60     │
 61     │
 62     │    #REGION 0
 63     │    REGIONAL m.currarea, m.talkstat, m.compstat
 64     │
 65     │   ┌IF SET("TALK") = "ON"
 66     │   │   SET TALK OFF
 67     │   │   m.talkstat = "ON"
 68     │   ├ELSE
 69     │   │   m.talkstat = "OFF"
 70     │   └ENDIF
 71     │    m.compstat = SET("COMPATIBLE")
 72     │    SET COMPATIBLE FOXPLUS
 73     │
 74     │    m.currarea = SELECT()
 75     │
 76     │
 77     │    *       ********************************************************
 78     │    *       *
 79     │    *       *                MS-DOS Window definitions
 80     │    *       *
 81     │    *       ********************************************************
 82     │    *
 83     │
 84     │   ┌IF NOT WEXIST("memowin") ;
 85     │   │       OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.PJX" ;
 86     │   │       OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.SCX" ;
 87     │   │       OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.MNX" ;
 88     │   │       OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.PRG" ;
 89     │   │       OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.FRX" ;
 90     │   │       OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.QPR"
 91     │   │    DEFINE WINDOW memowin ;
 92     │   │       FROM INT((SROW()-26)/2),INT((SCOL()-80)/2) ;
 93     │   │       TO INT((SROW()-26)/2)+25,INT((SCOL()-80)/2)+79 ;
 94     │   │       NOFLOAT ;
 95     │   │       NOCLOSE ;
 96     │   │       SHADOW ;
 97     │   │       nominimize ;
 98     │   │       NONE ;
 99     │   │       COLOR SCHEME 1
100     │   └ENDIF
101     │
102     │
103     │    *       ********************************************************
104     │    *       *
105     │    *       *              MEMOWIN/MS-DOS Screen Layout
106     │    *       *
107     │    *       ********************************************************
108     │    *
109     │
110     │    #REGION 1
111     │   ┌IF WVISIBLE("memowin")
112     │   │   ACTIVATE WINDOW memowin SAME
113     │   ├ELSE
114     │   │   ACTIVATE WINDOW memowin NOSHOW
115     │   └ENDIF
116     │    @ 1,2 GET m.title ;
117     │        SIZE 1,74 ;
118     │        DEFAULT " " ;
119     │        PICTURE "@I" ;
120     │        DISABLE
121     │    @ 2,2 EDIT m.text ;
122     │        SIZE 20,77,0 ;
123     │        DEFAULT " " ;
124     │        SCROLL;
125     │        NOMODIFY
126     │    @ 23,26 GET m.save ;
127     │        PICTURE "@*HT \<Print;\<Another;\<Quit" ;
128     │        SIZE 1,9,1 ;
129     │        DEFAULT 1 ;
130     │        VALID _qigOixoxd()
```

```
131
132     IF NOT WVISIBLE("memowin")
133        ACTIVATE WINDOW memowin
134     ENDIF
135
136     READ CYCLE
137
138     RELEASE WINDOW memowin
139     SELECT (m.currarea)
140
141
142     #REGION 0
143     IF m.talkstat = "ON"
144        SET TALK ON
145     ENDIF
146     IF m.compstat = "ON"
147        SET COMPATIBLE ON
148     ENDIF
149
150
151     *       ************************************************************
152     *       *
153     *       *                MEMOWIN/MS-DOS Cleanup Code
154     *       *
155     *       ************************************************************
156     *
157
158     #REGION 1
159 <---RETURN
160
161   CASE _windows
162
163
164     *       ************************************************************
165     *       *
166     *       *         MEMOWIN/Windows Setup Code - SECTION 1
167     *       *
168     *       ************************************************************
169     *
170
171     #REGION 1
172     PRIVATE ALL
173     DO CASE
174        CLEAR
175     CASE PARAMETER() = 0
176        m.title = ""
177        m.text = ""
178     CASE PARAMETER() = 1
179        m.title = ""
180     ENDCASE
181
182
183
184     #REGION 0
185     REGIONAL m.currarea, m.talkstat, m.compstat
186
187     IF SET("TALK") = "ON"
188        SET TALK OFF
189        m.talkstat = "ON"
190     ELSE
191        m.talkstat = "OFF"
192     ENDIF
193     m.compstat = SET("COMPATIBLE")
194     SET COMPATIBLE FOXPLUS
195
196     m.currarea = SELECT()
```

```
197
198
199     *       ************************************************************
200     *       *
201     *       *                Windows Window definitions
202     *       *
203     *       ************************************************************
204     *
205
206   IF NOT WEXIST("memowin") ;
207          OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.PJX" ;
208          OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.SCX" ;
209          OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.MNX" ;
210          OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.PRG" ;
211          OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.FRX" ;
212          OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.QPR"
213      DEFINE WINDOW memowin ;
214          AT  0.000, 0.000  ;
215          SIZE 26.769,80.000 ;
216          font "MS Sans Serif", 8 ;
217          STYLE "B" ;
218          NOFLOAT ;
219          NOCLOSE ;
220          SHADOW ;
221          nominimize ;
222          NONE
223      MOVE WINDOW memowin CENTER
224   ENDIF
225
226
227     *       ************************************************************
228     *       *
229     *       *                MEMOWIN/Windows Screen Layout
230     *       *
231     *       ************************************************************
232     *
233
234     #REGION 1
235     IF WVISIBLE("memowin")
236        ACTIVATE WINDOW memowin SAME
237     ELSE
238        ACTIVATE WINDOW memowin NOSHOW
239     ENDIF
240     @ 1.000,4.667 GET m.title ;
241          SIZE 1.154,77.000 ;
242          DEFAULT " " ;
243          font "MS Sans Serif", 8 ;
244          PICTURE "@I" ;
245          DISABLE
246     @ 3.692,5.000 EDIT m.text ;
247          SIZE 13.769,75.800,0.000 ;
248          DEFAULT " " ;
249          font "MS Sans Serif", 8 ;
250          SCROLL
251     @ 21.077,23.667 GET m.save ;
252          PICTURE "@*HT \<Print;\<Another;\<Quit" ;
253          SIZE 1.769,8.500,1.000 ;
254          DEFAULT 1 ;
255          font "MS Sans Serif", 8 ;
256          STYLE "B" ;
257          VALID _qig0ixpjo()
258
259     IF NOT WVISIBLE("memowin")
260        ACTIVATE WINDOW memowin
261     ENDIF
262
```

```
263         READ CYCLE
264
265         RELEASE WINDOW memowin
266         SELECT (m.currarea)
267
268
269         #REGION 0
270        ┌IF m.talkstat = "ON"
271        │   SET TALK ON
272        └ENDIF
273        ┌IF m.compstat = "ON"
274        │   SET COMPATIBLE ON
275        └ENDIF
276
277
278         *       ************************************************************
279         *       *
280         *       *                MEMOWIN/Windows Cleanup Code
281         *       *
282         *       ************************************************************
283         *
284
285         #REGION 1
286   <─────RETURN
287      │
288      └ENDCASE
289
290
291         *       ************************************************************
292         *       *
293         *       * _QIGOIXOXD            m.save VALID
294         *       *
295         *       * Function Origin:
296         *       *
297         *       * From Platform:        MS-DOS
298         *       * From Screen:          MEMOWIN,      Record Number:    4
299         *       * Variable:             m.save
300         *       * Called By:            VALID Clause
301         *       * Snippet Number:       1
302         *       *
303         *       ************************************************************
304         *
314      *!*************************************************************************
=> ******
315      *!
316      *!       Function: _QIGOIXOXD()
317      *!
318      *!      Called by: MEMOWIN.PRG
319      *!
320      *!          Calls: W_PRINT.SPR
321      *!
322      *!*************************************************************************
=> ******
314         FUNCTION _qigOixoxd     &&  m.save VALID
315         #REGION 1
316        ┌IF m.save = 1                 && Print
317        │   DO w_print.spr WITH m.text
318   <─────RETURN m.save
319      └ENDIF
320      ┌IF m.save = 2                  && Another
321   <─────RETURN m.save
322      └ENDIF
323      ┌IF m.save = 3                  && Quit
324   <─────RETURN m.save
325      └ENDIF
326
```

```
327
328         *       ************************************************************
329         *       *
330         *       * _QIGOIXPJO            m.save VALID
331         *       *
332         *       * Function Origin:
333         *       *
334         *       * From Platform:        Windows
335         *       * From Screen:          MEMOWIN,      Record Number:    8
336         *       * Variable:             m.save
337         *       * Called By:            VALID Clause
338         *       * Snippet Number:       2
339         *       *
340         *       ************************************************************
341         *
351      *!*************************************************************************
=> ******
352      *!
353      *!       Function: _QIGOIXPJO()
354      *!
355      *!      Called by: MEMOWIN.PRG
356      *!
357      *!          Calls: W_PRINT.SPR
358      *!
359      *!*************************************************************************
=> ******
351         FUNCTION _qigOixpjo     &&  m.save VALID
352         #REGION 1
353        ┌IF m.save = 1                 && Print
354        │   DO w_print.spr WITH m.text
355        │   m.quit = .F.
356        └ENDIF
357        ┌IF m.save = 2                  && Another
358        │   m.quit = .F.
359        └ENDIF
360        ┌IF m.save = 3                  && Quit
361        │   m.quit = .T.
362        └ENDIF
363         *: EOF: MEMOWIN.ACT
```

```
  1  *:*********************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\YESNO.PRG
  4  *:
  5  *:         System: Hazardous Material Life-Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 09/10/93      8:12
  9  *:
 10  *:         Set by: HMSC.PRG
 11  *:               : _QIPOVASCN()       (function  in HMSTEP.PRG)
 12  *:               : _QIPOVATWZ()       (function  in HMSTEP.PRG)
 13  *:
 14  *:          Calls: ERRMSG.PRG
 15  *:
 16  *:      Documented 12/01/93 at 11:32                FoxDoc  version 2.10f
 17  *:*********************************************************************
=> ******
 18      *       ************************************************************
 19      *       *                                                          *
 20      *       *                 YESNO.PRG                 22:21:11 *     *
 21      *       *                                                          *
 22      *       ************************************************************
 23      *       *                                                          *
 24      *       * Description:                                             *
 25      *       * This program was automatically generated by GENSCRN.     *
 26      *       *                                                          *
 27      *       ************************************************************
 28
 29
 30      *       ************************************************************
 31      *       *                                                          *
 32      *       *               YESNO Setup Code - SECTION 1               *
 33      *       *                                                          *
 34      *       ************************************************************
 35      *
 36
 37
 38      #REGION 1
 39
 40      PARAMETERS MESSAGE, ok, CANCEL
 41      PRIVATE ALL
 42
 43      ON ERROR DO errmsg WITH MESSAGE()
 44
 45      * If no message is sent then create message
 46
 47      IF PARAMETERS()=0
 48         m.message="is this correct?"
 49      ENDIF
 50
 51      * Ok and Cancel are used for prompts in the push buttons
 52      IF PARAMETERS() < 3
 53         * Default to normal prompts
 54         m.ok = "OK"
 55         m.cancel = "Cancel"
 56      ENDIF
 57
 58      * Truncate any message longer than 50 characters
 59      MESSAGE=IIF(TYPE("m.message")!="C","",m.message)
 60      IF LEN( m.message ) > 50
 61         * The message is centered in an @SAY
 62         m.message = SUBSTR( m.message, 1, 50 )
 63      ENDIF
 64
 65      m.message=UPPER(m.message)
 66      m.ok=UPPER(m.ok)
 67      m.cancel=UPPER(m.cancel)
 68
 69      PUSH KEY CLEAR
 70
 71      #REGION 0
 72      REGIONAL m.currarea, m.talkstat, m.compstat
 73
 74      m.talkstat=SET("TALK")
 75      SET TALK OFF
 76
 77      m.compstat = SET("COMPATIBLE")
 78      SET COMPATIBLE FOXPLUS
 79
 80      *       ************************************************************
 81      *       *                                                          *
 82      *       *                     Window definitions                   *
 83      *       *                                                          *
 84      *       ************************************************************
 85      *
 86
 87      IF NOT WEXIST("yesno")
 88         DEFINE WINDOW yesno ;
 89             FROM INT((SROW()-7)/2),INT((SCOL()-48)/2) ;
 90             TO INT((SROW()-7)/2)+6,INT((SCOL()-48)/2)+47 ;
 91             FLOAT ;
 92             NOCLOSE ;
 93             SHADOW ;
 94             DOUBLE ;
 95             COLOR SCHEME 5
 96      ENDIF
 97
 98
 99      *       ************************************************************
100      *       *                                                          *
101      *       *                     YESNO Screen Layout                  *
102      *       *                                                          *
103      *       ************************************************************
104      *
105
106      #REGION 1
107      IF WVISIBLE("yesno")
108         ACTIVATE WINDOW yesno SAME
109      ELSE
110         ACTIVATE WINDOW yesno NOSHOW
111      ENDIF
112      @ 1,2 SAY PADC( m.message, 40 ) ;
113          SIZE 1,41
114      @ 3,8 GET m.answer ;
115          PICTURE "@*HT \!\<&OK;\<&Cancel" ;
116          SIZE 1,12,5 ;
117          DEFAULT 1
118
119      IF NOT WVISIBLE("yesno")
120         ACTIVATE WINDOW yesno
121      ENDIF
122
123      READ CYCLE MODAL
124
125      RELEASE WINDOW yesno
126
127      #REGION 0
128
129      IF m.talkstat = "ON"
130         SET TALK ON
```

```
131     ELSE
132        SET TALK OFF
133     ENDIF
134
135     IF m.compstat = "ON"
136        SET COMPATIBLE ON
137     ENDIF
138
139
140       *       ************************************************************
141       *       *                                                          *
142       *       *                     YESNO Cleanup Code                   *
143       *       *                                                          *
144       *       ************************************************************
145       *
146
147       #REGION 1
148       POP KEY
149
150       * Convert the numeric value of m.Answer from 1|2 to .T.|.F.
151       * If the user selected OK and didn't exit with Escape
152     IF m.answer = 1 AND LASTKEY() <> 27
153  <------RETURN .T.
154     ELSE
155        * Cancel or Escape returns false
156  <------RETURN .F.
157     ENDIF
158       *: EOF: YESNO.ACT
```

```
*
*       ┌──────────────────────────────────────────────────────┐
*       │ 10/27/93             BROWSWIN.SPR            14:49:42 │
*       ├──────────────────────────────────────────────────────┤
*       │                                                      │
*       │ Author's Name                                        │
*       │                                                      │
*       │ Copyright (c) 1993 Company Name                      │
*       │ Address                                              │
*       │ City,     Zip                                        │
*       │                                                      │
*       │ Description:                                         │
*       │ This program was automatically generated by GENSCRN. │
*       │                                                      │
*       └──────────────────────────────────────────────────────┘


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*       ┌──────────────────────────────────────────────────────┐
*       │              MS-DOS Window definitions               │
*       └──────────────────────────────────────────────────────┘
*
*

IF NOT WEXIST("browsewin") ;
      OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.PJX" ;
      OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.SCX" ;
      OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.MNX" ;
      OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.PRG" ;
      OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.FRX" ;
      OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.QPR"
   DEFINE WINDOW browsewin ;
      FROM INT((SROW()-21)/2),INT((SCOL()-80)/2) ;
      TO INT((SROW()-21)/2)+20,INT((SCOL()-80)/2)+79 ;
      FLOAT ;
      NOCLOSE ;
      SHADOW ;
      nominimize ;
      SYSTEM ;
      COLOR SCHEME 1
ENDIF


*
*       ┌──────────────────────────────────────────────────────┐
*       │            BROWSWIN/MS-DOS Screen Layout             │
*       └──────────────────────────────────────────────────────┘
*
*

#REGION 1
IF WVISIBLE("browsewin")
   ACTIVATE WINDOW browsewin SAME
ELSE
   ACTIVATE WINDOW browsewin NOSHOW
ENDIF

IF NOT WVISIBLE("browsewin")
   ACTIVATE WINDOW browsewin
ENDIF

READ CYCLE

RELEASE WINDOW browsewin

#REGION 0
IF m.talkstat = "ON"
   SET TALK ON
ENDIF
IF m.compstat = "ON"
   SET COMPATIBLE ON
ENDIF
*: EOF: BROWSWIN.AC1
```

```
*
*   10/27/93            HMAT.SPR            14:48:51
*
*   Author's Name
*
*   Copyright (c) 1993 Company Name
*   Address
*   City,     Zip
*
*   Description:
*   This program was automatically generated by GENSCRN.
*

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*
*              MS-DOS Window definitions
*
*
*

IF NOT WEXIST("hmat") ;
      OR UPPER(WTITLE("HMAT")) == "HMAT.PJX" ;
      OR UPPER(WTITLE("HMAT")) == "HMAT.SCX" ;
      OR UPPER(WTITLE("HMAT")) == "HMAT.MNX" ;
      OR UPPER(WTITLE("HMAT")) == "HMAT.PRG" ;
      OR UPPER(WTITLE("HMAT")) == "HMAT.FRX" ;
      OR UPPER(WTITLE("HMAT")) == "HMAT.QPR"
   DEFINE WINDOW hmat ;
      FROM INT((SROW()-21)/2),INT((SCOL()-80)/2) ;
      TO INT((SROW()-21)/2)+20,INT((SCOL()-80)/2)+79 ;
      TITLE "HAZARDOUS MATERIALS" ;
      NOFLOAT ;
      NOCLOSE ;
      SHADOW ;
      nominimize ;
      DOUBLE ;
      COLOR SCHEME 1
ENDIF


*
*
*        HMAT/MS-DOS Setup Code - SECTION 2
*
*
*

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
m.oldescape = SET( "ESCAPE" )
SET ESCAPE OFF
m.adding = .F.
m.change = .F.
*allowedit = .f.

IF USED('hmat')
   SELECT hmat
   SET ORDER TO TAG hmatid
ELSE
   USE hmat AGAIN ORDER TAG hmatid
ENDIF
*************************************************
* Check see if the last record is defined
IF TYPE( "m.LastRec" ) = "U"

   * Start with the first record
   GO TOP
   m.lastrec = RECNO()

ELSE
   * Start on the last record used
   GO m.lastrec
ENDIF
**************************************************
SCATTER MEMVAR MEMO
m.hmcom = comname(ALLTRIM(m.niin))
m.oldhmcom = m.hmcom

*
*
*           HMAT/MS-DOS Screen Layout
*
*
*

#REGION 1
IF WVISIBLE("hmat")
   ACTIVATE WINDOW hmat SAME
ELSE
   ACTIVATE WINDOW hmat NOSHOW
ENDIF
@ 0,2 SAY "ID" ;
    SIZE 1,2, 0
@ 2,2 SAY "NIIN #:" ;
    SIZE 1,7, 0
@ 2,23 SAY "Common Name:" ;
    SIZE 1,12, 0
@ 1,2 SAY "Manufa" ;
    SIZE 1,6, 0
@ 0,14 SAY "Materials Name:" ;
    SIZE 1,15, 0
@ 0,5 GET m.hmatid ;
    SIZE 1,5 ;
    DEFAULT 0 ;
    DISABLE
@ 0,32 GET m.hmatname ;
    SIZE 1,44 ;
    DEFAULT " " ;
    PICTURE "@!" ;
    VALID _qkf0vr394()
@ 1,15 GET m.mfg ;
    SIZE 1,61 ;
    DEFAULT " " ;
    VALID _qkf0vr3ek()
```

```
133     @ 2,10 GET m.niin ;
134         SIZE 1,10 ;
135         DEFAULT " " ;
136         VALID _qkf0vr3hm()
137     @ 2,37 GET m.hmcom ;
138         SIZE 1,39 ;
139         DEFAULT " " ;
140         WHEN _qkf0vr3kq() ;
141         VALID _qkf0vr3me() ;
142         DISABLE
143     @ 4,0 EDIT m.msds ;
144         SIZE 13,78,0 ;
145         DEFAULT " " ;
146         SCROLL ;
147         VALID _qkf0vr3on()
148     @ 18,6 GET m.action ;
149         PICTURE "@*HN \<Add;\<Next;\<Previous;\?E\<xit" ;
150         SIZE 1,10,1 ;
151         DEFAULT 1 ;
152         VALID _qkf0vr3r9()
153     @ 18,51 GET m.save ;
154         PICTURE "@*HN \<Save;\<Cancel" ;
155         SIZE 1,8,1 ;
156         DEFAULT 1 ;
157         VALID _qkf0vr3xh() ;
158         DISABLE
159     @ 1,8 SAY "cturer:" ;
160         SIZE 1,7, 0
161
162   IF NOT WVISIBLE("hmat")
163       ACTIVATE WINDOW hmat
164   ENDIF
165
166     READ CYCLE MODAL
167
168     RELEASE WINDOW hmat
169
170     #REGION 0
171   IF m.talkstat = "ON"
172       SET TALK ON
173   ENDIF
174   IF m.compstat = "ON"
175       SET COMPATIBLE ON
176   ENDIF
177
178     *
179     *
180     *                  HMAT/MS-DOS Cleanup Code
181     *
182     *
183     *
184     *
185
186     #REGION 1
187     SELECT hmcom
188     USE
189     SELECT hmat
190     USE
191     POP KEY ALL
192     SET ESCAPE &oldescape
193     ********* End of Main Body - Entry Cleanup
194
195     ******************
196
197
198     *
```

```
199     *
200     *          HMAT/MS-DOS Supporting Procedures and Functions
201     *
202     *
203     *
204
205     #REGION 1
206     PROCEDURE CHANGE
207     ******************
208     m.oldexact = SET( "EXACT" )
209     SET EXACT ON
210     m.change =(TRIM(hmat.hmatname) <> TRIM( m.hmatname);
211         OR m.mfg <> hmat.mfg;
212         OR m.niin <> hmat.niin;
213         OR m.msds <> hmat.msds)
214     SET EXACT &oldexact
215     RETURN m.change
216
217     ******************
218     FUNCTION newline
219     ******************
220     PARAMETER TEXT
221     nl = CHR(10) + CHR(13)
222   IF EMPTY(m.text)
223       m.text = ""
224   ELSE
225       m.text = m.text + nl + nl
226   ENDIF
227     RETURN m.text
228
229     ******************
230     FUNCTION comname
231     ******************
232     PARAMETER mkey
233     PRIVATE msel
234     msel = SELECT()
235   IF !USED("HMCOM")
236       SELECT 0
237       USE hmcom ORDER TAG niin
238   ELSE
239       SELECT hmcom
240       SET ORDER TO TAG niin
241   ENDIF
242     SEEK mkey
243   IF FOUND()
244       m.name = common
245   ELSE
246       m.name = SPACE(30)
247   ENDIF
248     SELECT (msel)
249     RETURN m.name
250
251     ******************
252     PROCEDURE savecom
253     ******************
254     PARAMETER mkey, mname
255     PRIVATE msel
256     msel = SELECT()
257   IF !USED("HMCOM")
258       SELECT 0
259       USE hmcom ORDER TAG niin
260   ELSE
261       SELECT hmcom
262       SET ORDER TO TAG niin
263   ENDIF
264     SEEK mkey
```

```
265     IF FOUND()
266        REPLACE niin WITH mkey
267        REPLACE common WITH mname
268     ELSE
269        SET ORDER TO TAG hmcomid
270        GO BOTTOM
271        id = hmcomid + 1
272        REPLACE hmcomid WITH id
273        REPLACE niin WITH mkey
274        REPLACE common WITH mname
275     ENDIF
276     SELECT (msel)
277     RETURN
278
279
280
281     *
282     *
283     *        _QKF0VR394            M.hmatname VALID
284     *
285     *        Function Origin:
286     *
287     *        From Platform:        MS-DOS
288     *        From Screen:          HMAT,     Record Number:    8
289     *        Variable:             M.hmatname
290     *        Called By:            VALID Clause
291     *        Object Type:          Field
292     *        Snippet Number:       1
293     *
294     *
295     *
296     FUNCTION _qkf0vr394     &&  M.hmatname VALID
297     #REGION 1
298     IF m.adding = .T.
299        m.oldrec = RECNO()
300        SET ORDER TO TAG hmatname
301        SEEK(m.hmatname)
302        IF FOUND()
303           =errmsg("Material name already exist!",2)
304           SCATTER FIELD hmatname MEMVAR BLANK
305           CUROBJ = OBJNUM(m.hmatname)
306        ENDIF
307        SET ORDER TO TAG hmatid
308        GO m.oldrec
309     ENDIF
310     DO CHANGE
311     IF m.change
312        SHOW GET m.action disabled
313        SHOW GET m.save enabled
314     ENDIF
315     SHOW GETS
316
317     *
318     *
319     *        _QKF0VR3EK            m.mfg VALID
320     *
321     *        Function Origin:
322     *
323     *        From Platform:        MS-DOS
324     *        From Screen:          HMAT,     Record Number:    9
325     *        Variable:             m.mfg
326     *        Called By:            VALID Clause
327     *        Object Type:          Field
328     *        Snippet Number:       2
329     *
330     *
```



```
331     *
332     FUNCTION _qkf0vr3ek     &&  m.mfg VALID
333     #REGION 1
334     DO CHANGE
335     IF m.change
336        SHOW GET m.action disabled
337        SHOW GET m.save enabled
338     ENDIF
339     SHOW GETS
340
341     *
342     *
343     *        _QKF0VR3HM            m.niin VALID
344     *
345     *        Function Origin:
346     *
347     *        From Platform:        MS-DOS
348     *        From Screen:          HMAT,     Record Number:   10
349     *        Variable:             m.niin
350     *        Called By:            VALID Clause
351     *        Object Type:          Field
352     *        Snippet Number:       3
353     *
354     *
355     *
356     FUNCTION _qkf0vr3hm     &&  m.niin VALID
357     #REGION 1
358     m.hmcom = comname(ALLTRIM(m.niin))
359     IF EMPTY(m.hmcom)
360        SHOW GET m.hmcom enabled
361     ELSE
362        SHOW GET m.hmcom disabled
363     ENDIF
364     DO CHANGE
365     IF m.change
366        SHOW GET m.action disabled
367        SHOW GET m.save enabled
368     ENDIF
369     SHOW GETS
370
371     *
372     *
373     *        _QKF0VR3KQ            m.hmcom WHEN
374     *
375     *        Function Origin:
376     *
377     *        From Platform:        MS-DOS
378     *        From Screen:          HMAT,     Record Number:   11
379     *        Variable:             m.hmcom
380     *        Called By:            WHEN Clause
381     *        Object Type:          Field
382     *        Snippet Number:       4
383     *
384     *
385     *
386     FUNCTION _qkf0vr3kq     &&  m.hmcom WHEN
387     #REGION 1
388     m.oldhmcom = m.hmcom
389
390     *
391     *
392     *        _QKF0VR3ME            m.hmcom VALID
393     *
394     *        Function Origin:
395     *
396     *        From Platform:        MS-DOS
```

```
*       From Screen:         HMAT,      Record Number:   11
*       Variable:            m.hmcom
*       Called By:           VALID Clause
*       Object Type:         Field
*       Snippet Number:      5
*
*
*
FUNCTION _qkf0vr3me     &&  m.hmcom VALID
#REGION 1
IF m.hmcom <> m.oldhmcom
    SHOW GET m.action disabled
    SHOW GET m.save enabled
ENDIF

*
*
*       _QKFOVR3ON           m.msds VALID
*
*       Function Origin:
*
*       From Platform:       MS-DOS
*       From Screen:         HMAT,      Record Number:   12
*       Variable:            m.msds
*       Called By:           VALID Clause
*       Object Type:         Field
*       Snippet Number:      6
*
*
*
FUNCTION _qkf0vr3on     &&  m.msds VALID
#REGION 1
DO CHANGE
IF m.change
    SHOW GET m.action disabled
    SHOW GET m.save enabled
ENDIF

*
*
*       _QKFOVR3R9           m.Action VALID
*
*       Function Origin:
*
*       From Platform:       MS-DOS
*       From Screen:         HMAT,      Record Number:   13
*       Variable:            m.Action
*       Called By:           VALID Clause
*       Object Type:         Push Button
*       Snippet Number:      7
*
*
*
FUNCTION _qkf0vr3r9     &&  m.Action VALID
#REGION 1
IF m.action = 1
    * SCATTER FIELD hmatname MEMVAR BLANK
    SCATTER MEMVAR MEMO BLANK
    m.hmatid = RECCOUNT()+1
    SHOW GET m.hmatname enabled
    SHOW GET m.mfg enabled
    SHOW GET m.niin enabled
    SHOW GET action disabled
    SHOW GET SAVE enabled
    m.adding = .T.
    SHOW GETS
ELSE

    DO CASE

    CASE m.action = 2
        SKIP
        IF EOF()
            ?? CHR( 7 )
            WAIT "Last record" WINDOW NOWAIT
            GO BOTTOM
        ENDIF

    CASE m.action = 3
        SKIP -1
        IF BOF()
            ?? CHR( 7 )
            WAIT "First record" WINDOW NOWAIT
            GO TOP
        ENDIF

    CASE m.action = 4
        CLEAR READ
    ENDCASE

    IF m.action > 1 AND m.action < 4
        *      allowedit = .f.
        SCATTER MEMVAR MEMO
        m.hmcom = comname(ALLTRIM(m.niin))
        SHOW GETS
    ENDIF
ENDIF


*
*       _QKFOVR3XH           m.Save VALID
*
*       Function Origin:
*
*       From Platform:       MS-DOS
*       From Screen:         HMAT,      Record Number:   14
*       Variable:            m.Save
*       Called By:           VALID Clause
*       Object Type:         Push Button
*       Snippet Number:      8
*
*
*
* did the record change
FUNCTION _qkf0vr3xh     &&  m.Save VALID
#REGION 1
IF m.save = 1 && Selected Save Button
    IF EMPTY(m.hmatname)
        = errmsg("Please enter Materials Name !!...",1)
        CUROBJ = OBJNUM(m.hmatname)
        SHOW GETS
<------ RETURN
    ENDIF
    IF EMPTY(m.niin)
        = errmsg("Please enter Stock Number !!...",1)
        CUROBJ = OBJNUM(m.niin)
        SHOW GETS
<------ RETURN
    ENDIF

    IF m.adding && Adding a new record
```

```
529         APPEND BLANK MEMO
530         GATHER MEMVAR MEMO
531      ELSE
532         DO CHANGE
533         IF m.change && Changing an old record
534            GATHER MEMVAR MEMO
535         ENDIF
536      ENDIF
537      DO savecom WITH m.niin, m.hmcom
538   ELSE
539      SCATTER MEMVAR MEMO
540      m.hmcom = comname(ALLTRIM(m.niin))
541   ENDIF
542
543   SHOW GETS
544   SHOW GET m.hmcom disabled
545   SHOW GET action enabled
546   SHOW GET SAVE disabled
547
548   m.adding = .F.
549   m.change = .F.
550   *: EOF: HMAT.AC1
```

```
*
*       10/27/93              HMCF.SPR                14:49:05
*
*       Author's Name
*
*       Copyright (c) 1993 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*            MS-DOS Window definitions
*
*

IF NOT WEXIST("hmcf") ;
      OR UPPER(WTITLE("HMCF")) == "HMCF.PJX" ;
      OR UPPER(WTITLE("HMCF")) == "HMCF.SCX" ;
      OR UPPER(WTITLE("HMCF")) == "HMCF.MNX" ;
      OR UPPER(WTITLE("HMCF")) == "HMCF.PRG" ;
      OR UPPER(WTITLE("HMCF")) == "HMCF.FRX" ;
      OR UPPER(WTITLE("HMCF")) == "HMCF.QPR"
   DEFINE WINDOW hmcf ;
      FROM INT((SROW()-12)/2),INT((SCOL()-53)/2) ;
      TO INT((SROW()-12)/2)+11,INT((SCOL()-53)/2)+52 ;
      NOFLOAT ;
      NOCLOSE ;
      SHADOW ;
      nominimize ;
      DOUBLE ;
      COLOR SCHEME 1
ENDIF


*
*          HMCF/MS-DOS Setup Code - SECTION 2
*
*

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
```



```
m.oldescape = SET( "ESCAPE" )
SET ESCAPE OFF
m.adding = .F.
m.change = .F.

CLOSE ALL
SELECT 0
USE hmcf
SET ORDER TO TAG hmcf OF hmcf.cdx

************************************************
* Check see if the last record is defined
IF TYPE( "m.LastRec" ) = "U"

   * Start with the first record
   GO TOP
   m.lastrec = RECNO()

ELSE
   * Start on the last record used
   GO m.lastrec
ENDIF
************************************************
SCATTER MEMVAR


*
*            HMCF/MS-DOS Screen Layout
*
*

#REGION 1
IF WVISIBLE("hmcf")
   ACTIVATE WINDOW hmcf SAME
ELSE
   ACTIVATE WINDOW hmcf NOSHOW
ENDIF
@ 1,40 GET m.action ;
    PICTURE "@*VN \<Add;\<Edit;\!\<Next;\<Previous;\?E\<xit" ;
    SIZE 1,10,1 ;
    DEFAULT 1 ;
    VALID _qkf0vrdyq()
@ 2,14 GET m.hmcfid ;
    SIZE 1,10 ;
    DEFAULT 0 ;
    DISABLE
@ 4,14 GET m.hmcf ;
    SIZE 1,25 ;
    DEFAULT " " ;
    PICTURE "@!" ;
    VALID _qkf0vre5a() ;
    DISABLE
@ 8,11 GET m.save ;
    PICTURE "@*HN \<Save;\<Cancel" ;
    SIZE 1,8,1 ;
    DEFAULT 1 ;
    VALID _qkf0vre7o() ;
    DISABLE
@ 1,0 TO 9,39
@ 0,11 SAY "HM COST FACTORS" ;
    SIZE 1,15, 0
@ 2,5 SAY "ID NUM:" ;
    SIZE 1,7, 0
@ 4,1 SAY "COST FACTOR:" ;
```

```
133         SIZE 1,12, 0
134
135     IF NOT WVISIBLE("hmcf")
136        ACTIVATE WINDOW hmcf
137     ENDIF
138
139     READ CYCLE
140
141     RELEASE WINDOW hmcf
142
143     #REGION 0
144     IF m.talkstat = "ON"
145        SET TALK ON
146     ENDIF
147     IF m.compstat = "ON"
148        SET COMPATIBLE ON
149     ENDIF
150
151
152     *
153     *
154     *                HMCF/MS-DOS Cleanup Code
155     *
156     *
157     *
158
159     #REGION 1
160     POP KEY ALL
161     SET ESCAPE &oldescape
162     ********* End of Main Body - Entry Cleanup
163
164     ******************
165
166
167     *
168     *
169     *          HMCF/MS-DOS Supporting Procedures and Functions
170     *
171     *
172     *
173
174     #REGION 1
175     PROCEDURE CHANGE
176     ******************
177     m.oldexact = SET( "EXACT" )
178     SET EXACT ON
179     m.change =(TRIM(hmcf.hmcf) <> TRIM( m.hmcf))
180     SET EXACT &oldexact
181     RETURN m.change
182
183
184     *
185     *
186     *        _QKFOVRDYQ            m.Action VALID
187     *
188     *        Function Origin:
189     *
190     *        From Platform:        MS-DOS
191     *        From Screen:          HMCF,     Record Number:    4
192     *        Variable:             m.Action
193     *        Called By:            VALID Clause
194     *        Object Type:          Push Button
195     *        Snippet Number:       1
196     *
197     *
198     *
199     FUNCTION _qkf0vrdyq     && m.Action VALID
200     #REGION 1
201     IF m.action = 1
202        SCATTER MEMVAR BLANK
203        m.hmcfid=RECCOUNT()+1
204        SHOW GETS
205        SHOW GET m.hmcf enabled
206        SHOW GET action disabled
207        SHOW GET SAVE enabled
208        m.adding = .T.
209
210     ELSE
211
212        DO CASE
213
214        CASE m.action = 2
215           SHOW GETS
216           SHOW GET action disabled
217           SHOW GET SAVE enabled
218           SHOW GET m.hmcf enabled
219
220        CASE m.action = 3
221           SKIP
222           IF EOF()
223              ?? CHR( 7 )
224              WAIT "Last record" WINDOW NOWAIT
225              SKIP -1
226           ELSE
227              SCATTER MEMVAR
228              SHOW GETS
229           ENDIF
230
231        CASE m.action = 4
232           SKIP -1
233           IF BOF()
234              ?? CHR( 7 )
235              WAIT "First record" WINDOW NOWAIT
236              SKIP
237           ELSE
238              SCATTER MEMVAR
239              SHOW GETS
240           ENDIF
241
242        CASE m.action = 5
243           CLEAR READ
244        ENDCASE
245     ENDIF
246
247
248     *
249     *
250     *        _QKFOVRE5A            m.hmcf VALID
251     *
252     *        Function Origin:
253     *
254     *        From Platform:        MS-DOS
255     *        From Screen:          HMCF,     Record Number:    6
256     *        Variable:             m.hmcf
257     *        Called By:            VALID Clause
258     *        Object Type:          Field
259     *        Snippet Number:       2
260     *
261     *
262     *
263     FUNCTION _qkf0vre5a     && m.hmcf VALID
264     #REGION 1
```

```
265     IF m.adding
266        m.oldrec = RECNO()
267        SEEK m.hmcf
268        IF FOUND()
269           =errmsg("Record already exists",1)
270           SCATTER MEMVAR BLANK FIELD hmcf
271        ENDIF
272        GO m.oldrec
273     ENDIF
274
275
276       *
277       *
278       *        _QKF0VRE7O           m.Save VALID
279       *
280       *        Function Origin:
281       *
282       *        From Platform:       MS-DOS
283       *        From Screen:         HMCF,     Record Number:    7
284       *        Variable:            m.Save
285       *        Called By:           VALID Clause
286       *        Object Type:         Push Button
287       *        Snippet Number:      3
288       *
289       *
290       *
291       FUNCTION _qkf0vre7o     &&  m.Save VALID
292       #REGION 1
293       m.notsave = .F.
294     IF m.save = 1 && Selected Save Button
295        IF EMPTY(m.hmcf)
296           =errmsg("Data empty, could not save!!",2)
297           _CUROBJ = OBJNUM(m.hmcf)
298           m.notsave = .T.
299        ELSE
300           IF m.adding && Adding a new record
301              APPEND BLANK
302              GATHER MEMVAR
303           ELSE
304
305              IF m.change && Changing an old record
306                 GATHER MEMVAR
307              ENDIF
308           ENDIF
309        ENDIF
310     ELSE
311        SCATTER MEMVAR
312     ENDIF
313
314       SHOW GETS
315     IF m.notsave
316        SHOW GET m.hmcf enabled
317        SHOW GET action disabled
318        SHOW GET SAVE enabled
319     ELSE
320        SHOW GET m.hmcf disabled
321        SHOW GET action enabled
322        SHOW GET SAVE disabled
323        m.adding = .f.
324        m.change = .F.
325     ENDIF
326       *: EOF: HMCF.AC1
```

```
*
*       10/27/93            HMCFE.SPR            14:49:08
*
*
*
*       Author's Name
*
*       Copyright (c) 1993 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*
*                MS-DOS Window definitions
*
*
*

IF NOT WEXIST("hmcfe") ;
        OR UPPER(WTITLE("HMCFE")) == "HMCFE.PJX" ;
        OR UPPER(WTITLE("HMCFE")) == "HMCFE.SCX" ;
        OR UPPER(WTITLE("HMCFE")) == "HMCFE.MNX" ;
        OR UPPER(WTITLE("HMCFE")) == "HMCFE.PRG" ;
        OR UPPER(WTITLE("HMCFE")) == "HMCFE.FRX" ;
        OR UPPER(WTITLE("HMCFE")) == "HMCFE.QPR"
   DEFINE WINDOW hmcfe ;
        FROM INT((SROW()-16)/2),INT((SCOL()-67)/2) ;
        TO INT((SROW()-16)/2)+15,INT((SCOL()-67)/2)+66 ;
        NOFLOAT ;
        NOCLOSE ;
        SHADOW ;
        nominimize ;
        DOUBLE ;
        COLOR SCHEME 1
ENDIF


*
*
*             HMCFE/MS-DOS Setup Code - SECTION 2
*
*
*

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
m.oldescape = SET( "ESCAPE" )
SET ESCAPE OFF
m.adding = .F.
m.change = .F.
m.action=5

CLOSE ALL

IF USED("HMCFE")
   SELECT hmcfe
   SET ORDER TO hmcfid
ELSE
   SELECT 0
   USE (LOCFILE("HMCFE.dbf","DBF","Where is hmmt?"));
       AGAIN ALIAS hmcfe ;
       ORDER hmcfid
ENDIF

*USE HMCFE AGAIN
***************************************************
* Check see if the last record is defined
GO TOP
*****************************************************
SCATTER MEMVAR
IF m.hmcfid>0
   m.answr=get_hmcf(m.hmcfid)
ELSE
   m.answr=""
ENDIF

m.oldanswr=m.answr

SHOW GETS


*
*
*             HMCFE/MS-DOS Screen Layout
*
*
*

#REGION 1
IF WVISIBLE("hmcfe")
   ACTIVATE WINDOW hmcfe SAME
ELSE
   ACTIVATE WINDOW hmcfe NOSHOW
ENDIF
@ 1,54 GET m.action ;
    PICTURE "@*VN \<Add;\<Edit;\<Next;\<Previous;\!\<Top;\<Bottom;\?E
=> \<xit" ;
    SIZE 1,10,1 ;
    DEFAULT 1 ;
    VALID _qkf0vrggz()
@ 5,3 GET m.factor ;
    PICTURE "@*IHK " ;
    SIZE 1,10,1 ;
    DEFAULT 0 ;
    WHEN _qkf0vrgrp() ;
    VALID _qkf0vrgtp()
@ 7,3 GET m.element ;
    PICTURE "@*IHN " ;
    SIZE 1,10,1 ;
    DEFAULT 0 ;
    WHEN _qkf0vrgx4() ;
    VALID _qkf0vrgyg()
```

```
@ 5,15 GET m.hmcfid ;
    SIZE 1,5 ;
    DEFAULT " " ;
    DISABLE
@ 5,22 GET m.answr ;
    SIZE 1,30 ;
    DEFAULT " " ;
    WHEN _qkf0vrh1g() ;
    VALID _qkf0vrh34() ;
    DISABLE
@ 7,15 GET m.hmcfeid ;
    SIZE 1,5 ;
    DEFAULT 0 ;
    DISABLE
@ 7,22 GET m.hmcfe ;
    SIZE 1,30 ;
    DEFAULT " " ;
    PICTURE "@!" ;
    DISABLE
@ 11,21 GET m.save ;
    PICTURE "@*HN \<Save;\<Cancel" ;
    SIZE 1,8,1 ;
    DEFAULT 1 ;
    VALID _qkf0vrh6d() ;
    DISABLE
@ 1,1 TO 12,52
@ 0,18 SAY "HM COST FACTOR ELEMENTS" ;
    SIZE 1,23, 0
@ 7,4 SAY "ELEMENT:" ;
    SIZE 1,8, 0
@ 5,5 SAY "FACTOR:" ;
    SIZE 1,7, 0
@ 3,23 SAY "." ;
    SIZE 1,1, 0

IF NOT WVISIBLE("hmcfe")
   ACTIVATE WINDOW hmcfe
ENDIF

READ CYCLE

RELEASE WINDOW hmcfe

#REGION 0
IF m.talkstat = "ON"
   SET TALK ON
ENDIF
IF m.compstat = "ON"
   SET COMPATIBLE ON
ENDIF


*
*
*          HMCFE/MS-DOS Cleanup Code
*
*
*

#REGION 1
POP KEY ALL
CLOSE DATABASE cfetmp,cftmp
SET ESCAPE &oldescape
********* End of Main Body - Entry Cleanup *****
*********************
```



```
*
*
*          HMCFE/MS-DOS Supporting Procedures and Functions
*
*
*

#REGION 1
PROCEDURE escpressed
*********************
RETURN
*********************
PROCEDURE CHANGE
*********************
m.oldexact = SET( "EXACT" )
SET EXACT ON
m.change =(TRIM(hmcfe.hmcfe) <> TRIM( m.hmcfe);
    OR hmcfe.hmcfid <> m.hmcfid)
SET EXACT &oldexact
RETURN m.change

******************
PROCEDURE get_hmcf
******************
PARAMETER m.hmcfid
m.oldfile=SELECT()

SELECT DISTINCT hmcf.hmcf;
    FROM hmcf;
    WHERE hmcf.hmcfid = (m.hmcfid);
    INTO ARRAY X
m.hmcf=X[1]

SELECT(m.oldfile)
RETURN m.hmcf
*************************
PROCEDURE get_cf
*************************
PARAMETER m.answr
DIMENSION cfearr[1]
cfearr[1]=""
m.oldfile=SELECT()

SELECT DISTINCT hmcf.hmcf;
    FROM hmcf;
    WHERE hmcf.hmcf IN (m.answr);
    ORDER BY hmcf.hmcfid;
    INTO ARRAY cfearr

IF NOT EMPTY(cfearr[1])

   m.answr=chooser(@cfearr,"Select a Cost Factor")

ELSE
   =errmsg(m.answr + " was not found",1)
   m.hmcfid=0
   m.answr=""

ENDIF

SELECT(m.oldfile)
RETURN m.answr

***************************
PROCEDURE get_cfid
```

```
264     ****************************
265     PARAMETERS m.answr
266     IF PARAMETER()=0
267        X=0
268     ELSE
269
270        m.oldfile=SELECT()
271
272        SELECT DISTINCT hmcf.hmcfid;
273            FROM hmcf;
274            WHERE hmcf.hmcf IN (m.answr);
275            INTO ARRAY X
276        X=X[1]
277        SELECT (m.oldfile)
278     ENDIF
279     RETURN X
280
281     ************************
282     PROCEDURE rel
283     * DISPLAY RELATIONSHIP
284     * OF HMCFE AND HMCF
285     ************************
286     PARAMETER m.id
287     SET TALK OFF
288
289     IF PARAMETER()=0
290        m.id=0
291     ENDIF
292
293     m.oldfile=SELECT()
294
295     &&OPEN FILE # 1
296
297     IF USED("HMCFEI")
298        SELECT hmcfei
299        SET ORDER TO hmcfeid
300
301     ELSE
302        SELECT 0
303        USE hmcfei ORDER hmcfeid
304     ENDIF
305
306     &&OPEN FILE # 2
307
308     IF USED("HMCFE")
309        SELECT hmcfe
310        SET ORDER TO hmcfid
311     ELSE
312        SELECT 0
313        USE hmcfe AGAIN ORDER hmcfid
314     ENDIF
315
316     && OPEN FILE #3
317
318     IF USED("HMCF")
319        SELECT hmcf
320        SET ORDER TO hmcfid
321     ELSE
322        SELECT 0                    &&OPEN FILE #3
323        USE hmcf ORDER hmcfid
324     ENDIF
325
326     IF m.id>0
327
328        SET FILTER TO hmcfid = m.id
329     ENDIF
```



```
330
331     SELECT hmcfe
332     SET RELATION TO hmcfeid INTO hmcfei ADDITIVE
333
334     SELECT hmcf
335
336     SET RELATION TO hmcfid INTO hmcfe ADDITIVE
337     SET SKIP TO hmcfei, hmcfe
338
339     ** Show fields from grandparent (HMCFEI), parent (HMCFE) and child (H
=> MCF)**
340     BROWSE FIELDS hmcf.hmcf:H="FACTORS", hmcfe.hmcfe:H="ELEMENTS",;
341         hmcfei.hmcfei:H="ITEMS", hmcfei.hmcfeicost:H="COSTS";
342         FOR hmcf.hmcfid = hmcfe.hmcfid NOMODIFY NOAPPEND NODELETE NORMAL
=> TITLE "COST FACTORS"
343
344     m.x=hmcfe.hmcfeid
345     SET FILTER TO
346     SELECT hmcf
347     SET RELATION TO
348     SELECT hmcfe
349     SET RELATION TO
350     SELECT (m.oldfile)
351     RETURN m.x
352
353
354     *
355     *
356     *       _QKF0VRGGZ              m.Action VALID
357     *
358     *       Function Origin:
359     *
360     *       From Platform:          MS-DOS
361     *       From Screen:            HMCFE,      Record Number:    4
362     *       Variable:               m.Action
363     *       Called By:              VALID Clause
364     *       Object Type:            Push Button
365     *       Snippet Number:         1
366     *
367     *
368     *
369     FUNCTION _qkf0vrggz     &&  m.Action VALID
370     #REGION 1
371     IF m.action = 1
372        GO BOTTOM
373        SCATTER MEMVAR FIELD hmcfe,hmcfeid BLANK
374        m.hmcfeid = RECCOUNT()+1
375        SHOW GETS
376        SHOW GET action disabled
377        SHOW GET SAVE enabled
378        SHOW GET m.answr enabled
379        SHOW GET m.hmcfe enabled
380        SHOW GET m.factor disabled
381        SHOW GET m.element disabled
382        m.adding = .T.
383     ELSE
384
385        DO CASE
386
387        CASE m.action = 2
388           SHOW GETS
389           SHOW GET action disabled
390           SHOW GET SAVE enabled
391           SHOW GET m.answr enabled
392           SHOW GET m.hmcfe enabled
393           SHOW GET m.factor disabled
```

```
394         SHOW GET m.element disabled
395
396      CASE m.action = 3
397         SKIP
398         IF EOF()
399            ?? CHR( 7 )
400            WAIT "Last record" WINDOW NOWAIT
401            SKIP -1
402         ENDIF
403
404      CASE m.action = 4
405         SKIP -1
406         IF BOF()
407            ?? CHR( 7 )
408            WAIT "First record" WINDOW NOWAIT
409            SKIP
410         ENDIF
411
412      CASE m.action = 5
413         GO TOP
414
415      CASE m.action = 6
416         GO BOTTOM
417
418      CASE m.action = 7
419         CLEAR READ
420      ENDCASE
421
422      IF m.action >2 AND m.action < 7
423         SCATTER MEMVAR
424         IF m.hmcfid>0
425            m.answr=get_hmcf(m.hmcfid)
426            m.oldanswr=m.answr
427         ELSE
428            m.answr=""
429            m.oldanswr=""
430         ENDIF
431         SHOW GETS
432         SHOW GET m.hmcfe disabled
433         SHOW GET m.answr disabled
434         SHOW GET m.factor enabled
435         SHOW GET m.element enabled
436      ENDIF
437   ENDIF
438
439
440   *
441   *
442   *        _QKF0VRGRP            m.factor WHEN
443   *
444   *        Function Origin:
445   *
446   *        From Platform:        MS-DOS
447   *        From Screen:          HMCFE,     Record Number:    5
448   *        Variable:             m.factor
449   *        Called By:            WHEN Clause
450   *        Snippet Number:       2
451   *
452   *
453   *
454   FUNCTION _qkf0vrgrp     &&  m.factor WHEN
455   #REGION 1
456   m.factor=0
457
458
459   *
460   *
461   *        _QKF0VRGTP            m.factor VALID
462   *
463   *        Function Origin:
464   *
465   *        From Platform:        MS-DOS
466   *        From Screen:          HMCFE,     Record Number:    5
467   *        Variable:             m.factor
468   *        Called By:            VALID Clause
469   *        Snippet Number:       3
470   *
471   *
472   *
473   FUNCTION _qkf0vrgtp     &&  m.factor VALID
474   #REGION 1
475   m.factor=rel(0)
476   IF m.factor>0
477      LOCATE FOR hmcfeid = m.factor
478      IF FOUND()
479         GO RECNO()
480         SCATTER MEMVAR
481         IF m.hmcfid>0
482            m.answr=get_hmcf(m.hmcfid)
483         ELSE
484            m.answr=""
485         ENDIF
486         m.oldanswr=m.answr
487         SHOW GETS
488      ENDIF
489   ENDIF
490
491
492   *
493   *
494   *        _QKF0VRGX4            m.element WHEN
495   *
496   *        Function Origin:
497   *
498   *        From Platform:        MS-DOS
499   *        From Screen:          HMCFE,     Record Number:    6
500   *        Variable:             m.element
501   *        Called By:            WHEN Clause
502   *        Snippet Number:       4
503   *
504   *
505   *
506   FUNCTION _qkf0vrgx4     &&  m.element WHEN
507   #REGION 1
508   m.element=m.hmcfid
509
510
511   *
512   *
513   *        _QKF0VRGYG            m.element VALID
514   *
515   *        Function Origin:
516   *
517   *        From Platform:        MS-DOS
518   *        From Screen:          HMCFE,     Record Number:    6
519   *        Variable:             m.element
520   *        Called By:            VALID Clause
521   *        Snippet Number:       5
522   *
523   *
524   *
525   FUNCTION _qkf0vrgyg     &&  m.element VALID
```

```
526     #REGION 1
527     m.element=rel(m.hmcfid)
528     IF m.element>0
529        LOCATE FOR hmcfeid = m.element
530        IF FOUND()
531           GO RECNO()
532           SCATTER MEMVAR
533           IF m.hmcfid>0
534              m.answr=get_hmcf(m.hmcfid)
535           ELSE
536              m.answr=""
537           ENDIF
538           m.oldanswr=m.answr
539           SHOW GETS
540        ENDIF
541     ENDIF
542
543
544     *
545     *
546     *           _QKF0VRH1G            m.answr WHEN
547     *
548     *           Function Origin:
549     *
550     *           From Platform:        MS-DOS
551     *           From Screen:          HMCFE,      Record Number:    8
552     *           Variable:             m.answr
553     *           Called By:            WHEN Clause
554     *           Object Type:          Field
555     *           Snippet Number:       6
556     *
557     *
558     *
559     FUNCTION _qkf0vrh1g     &&  m.answr WHEN
560     #REGION 1
561     m.oldanswr=m.answr
562
563
564     *
565     *
566     *           _QKF0VRH34            m.answr VALID
567     *
568     *           Function Origin:
569     *
570     *           From Platform:        MS-DOS
571     *           From Screen:          HMCFE,      Record Number:    8
572     *           Variable:             m.answr
573     *           Called By:            VALID Clause
574     *           Object Type:          Field
575     *           Snippet Number:       7
576     *
577     *
578     *
579     FUNCTION _qkf0vrh34     &&  m.answr VALID
580     #REGION 1
581     IF m.oldanswr <> m.answr
582        IF NOT EMPTY(m.answr)
583           m.answr=ALLTRIM(m.answr)
584           m.answr=IIF(m.answr="?","",UPPER(m.answr))
585           m.answr=ALLTRIM(get_cf(m.answr))
586           IF NOT EMPTY(m.answr)
587              m.hmcfid=get_cfid(m.answr)
588              m.oldanswr=m.answr
589           ENDIF
590        ENDIF
591        SHOW GETS
592     ENDIF
593
594
595     *
596     *
597     *           _QKF0VRH6D            m.Save VALID
598     *
599     *           Function Origin:
600     *
601     *           From Platform:        MS-DOS
602     *           From Screen:          HMCFE,      Record Number:    11
603     *           Variable:             m.Save
604     *           Called By:            VALID Clause
605     *           Object Type:          Push Button
606     *           Snippet Number:       8
607     *
608     *
609     *
610     FUNCTION _qkf0vrh6d     &&  m.Save VALID
611     #REGION 1
612     IF m.save = 1 && Selected Save Button
613
614        IF m.adding && Adding a new record
615           APPEND BLANK
616           GATHER MEMVAR
617        ELSE
618           IF m.change && Changing an old record
619              GATHER MEMVAR
620           ENDIF
621        ENDIF
622     ENDIF
623
624     SCATTER MEMVAR
625     IF m.hmcfid>0
626        m.answr=get_hmcf(m.hmcfid)
627     ELSE
628        m.hmcf=""
629     ENDIF
630     m.oldanswr=m.answr
631     SHOW GETS
632     SHOW GET action enabled
633     SHOW GET m.factor enabled
634     SHOW GET m.element enabled
635     SHOW GET SAVE disabled
636     SHOW GET m.hmcfe disabled
637     SHOW GET m.answr disabled
638
639     m.adding=.F.
640     m.change=.F.
641     *: EOF: HMCFE.AC1
```

```
*
*
*    10/27/93            HMCFEI.SPR            14:49:13
*
*
*
*    Author's Name
*
*    Copyright (c) 1993 Company Name
*    Address
*    City,     Zip
*
*    Description:
*    This program was automatically generated by GENSCRN.
*
*


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*
*              MS-DOS Window definitions
*
*
*

IF NOT WEXIST("hmcfei") ;
        OR UPPER(WTITLE("HMCFEI")) == "HMCFEI.PJX" ;
        OR UPPER(WTITLE("HMCFEI")) == "HMCFEI.SCX" ;
        OR UPPER(WTITLE("HMCFEI")) == "HMCFEI.MNX" ;
        OR UPPER(WTITLE("HMCFEI")) == "HMCFEI.PRG" ;
        OR UPPER(WTITLE("HMCFEI")) == "HMCFEI.FRX" ;
        OR UPPER(WTITLE("HMCFEI")) == "HMCFEI.QPR"
   DEFINE WINDOW hmcfei ;
        FROM INT((SROW()-17)/2),INT((SCOL()-69)/2) ;
        TO INT((SROW()-17)/2)+16,INT((SCOL()-69)/2)+68 ;
        NOFLOAT ;
        NOCLOSE ;
        SHADOW ;
        nominimize ;
        DOUBLE ;
        COLOR SCHEME 1
ENDIF


*
*
*         HMCFEI/MS-DOS Setup Code - SECTION 2
*
*
*

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
m.oldescape = SET( "ESCAPE" )
SET ESCAPE OFF
m.adding = .F.
m.change = .F.
m.editing = .F.
m.action=5

CLOSE ALL

IF USED("HMCFEI")
   SELECT hmcfei
   SET ORDER TO hmcfid
ELSE
   SELECT 0
   USE (LOCFILE("HMCFEI.dbf","DBF","Where is HMCFEI?"));
       AGAIN ALIAS hmcfei ;
       ORDER hmcfid
ENDIF

*USE HMCFE AGAIN
**************************************************
* Check see if the last record is defined
GO TOP
***************************************************
SCATTER MEMVAR
IF m.hmcfid>0
   m.answr=get_hmcf(m.hmcfid)
ELSE
   m.answr=""
ENDIF
IF m.hmcfeid>0
   m.hmcfe=get_hmcfe(m.hmcfeid)
ELSE
   m.hmcfe=""
ENDIF
m.oldhmcfe=m.hmcfe
m.oldanswr=m.answr
m.action = 1
SHOW GETS


*
*
*              HMCFEI/MS-DOS Screen Layout
*
*
*

#REGION 1
IF WVISIBLE("hmcfei")
   ACTIVATE WINDOW hmcfei SAME
ELSE
   ACTIVATE WINDOW hmcfei NOSHOW
ENDIF
@ 0,55 GET m.action ;
    PICTURE "@*VN \<Add;\<Edit;\<Next;\<Previous;\!\<Top;\<Bottom;B\<
=> rowse;\?E\<xit" ;
    SIZE 1,10,1 ;
    DEFAULT 1 ;
    VALID _qkf0vrk34
@ 3,2 GET m.factor ;
    PICTURE "@*IHN " ;
    SIZE 1,10,1 ;
    DEFAULT 0 ;
    WHEN _qkf0vrkgo() ;
    VALID _qkf0vrkiq()
```

```
@ 5,2 GET m.element ;
    PICTURE "@*IHN " ;
    SIZE 1,10,1 ;
    DEFAULT 0 ;
    WHEN _qkf0vrkmh() ;
    VALID _qkf0vrknv() ;
    DISABLE
@ 3,13 GET m.hmcfid ;
    SIZE 1,5 ;
    DEFAULT " " ;
    DISABLE
@ 3,20 GET m.answr ;
    SIZE 1,30 ;
    DEFAULT " " ;
    WHEN _qkf0vrks5() ;
    VALID _qkf0vrktm() ;
    DISABLE
@ 5,13 GET m.hmcfeid ;
    SIZE 1,5 ;
    DEFAULT 0 ;
    DISABLE
@ 5,20 GET m.hmcfe ;
    SIZE 1,30 ;
    DEFAULT " " ;
    PICTURE "@!" ;
    WHEN _qkf0vrkwj() ;
    VALID _qkf0vrkxz() ;
    DISABLE
@ 7,13 GET m.hmcfeiid ;
    SIZE 1,5 ;
    DEFAULT 0 ;
    DISABLE
@ 7,20 GET m.hmcfei ;
    SIZE 1,30 ;
    DEFAULT " " ;
    PICTURE "@!" ;
    DISABLE
@ 9,14 GET m.hmcfeicost ;
    SIZE 1,10 ;
    DEFAULT 0 ;
    PICTURE "@$" ;
    DISABLE
@ 12,19 GET m.save ;
    PICTURE "@*HN \<Save;\<Cancel" ;
    SIZE 1,8,1 ;
    DEFAULT 1 ;
    VALID _qkf0vrilj() ;
    DISABLE
@ 1,0 TO 14,53
@ 0,14 SAY "COST FACTOR ELEMENT ITEMS" ;
    SIZE 1,25, 0
@ 5,4 SAY "ELEMENT:" ;
    SIZE 1,8, 0
@ 3,5 SAY "FACTOR:" ;
    SIZE 1,7, 0
@ 7,7 SAY "ITEM:" ;
    SIZE 1,5, 0
@ 9,7 SAY "COST:" ;
    SIZE 1,5, 0

IF NOT WVISIBLE("hmcfei")
    ACTIVATE WINDOW hmcfei
ENDIF

READ CYCLE
```



```
RELEASE WINDOW hmcfei

#REGION 0
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF


*
*                  HMCFEI/MS-DOS Cleanup Code
*
*
*

#REGION 1
POP KEY ALL
SET ESCAPE &oldescape
********* End of Main Body - Entry Cleanup *****

**********************


*
*            HMCFEI/MS-DOS Supporting Procedures and Functions
*
*
*

#REGION 1
PROCEDURE escpressed
**********************
RETURN

**********************
PROCEDURE CHANGE
**********************
m.oldexact = SET( "EXACT" )
SET EXACT ON
m.change =(TRIM(hmcfei.hmcfei) <> TRIM( m.hmcfei);
    OR hmcfei.hmcfid <> m.hmcfid;
    OR hmcfei.hmcfeid <> m.hmcfeid;
    OR hmcfei.hmcfeino <> m.hmcfeino;
    OR hmcfei.hmcfeicost <> m.hmcfeicost)
SET EXACT &oldexact
RETURN m.change

******************
PROCEDURE get_hmcf
******************
PARAMETER m.hmcfid
m.oldfile=SELECT()
DIMENSION X[1]

SELECT DISTINCT hmcf.hmcf;
    FROM hmcf;
    WHERE hmcf.hmcfid = (m.hmcfid);
    INTO ARRAY X
IF NOT EMPTY(X[1])
    m.hmcf=X[1]
ELSE
    m.hmcf = ""
```

```
264     ENDIF
265
266     SELECT(m.oldfile)
267     RETURN m.hmcf
268     ************************
269     PROCEDURE get_cf
270     ************************
271     PARAMETER m.answr
272     DIMENSION hcf[1]
273     hcf[1]=""
274     m.oldfile=SELECT()
275
276     SELECT DISTINCT hmcf.hmcf;
277         FROM hmcf;
278         WHERE hmcf.hmcf IN (m.answr);
279         ORDER BY hmcf.hmcfid;
280         INTO ARRAY hcf
281
282     IF NOT EMPTY(hcf[1])
283
284        m.answr=chooser(@hcf,"Select a Cost Factor")
285
286     ELSE
287        =errmsg(m.answr + " was not found",1)
288        m.hmcfid=0
289        m.answr=""
290
291     ENDIF
292
293     SELECT(m.oldfile)
294     RETURN m.answr
295
296     ***************************
297     PROCEDURE get_cfid
298     ************************
299     PARAMETERS m.answr
300     IF PARAMETER()=0
301        X=0
302     ELSE
303        DIMENSION X[1]
304        m.oldfile=SELECT()
305
306        SELECT DISTINCT hmcf.hmcfid;
307            FROM hmcf;
308            WHERE hmcf.hmcf IN (m.answr);
309            INTO ARRAY X
310        IF NOT EMPTY(X[1])
311           X=X[1]
312        ELSE
313           X = 0
314        ENDIF
315        SELECT (m.oldfile)
316     ENDIF
317     RETURN X
318
319     ************************************************
320     *    COST FACTOR ELEMENTS -- CFE               *
321     ************************************************
322
323     ******************
324     PROCEDURE get_hmcfe
325     ************************
326     PARAMETER m.hmcfeid
327     DIMENSION X[1]
328     m.oldfile=SELECT()
329
330     SELECT DISTINCT hmcfe.hmcfe;
331         FROM hmcfe;
332         WHERE hmcfe.hmcfeid = (m.hmcfeid);
333         INTO ARRAY X
334     IF !EMPTY(X[1])
335        m.hmcfe=X[1]
336     ELSE
337        m.hmcfe = ""
338     ENDIF
339
340     SELECT(m.oldfile)
341     RETURN m.hmcfe
342     ************************
343     PROCEDURE get_cfe
344     ************************
345     PARAMETER m.hmcfe
346     DIMENSION hcfe[1]
347     hcfe[1]=""
348     m.oldfile=SELECT()
349
350     SELECT DISTINCT hmcfe.hmcfe;
351         FROM hmcfe;
352         WHERE hmcfe.hmcfe IN (m.hmcfe);
353         ORDER BY hmcfe.hmcfeid;
354         INTO ARRAY hcfe
355
356     IF NOT EMPTY(hcfe[1])
357
358        m.hmcfe=chooser(@hcfe,"Select a Cost Factor")
359
360     ELSE
361        =errmsg(m.hmcfe + " was not found",1)
362        m.hmcfeid=0
363        m.hmcfe=""
364
365     ENDIF
366
367     SELECT(m.oldfile)
368     RETURN m.hmcfe
369
370     ***************************
371     PROCEDURE get_cfeid
372     ************************
373     PARAMETERS m.hmcfe
374     IF PARAMETER()=0
375        X=0
376     ELSE
377        DIMENSION X[1]
378        m.oldfile=SELECT()
379
380        SELECT DISTINCT hmcfe.hmcfeid;
381            FROM hmcfe;
382            WHERE hmcfe.hmcfe IN (m.hmcfe);
383            INTO ARRAY X
384        IF !EMPTY(X[1])
385           X=X[1]
386        ELSE
387           X = 0
388        ENDIF
389        SELECT (m.oldfile)
390     ENDIF
391     RETURN X
392
393     ************************
394     PROCEDURE rel
395     * DISPLAY RELATIONSHIP
```

```
396     * OF HMCFE AND HMCF
397     ***************************
398     PARAMETER m.id
399
400     IF PARAMETER()=0
401        m.id=0
402     ENDIF
403
404     m.oldfile=SELECT()
405     SET TALK OFF
406
407     &&OPEN FILE # 1
408
409     IF USED("HMCFEI")
410        SELECT hmcfei
411        SET ORDER TO TAG hmcfeid OF hmcfei.cdx
412
413     ELSE
414
415        SELECT 0
416        USE hmcfei ALIAS hmcfei AGAIN
417        SET ORDER TO TAG hmcfeid OF hmcfei.cdx
418
419     ENDIF
420
421     &&OPEN FILE # 2
422
423     IF USED("HMCFE")
424        SELECT hmcfe
425        SET ORDER TO TAG hmcfid OF hmcfe.cdx
426
427     ELSE
428
429        SELECT 0
430        USE hmcfe ALIAS hmcfe AGAIN
431        SET ORDER TO TAG hmcfid OF hmcfe.cdx
432
433     ENDIF
434
435     && OPEN FILE #3
436
437     IF USED("HMCF")
438        SELECT hmcf
439        SET ORDER TO TAG hmcfid OF hmcf.cdx
440
441     ELSE
442
443        SELECT 0
444        USE hmcf ALIAS hmcf AGAIN
445        SET ORDER TO hmcfid OF hmcf.cdx
446     ENDIF
447
448     IF m.id>0
449
450        SET FILTER TO hmcfid = m.id
451
452     ENDIF
453
454
455     SELECT hmcfe
456     SET RELATION TO hmcfeid INTO hmcfei && ADDITIVE
457
458     SELECT hmcf
459     SET RELATION TO hmcfid INTO hmcfe ADDITIVE
460     SET SKIP TO hmcfei, hmcfe
461
462
463        ** Show fields from grandparent (HMCFEI), parent (HMCFE) and child (H
=> MCF)**
464        BROWSE FIELDS hmcf.hmcf:H="FACTORS", hmcfe.hmcfe:H="ELEMENTS",;
465            hmcfei.hmcfei:H="ITEMS", hmcfei.hmcfeicost:H="COSTS";
466            FOR hmcf.hmcfid = hmcfe.hmcfid NOMODIFY NOAPPEND NODELETE TITLE "
=> COST FACTORS";
467            NORMAL
468
469        m.x=hmcfe.hmcfeid
470
471        SET FILTER TO
472        SELECT hmcf
473        SET RELATION TO
474        SELECT hmcfe
475        SET RELATION TO
476        SELECT(m.oldfile)
477        RETURN m.x
478        ***************************
479        PROCEDURE browseitem
480        * DISPLAY RELATIONSHIP
481        * OF HMCFE AND HMCF
482        ***************************
483        PARAMETER m.id
484
485        IF PARAMETER()=0
486           m.id=0
487        ENDIF
488
489        m.oldfile=SELECT()
490        SET TALK OFF
491
492        &&OPEN FILE # 1
493
494        IF USED("HMCFEI")
495           SELECT hmcfei
496           SET ORDER TO TAG hmcfeid OF hmcfei.cdx
497
498        ELSE
499
500           SELECT 0
501           USE hmcfei ALIAS hmcfei AGAIN
502           SET ORDER TO TAG hmcfeid OF hmcfei.cdx
503
504        ENDIF
505
506        &&OPEN FILE # 2
507
508        IF USED("HMCFE")
509           SELECT hmcfe
510           SET ORDER TO TAG hmcfid OF hmcfe.cdx
511
512        ELSE
513
514           SELECT 0
515           USE hmcfe ALIAS hmcfe AGAIN
516           SET ORDER TO TAG hmcfid OF hmcfe.cdx
517
518        ENDIF
519
520        && OPEN FILE #3
521
522        IF USED("HMCF")
523           SELECT hmcf
524           SET ORDER TO TAG hmcfid OF hmcf.cdx
525
```

```
526     ELSE
527
528        SELECT 0
529        USE hmcf ALIAS hmcf AGAIN
530        SET ORDER TO hmcfid OF hmcf.cdx
531     ENDIF
532
533     IF m.id>0
534
535        SET FILTER TO hmcfid = m.id
536
537     ENDIF
538
539
540     SELECT hmcfe
541     SET RELATION TO hmcfeid INTO hmcfei && ADDITIVE
542
543     SELECT hmcf
544     SET RELATION TO hmcfid INTO hmcfe ADDITIVE
545     SET SKIP TO hmcfei, hmcfe
546
547
548     ** Show fields from grandparent (HMCFEI), parent (HMCFE) and child (H
=> MCF)**
549     BROWSE FIELDS hmcf.hmcf:H="FACTORS", hmcfe.hmcfe:H="ELEMENTS",;
550         hmcfei.hmcfei:H="ITEMS", hmcfei.hmcfeicost:H="COSTS";
551         FOR !EMPTY(hmcfei.hmcfeiid) NOMODIFY NOAPPEND NODELETE TITLE "COS
=> T FACTORS";
552         NORMAL
553
554     m.x=hmcfei.hmcfeiid
555
556     SET FILTER TO
557     SET RELATION TO
558     SELECT(m.oldfile)
559     RETURN m.x
560
561
562     *
563     *
564     *        _QKF0VRK34            m.Action VALID
565     *
566     *        Function Origin:
567     *
568     *        From Platform:        MS-DOS
569     *        From Screen:          HMCFEI,     Record Number:    2
570     *        Variable:             m.Action
571     *        Called By:            VALID Clause
572     *        Object Type:          Push Button
573     *        Snippet Number:       1
574     *
575     *
576     *
577     FUNCTION _qkf0vrk34     &&  m.Action VALID
578     #REGION 1
579     IF m.action = 1
580        GO BOTTOM
581        SCATTER MEMVAR FIELD hmcfei,hmcfeicosts,hmcfeino BLANK
582        m.oldhmcfe=m.hmcfe
583        m.oldanswr=m.answr
584        m.hmcfeiid = RECCOUNT()+1
585        SHOW GETS
586        SHOW GET action disabled
587        SHOW GET SAVE enabled
588        SHOW GET m.answr enabled
589        SHOW GET m.hmcfe enabled
590        SHOW GET m.hmcfei enabled
591        SHOW GET m.hmcfeino enabled
592        SHOW GET m.hmcfeicost enabled
593        SHOW GET m.element enabled
594        m.adding = .T.
595     ELSE
596
597        DO CASE
598
599        CASE m.action = 2
600           SHOW GETS
601           SHOW GET action disabled
602           SHOW GET m.element enabled
603           SHOW GET SAVE enabled
604           SHOW GET m.answr enabled
605           SHOW GET m.hmcfe enabled
606           SHOW GET m.hmcfei enabled
607           SHOW GET m.hmcfeino enabled
608           SHOW GET m.hmcfeicost enabled
609           m.editing = .T.
610        CASE m.action = 3
611           SKIP
612           IF EOF()
613              ?? CHR( 7 )
614              WAIT "Last record" WINDOW NOWAIT
615              SKIP -1
616           ENDIF
617
618        CASE m.action = 4
619           SKIP -1
620           IF BOF()
621              ?? CHR( 7 )
622              WAIT "First record" WINDOW NOWAIT
623              SKIP
624           ENDIF
625
626        CASE m.action = 5
627           GO TOP
628
629        CASE m.action = 6
630           GO BOTTOM
631
632        CASE m.action = 7
633           m.item = browseitem(0)
634           IF m.item>0
635              LOCATE FOR hmcfeiid =m.item
636              IF FOUND()
637                 GO RECNO()
638                 SCATTER MEMVAR
639                 IF m.hmcfid>0
640                    m.answr=get_hmcf(m.hmcfid)
641                 ELSE
642                    m.answr=""
643                 ENDIF
644
645                 IF m.hmcfeid>0
646                    m.hmcfe=get_hmcfe(m.hmcfeid)
647                 ELSE
648                    m.hmcfe=""
649                 ENDIF
650                 m.oldhmcfe=m.hmcfe
651                 m.oldanswr=m.answr
652                 SHOW GETS
653              ENDIF
654           ENDIF
655
```

```
656     |  CASE m.action = 8
657     |     CLEAR READ
658     |  ENDCASE
659     |
660     |  IF m.action >2 AND m.action < 7
661     |     SCATTER MEMVAR
662     |     IF m.hmcfid>0
663     |        m.answr=get_hmcf(m.hmcfid)
664     |        m.oldanswr=m.answr
665     |     ELSE
666     |        m.answr=""
667     |        m.oldanswr=""
668     |     ENDIF
669     |
670     |     IF m.hmcfeid>0
671     |        m.hmcfe=get_hmcfe(m.hmcfeid)
672     |        m.oldhmcfe=m.hmcfe
673     |     ELSE
674     |        m.hmcfe=""
675     |        m.oldhmcfe=""
676     |     ENDIF
677     |     SHOW GETS
678     |     SHOW GET m.hmcfe disabled
679     |     SHOW GET m.answr disabled
680     |     SHOW GET m.hmcfei disabled
681     |     SHOW GET m.hmcfeino disabled
682     |     SHOW GET m.hmcfeicost disabled
683     |     SHOW GET m.factor enabled
684     |     SHOW GET m.element disabled
685     |  ENDIF
686     ENDIF
687
688
689     *
690     *
691     *        _QKFOVRKGO            m.factor WHEN
692     *
693     *        Function Origin:
694     *
695     *        From Platform:        MS-DOS
696     *        From Screen:          HMCFEI,     Record Number:    3
697     *        Variable:             m.factor
698     *        Called By:            WHEN Clause
699     *        Snippet Number:       2
700     *
701     *
702     *
703     FUNCTION _qkf0vrkgo     &&  m.factor WHEN
704     #REGION 1
705     m.factor=0
706
707
708     *
709     *
710     *        _QKFOVRKIQ            m.factor VALID
711     *
712     *        Function Origin:
713     *
714     *        From Platform:        MS-DOS
715     *        From Screen:          HMCFEI,     Record Number:    3
716     *        Variable:             m.factor
717     *        Called By:            VALID Clause
718     *        Snippet Number:       3
719     *
720     *
721     *
```



```
722     FUNCTION _qkf0vrkiq     &&  m.factor VALID
723     #REGION 1
724     m.factor=rel(0)
725     IF m.adding = .T. OR m.editing = .T.
726        IF m.factor>0
727           msel = SELECT()
728           SELECT hmcfe
729           LOCATE FOR hmcfeid =m.factor
730           IF FOUND()
731              GO RECNO()
732              SCATTER MEMVAR
733              IF m.hmcfid>0
734                 m.answr=get_hmcf(m.hmcfid)
735              ELSE
736                 m.answr=""
737              ENDIF
738
739              IF m.hmcfeid>0
740                 m.hmcfe=get_hmcfe(m.hmcfeid)
741              ELSE
742                 m.hmcfe=""
743              ENDIF
744              m.oldhmcfe=m.hmcfe
745              m.oldanswr=m.answr
746              SHOW GETS
747              SELECT (msel)
748           ENDIF
749        ENDIF
750     ENDIF
751
752
753     *
754     *
755     *        _QKFOVRKMH            m.element WHEN
756     *
757     *        Function Origin:
758     *
759     *        From Platform:        MS-DOS
760     *        From Screen:          HMCFEI,     Record Number:    4
761     *        Variable:             m.element
762     *        Called By:            WHEN Clause
763     *        Snippet Number:       4
764     *
765     *
766     *
767     FUNCTION _qkf0vrkmh     &&  m.element WHEN
768     #REGION 1
769     m.element=m.hmcfid
770
771
772     *
773     *
774     *        _QKFOVRKNV            m.element VALID
775     *
776     *        Function Origin:
777     *
778     *        From Platform:        MS-DOS
779     *        From Screen:          HMCFEI,     Record Number:    4
780     *        Variable:             m.element
781     *        Called By:            VALID Clause
782     *        Snippet Number:       5
783     *
784     *
785     *
786     FUNCTION _qkf0vrknv     &&  m.element VALID
787     #REGION 1
```

```
788     m.element=rel(m.hmcfid)
789   IF m.adding = .T. OR m.editing = .T.
790     IF m.element>0
791        msel = SELECT()
792        SELECT hmcfe
793        LOCATE FOR hmcfeid = m.element
794       IF FOUND()
795          GO RECNO()
796          SCATTER MEMVAR
797         IF m.hmcfid>0
798            m.answr=get_hmcf(m.hmcfid)
799         ELSE
800            m.answr=""
801         ENDIF
802         IF m.hmcfeid>0
803            m.hmcfe=get_hmcfe(m.hmcfeid)
804         ELSE
805            m.hmcfe=""
806         ENDIF
807          m.oldhmcfe=m.hmcfe
808          m.oldanswr=m.answr
809          SELECT (msel)
810          SHOW GETS
811       ENDIF
812     ENDIF
813   ENDIF
814
815
816     *
817     *
818     *          _QKF0VRKS5            m.answr WHEN
819     *
820     *          Function Origin:
821     *
822     *          From Platform:        MS-DOS
823     *          From Screen:          HMCFEI,     Record Number:    6
824     *          Variable:             m.answr
825     *          Called By:            WHEN Clause
826     *          Object Type:          Field
827     *          Snippet Number:       6
828     *
829     *
830     *
831     FUNCTION _qkf0vrks5     &&  m.answr WHEN
832     #REGION 1
833     m.oldanswr=m.answr
834
835
836     *
837     *
838     *          _QKF0VRKTM            m.answr VALID
839     *
840     *          Function Origin:
841     *
842     *          From Platform:        MS-DOS
843     *          From Screen:          HMCFEI,     Record Number:    6
844     *          Variable:             m.answr
845     *          Called By:            VALID Clause
846     *          Object Type:          Field
847     *          Snippet Number:       7
848     *
849     *
850     *
851     FUNCTION _qkf0vrktm     &&  m.answr VALID
852     #REGION 1
853   IF m.oldanswr <> m.answr
```



```
854     IF NOT EMPTY(m.answr)
855        m.answr=ALLTRIM(m.answr)
856        m.answr=IIF(m.answr="?","",UPPER(m.answr))
857        m.answr=ALLTRIM(get_cf(m.answr))
858       IF NOT EMPTY(m.answr)
859          m.hmcfid=get_cfid(m.answr)
860          m.oldanswr=m.answr
861       ENDIF
862     ENDIF
863     SHOW GETS
864   ENDIF
865
866
867     *
868     *
869     *          _QKF0VRKWJ            m.hmcfe WHEN
870     *
871     *          Function Origin:
872     *
873     *          From Platform:        MS-DOS
874     *          From Screen:          HMCFEI,     Record Number:    8
875     *          Variable:             m.hmcfe
876     *          Called By:            WHEN Clause
877     *          Object Type:          Field
878     *          Snippet Number:       8
879     *
880     *
881     *
882     FUNCTION _qkf0vrkwj     &&  m.hmcfe WHEN
883     #REGION 1
884     m.oldhmcfe=m.hmcfe
885
886
887     *
888     *
889     *          _QKF0VRKXZ            m.hmcfe VALID
890     *
891     *          Function Origin:
892     *
893     *          From Platform:        MS-DOS
894     *          From Screen:          HMCFEI,     Record Number:    8
895     *          Variable:             m.hmcfe
896     *          Called By:            VALID Clause
897     *          Object Type:          Field
898     *          Snippet Number:       9
899     *
900     *
901     *
902     FUNCTION _qkf0vrkxz     &&  m.hmcfe VALID
903     #REGION 1
904   IF m.oldhmcfe <> m.hmcfe
905     IF NOT EMPTY(m.hmcfe)
906        m.hmcfe=ALLTRIM(m.hmcfe)
907        m.hmcfe=IIF(m.hmcfe="?","",UPPER(m.hmcfe))
908        m.hmcfe=ALLTRIM(get_cfe(m.hmcfe))
909       IF NOT EMPTY(m.hmcfe)
910          m.hmcfeid=get_cfeid(m.hmcfe)
911          m.oldhmcfe=m.hmcfe
912       ENDIF
913     ENDIF
914     SHOW GETS
915   ENDIF
916
917
918     *
919     *
```

```
920     *          _QKF0VRL1J           m.Save VALID
921     *
922     *          Function Origin:
923     *
924     *          From Platform:       MS-DOS
925     *          From Screen:         HMCFEI,     Record Number:   12
926     *          Variable:            m.Save
927     *          Called By:           VALID Clause
928     *          Object Type:         Push Button
929     *          Snippet Number:      10
930     *
931     *
932     *
933     * did the record change
934     FUNCTION _qkf0vrl1j     &&  m.Save VALID
935     #REGION 1
936     DO CHANGE
937
938     IF m.save = 1 && Selected Save Button
939     |
940     |  IF m.adding && Adding a new record
941     |     APPEND BLANK
942     |     GATHER MEMVAR
943     |  ENDIF
944     |
945     |  IF m.change && Changing an old record
946     |     GATHER MEMVAR
947     |  ENDIF
948     ENDIF
949
950     SCATTER MEMVAR
951     IF m.hmcfid>0
952        m.answr=get_hmcf(m.hmcfid)
953     ELSE
954        m.hmcf=""
955     ENDIF
956
957     IF m.hmcfeid>0
958        m.hmcfe=get_hmcfe(m.hmcfeid)
959     ELSE
960        m.hmcfe=""
961     ENDIF
962     m.oldhmcfe=m.hmcfe
963     m.oldanswr=m.answr
964     SHOW GETS
965     SHOW GET action enabled
966     SHOW GET m.factor enabled
967     SHOW GET m.element disabled
968     SHOW GET SAVE disabled
969     SHOW GET m.hmcfe disabled
970     SHOW GET m.hmcfei disabled
971     SHOW GET m.hmcfeino disabled
972     SHOW GET m.hmcfeicost disabled
973     SHOW GET m.answr disabled
974
975     m.adding=.F.
976     m.change=.F.
977     m.editing = .F.
978     *: EOF: HMCFEI.AC1
```

```
*
*     ┌──────────────────────────────────────────────────────┐
*     │ 12/01/93            HMCOMP.SPR              09:52:44 │
*     │                                                      │
*     │ Author's Name                                        │
*     │                                                      │
*     │ Copyright (c) 1993 Company Name                      │
*     │ Address                                              │
*     │ City,     Zip                                        │
*     │                                                      │
*     │ Description:                                         │
*     │ This program was automatically generated by GENSCRN. │
*     └──────────────────────────────────────────────────────┘
*

PARAMETERS chmscid, chmscname

*
*     ┌──────────────────────────────────────────────────────┐
*     │       HMCOMP/MS-DOS Setup Code - SECTION 1           │
*     └──────────────────────────────────────────────────────┘
*

#REGION 1

IF PARAMETER() = 0
<--  RETURN
ENDIF


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*     ┌──────────────────────────────────────────────────────┐
*     │              MS-DOS Window definitions               │
*     └──────────────────────────────────────────────────────┘
*

IF NOT WEXIST("w_hmcost") ;
       OR UPPER(WTITLE("W_HMCOST")) == "W_HMCOST.PJX" ;
       OR UPPER(WTITLE("W_HMCOST")) == "W_HMCOST.SCX" ;
       OR UPPER(WTITLE("W_HMCOST")) == "W_HMCOST.MNX" ;
       OR UPPER(WTITLE("W_HMCOST")) == "W_HMCOST.PRG" ;
       OR UPPER(WTITLE("W_HMCOST")) == "W_HMCOST.FRX" ;
       OR UPPER(WTITLE("W_HMCOST")) == "W_HMCOST.QPR"
   DEFINE WINDOW w_hmcost ;
       FROM INT((SROW()-22)/2),INT((SCOL()-76)/2) ;
       TO INT((SROW()-22)/2)+21,INT((SCOL()-76)/2)+75 ;
       TITLE "COMPUTE COST VALUE" ;
       NOFLOAT ;
       NOCLOSE ;
       SHADOW ;
       nominimize ;
       DOUBLE ;
       COLOR SCHEME 1
ENDIF


*
*     ┌──────────────────────────────────────────────────────┐
*     │       HMCOMP/MS-DOS Setup Code - SECTION 2           │
*     └──────────────────────────────────────────────────────┘
*

#REGION 1
molderr = ON("ERROR")
msafe = SET("safety")
SET SAFETY OFF
ON ERROR
DO OPEN
DECLARE steparray[1]
m.step = 1

DECLARE hmatarray[1]
m.hmat = 1

DECLARE factarray[1]
m.fact = 1

DECLARE phasearray[1]
m.phase = 1

newline = CHR(10) + CHR(13)
m.finalstr = ""
m.sample = 200
STORE 1 TO m.bbstep, m.bbhmat, m.bbfact, m.bbwfact, m.bbwstep, m.bbph
=> ase
SELECT hmtab
COPY STRUC TO hmcomp
USE
SELECT 0
USE hmcomp
DO get_hmarray


*
*     ┌──────────────────────────────────────────────────────┐
*     │          HMCOMP/MS-DOS Screen Layout                 │
*     └──────────────────────────────────────────────────────┘
*

#REGION 1
IF WVISIBLE("w_hmcost")
   ACTIVATE WINDOW w_hmcost SAME
ELSE
   ACTIVATE WINDOW w_hmcost NOSHOW
ENDIF
@ 12,40 SAY "at step:" ;
   SIZE 1,8, 0
@ 6,8 SAY "for Step:" ;
   SIZE 1,9, 0
@ 9,8 SAY "for Factor:" ;
   SIZE 1,11, 0
@ 12,8 SAY "for Factor:" ;
   SIZE 1,11, 0
@ 15,8 SAY "for Phase:" ;
```

```
132         SIZE 1,10, 0
133      a 1,25 SAY "Scenario:" ;
134         SIZE 1,9, 0
135      a 17,21 SAY "Total of Scenario cost:" ;
136         SIZE 1,23, 0
137      a 16,1 TO 19,19
138      a 0,1 SAY "Estimated" ;
139         SIZE 1,9, 0
140      a 1,2 SAY "Cost" ;
141         SIZE 1,4, 0
142      a 0,64 SAY "Estimated" ;
143         SIZE 1,9, 0
144      a 1,65 SAY "Variance" ;
145         SIZE 1,8, 0
146      a 17,3 SAY "# of Iterations" ;
147         SIZE 1,15, 0
148      a 1,34 GET m.chmscid ;
149         SIZE 1,3 ;
150         DEFAULT 0 ;
151         PICTURE "@Z" ;
152         DISABLE
153      a 1,38 GET m.chmscname ;
154         SIZE 1,15 ;
155         DEFAULT " " ;
156         PICTURE "@!T" ;
157         DISABLE
158      a 3,8 SAY "for Product:" ;
159         SIZE 1,12, 0
160      a 2,21 GET m.hmat ;
161         PICTURE "@^" ;
162         FROM hmatarray ;
163         SIZE 3,42 ;
164         DEFAULT 1 ;
165         COLOR SCHEME 1, 2
166      a 5,21 GET m.step ;
167         PICTURE "@^" ;
168         FROM steparray ;
169         SIZE 3,42 ;
170         DEFAULT 1 ;
171         COLOR SCHEME 1, 2
172      a 6,2 GET m.cstep ;
173         PICTURE "@*C  " ;
174         SIZE 1,5 ;
175         DEFAULT 0
176      a 6,67 GET m.bstep ;
177         PICTURE "@*C  " ;
178         SIZE 1,5 ;
179         DEFAULT 0 ;
180         VALID _qle0l6b8t()
181      a 8,21 GET m.fact ;
182         PICTURE "@^" ;
183         FROM factarray ;
184         SIZE 3,42 ;
185         DEFAULT 1 ;
186         COLOR SCHEME 1, 2
187      a 9,2 GET m.cfact ;
188         PICTURE "@*C  " ;
189         SIZE 1,5 ;
190         DEFAULT 0
191      a 9,67 GET m.bfact ;
192         PICTURE "@*C  " ;
193         SIZE 1,5 ;
194         DEFAULT 0 ;
195         VALID _qle0l6biv()
196      a 11,21 GET m.wfact ;
197         PICTURE "@^" ;
198         FROM factarray ;
199         SIZE 3,19 ;
200         DEFAULT 1 ;
201         COLOR SCHEME 1, 2
202      a 11,48 GET m.wstep ;
203         PICTURE "@^" ;
204         FROM steparray ;
205         SIZE 3,15 ;
206         DEFAULT 1 ;
207         COLOR SCHEME 1, 2
208      a 12,2 GET m.cwstep ;
209         PICTURE "@*C  " ;
210         SIZE 1,5 ;
211         DEFAULT 0
212      a 12,67 GET m.bwstep ;
213         PICTURE "@*C  " ;
214         SIZE 1,5 ;
215         DEFAULT 0 ;
216         VALID _qle0l6br7()
217      a 14,21 GET m.phase ;
218         PICTURE "@^" ;
219         FROM phasearray ;
220         SIZE 3,42 ;
221         DEFAULT 1 ;
222         COLOR SCHEME 1, 2
223      a 15,2 GET m.cphase ;
224         PICTURE "@*C  " ;
225         SIZE 1,5 ;
226         DEFAULT 0
227      a 15,67 GET m.bphase ;
228         PICTURE "@*C  " ;
229         SIZE 1,5 ;
230         DEFAULT 0 ;
231         VALID _qle0l6byn()
232      a 17,47 GET m.scost ;
233         PICTURE "@*RHN Yes   ;No" ;
234         SIZE 1,10,0 ;
235         DEFAULT 2
236      a 18,3 GET m.sample ;
237         SIZE 1,12 ;
238         DEFAULT 0 ;
239         PICTURE "@Z" ;
240         VALID _qle0l6c5n()
241      a 19,24 GET m.action ;
242         PICTURE "@*HN \<Ok;\<Browse;\<Cancel" ;
243         SIZE 1,8,3 ;
244         DEFAULT 1 ;
245         VALID _qle0l6cb0()
246
247      IF NOT WVISIBLE("w_hmcost")
248         ACTIVATE WINDOW w_hmcost
249      ENDIF
250
251      READ CYCLE MODAL
252
253      RELEASE WINDOW w_hmcost
254
255      #REGION 0
256      IF m.talkstat = "ON"
257         SET TALK ON
258      ENDIF
259      IF m.compstat = "ON"
260         SET COMPATIBLE ON
261      ENDIF
262
263
```

```
*
*                    HMCOMP/MS-DOS Cleanup Code
*
*

#REGION 1
=delfile("hmcomp")
=delfile("hmtemp")
=delfile("test")
=delfile("temp")
SET SAFETY &msafe
ON ERROR &molderr
RETURN
**********************


*
*                HMCOMP/MS-DOS Supporting Procedures and Functions
*
*

#REGION 1
PROCEDURE OPEN
**********************

filesuccess = openfile("hmstep")
DO CANCEL WITH filesuccess

filesuccess = openfile("hmtab")
DO CANCEL WITH filesuccess

filesuccess = openfile("hmat")
DO CANCEL WITH filesuccess
SET ORDER TO TAG hmatid

filesuccess = openfile("hmlc")
DO CANCEL WITH filesuccess
SET ORDER TO TAG hmlcid

filesuccess = openfile("hmwp")
DO CANCEL WITH filesuccess
SET ORDER TO TAG hmwpid

filesuccess = openfile("hmcf")
DO CANCEL WITH filesuccess

RETURN

*****************************
FUNCTION delfile
*****************************
PARAMETER FILE
IF USED(FILE)
   SELECT (FILE)
   USE
   DELETE FILE (FILE + ".dbf")
ENDIF
RETURN

**********************
PROCEDURE CANCEL
**********************
PARAMETER success
IF !m.success
   =errmsg("File doesn's Exist !!!!", 2)
<--CANCEL
ENDIF
RETURN

**********************
PROCEDURE get_hmarray
**********************
PRIVATE msel
msel = SELECT()
SELECT *;
    FROM hmstep;
    WHERE hmscid = chmscid;
    INTO TABLE hmtemp

SELECT hmtemp
SET RELATION TO hmlcid INTO hmlc
IF RECCOUNT() > 0
   DECLARE steparray[RECCOUNT(), 2]
   i = 0
   j = 0
   SCAN
      i = i + 1
      steparray[i,1] = STR(hmstep)
      steparray[i,2] = RECNO()
      DO get_table WITH hmcomid, hmlcid, hmwpid
      m.lcphase = ALLTRIM(hmlc.hmlc)
      IF ASCAN(phasearray, lcphase) = 0
         j = j + 1
         DECLARE phasearray[j,2]
         phasearray[j,1] = ALLTRIM(hmlc.hmlc)
         phasearray[j,2] = hmlcid
      ENDIF
   ENDSCAN
   IF i > 1
      DECLARE steparray[i+1,2]
      =AINS(steparray,1)
      steparray[1,1] = "All Steps"
      steparray[1,2] = 0
   ENDIF

   IF j > 1
      DECLARE phasearray[j + 1,2]
      =AINS(phasearray,1)
      phasearray[1,1] = "All phases"
      phasearray[1,2] = 0
   ENDIF
   SET RELATION TO           && close relationship

   SELECT hmcomp
   IF RECCOUNT() > 0
      *  Creat array for factor
      SET RELATION TO hmcfid INTO hmcf
      i = 0
      SCAN
         factor = ALLTRIM(hmcf.hmcf)
         IF ASCAN(factarray, factor) = 0
            i = i + 1
            DECLARE factarray[i, 2]
            factarray[i,1] = ALLTRIM(hmcf.hmcf)
            factarray[i,2] = hmcfid
         ENDIF
      ENDSCAN
      ASORT = (factarray)
```

```
396         IF i > 1
397            DECLARE factarray[i+1,2]
398            =AINS(factarray,1)
399            factarray[1,1] = "All factors"
400            factarray[1,2] = 0
401         ENDIF
402
403           * Create array for material
404
405           SET RELATION TO hmatid INTO hmat
406           i = 0
407         SCAN
408            material = ALLTRIM(hmat.hmatname) + IIF(EMPTY(hmat.mfg),"","
=>  [" + ALLTRIM(hmat.mfg) + "] ")
409            IF ASCAN(hmatarray, material) = 0
410               i = i + 1
411               DECLARE hmatarray[i, 2]
412               hmatarray[i,1] = ALLTRIM(hmat.hmatname) + IIF(EMPTY(hmat.
=> mfg),"","  [" + ALLTRIM(hmat.mfg) + "] ")
413               hmatarray[i,2] = hmatid
414            ENDIF
415         ENDSCAN
416           ASORT = (hmatarray)
417         IF i > 1
418            DECLARE hmatarray[i+1,2]
419            =AINS(hmatarray,1)
420            hmatarray[1,1] = "All products"
421            hmatarray[1,2] = 0
422         ENDIF
423      ENDIF
424   ENDIF
425   SELECT (msel)
426   RETURN
427
428   **********************
429   PROCEDURE get_table
430   **********************
431   PARAMETER comid, lcid, wpid
432
433   PRIVATE msel
434   msel = SELECT()
435   SELECT *;
436       FROM hmtab;
437       WHERE hmcomid = m.comid AND hmlcid = m.lcid AND hmwpid = m.wpid;
438       INTO TABLE temp
439
440   SELECT hmcomp
441   APPEND FROM temp
442   SELECT (msel)
443
444   RETURN
445   **********************
446   PROCEDURE compute
447   **********************
448   PUBLIC asize
449   IF hmatarray[m.hmat,2] = 0
450      asize = ALEN(hmatarray,1)
451      FOR INDEX = 2 TO asize
452         = subcomput(ALLTRIM(hmatarray[index,1]), hmatarray[index,2])
453      ENDFOR
454   ELSE
455      = subcomput(ALLTRIM(hmatarray[m.hmat,1]), hmatarray[m.hmat,2])
456   ENDIF
457   DO DISPLAY WITH m.finalstr
458   RETURN
459
460   **********************
461   PROCEDURE subcomput
462   **********************
463   PARAMETER m.ihmatname, m.ihmatid
464
465   PRIVATE msel
466   msel = SELECT()
467   scentotal = 0
468   DIMENSION steptotal[1], hmattotal[1], factotal[1], wfactotal[1], phas
=> etotal[1]
469
470   IF m.cstep = 1
471      IF steparray[m.step,2] = 0
472         SELECT hmtemp
473         GO TOP
474         DIMENSION steptotal[RECCOUNT()]
475         i = 0
476         SCAN
477            i = i + 1
478            m.stotal = computstep(hmtemp.hmcomid, m.ihmatid, hmtemp.hmlc
=> id, hmtemp.hmwpid, hmtemp.pernum, hmtemp.durnum, hmtemp.qtynum)
479            steptotal[i] = ALLTRIM(STR(hmstep)) + u + ALLTRIM(STR(m.stot
=> al,8,2))
480            scentotal = scentotal + m.stotal
481         ENDSCAN
482      ELSE
483         DIMENSION steptotal[1]
484         SELECT hmtemp
485         GO steparray[m.step,2]
486         m.stotal = computstep(hmtemp.hmcomid, m.ihmatid, hmtemp.hmlcid,
=>  hmtemp.hmwpid, hmtemp.pernum, hmtemp.durnum, hmtemp.qtynum)
487         steptotal[1] = ALLTRIM(STR(hmtemp.hmstep)) + u + ALLTRIM(STR(m.
=> stotal,8,2))
488      ENDIF
489   ENDIF
490
491   IF m.scost = 1
492      IF EMPTY(scentotal)
493         SELECT hmtemp
494         GO TOP
495         SCAN
496            SCATTER MEMVAR
497            m.stotal = computstep(m.hmcomid, m.ihmatid, m.hmlcid, m.hmwp
=> id, m.pernum, m.durnum, m.qtynum)
498            scentotal = scentotal + m.stotal
499         ENDSCAN
500      ENDIF
501   ENDIF
502
503
504   IF m.cfact = 1
505      IF factarray[m.fact,2] = 0
506         SIZE = ALEN(factarray,1)
507         DIMENSION factotal[SIZE - 1]
508         FOR j = 2 TO SIZE
509            m.stotal = computfact(factarray[j,2], m.ihmatid)
510            factotal[j - 1] = ALLTRIM(factarray[j,1]) + u  + ALLTRIM(STR
=> (m.stotal,8,2))
511         ENDFOR
512      ELSE
513         DIMENSION factotal[1]
514         m.stotal = computfact(factarray[m.fact,2], m.ihmatid)
515         factotal = ALLTRIM(factarray[m.fact,1]) + u + ALLTRIM(STR(m.sto
=> tal,8,2))
516      ENDIF
517   ENDIF
```

```
518
519     IF m.cwstep = 1
520        IF factarray[m.wfact,2] = 0
521           SIZE = ALEN(factarray,1)
522           DIMENSION wfactotal[SIZE - 1]
523           FOR j = 2 TO SIZE
524              m.result = computwfact(factarray[j,2], m.ihmatid)
525              wfactotal[j - 1] = ALLTRIM(factarray[j,1]) + u  + ALLTRIM(m.
=> result)
526           ENDFOR
527        ELSE
528           DIMENSION wfactotal[1]
529           m.result = computwfact(factarray[m.wfact,2], m.ihmatid)
530           wfactotal[1] = ALLTRIM(factarray[m.wfact,1]) + u + ALLTRIM(m.re
=> sult)
531        ENDIF
532     ENDIF
533
534     IF m.cphase = 1
535        IF phasearray[m.phase,2] = 0
536           SIZE = ALEN(phasearray,1)
537           DIMENSION phasetotal[SIZE-1]
538           FOR j = 2 TO SIZE
539              m.stotal = computphase(phasearray[j,2], m.ihmatid)
540              phasetotal[j - 1] = ALLTRIM(phasearray[j,1]) + u  + ALLTRIM(
=> STR(m.stotal,8,2))
541           ENDFOR
542        ELSE
543           DIMENSION phasetotal[1]
544           m.stotal = computphase(phasearray[m.phase,2], m.ihmatid)
545           phasetotal[1] = ALLTRIM(phasearray[m.phase,1]) + u + ALLTRIM(ST
=> R(m.stotal,8,2))
546        ENDIF
547     ENDIF
548       bootstrap = ""
549       nl = CHR(10) + CHR(13)
550     DO CASE
551     CASE m.bstep = 1
552        SELECT hmtemp
553        GO steparray[m.bbstep,2]
554        m.btotal = bstep(m.ihmatid, hmtemp.hmlcid, hmtemp.hmwpid, hmtemp.p
=> ernum, hmtemp.durnum, hmtemp.qtynum)
555        bootstrap = "The bootstrap for step "+ ALLTRIM(STR(hmtemp.hmstep))
=>  + ": " + + nl + SPACE(5) + "Standard deviation  -  " + dp(m.btotal,u,1) +
=> SPACE(10) + "Mean - " + dp(m.btotal, u,2)
556
557     CASE m.bfact = 1
558        m.btotal = bfact(factarray[m.bbfact,2], m.ihmatid)
559        bootstrap = "The bootstrap for factor "+ ALLTRIM(factarray[m.bbfac
=> t,1]) + ": "+ nl + SPACE(5) + "Standard deviation  -  " + dp(m.btotal,u,1)
=> + SPACE(10) + "Mean - " + dp(m.btotal, u,2)
560
561     CASE m.bwstep = 1
562        m.btotal = bwstep(factarray[m.bbwfact,2], steparray[m.bbwstep,2],
=> m.ihmatid)
563        bootstrap = "The bootstrap for factor " + ALLTRIM(factarray[m.bbwf
=> act,1]) + " within step " + ALLTRIM(steparray[m.bbwstep,1]) + ": " + nl + S
=> PACE(5) + "Standard deviation  -  " + dp(m.btotal,u,1) + SPACE(10) + "Mean
=> - " + dp(m.btotal, u,2)
564
565     CASE m.bphase = 1
566        m.btotal = bphase(phasearray[m.bbphase,2], m.ihmatid)
567        bootstrap = "The bootstrap for phase "+ ALLTRIM(phasearray[m.bbpha
=> se,1]) +": " + nl + SPACE(5) + "Standard deviation  -  " + dp(m.btotal,u,1)
=>  + SPACE(10) + "Mean - " + dp(m.btotal, u,2)
568     ENDCASE
```



```
569       finalstr = IIF(!EMPTY(finalstr), finalstr + newline + newline,"")
570       finalstr = finalstr + getstr(m.ihmatname, scentotal, @steptotal, @fac
=> total, @wfactotal, @phasetotal, bootstrap)
571       SELECT (msel)
572       RETURN
573
574       *********************
575       PROCEDURE hmbrowse
576       *********************
577       PRIVATE msel
578       msel = SELECT()
579       m.title = "SCENARIO: " + ALLTRIM(UPPER(chmscname)     )
580       SELECT hmtemp
581       SET RELATION TO hmcomid INTO hmcom
582       SET RELATION TO hmlcid INTO hmlc ADDITIVE
583       SET RELATION TO hmwpid INTO hmwp ADDITIVE
584       BROWSE FIELDS hmstep:4:H="STEP",hmcom.common:30:H="Material",;
585           hmlc.hmlc:30:H="Life cycle Phase",;
586           hmwp.hmwp:30:H="working process",;
587           pernum:5:H="# of Employee",;
588           durnum:5:H="Number of Time",;
589           qtynum:5:H="# of Quatity" NOMODIFY NORMAL TITLE m.title
590
591       SELECT (msel)
592       RETURN
593
594       ****************************
595       FUNCTION computstep
596       ****************************
597       PARAMETER com, hmat, lcphase, wp, sn, sd, sq
598
599       PRIVATE msel
600       msel = SELECT()
601       SELECT SUM(hmcomp.wtaverage),0000.00, hmcomp.hmetid, hmcomp.hmetprob,
=>  hmcomp.hmcfid,;
602            hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd, hmcomp.perq, hmcomp.hmu
=> nit;
603            FROM hmcomp;
604            WHERE hmcomp.hmcomid = m.com;
605            AND hmcomp.hmatid = m.hmat;
606            AND hmcomp.hmlcid = m.lcphase;
607            AND hmcomp.hmwpid = m.wp;
608            GROUP BY hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmcfeid;
609            ORDER BY hmcomp.hmetid;
610            INTO TABLE test
611       m.sum = CALCULATE(sn, sd, sq)
612
613       SELECT (msel)
614       RETURN m.sum
615
616       ****************************
617       FUNCTION factname
618       ****************************
619       PARAMETER mfactid
620       PRIVATE j
621       mname = ""
622       j = ASCAN(factarray, mfactid)
623       * FOR j = 1 to alen(factarray,1)
624       mname = ALLTRIM(factarray[j - 1])
625       RETURN mname
626
627       ****************************
628       FUNCTION computfact
629       ****************************
630       PARAMETER m.factid, m.hmatid
631       PRIVATE msel
```

```
632     IF PARAMETER() = 0
633   <-----RETURN 0
634     ENDIF
635       msel = SELECT()
636       SELECT hmtemp
637       GO TOP
638       m.sum = 0
639     SCAN
640         m.wpid = hmtemp.hmwpid
641         m.comid = hmtemp.hmcomid
642         m.lcid = hmtemp.hmlcid
643         SELECT SUM(hmcomp.wtaverage),0000.00, hmcomp.hmetid, hmcomp.hmetpr
=> ob,;
644                hmcomp.hmcfid,        hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd
=> , hmcomp.perq,;
645                hmcomp.hmunit;
646                FROM hmcomp;
647                WHERE hmcomp.hmcomid = m.comid;
648                AND hmcomp.hmatid = m.hmatid;
649                AND hmcomp.hmlcid = m.lcid;
650                AND hmcomp.hmwpid = m.wpid;
651                AND hmcomp.hmcfid = m.factid;
652                GROUP BY hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmcfeid;
653                ORDER BY hmcomp.hmetid;
654                INTO TABLE test
655         m.sum = m.sum + CALCULATE(hmtemp.pernum, hmtemp.durnum, hmtemp.qty
=> num)
656     ENDSCAN
657       SELECT (msel)
658       RETURN m.sum
659
660       ****************************
661       FUNCTION computwfact
662       ****************************
663       PARAMETER m.factid, m.hmatid
664       PRIVATE msel
665     IF PARAMETER() = 0
666   <-----RETURN 0
667     ENDIF
668       msel = SELECT()
669     IF steparray[m.wstep,2] = 0
670         SELECT hmtemp
671         m.resul = ""
672         GO TOP
673       SCAN
674           m.comid = hmtemp.hmcomid
675           m.lcid = hmtemp.hmlcid
676           m.wpid = hmtemp.hmwpid
677           SELECT SUM(hmcomp.wtaverage),0000.00, hmcomp.hmetid, hmcomp.hme
=> tprob, hmcomp.hmcfid,;
678                  hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd, hmcomp.perq, hmco
=> mp.hmunit;
679                  FROM hmcomp;
680                  WHERE hmcomp.hmcomid = m.comid;
681                  AND hmcomp.hmatid = m.hmatid;
682                  AND hmcomp.hmlcid = m.lcid;
683                  AND hmcomp.hmwpid = m.wpid;
684                  AND hmcomp.hmcfid = m.factid;
685                  GROUP BY hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmcfeid;
686                  ORDER BY hmcomp.hmetid;
687                  INTO TABLE test
688           m.resul = IIF(EMPTY(m.resul), "", m.resul + u) + ALLTRIM(STR(hm
=> temp.hmstep)) + ":" + ALLTRIM(STR(CALCULATE(hmtemp.pernum, hmtemp.durnum, h
=> mtemp.qtynum),8,2))
689       ENDSCAN
690     ELSE
691         SELECT hmtemp
692         GO (steparray[m.wstep,2])
693         m.comid = hmtemp.hmcomid
694         m.lcid = hmtemp.hmlcid
695         m.wpid = hmtemp.hmwpid
696         SELECT SUM(hmcomp.wtaverage),0000.00, hmcomp.hmetid, hmcomp.hmetpr
=> ob, hmcomp.hmcfid,;
697                hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd, hmcomp.perq, hmcomp.
=> hmunit;
698                FROM hmcomp;
699                WHERE hmcomp.hmcomid = m.comid;
700                AND hmcomp.hmatid = m.hmatid;
701                AND hmcomp.hmlcid = m.lcid;
702                AND hmcomp.hmwpid = m.wpid;
703                AND hmcomp.hmcfid = m.factid;
704                GROUP BY hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmcfeid;
705                ORDER BY hmcomp.hmetid;
706                INTO TABLE test
707           m.resul = ALLTRIM(STR(hmtemp.hmstep)) + ":" + ALLTRIM(STR(CALCULAT
=> E(hmtemp.pernum, hmtemp.durnum, hmtemp.qtynum),8,2))
708       ENDIF
709         SELECT (msel)
710         RETURN m.resul
711
712         ****************************
713         FUNCTION computphase
714         ****************************
715         PARAMETER m.phaseid, m.hmatid
716
717         PRIVATE m.sum,msel
718       IF PARAMETER() = 0
719     <-----RETURN 0
720       ENDIF
721         msel = SELECT()
722         SELECT hmtemp
723         GO TOP
724         m.sum = 0
725       SCAN
726         IF hmlcid = m.phaseid
727             m.comid = hmtemp.hmcomid
728             m.lcid = hmtemp.hmlcid
729             m.wpid = hmtemp.hmwpid
730             SELECT SUM(hmcomp.wtaverage),0000.00, hmcomp.hmetid, hmcomp.hme
=> tprob, hmcomp.hmcfid,;
731                    hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd, hmcomp.perq, hmco
=> mp.hmunit;
732                    FROM hmcomp;
733                    WHERE hmcomp.hmcomid = m.comid;
734                    AND hmcomp.hmatid = m.hmatid;
735                    AND hmcomp.hmlcid = m.lcid;
736                    AND hmcomp.hmwpid = m.wpid;
737                    GROUP BY hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmcfeid;
738                    ORDER BY hmcomp.hmetid;
739                    INTO TABLE test
740             m.sum = m.sum + CALCULATE(hmtemp.pernum, hmtemp.durnum, hmtemp.
=> qtynum)
741         ENDIF
742       ENDSCAN
743         SELECT (msel)
744         RETURN m.sum
745
746         ****************************
747         FUNCTION CALCULATE
748         ****************************
749         PARAMETER sn, sd, sq
750         PRIVATE msel
```

```
751       msel = SELECT()
752     ┌─IF !USED("test")
753   <─┴─────RETURN 0
754     └─ENDIF
755       SELECT test
756       t = 0
757     ┌─IF RECCOUNT() > 0
758     │   ┌─SCAN
759     │   │     s = 1
760     │   │   ┌─IF perp = 1
761     │   │   │     s = s * sn
762     │   │   └─ENDIF
763     │   │   ┌─IF perd = 1
764     │   │   │     s = s * sd
765     │   │   └─ENDIF
766     │   │   ┌─IF perq = 1
767     │   │   │     s = s * sq
768     │   │   └─ENDIF
769     │   │     s = sum_wtaver * s
770     │   │     REPLACE exp_2 WITH s
771     │   └─ENDSCAN
772     └─ENDIF
773
774       SELECT SUM(exp_2), hmetprob;
775           FROM test;
776           GROUP BY hmetid;
777           INTO CURSOR EXP
778
779       SELECT EXP
780     ┌─IF RECCOUNT() > 0
781     │     t = 0
782     │   ┌─SCAN
783     │   │   ┌─IF EMPTY(hmetprob)
784     │   │   │     hmetprob = IIF(RECNO()=1,1,.01)
785     │   │   └─ENDIF
786     │   │     t = t + (sum_exp_2 * hmetprob)
787     │   └─ENDSCAN
788     └─ENDIF
789       =delfile('EXP')
790       SELECT (msel)
791       RETURN t
792
793       ****************************
794       FUNCTION resetwt
795       ****************************
796       PARAMETER matid, lcphase, wp
797       msel = SELECT()
798       SELECT test
799     ┌─SCAN
800     │     m.wt = selwt(m.matid, m.lcphase, m.wp, test.hmetid, test.hmcfid, t
=> est.hmcfeid)
801     │     REPLACE sum_wtaver WITH m.wt
802     └─ENDSCAN
803       SELECT (msel)
804       RETURN
805
806       **********************
807       FUNCTION selwt
808       **********************
809       PARAMETER matid, lcphase, wp, metid, cfid, cfeid
810       msel = SELECT()
811       SELECT hmcfecost, prob;
812           FROM hmtab;
813           WHERE hmtab.hmatid = m.matid;
814           AND hmtab.hmlcid = m.lcphase;
815           AND hmtab.hmwpid = m.wp;
816           AND hmtab.hmetid = m.metid;
817           AND hmtab.hmcfid = m.cfid;
818           AND hmtab.hmcfeid = m.cfeid;
819           INTO TABLE test1
820
821       SELECT test1
822       m.randnum = RAND()
823       m.wt = 0
824       m.start = 0
825       m.end = 0
826     ┌─SCAN
827     │     m.end = IIF(EMPTY(m.end),test1.prob, END + test1.prob)
828     │   ┌─IF m.randnum > m.start AND m.randnum < m.end
829     │   │     m.wt = test1.hmcfecost
830   v─┴───┴─────EXIT
831     │   └─ENDIF
832     │     m.start = m.start + test1.prob
833     └─ENDSCAN
834       =delfile('test1')
835       SELECT (msel)
836       RETURN m.wt
837
838       ****************************
839       FUNCTION setupboot
840       ****************************
841     ┌─DO CASE
842     ├─CASE m.bstep = 1
843     │   ┌─IF steparray[m.step,2] = 0
844     │   │     =ACOPY(steparray, temparray)
845     │   │     =ADEL(temparray,1)
846     │   │     DECLARE temparray[ALEN(temparray,1) - 1,2]
847     │   │     m.temp = bselect(@temparray,"Select step: ")
848     │   │   ┌─IF EMPTY(m.temp)
849     │   │   │     =errmsg("No Bootstrap computing...",2)
850     │   │   │     m.bstep = 0
851     │   │   ├─ELSE
852     │   │   │     m.pos = ASCAN(m.steparray,m.temp)
853     │   │   │     m.bbstep = ASUBSCRIPT(m.steparray,m.pos,1)
854     │   │   └─ENDIF
855     │   │     RELEASE temparray
856     │   ├─ELSE
857     │   │     m.bbstep = m.step
858     │   └─ENDIF
859     │
860     ├─CASE m.bfact = 1
861     │   ┌─IF factarray[m.fact,2] = 0
862     │   │     =ACOPY(factarray, temparray)
863     │   │     =ADEL(temparray,1)
864     │   │     DECLARE temparray[ALEN(temparray,1) - 1,2]
865     │   │     m.temp = bselect(@temparray,"Select factor : ")
866     │   │   ┌─IF EMPTY(m.temp)
867     │   │   │     =errmsg("No Bootstrap computing...",2)
868     │   │   │     m.bfact = 0
869     │   │   ├─ELSE
870     │   │   │     m.pos = ASCAN(m.factarray,m.temp)
871     │   │   │     m.bbfact = ASUBSCRIPT(m.factarray,m.pos,1)
872     │   │   │
873     │   │   └─ENDIF
874     │   │     RELEASE temparray
875     │   ├─ELSE
876     │   │     m.bbfact = m.fact
877     │   └─ENDIF
878     ├─CASE m.bwstep = 1
879     │   ┌─IF steparray[m.wstep,2] = 0 OR factarray[m.wfact,2] = 0
880     │   │     =ACOPY(factarray, temparray)
881     │   │   ┌─IF temparray[1,2] = 0
```

```
882             =ADEL(temparray,1)
883             DECLARE temparray[ALEN(temparray,1) - 1,2]
884          ENDIF
885          =ACOPY(steparray, temparray1)
886          IF temparray1[1,2]= 0
887             =ADEL(temparray1,1)
888             DECLARE temparray1[ALEN(temparray1,1) - 1,2]
889          ENDIF
890          m.temp = bwfact(@temparray,@temparray1)
891          IF EMPTY(m.temp)
892             =errmsg("No Bootstrap computing...",2)
893             m.bwstep = 0
894          ELSE
895             m.pos = ASCAN(factarray,dp(m.temp, u,1))
896             m.bbwfact = ASUBSCRIPT(m.factarray,m.pos,1)
897             m.pos = ASCAN(steparray,dp(m.temp, u,2))
898             m.bbwstep = ASUBSCRIPT(m.steparray,m.pos,1)
899          ENDIF
900          RELEASE temparray, temparray1
901       ELSE
902          m.bbwstep = m.step
903          m.bbwfact = m.fact
904       ENDIF
905
906    CASE m.bphase = 1
907       IF phasearray[m.phase,2] = 0
908          =ACOPY(phasearray, temparray)
909          =ADEL(temparray,1)
910          DECLARE temparray[ALEN(temparray,1) - 1,2]
911          m.temp = bselect(@temparray,"Select Life cycle phase:")
912          IF EMPTY(m.temp)
913             =errmsg("No Bootstrap computing...",2)
914             m.bphase = 0
915          ELSE
916             m.pos = ASCAN(phasearray, m.temp)
917             m.bbphase = ASUBSCRIPT(m.phasearray,m.pos,1)
918          ENDIF
919       ELSE
920          m.bbphase = m.phase
921       ENDIF
922    ENDCASE
923    SHOW GETS
924    RETURN
925
926    ******************************
927    FUNCTION bstep
928    ******************************
929    PARAMETER mat, lcphase, wp, sn, sd, sq
930    CREAT TABLE boot (cost N(10,2))
931    m.boot = ""
932    SELECT SUM(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.hmetprob
=> , hmcomp.hmcfid,;
933           hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd, hmcomp.perq, hmcomp.hmu
=> nit;
934           FROM hmcomp;
935           WHERE hmcomp.hmatid = m.mat;
936           AND hmcomp.hmlcid = m.lcphase;
937           AND hmcomp.hmwpid = m.wp;
938           GROUP BY hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmcfeid;
939           ORDER BY hmcomp.hmetid;
940           INTO TABLE test
941
942    FOR i = 1 TO m.sample
943       = resetwt(m.mat, m.lcphase, m.wp)
944       m.sum = CALCULATE(sn, sd, sq)
945       SELECT boot
946       APPEND BLANK
947       REPLACE cost WITH m.sum
948    ENDFOR
949    CALCULATE STD(cost), AVG(cost) TO m.std, m.mean
950    m.boot = ALLTRIM(STR(m.std,8,4)) + u + ALLTRIM(STR(m.mean,8,2))
951    =delfile("boot")
952    RETURN  m.boot
953
954    ******************************
955    FUNCTION bfact
956    ******************************
957    PARAMETER m.factid, m.hmatid
958    PRIVATE msel
959    IF PARAMETER() = 0
960 <------RETURN 0
961    ENDIF
962    m.boot = ""
963    msel = SELECT()
964    CREATE TABLE boot (cost N(10,2))
965    FOR i = 1 TO m.sample
966       SELECT hmtemp
967       GO TOP
968       m.sum = 0
969       SCAN
970          m.wpid = hmtemp.hmwpid
971          m.comid = hmtemp.hmcomid
972          m.lcid = hmtemp.hmlcid
973          SELECT SUM(hmcomp.wtaverage),0000.00, hmcomp.hmetid, hmcomp.hme
=> tprob,;
974              hmcomp.hmcfid,hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd,;
975              hmcomp.perq, hmcomp.hmunit;
976              FROM hmcomp;
977              WHERE hmcomp.hmcomid = m.comid;
978              AND hmcomp.hmatid = m.hmatid;
979              AND hmcomp.hmlcid = m.lcid;
980              AND hmcomp.hmwpid = m.wpid;
981              AND hmcomp.hmcfid = m.factid;
982              GROUP BY hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmcfeid;
983              ORDER BY hmcomp.hmetid;
984              INTO TABLE test
985          =resetwt(m.hmatid, m.lcid, m.wpid)
986          m.sum = m.sum + CALCULATE(hmtemp.pernum, hmtemp.durnum, hmtemp.
=> qtynum)
987       ENDSCAN
988       SELECT boot
989       APPEND BLANK
990       REPLACE cost WITH m.sum
991    ENDFOR
992    SELECT boot
993    CALCULATE STD(cost), AVG(cost) TO m.std, m.mean
994    m.boot = ALLTRIM(STR(m.std,8,4)) + u + ALLTRIM(STR(m.mean,8,2))
995    =delfile('boot')
996    SELECT (msel)
997    RETURN m.boot
998
999    ******************************
1000   FUNCTION bwstep
1001   ******************************
1002   PARAMETER m.factid, m.stepid, m.hmatid
1003   PRIVATE msel
1004   IF PARAMETER() = 0
1005 <------RETURN 0
1006   ENDIF
1007   m.boot = ""
1008   msel = SELECT()
1009   CREATE TABLE boot (cost N(10,2))
```

```
1010        SELECT hmtemp
1011        GO (m.stepid)
1012        m.comid = hmtemp.hmcomid
1013        m.lcid = hmtemp.hmlcid
1014        m.wpid = hmtemp.hmwpid
1015        SELECT SUM(hmcomp.wtaverage),0000.00, hmcomp.hmetid, hmcomp.hmetprob,
 => hmcomp.hmcfid,;
1016              hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd, hmcomp.perq, hmcomp.hmu
 => nit;
1017              FROM hmcomp;
1018              WHERE hmcomp.hmcomid = m.comid;
1019              AND hmcomp.hmatid = m.hmatid;
1020              AND hmcomp.hmlcid = m.lcid;
1021              AND hmcomp.hmwpid = m.wpid;
1022              AND hmcomp.hmcfid = m.factid;
1023              GROUP BY hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmcfeid;
1024              ORDER BY hmcomp.hmetid;
1025              INTO TABLE test
1026        FOR i = 1 TO m.sample
1027           =resetwt(m.hmatid, m.lcid, m.wpid)
1028           m.sum = CALCULATE(hmtemp.pernum, hmtemp.durnum, hmtemp.qtynum)
1029           SELECT boot
1030           APPEND BLANK
1031           REPLACE cost WITH m.sum
1032        ENDFOR
1033        SELECT boot
1034
1035        CALCULATE STD(cost), AVG(cost) TO m.std, m.mean
1036        m.boot = ALLTRIM(STR(m.std,8,4)) + u + ALLTRIM(STR(m.mean,8,2))
1037
1038        =delfile('boot')
1039        SELECT (msel)
1040        RETURN m.boot
1041
1042        ****************************
1043        FUNCTION bphase
1044        ****************************
1045        PARAMETER m.phaseid, m.hmatid
1046
1047        PRIVATE m.sum,msel
1048        IF PARAMETER() = 0
1049           RETURN 0
1050        ENDIF
1051        m.boot = ""
1052        msel = SELECT()
1053        CREATE TABLE boot (cost N(10,2))
1054        FOR i = 1 TO m.sample
1055           SELECT hmtemp
1056           GO TOP
1057           m.sum = 0
1058           SCAN
1059              IF hmlcid = m.phaseid
1060                 m.comid = hmtemp.hmcomid
1061                 m.lcid = hmtemp.hmlcid
1062                 m.wpid = hmtemp.hmwpid
1063                 SELECT SUM(hmcomp.wtaverage),0000.00, hmcomp.hmetid, hmcomp.
 => hmetprob, hmcomp.hmcfid,;
1064                       hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd, hmcomp.perq, h
 => mcomp.hmunit;
1065                       FROM hmcomp;
1066                       WHERE hmcomp.hmcomid = m.comid;
1067                       AND hmcomp.hmatid = m.hmatid;
1068                       AND hmcomp.hmlcid = m.lcid;
1069                       AND hmcomp.hmwpid = m.wpid;
1070                       GROUP BY hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmcfeid;
1071                       ORDER BY hmcomp.hmetid;
1072                       INTO TABLE test
1073                    =resetwt(m.hmatid, m.lcid, m.wpid)
1074                    m.sum = m.sum + CALCULATE(hmtemp.pernum, hmtemp.durnum, hmte
 => mp.qtynum)
1075              ENDIF
1076           ENDSCAN
1077           SELECT boot
1078           APPEND BLANK
1079           REPLACE cost WITH m.sum
1080        ENDFOR
1081        SELECT boot
1082        CALCULATE STD(cost), AVG(cost) TO m.std, m.mean
1083        m.boot = ALLTRIM(STR(m.std,8,4)) + u + ALLTRIM(STR(m.mean,8,2))
1084        =delfile('boot')
1085        SELECT (msel)
1086        RETURN m.boot
1087
1088        ****************************
1089        FUNCTION getstr
1090        ****************************
1091        PARAMETERS hmatname, scentotal, steptotal, factotal, wfactotal, phase
 => total, bootstrap
1092        EXTERNAL ARRAY steptotal, factotal, wfactotal, phasetotal
1093        m.text = ""
1094        u = "~"
1095        mindent = SPACE(5)
1096        m.text = hmatname + ":" + newline
1097        IF !EMPTY(scentotal)
1098           m.text = m.text + mindent + "Total Cost of scenario"     + ": " +
 => STR(scentotal,8,2) + newline
1099           m.text = m.text + newline
1100        ENDIF
1101        IF !EMPTY(steptotal)
1102           FOR i = 1 TO ALEN(steptotal)
1103              m.text = m.text + mindent + "Cost of STEP " + dp(steptotal[i],u
 => ,1) + ": " + dp(steptotal[i],u,2) + newline
1104           ENDFOR
1105           m.text = m.text + newline
1106        ENDIF
1107
1108        IF !EMPTY(factotal)
1109           FOR i = 1 TO ALEN(factotal)
1110              m.text = m.text + mindent + "Cost of FACTOR " + dp(factotal[i],
 => u,1) + ": " + dp(factotal[i],u,2) + newline
1111           ENDFOR
1112           m.text = m.text + newline
1113        ENDIF
1114        IF !EMPTY(wfactotal)
1115           FOR i = 1 TO ALEN(wfactotal)
1116              m.wd = wfactotal[i]
1117              IF !EMPTY(m.wd)
1118                 m.fd = dp(m.wd,u,1)
1119                 m.sd = dp(m.wd,u,2,999)
1120                 DO WHILE !EMPTY(m.sd)
1121                    m.dd = dp(m.sd,u,1)
1122                    IF OCCURS(u, m.sd) > 0
1123                       m.sd = dp(m.sd,u,2,999)
1124                    ELSE
1125                       m.sd = ""
1126                    ENDIF
1127                    m.text = m.text + mindent + "Cost of FACTOR " + m.fd + "
 => within STEP " + dp(m.dd,":", 1) + ": " + dp(m.dd, ":", 2) + newline
1128                 ENDDO
1129              ENDIF
1130           ENDFOR
1131           m.text = m.text + newline
```

```
1132    ENDIF
1133    IF !EMPTY(phasetotal)
1134       FOR i = 1 TO ALEN(phasetotal)
1135          m.text = m.text + mindent + "Cost of PHASE " + dp(phasetotal[i
 => ,u,1) + ": " + dp(phasetotal[i],u,2) + newline
1136       ENDFOR
1137       m.text = m.text + newline
1138    ENDIF
1139    IF !EMPTY(bootstrap)
1140       m.text = m.text + mindent + bootstrap + newline
1141    ENDIF
1142
1143    RETURN m.text
1144
1145    *
1146    *       _QLE0L6B8T              m.bstep VALID
1147    *
1148    *       Function Origin:
1149    *
1150    *       From Platform:          MS-DOS
1151    *       From Screen:            HMCOMP,     Record Number:   21
1152    *       Variable:               m.bstep
1153    *       Called By:              VALID Clause
1154    *       Object Type:            Check Box
1155    *       Snippet Number:         1
1156    *
1157    *
1158    *
1159    FUNCTION _qle0l6b8t     &&  m.bstep VALID
1160    #REGION 1
1161    IF bstep = 1
1162       bfact = 0
1163       bwstep = 0
1164       bphase = 0
1165       SHOW GETS
1166    ENDIF
1167
1168
1169    *
1170    *       _QLE0L68IV              m.bfact VALID
1171    *
1172    *       Function Origin:
1173    *
1174    *       From Platform:          MS-DOS
1175    *       From Screen:            HMCOMP,     Record Number:   24
1176    *       Variable:               m.bfact
1177    *       Called By:              VALID Clause
1178    *       Object Type:            Check Box
1179    *       Snippet Number:         2
1180    *
1181    *
1182    *
1183    FUNCTION _qle0l6biv     &&  m.bfact VALID
1184    #REGION 1
1185    IF bfact = 1
1186       bstep = 0
1187       bwstep = 0
1188       bphase = 0
1189       SHOW GETS
1190    ENDIF
1191
1192
1193    *
1194    *
1195
```

```
1197    *       _QLE0L6BR7              m.bwstep VALID
1198    *
1199    *       Function Origin:
1200    *
1201    *       From Platform:          MS-DOS
1202    *       From Screen:            HMCOMP,     Record Number:   28
1203    *       Variable:               m.bwstep
1204    *       Called By:              VALID Clause
1205    *       Object Type:            Check Box
1206    *       Snippet Number:         3
1207    *
1208    *
1209    *
1210    FUNCTION _qle0l6br7     &&  m.bwstep VALID
1211    #REGION 1
1212    IF bwstep = 1
1213       bfact = 0
1214       bstep = 0
1215       bphase = 0
1216       SHOW GETS
1217    ENDIF
1218
1219
1220    *
1221    *       _QLE0L6BYN              m.bphase VALID
1222    *
1223    *       Function Origin:
1224    *
1225    *       From Platform:          MS-DOS
1226    *       From Screen:            HMCOMP,     Record Number:   31
1227    *       Variable:               m.bphase
1228    *       Called By:              VALID Clause
1229    *       Object Type:            Check Box
1230    *       Snippet Number:         4
1231    *
1232    *
1233    *
1234    FUNCTION _qle0l6byn     &&  m.bphase VALID
1235    #REGION 1
1236    IF bphase = 1
1237       bfact = 0
1238       bwstep = 0
1239       bstep = 0
1240       SHOW GETS
1241    ENDIF
1242
1243
1244    *
1245    *       _QLE0L6C5N              m.sample VALID
1246    *
1247    *       Function Origin:
1248    *
1249    *       From Platform:          MS-DOS
1250    *       From Screen:            HMCOMP,     Record Number:   33
1251    *       Variable:               m.sample
1252    *       Called By:              VALID Clause
1253    *       Object Type:            Field
1254    *       Snippet Number:         5
1255    *
1256    *
1257    *
1258    FUNCTION _qle0l6c5n     &&  m.sample VALID
1259    #REGION 1
1260    IF m.sample > 300
1261
1262
```

```
1263  <-------RETURN .F.
1264    L-ENDIF
1265
1266
1267      *
1268      *
1269      *       _QLE0L6CB0              m.action VALID
1270      *
1271      *       Function Origin:
1272      *
1273      *       From Platform:          MS-DOS
1274      *       From Screen:            HMCOMP,     Record Number:   34
1275      *       Variable:               m.action
1276      *       Called By:              VALID Clause
1277      *       Object Type:            Push Button
1278      *       Snippet Number:         6
1279      *
1280      *
1281      *
1282      FUNCTION _qle0l6cb0     &&  m.action VALID
1283      #REGION 1
1284     +-DO CASE
1285     +-CASE m.action = 1
1286     |    =setupboot()
1287     |    wpop = popupshow("Calculating....")
1288     |    DO compute
1289     |    =popuphide(wpop)
1290     |    CLEAR READ
1291     +-CASE m.action = 2
1292     |    DO hmbrowse
1293     +-CASE m.action = 3
1294     |    CLEAR READ
1295     L-ENDCASE
1296      *: EOF: HMCOMP.AC1
```

```
*
*       10/27/93              HMET.SPR                  14:49:02
*
*       Author's Name
*
*       Copyright (c) 1993 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*               MS-DOS Window definitions
*

IF NOT WEXIST("hmet") ;
      OR UPPER(WTITLE("HMET")) == "HMET.PJX" ;
      OR UPPER(WTITLE("HMET")) == "HMET.SCX" ;
      OR UPPER(WTITLE("HMET")) == "HMET.MNX" ;
      OR UPPER(WTITLE("HMET")) == "HMET.PRG" ;
      OR UPPER(WTITLE("HMET")) == "HMET.FRX" ;
      OR UPPER(WTITLE("HMET")) == "HMET.QPR"
   DEFINE WINDOW hmet ;
      FROM INT((SROW()-12)/2),INT((SCOL()-52)/2) ;
      TO INT((SROW()-12)/2)+11,INT((SCOL()-52)/2)+51 ;
      NOFLOAT ;
      NOCLOSE ;
      SHADOW ;
      nominimize ;
      DOUBLE ;
      COLOR SCHEME 1
ENDIF

*
*           HMET/MS-DOS Setup Code - SECTION 2
*

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
```



```
m.oldescape = SET( "ESCAPE" )
SET ESCAPE OFF
m.adding = .F.
m.change = .F.

CLOSE ALL
SELECT 0
USE hmet
SET ORDER TO TAG hmet OF hmet.cdx

************************************************
* Check see if the last record is defined
IF TYPE( "m.LastRec" ) = "U"

   * Start with the first record
   GO TOP
   m.lastrec = RECNO()

ELSE
   * Start on the last record used
   GO m.lastrec
ENDIF
************************************************
SCATTER MEMVAR

*
*               HMET/MS-DOS Screen Layout
*

#REGION 1
IF WVISIBLE("hmet")
   ACTIVATE WINDOW hmet SAME
ELSE
   ACTIVATE WINDOW hmet NOSHOW
ENDIF
@ 1,40 GET m.action ;
    PICTURE "@*VN \<Add;\<Edit;\!\<Next;\<Previous;\?E\<xit" ;
    SIZE 1,10,1 ;
    DEFAULT 1 ;
    VALID _qkf0vrbit()
@ 8,10 GET m.save ;
    PICTURE "@*HN \<Save;\<Cancel" ;
    SIZE 1,8,1 ;
    DEFAULT 1 ;
    VALID _qkf0vrbpa() ;
    DISABLE
@ 2,10 GET m.hmetid ;
    SIZE 1,10 ;
    DEFAULT 0 ;
    DISABLE
@ 4,10 GET m.hmet ;
    SIZE 1,28 ;
    DEFAULT " " ;
    PICTURE "@!" ;
    VALID _qkf0vrbt3()
@ 1,0 TO 9,39
@ 0,11 SAY "HM EXPOSURE TYPES" ;
    SIZE 1,17, 0
@ 2,2 SAY "ID NUM:" ;
    SIZE 1,7, 0
@ 4,3 SAY " TYPE:" ;
    SIZE 1,6, 0
```

```
133
134     IF NOT WVISIBLE("hmet")
135        ACTIVATE WINDOW hmet
136     ENDIF
137
138       READ CYCLE
139
140       RELEASE WINDOW hmet
141
142       #REGION 0
143     IF m.talkstat = "ON"
144        SET TALK ON
145     ENDIF
146     IF m.compstat = "ON"
147        SET COMPATIBLE ON
148     ENDIF
149
150
151       *
152       *
153       *              HMET/MS-DOS Cleanup Code
154       *
155       *
156       *
157
158       #REGION 1
159       POP KEY ALL
160       SET ESCAPE &oldescape
161       ********* End of Main Body - Entry Cleanup
162
163       ******************
164
165
166       *
167       *
168       *           HMET/MS-DOS Supporting Procedures and Functions
169       *
170       *
171       *
172
173       #REGION 1
174       PROCEDURE CHANGE
175       ******************
176       m.oldexact = SET( "EXACT" )
177       SET EXACT ON
178       m.change =(TRIM(hmet.hmet) <> TRIM( m.hmet))
179       SET EXACT &oldexact
180       RETURN m.change
181
182
183       *
184       *
185       *        _QKF0VRBIT            m.Action VALID
186       *
187       *        Function Origin:
188       *
189       *        From Platform:        MS-DOS
190       *        From Screen:          HMET,     Record Number:    3
191       *        Variable:             m.Action
192       *        Called By:            VALID Clause
193       *        Object Type:          Push Button
194       *        Snippet Number:       1
195       *
196       *
197       *
198       FUNCTION _qkf0vrbit     &&  m.Action VALID
```

```
199       #REGION 1
200     IF m.action = 1
201        SCATTER MEMVAR BLANK
202        m.hmetid=RECCOUNT()+1
203        SHOW GETS
204        SHOW GET action disabled
205        SHOW GET SAVE enabled
206        m.adding = .T.
207
208     ELSE
209
210        DO CASE
211
212        CASE m.action = 2
213           SHOW GETS
214           SHOW GET action disabled
215           SHOW GET SAVE enabled
216
217        CASE m.action = 3
218           SKIP
219           IF EOF()
220              ?? CHR( 7 )
221              WAIT "Last record" WINDOW NOWAIT
222              SKIP -1
223           ELSE
224              SCATTER MEMVAR
225              SHOW GETS
226           ENDIF
227
228        CASE m.action = 4
229           SKIP -1
230           IF BOF()
231              ?? CHR( 7 )
232              WAIT "First record" WINDOW NOWAIT
233              SKIP
234           ELSE
235              SCATTER MEMVAR
236              SHOW GETS
237           ENDIF
238
239        CASE m.action = 5
240           CLEAR READ
241        ENDCASE
242     ENDIF
243
244
245
246       *
247       *
248       *        _QKF0VRBPA            m.Save VALID
249       *
250       *        Function Origin:
251       *
252       *        From Platform:        MS-DOS
253       *        From Screen:          HMET,     Record Number:    4
254       *        Variable:             m.Save
255       *        Called By:            VALID Clause
256       *        Object Type:          Push Button
257       *        Snippet Number:       2
258       *
259       *
260       *
261       * did the record change
262       FUNCTION _qkf0vrbpa     &&  m.Save VALID
263       #REGION 1
264       DO CHANGE
```

```
265
266     IF m.save = 1 && Selected Save Button
267
268       IF m.adding && Adding a new record
269          APPEND BLANK
270          GATHER MEMVAR
271       ENDIF
272
273       IF m.change && Changing an old record
274          GATHER MEMVAR
275       ENDIF
276
277     ELSE
278        SCATTER MEMVAR
279     ENDIF
280
281      SHOW GETS
282      SHOW GET action enabled
283      SHOW GET SAVE disabled
284
285      m.adding = .F.
286      m.change = .F.
287
288
289      *
290      *
291      *          _QKF0VRBT3            m.hmet VALID
292      *
293      *          Function Origin:
294      *
295      *          From Platform:        MS-DOS
296      *          From Screen:          HMET,     Record Number:    6
297      *          Variable:             m.hmet
298      *          Called By:            VALID Clause
299      *          Object Type:          Field
300      *          Snippet Number:       3
301      *
302      *
303      *
304      FUNCTION _qkf0vrbt3     &&  m.hmet VALID
305      #REGION 1
306     IF m.adding
307        SEEK m.hmet
308       IF FOUND()
309          =errmsg("Record already exists",1)
310          SCATTER MEMVAR BLANK FIELD hmet
311       ENDIF
312     ENDIF
313      *: EOF: HMET.AC1
```

```
*
*       10/27/93                HMLC.SPR                14:48:55
*
*       Author's Name
*
*       Copyright (c) 1993 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*               MS-DOS Window definitions
*

IF NOT WEXIST("hmlc") ;
      OR UPPER(WTITLE("HMLC")) == "HMLC.PJX" ;
      OR UPPER(WTITLE("HMLC")) == "HMLC.SCX" ;
      OR UPPER(WTITLE("HMLC")) == "HMLC.MNX" ;
      OR UPPER(WTITLE("HMLC")) == "HMLC.PRG" ;
      OR UPPER(WTITLE("HMLC")) == "HMLC.FRX" ;
      OR UPPER(WTITLE("HMLC")) == "HMLC.QPR"
   DEFINE WINDOW hmlc ;
      FROM INT((SROW()-12)/2),INT((SCOL()-52)/2) ;
      TO INT((SROW()-12)/2)+11,INT((SCOL()-52)/2)+51 ;
      NOFLOAT ;
      NOCLOSE ;
      SHADOW ;
      nominimize ;
      DOUBLE ;
      COLOR SCHEME 1
ENDIF


*
*           HMLC/MS-DOS Setup Code - SECTION 2
*

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
m.oldescape = SET( "ESCAPE" )
SET ESCAPE OFF
m.adding = .F.
m.change = .F.

CLOSE ALL
SELECT 0
USE hmlc
SET ORDER TO TAG hmlcid OF hmlc.cdx

************************************************
* Check see if the last record is defined
IF TYPE( "m.LastRec" ) = "U"

   * Start with the first record
   GO TOP
   m.lastrec = RECNO()

ELSE
   * Start on the last record used
   GO m.lastrec
ENDIF
************************************************
SCATTER MEMVAR


*
*               HMLC/MS-DOS Screen Layout
*

#REGION 1
IF WVISIBLE("hmlc")
   ACTIVATE WINDOW hmlc SAME
ELSE
   ACTIVATE WINDOW hmlc NOSHOW
ENDIF
@ 1,40 GET m.action ;
   PICTURE "@*VN \<Add;\<Edit;\!\<Next;\<Previous;\?E\<xit" ;
   SIZE 1,10,1 ;
   DEFAULT 1 ;
   VALID  qkf0vr6jx()
@ 1,0 TO 9,39
@ 4,11 SAY ":" ;
   SIZE 1,1, 0
@ 0,11 SAY "MM LIFE CYCLE PHASES" ;
   SIZE 1,20, 0
@ 2,3 SAY "PHASE ID:" ;
   SIZE 1,9, 0
@ 4,1 SAY "PHASE NAME:" ;
   SIZE 1,11, 0
@ 2,13 GET m.hmlcid ;
   SIZE 1,10 ;
   DEFAULT 0 ;
   DISABLE
@ 4,13 GET m.hmlc ;
   SIZE 1,25 ;
   DEFAULT " " ;
   PICTURE "@!" ;
   VALID  qkf0vr6te() ;
   DISABLE
@ 8,10 GET m.save ;
   PICTURE "@*HN \<Save;\<Cancel" ;
   SIZE 1,8,1 ;
```

```
133         DEFAULT 1 ;
134         VALID _qkf0vr6vw() ;
135         DISABLE
136
137     IF NOT WVISIBLE("hmlc")
138        ACTIVATE WINDOW hmlc
139     ENDIF
140
141     READ CYCLE
142
143     RELEASE WINDOW hmlc
144
145     #REGION 0
146     IF m.talkstat = "ON"
147        SET TALK ON
148     ENDIF
149     IF m.compstat = "ON"
150        SET COMPATIBLE ON
151     ENDIF
152
153
154     *
155     *
156     *              HMLC/MS-DOS Cleanup Code
157     *
158     *
159     *
160
161     #REGION 1
162     POP KEY ALL
163     SET ESCAPE &oldescape
164     ********* End of Main Body - Entry Cleanup
165
166     ******************
167
168
169     *
170     *
171     *          HMLC/MS-DOS Supporting Procedures and Functions
172     *
173     *
174     *
175
176     #REGION 1
177     PROCEDURE CHANGE
178     ******************
179     m.oldexact = SET( "EXACT" )
180     SET EXACT ON
181     m.change =(TRIM(hmlc.hmlc) <> TRIM( m.hmlc))
182     SET EXACT &oldexact
183     RETURN m.change
184
185
186     *
187     *
188     *          _QKF0VR6JX          m.Action VALID
189     *
190     *          Function Origin:
191     *
192     *          From Platform:      MS-DOS
193     *          From Screen:        HMLC,     Record Number:    3
194     *          Variable:           m.Action
195     *          Called By:          VALID Clause
196     *          Object Type:        Push Button
197     *          Snippet Number:     1
198     *
199     *
200     *
201     FUNCTION _qkf0vr6jx     && m.Action VALID
202     #REGION 1
203     IF m.action = 1
204        SCATTER MEMVAR BLANK
205        m.hmlcid=RECCOUNT()+1
206        SHOW GETS
207        SHOW GET m.hmlc enabled
208        SHOW GET action disabled
209        SHOW GET SAVE enabled
210        m.adding = .T.
211
212     ELSE
213
214        DO CASE
215
216        CASE m.action = 2
217           SHOW GETS
218           SHOW GET m.hmlc enabled
219           SHOW GET action disabled
220           SHOW GET SAVE enabled
221
222        CASE m.action = 3
223           SKIP
224           IF EOF()
225              ?? CHR( 7 )
226              WAIT "Last record" WINDOW NOWAIT
227              SKIP -1
228           ELSE
229              SCATTER MEMVAR
230              SHOW GETS
231           ENDIF
232
233        CASE m.action = 4
234           SKIP -1
235           IF BOF()
236              ?? CHR( 7 )
237              WAIT "First record" WINDOW NOWAIT
238              SKIP
239           ELSE
240              SCATTER MEMVAR
241              SHOW GETS
242           ENDIF
243
244        CASE m.action = 5
245           CLEAR READ
246        ENDCASE
247     ENDIF
248
249
250
251     *
252     *
253     *          _QKF0VR6TE          m.hmlc VALID
254     *
255     *          Function Origin:
256     *
257     *          From Platform:      MS-DOS
258     *          From Screen:        HMLC,     Record Number:   10
259     *          Variable:           m.hmlc
260     *          Called By:          VALID Clause
261     *          Object Type:        Field
262     *          Snippet Number:     2
263     *
264     *
```

```
265     *
266     FUNCTION _qkf0vr6te     &&  m.hmlc VALID
267     #REGION 1
268     IF m.adding
269        m.oldrec = RECNO()
270        GO TOP
271        SEEK m.hmlc
272        IF FOUND()
273           =errmsg("Record already exists",1)
274           SCATTER FIELD hmlc MEMVAR BLANK
275        ENDIF
276        GO m.oldrec
277     ENDIF
278
279
280     *
281     *
282     *
283     *
284     *
285     *
286     *
287     *
288     *
289     *
290     *
291     *
292     *
293     *
294     *
295     FUNCTION _qkf0vr6vw     &&  m.Save VALID
296     #REGION 1
297     IF m.save = 1 && Selected Save Button
298
299        IF m.adding && Adding a new record
300           APPEND BLANK
301           GATHER MEMVAR
302        ELSE
303           DO CHANGE
304           IF m.change && Changing an old record
305              GATHER MEMVAR
306           ENDIF
307        ENDIF
308     ENDIF
309     SCATTER MEMVAR
310     SHOW GETS
311     SHOW GET m.hmlc disabled
312     SHOW GET action enabled
313     SHOW GET SAVE disabled
314
315     m.adding = .F.
316     m.change = .F.
317     *: EOF: HMLC.AC1
```

_QKF0VR6VW m.Save VALID

Function Origin:

From Platform:	MS-DOS
From Screen:	HMLC, Record Number: 11
Variable:	m.Save
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	3

```
*
*       10/27/93              HMTAB.SPR                14:49:19
*
*       Author's Name
*
*       Copyright (c) 1993 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*              MS-DOS Window definitions
*

IF NOT WEXIST("hmtab") ;
      OR UPPER(WTITLE("HMTAB")) == "HMTAB.PJX" ;
      OR UPPER(WTITLE("HMTAB")) == "HMTAB.SCX" ;
      OR UPPER(WTITLE("HMTAB")) == "HMTAB.MNX" ;
      OR UPPER(WTITLE("HMTAB")) == "HMTAB.PRG" ;
      OR UPPER(WTITLE("HMTAB")) == "HMTAB.FRX" ;
      OR UPPER(WTITLE("HMTAB")) == "HMTAB.QPR"
   DEFINE WINDOW hmtab ;
      FROM INT((SROW()-22)/2),INT((SCOL()-79)/2) ;
      TO INT((SROW()-22)/2)+21,INT((SCOL()-79)/2)+78 ;
      FLOAT ;
      NOCLOSE ;
      SHADOW ;
      nominimize ;
      DOUBLE ;
      COLOR SCHEME 1
ENDIF

*
*           HMTAB/MS-DOS Setup Code - SECTION 2
*

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
```

```
   m.oldescape = SET( "ESCAPE" )
   SET ESCAPE OFF
   m.adding = .F.
   m.change = .F.
   m.action=5


   ****************************
   * SELECT HAZARDOUS MATERIAL
   ****************************
   m.data=hmlu()
   m.hmname = dp(m.data,"^",1)
   m.hmatid = INT(VAL(dp(m.data,"^",2)))
   IF EMPTY(m.hmname)
      =errmsg("No material selected",1)
      RETURN
   ENDIF

   CLOSE ALL
   ****************************
   * GET UNIT LIST
   ****************************
   IF !USED("HMUNIT")
      SELECT 0
      USE hmunit ORDER TAG hmunit
   ELSE
      SELECT hmunit
   ENDIF
   COPY TO ARRAY unitlist FIELD hmunit
   USE
   m.unit = "GALLON"

   ****************************
   * GET THE MATERIAL RECORDS
   ****************************
   *m.hmatid=get_hmid(m.hmname)
   m.hmcomid = get_hmcomid(m.hmatid)
   SELECT 0
   USE hmtab ALIAS hmtab
   SET FILTER TO hmatid = m.hmatid


   *What if the file is empty?

   GO TOP
   m.oldhmcomid = m.hmcomid
   m.oldhmatid=m.hmatid
   SCATTER MEMVAR
   m.oldunit = m.hmunit

   IF m.tabid<1
      m.action=1
      m.adding=.T.
      m.tabid=RECCOUNT()+1
      m.hmatid=m.oldhmatid
      m.hmcomid = m.oldhmcomid
      m.hmlc=""
      m.hmlcid=0
      m.oldhmlc=""
      m.hmet=""
      m.oldhmet=""
      m.hmetid=0
      m.hmetprob=0.000
      m.hmwp=""
      m.oldhmwp=""
      m.hmwpid=0
```

```
133         m.hmcf=""
134         m.hmcfid=0
135         m.oldhmcf=""
136         m.hmcfeid=0
137         m.hmcfe=""
138         m.oldhmcfe=""
139         m.hmcfeiid=0
140         m.hmcfei=""
141         m.hmcfecost=0.00
142         m.hmunit=IIF(EMPTY(m.oldunit),"GALLON",m.oldunit)
143         m.prob=0.00
144         m.wtavgerage=0.00
145
146      ELSE
147
148         IF m.hmlcid>0
149            m.hmlc=get_hmlcn(m.hmlcid)
150            m.oldhmlc=m.hmlc
151         ELSE
152            m.hmlc=""
153            m.oldhmlc=""
154         ENDIF
155
156         IF m.hmwpid>0
157            m.hmwp=get_hmwpn(m.hmwpid)
158            m.oldhmwp=m.hmwp
159         ELSE
160            m.hmwp=""
161            m.oldhmwp=""
162         ENDIF
163
164         IF m.hmetid>0
165            m.hmet=get_hmetn(hmetid)
166            m.oldhmet=m.hmet
167         ELSE
168            m.hmet=""
169            m.oldhmet=""
170         ENDIF
171
172         IF m.hmcfid>0
173            m.hmcf=get_cfar(m.hmcfid)
174            m.oldhmcf=m.hmcf
175         ELSE
176            m.hmcf=""
177            m.oldhmcf=""
178         ENDIF
179
180         IF m.hmcfeid>0
181            m.hmcfe=get_cfar1(m.hmcfeid)
182            m.oldhmcfe=m.hmcfe
183         ELSE
184            m.hmcfe=""
185            m.oldhmcfe=""
186         ENDIF
187
188         IF m.hmcfeiid>0
189            m.hmcfei=get_ei(m.hmcfeiid)
190            m.oldhmcfei=m.hmcfei
191         ELSE
192            m.hmcfei=""
193            m.oldhmcfei=""
194         ENDIF
195      ENDIF
196
197      SHOW GETS
198
199
200
201
202
203      *
204      *
205      *        HMTAB/MS-DOS Screen Layout
206      *
207      *
208      *
209
210      #REGION 1
211      IF WVISIBLE("hmtab")
212         ACTIVATE WINDOW hmtab SAME
213      ELSE
214         ACTIVATE WINDOW hmtab NOSHOW
215      ENDIF
216      @ 1,67 GET m.action ;
217            PICTURE "@*VN \<Add;\<Edit;\!\<Next;\<Previous;\<Top;\<Bottom;\?E
=> \<xit" ;
218            SIZE 1,10,1 ;
219            DEFAULT 1 ;
220            VALID _qkf0vrpcs()
221      @ 0,5 GET m.tabid ;
222            SIZE 1,4 ;
223            DEFAULT 0 ;
224            DISABLE
225      @ 1,14 GET m.hmatid ;
226            SIZE 1,4 ;
227            DEFAULT 0 ;
228            PICTURE "@Z" ;
229            DISABLE
230      @ 1,19 GET m.hmname ;
231            SIZE 3,21 ;
232            DEFAULT " " ;
233            PICTURE "@!" ;
234            DISABLE
235      @ 1,47 GET m.hmunit ;
236            PICTURE "@^" ;
237            FROM unitlist ;
238            SIZE 3,18 ;
239            DEFAULT 1 ;
240            DISABLE ;
241            COLOR SCHEME 1, 2
242      @ 4,14 GET m.hmlcid ;
243            SIZE 1,4 ;
244            DEFAULT 0 ;
245            PICTURE "@Z" ;
246            DISABLE
247      @ 4,19 GET m.hmlc ;
248            SIZE 1,46 ;
249            DEFAULT " " ;
250            PICTURE "@!" ;
251            WHEN _qkf0vrpys() ;
252            VALID _qkf0vrq1c() ;
253            DISABLE
254      @ 5,14 GET m.hmwpid ;
255            SIZE 1,4 ;
256            DEFAULT 0 ;
257            PICTURE "@Z" ;
258            DISABLE
259      @ 5,19 GET m.hmwp ;
260            SIZE 2,46 ;
261            DEFAULT " " ;
262            PICTURE "@!" ;
263            WHEN _qkf0vrq4t() ;
```

```
        VALID _qkf0vrq72() ;
        DISABLE
    @ 7,18 GET m.hmetid ;
        SIZE 1,4 ;
        DEFAULT 0 ;
        PICTURE "@Z" ;
        DISABLE
    @ 7,24 GET m.hmet ;
        SIZE 1,41 ;
        DEFAULT " " ;
        PICTURE "@!" ;
        WHEN _qkf0vrqal() ;
        VALID _qkf0vrqc2() ;
        DISABLE
    @ 8,32 GET m.hmetprob ;
        SIZE 1,9 ;
        DEFAULT 0 ;
        DISABLE
    @ 10,13 GET m.hmcfid ;
        SIZE 1,3 ;
        DEFAULT 0 ;
        PICTURE "@Z" ;
        DISABLE
    @ 10,17 GET m.hmcf ;
        SIZE 1,39 ;
        DEFAULT " " ;
        WHEN _qkf0vrqfr() ;
        VALID _qkf0vrqh7() ;
        DISABLE
    @ 11,4 GET m.element ;
        PICTURE "@*IHN " ;
        SIZE 1,9,1 ;
        DEFAULT 0 ;
        VALID _qkf0vrqkz() ;
        DISABLE
    @ 11,13 GET m.hmcfeid ;
        SIZE 1,3 ;
        DEFAULT 0 ;
        PICTURE "@Z" ;
        DISABLE
    @ 11,17 GET m.hmcfe ;
        SIZE 1,39 ;
        DEFAULT " " ;
        PICTURE "@!" ;
        DISABLE
    @ 12,13 GET m.hmcfeiid ;
        SIZE 1,3 ;
        DEFAULT 0 ;
        PICTURE "@Z" ;
        DISABLE
    @ 12,17 GET m.hmcfei ;
        SIZE 1,39 ;
        DEFAULT " " ;
        PICTURE "@!" ;
        DISABLE
    @ 14,13 GET m.hmcfecost ;
        SIZE 1,9 ;
        DEFAULT 0 ;
        PICTURE "@$$$,$$$.99" ;
        DISABLE
    @ 14,46 GET m.prob ;
        SIZE 1,5 ;
        DEFAULT 0 ;
        PICTURE "9.99" ;
        DISABLE
    @ 16,33 GET m.perp ;
        PICTURE "@*RHN YES;NO" ;
        SIZE 1,7,0 ;
        DEFAULT 2 ;
        DISABLE
    @ 17,33 GET m.perd ;
        PICTURE "@*RHN YES;NO" ;
        SIZE 1,7,0 ;
        DEFAULT 2 ;
        DISABLE
    @ 18,33 GET m.perq ;
        PICTURE "@*RHN YES;NO" ;
        SIZE 1,7,0 ;
        DEFAULT 2 ;
        DISABLE
    @ 16,68 GET m.save ;
        PICTURE "@*VN \<Save;\<Cancel" ;
        SIZE 1,8,1 ;
        DEFAULT 1 ;
        VALID _qkf0vrqrg() ;
        DISABLE
    @ 11,4 SAY "Element:" ;
        SIZE 1,8, 0
    @ 9,3 TO 19,63
    @ 14,33 SAY "Probability" ;
        SIZE 1,11, 0
    @ 16,19 SAY "By Person" ;
        SIZE 1,9, 0
    @ 1,4 SAY "Material:" ;
        SIZE 1,9, 0
    @ 0,26 SAY "HAZARDOUS MATERIALS TABLE" ;
        SIZE 1,25, 0
    @ 4,4 SAY "Phase:" ;
        SIZE 1,6, 0
    @ 5,4 SAY "Process:" ;
        SIZE 1,8, 0
    @ 7,4 SAY "Exposure Type:" ;
        SIZE 1,14, 0
    @ 8,4 SAY "   Probability of exposure: " ;
        SIZE 1,28, 0
    @ 12,7 SAY "Item:" ;
        SIZE 1,5, 0
    @ 14,7 SAY "Cost:" ;
        SIZE 1,5, 0
    @ 10,5 SAY "Factor:" ;
        SIZE 1,7, 0
    @ 0,4 SAY "#" ;
        SIZE 1,1, 0
    @ 2,42 SAY "Per:" ;
        SIZE 1,4, 0
    @ 18,19 SAY "By Quantity" ;
        SIZE 1,11, 0
    @ 17,19 SAY "By Day" ;
        SIZE 1,6, 0

IF NOT WVISIBLE("hmtab")
    ACTIVATE WINDOW hmtab
ENDIF

READ CYCLE

RELEASE WINDOW hmtab

#REGION 0
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
```

```
IF m.compstat = "ON"
   SET COMPATIBLE ON
ENDIF


*
*                    HMTAB/MS-DOS Cleanup Code
*
*
*

#REGION 1
CLOSE ALL
POP KEY ALL
SET ESCAPE &oldescape
RETURN
********* End of Main Body - Entry Cleanup
**********************


*
*                HMTAB/MS-DOS Supporting Procedures and Functions
*
*
*

#REGION 1
PROCEDURE escpressed
RETURN
**********************


******************
PROCEDURE CHANGE
******************
IF USED("HMTAB")
   SELECT hmtab
   SET FILTER TO hmatid = m.hmatid
ELSE
   SELECT 0
   USE hmtab ALIAS hmtab
   SET FILTER TO hmatid = m.hmatid
ENDIF

m.oldexact = SET( "EXACT" )
SET EXACT ON
m.change =(hmtab.hmatid <> m.hmatid;
    OR hmtab.hmlcid <> m.hmlcid;
    OR hmtab.hmwpid <> m.hmwpid;
    OR hmtab.hmetid <> m.hmetid;
    OR hmtab.hmetprob <> m.hmetprob;
    OR hmtab.hmcfid <> m.hmcfid;
    OR hmtab.hmcfeid <>      m.hmcfeid;
    OR hmtab.hmcfeiid <> m.hmcfeiid;
    OR hmtab.hmcfecost <> m.hmcfecost;
    OR hmtab.hmunit<>m.hmunit;
    OR hmtab.prob <> m.prob;
    OR hmtab.wtaverage <> m.wtaverage;
    OR hmtab.perd <> m.perd;
    OR hmtab.perq <> m.perq;
    OR hmtab.perp <> m.perp)

SET EXACT &oldexact
RETURN m.change


**************************
*FUNCTION GET_HMNAME
**************************
*RELEASE hmn
*DIMENSION hmn[1]
*hmn[1]=""
*M.oldfile=SELECT()

*SELECT HMAT.HMATNAME;
    *       FROM HMAT;
    *       WHERE HMAT.HMATNAME IN (alltrim(m.hmname));
    *       INTO ARRAY hmn

*m.ans=""
*IF NOT EMPTY(hmn[1])
*    m.ans=CHOOSER(@hmn,"Select a Material")
*ELSE
*    =ERRMSG(m.hmname + " was not found",1)
*    m.ans=""
*ENDIF

*SELECT (M.oldfile)
*RETURN m.ans

**************************
*FUNCTION get_hmid
**************************
*PARAMETER m.hmname

*m.oldfile=SELECT()

*SELECT DISTINCT HMAT.HMATID;
    *       FROM HMAT;
    *       WHERE HMAT.HMATNAME IN (alltrim(m.hmname));
    *       INTO ARRAY X

*M.HMATID=X[1]
*SELECT (m.oldfile)
*RETURN M.HMATID

**************************
FUNCTION get_hmcomid
**************************
PARAMETER m.hmatid

m.oldfile=SELECT()

SELECT DISTINCT hmat.niin;
    FROM hmat;
    WHERE hmat.hmatid = m.hmatid;
    INTO ARRAY X
IF TYPE("X") != "U"
   m.stock = X[1]
ELSE
   m.stock = 0
ENDIF
IF !USED("HMCOM")
   SELECT 0
   USE hmcom
ELSE
   SELECT hmcom
ENDIF
SET ORDER TO TAG niin
SEEK m.stock
IF FOUND()
```

```
528     |   m.comid = hmcomid
529     |ELSE
530     |   m.comid = 0
531     |ENDIF
532     SELECT (m.oldfile)
533     RETURN m.comid
534
535     **************************
536     *GET LIFE CYCLE PHASE
537     FUNCTION get_hmlc
538     **************************
539     PARAMETER m.hmlc
540
541     RELEASE hmn
542     DIMENSION hmn[1]
543     hmn[1]=""
544     m.oldfile=SELECT()
545
546     SELECT hmlc.hmlc;
547         FROM hmlc;
548         WHERE hmlc.hmlc IN (ALLTRIM(m.hmlc));
549         ORDER BY hmlcid;
550         INTO ARRAY hmn
551
552     m.ans=""
553     IF NOT EMPTY(hmn[1])
554        m.ans=chooser(@hmn,"Select a Life Cycle Phase")
555     ELSE
556        =errmsg(m.hmlc + " was not found",1)
557        m.ans = ""
558     ENDIF
559
560     SELECT (m.oldfile)
561     RETURN m.ans
562
563     ***********************
564     FUNCTION get_hmlcn
565     PARAMETER m.hmlcid
566     m.oldfile=SELECT()
567
568     SELECT DISTINCT hmlc.hmlc;
569         FROM hmlc;
570         WHERE hmlc.hmlcid = (m.hmlcid);
571         INTO ARRAY X
572
573     m.hmlc=X[1]
574     SELECT (m.oldfile)
575     RETURN m.hmlc
576
577     **************************
578     *Working Processes
579     FUNCTION get_hmwp
580     **************************
581     PARAMETER m.hmwp
582     RELEASE hmn
583     DIMENSION hmn[1]
584     hmn[1]=""
585     m.oldfile=SELECT()
586
587     SELECT hmwp.hmwp;
588         FROM hmwp;
589         WHERE hmwp.hmwp IN (ALLTRIM(m.hmwp));
590         INTO ARRAY hmn
591
592     m.ans=""
593     IF NOT EMPTY(hmn[1])
```

```
594     |   m.ans=chooser(@hmn,"Select a Process")
595     |ELSE
596     |   =errmsg(m.hmwp + " was not found",1)
597     |   m.ans = ""
598     |ENDIF
599
600     SELECT (m.oldfile)
601
602     RETURN m.ans
603
604     ***********************
605     FUNCTION get_hmwpn
606     PARAMETER m.hmwpid
607     m.oldfile=SELECT()
608
609     SELECT DISTINCT hmwp.hmwp;
610         FROM hmwp;
611         WHERE hmwp.hmwpid = m.hmwpid;
612         INTO ARRAY X
613
614     m.hmwp=X[1]
615     SELECT(m.oldfile)
616     RETURN m.hmwp
617
618     **************************
619     FUNCTION get_hmet
620     **************************
621     PARAMETER m.hmet
622
623     RELEASE hmn
624     DIMENSION hmn[1]
625     hmn[1]=""
626     m.oldfile=SELECT()
627
628     SELECT hmet.hmet;
629         FROM hmet;
630         WHERE hmet.hmet IN (ALLTRIM(m.hmet));
631         INTO ARRAY hmn
632
633     m.ans=""
634     IF NOT EMPTY(hmn[1])
635        m.ans=chooser(@hmn,"Select an Exposure Type")
636     ELSE
637        =errmsg(m.hmet + " was not found",1)
638     ENDIF
639
640     SELECT (m.oldfile)
641     RETURN m.ans
642
643     ***************************
644     FUNCTION get_hmetid
645     **************************
646     PARAMETER m.hmet
647     m.oldfile=SELECT()
648
649     SELECT DISTINCT hmet.hmetid;
650         FROM hmet;
651         WHERE hmet.hmet IN (ALLTRIM(m.hmet));
652         INTO ARRAY X
653
654     m.hmetid=X[1]
655     SELECT (m.oldfile)
656     RETURN m.hmetid
657
658     **************************
659     FUNCTION get_hmetn
```

```
660  ****************************
661  PARAMETER m.hmetid
662  m.oldfile=SELECT()
663
664  SELECT DISTINCT hmet.hmet;
665      FROM hmet;
666      WHERE hmet.hmetid = (m.hmetid);
667      INTO ARRAY X
668
669  m.hmet=X[1]
670  SELECT (m.oldfile)
671  RETURN m.hmet
672
673
674  ***********   COST FACTORS ******************
675
676  **********************
677  FUNCTION get_cfar
678  *get the cost factors
679  **********************
680  PARAMETER m.hmcfid
681  m.oldfile=SELECT()
682
683  DECLARE cfarr[1]
684  cfarr[1]=""
685
686  SELECT hmcf.hmcf;
687      FROM hmcf;
688      WHERE hmcf.hmcfid = (m.hmcfid);
689      INTO ARRAY cfarr
690
691  m.hmcf=cfarr[1]
692  SELECT(m.oldfile)
693  RETURN m.hmcf
694
695  *************************
696  FUNCTION get_cfar1
697  *get cost factor elements
698  *************************
699  PARAMETER m.hmcfeid
700  m.oldfile=SELECT()
701  DECLARE cfar1[1]
702  cfar1[1]=""
703
704  SELECT hmcfe.hmcfe;
705      FROM hmcfe;
706      WHERE hmcfe.hmcfeid = (m.hmcfeid);
707      INTO ARRAY cfar1
708
709  m.hmcfe=cfar1[1]
710  SELECT (m.oldfile)
711  RETURN m.hmcfe
712
713  ************************
714  FUNCTION get_cf
715  ************************
716  PARAMETER m.hmcf
717
718  RELEASE MEMO LIKE cfearr
719  DIMENSION cfearr[1]
720  cfearr[1]=""
721
722  m.oldfile=SELECT()
723
724  SELECT hmcf.hmcf;
725      FROM hmcf;
726      WHERE hmcf.hmcf IN (ALLTRIM(m.hmcf));
727      ORDER BY hmcf.hmcfid;
728      INTO ARRAY cfearr
729
730  IF NOT EMPTY(cfearr[1])
731
732     m.x=chooser(@cfearr,"Select a Cost Factor")
733  ELSE
734     =errmsg(m.x + " was not found",1)
735     m.hmcfid=0
736     m.x=""
737
738  ENDIF
739
740  SELECT (m.oldfile)
741  RETURN m.x
742
743  ***************************
744  FUNCTION get_hmcfid
745  ***************************
746  PARAMETER m.hmcf
747  m.oldfile = SELECT()
748  SELECT DISTINCT hmcf.hmcfid;
749      FROM hmcf;
750      WHERE hmcf.hmcf IN (ALLTRIM(m.hmcf));
751      INTO ARRAY X
752  m.hmcfid=X[1]
753  SELECT (m.oldfile)
754  RETURN m.hmcfid
755
756  ************************
757  FUNCTION get_cf1
758  ************************
759  PARAMETER m.hmcfe
760
761  RELEASE MEMO LIKE cfearr1
762  DIMENSION cfearr1[1]
763  cfearr1[1]=""
764  m.oldfile=SELECT()
765
766  SELECT hmcfe.hmcfe;
767      FROM hmcfe;
768      WHERE hmcfe.hmcfe IN (ALLTRIM(m.hmcfe));
769      ORDER BY hmcfe.hmcfeid;
770      INTO ARRAY cfearr1
771
772  IF NOT EMPTY(cfearr1[1])
773
774     m.x=chooser(@cfearr1,"Select a Cost Factor Element")
775
776  ELSE
777     =errmsg(m.x + " was not found",1)
778     m.hmcfeid=0
779     m.x=""
780
781  ENDIF
782  SELECT (m.oldfile)
783  RETURN m.x
784
785  ***************************
786  FUNCTION get_eid
787  ***************************
788  PARAMETER m.hmcfe
789
790  m.oldfile = SELECT()
791
```

```
SELECT DISTINCT hmcfe.hmcfeid;
    FROM hmcfe;
    WHERE hmcfe.hmcfe IN (ALLTRIM(m.hmcfe));
    INTO ARRAY X

m.hmcfeid=X[1]
SELECT (m.oldfile)
RETURN m.hmcfeid

****************************
FUNCTION get_ei
****************************
PARAMETER m.hmcfeiid
m.oldfile=SELECT()

SELECT hmcfei.hmcfei;
    FROM hmcfei;
    WHERE hmcfei.hmcfeiid = (m.hmcfeiid);
    INTO ARRAY X

m.hmcfei=X[1]
SELECT (m.oldfile)
RETURN m.hmcfei

****************************
FUNCTION get_eiid
****************************
PARAMETER m.hmcfei
m.oldfile=SELECT()

SELECT hmcfei.hmcfeiid;
    FROM hmcfei;
    WHERE hmcfei.hmcfei IN (ALLTRIM(m.hmcfei));
    INTO ARRAY X

m.hmcfeiid=X[1]
SELECT (m.oldfile)
RETURN m.hmcfeiid

************************
FUNCTION rel
************************
PARAMETER m.id
EXTERNAL ARRAY cost
m.oldfile=SELECT()

IF PARAMETER()=0
   m.id=0
ENDIF

SET TALK OFF

&&OPEN FILE # 1

IF USED("CFEITMP")
   SELECT cfeitmp
   SET ORDER TO TAG hmcfeid OF hmcfei.cdx

ELSE

   SELECT 0
   USE hmcfei ALIAS cfeitmp AGAIN
   SET ORDER TO TAG hmcfeid OF hmcfei.cdx

ENDIF

&&OPEN FILE # 2

IF USED("CFETMP")
   SELECT cfetmp
   SET ORDER TO TAG hmcfid OF hmcfe.cdx

ELSE

   SELECT 0
   USE hmcfe ALIAS cfetmp AGAIN
   SET ORDER TO TAG hmcfid OF hmcfe.cdx

ENDIF

&& OPEN FILE #3

IF USED("CFTMP")
   SELECT cftmp
   SET ORDER TO TAG hmcfid OF hmcf.cdx

ELSE

   SELECT 0
   USE hmcf ALIAS cftmp AGAIN
   SET ORDER TO hmcfid OF hmcf.cdx
ENDIF

IF m.id>0

   SET FILTER TO hmcfid = m.id

ENDIF

  SELECT cfetmp
  SET RELATION TO hmcfeid INTO cfeitmp ADDITIVE

  SELECT cftmp
  SET RELATION TO hmcfid INTO cfetmp ADDITIVE
  SET SKIP TO cfeitmp, cfetmp

  ** Show fields from grandparent (HMCFEI), parent (HMCFE) and child (H
=> MCF)**
  BROWSE FIELDS cftmp.hmcf:H="FACTORS", cfetmp.hmcfe:H="ELEMENTS",;
      cfeitmp.hmcfei:H="ITEMS", cfeitmp.hmcfeicost:H="COSTS", cfeitmp.h
=> mcfeiunit:H="UNITS";
      NOMODIFY NOAPPEND NODELETE NORMAL TITLE "COST FACTORS"

  SCATTER FIELDS cftmp.hmcf, cfetmp.hmcfe, cfeitmp.hmcfei, cfeitmp.hmcf
=> eicost, cfeitmp.hmcfeiunit;
      TO cost

  SET RELATION TO
  IF USED('cftmp')
     SELECT cftmp
     USE
  ENDIF
  IF USED('cfetmp')
     SELECT cfetmp
     USE
  ENDIF
  IF USED('cfeitmp')
     SELECT cfeitmp
     USE
  ENDIF
  *CLOSE DATABASE CFTMP,CFETMP,CFEITMP
  SELECT (m.oldfile)
```

```
RETURN
******* END ************


*
*       _QKFOVRPCS           m.Action VALID
*
*       Function Origin:
*
*       From Platform:       MS-DOS
*       From Screen:         HMTAB,     Record Number:   2
*       Variable:            m.Action
*       Called By:           VALID Clause
*       Object Type:         Push Button
*       Snippet Number:      1
*
*
*
FUNCTION _qkf0vrpcs     &&  m.Action VALID
#REGION 1
IF m.action = 1
   m.tabid = RECCOUNT()+1
   m.hmcomid = m.oldhmcomid
   m.hmatid=m.oldhmatid
   m.oldhmlc=""
   m.hmet=""
   m.hmetid=0
   m.hmetprob=0.000
   m.oldhmet=""
   m.hmwp=SPACE(80)
   m.hmwpid=0
   m.oldhmwp=""
   m.hmcf=""
   m.hmcfid=0
   m.oldhmcf=""
   m.hmcfeid=0
   m.hmcfe=""
   m.hmcfeiid=0
   m.hmcfei=""
   m.nmcfecost=0.00
   m.hmunit=m.oldunit
   m.prob=0.00
   m.wtavgerage=0.00
   IF EMPTY(m.hmunit)
      SHOW GET m.hmunit enabled
      m.hmunit = "GALLON"
   ENDIF
   SHOW GETS
   SHOW GET action disabled
   SHOW GET m.element enabled
   SHOW GET SAVE enabled
   SHOW GET m.hmlc enabled
   SHOW GET m.hmwp enabled
   SHOW GET m.hmet enabled
   SHOW GET m.hmetprob enabled
   SHOW GET m.hmcf enabled
   SHOW GET m.hmcfecost enabled
   SHOW GET m.prob enabled
   SHOW GET m.wtaverage enabled
   SHOW GET m.perp enabled
   SHOW GET m.perd enabled
   SHOW GET m.perq enabled
   m.adding = .T.

ELSE

   DO CASE

   CASE m.action = 2
      SHOW GETS
      SHOW GET action disabled
      SHOW GET m.element enabled
      SHOW GET SAVE enabled
      SHOW GET m.hmlc enabled
      SHOW GET m.hmwp enabled
      SHOW GET m.hmet enabled
      SHOW GET m.hmetprob enabled
      SHOW GET m.hmcf enabled
      SHOW GET m.hmcfecost enabled
      SHOW GET m.prob enabled
      SHOW GET m.wtaverage enabled
      *     SHOW GET m.hmunit ENABLED
      SHOW GET m.perp enabled
      SHOW GET m.perd enabled
      SHOW GET m.perq enabled

   CASE m.action = 3
      IF !EOF()
         SKIP
      ENDIF
      IF EOF()
         ?? CHR( 7 )
         WAIT "Last record" WINDOW NOWAIT
         GO BOTTOM
      ENDIF

   CASE m.action = 4
      IF !BOF()
         SKIP -1
      ENDIF
      IF BOF()
         ?? CHR( 7 )
         WAIT "First record" WINDOW NOWAIT
         GO TOP
      ENDIF

   CASE m.action = 5
      GO TOP

   CASE m.action=6
      GO BOTTOM

   CASE m.action = 7
      CLEAR READ
   ENDCASE
ENDIF

IF m.action >2 AND m.action < 7

   SCATTER MEMVAR

   IF m.hmlcid>0
      m.hmlc=get_hmlcn(m.hmlcid)
      m.oldhmlc=m.hmlc
   ELSE
      m.hmlc=""
      m.oldhmlc=""
   ENDIF

   IF m.hmwpid>0
      m.hmwp=get_hmwpn(m.hmwpid)
```

```
1053        m.oldhmwp=m.hmwp
1054     ELSE
1055        m.hmwp=""
1056        m.oldhmwp=""
1057     ENDIF
1058
1059     IF m.hmetid>0
1060        m.hmet=get_hmetn(m.hmetid)
1061     ELSE
1062        m.hmet=""
1063        m.oldhmet=""
1064     ENDIF
1065
1066     IF m.hmcfid>0
1067        m.hmcf=get_cfar(m.hmcfid)
1068        m.oldhmcf=m.hmcf
1069     ELSE
1070        m.hmcf=""
1071        m.oldhmcf=""
1072     ENDIF
1073
1074     IF m.hmcfeid>0
1075        m.hmcfe=get_cfar1(m.hmcfeid)
1076
1077     ELSE
1078        m.hmcfe=""
1079
1080     ENDIF
1081
1082     IF m.hmcfeiid>0
1083        m.hmcfei=get_ei(m.hmcfeiid)
1084
1085     ELSE
1086        m.hmcfei=""
1087
1088     ENDIF
1089
1090       SHOW GETS
1091       SHOW GET action enabled
1092       SHOW GET m.element disabled
1093       SHOW GET SAVE disabled
1094       SHOW GET m.hmlc disabled
1095       SHOW GET m.hmwp disabled
1096       SHOW GET m.hmet disabled
1097       SHOW GET m.hmetprob disabled
1098       SHOW GET m.hmcf disabled
1099       SHOW GET m.hmcfecost disabled
1100       SHOW GET m.prob disabled
1101       * SHOW GET m.hmunit DISABLED
1102       SHOW GET m.perp disabled
1103       SHOW GET m.perd disabled
1104       SHOW GET m.perq disabled
1105  ENDIF
1106
1107
1108  *
1109  *        _QKF0VRPYS            m.hmlc WHEN
1110  *
1111  *        Function Origin:
1112  *
1113  *        From Platform:        MS-DOS
1114  *        From Screen:          HMTAB,      Record Number:    8
1115  *        Variable:             m.hmlc
1116  *        Called By:            WHEN Clause
1117  *        Object Type:          Field
1118  *        Snippet Number:       2
1119  *
1120  *
1121  *
1122  *
1123  FUNCTION _qkf0vrpys     && m.hmlc WHEN
1124  #REGION 1
1125  m.oldhmlc=m.hmlc
1126
1127
1128  *
1129  *
1130  *        _QKF0VRQ1C            m.hmlc VALID
1131  *
1132  *        Function Origin:
1133  *
1134  *        From Platform:        MS-DOS
1135  *        From Screen:          HMTAB,      Record Number:    8
1136  *        Variable:             m.hmlc
1137  *        Called By:            VALID Clause
1138  *        Object Type:          Field
1139  *        Snippet Number:       3
1140  *
1141  *
1142  *
1143  FUNCTION _qkf0vrq1c     && m.hmlc VALID
1144  #REGION 1
1145  IF m.oldhmlc <> m.hmlc
1146     IF NOT EMPTY(m.hmlc)
1147        m.hmlc=ALLTRIM(m.hmlc)
1148        m.hmlc=IIF(m.hmlc="?","",UPPER(m.hmlc))
1149        m.hmlc=get_hmlc(m.hmlc)
1150        IF NOT EMPTY(m.hmlc)
1151
1152           SELECT DISTINCT hmlc.hmlcid;
1153               FROM hmlc;
1154               WHERE hmlc.hmlc IN (m.hmlc);
1155               INTO ARRAY X
1156           m.hmlcid=X[1]
1157        ENDIF
1158     ENDIF
1159     SHOW GETS
1160  ENDIF
1161
1162
1163  *
1164  *
1165  *        _QKF0VRQ4T            m.hmwp WHEN
1166  *
1167  *        Function Origin:
1168  *
1169  *        From Platform:        MS-DOS
1170  *        From Screen:          HMTAB,      Record Number:   10
1171  *        Variable:             m.hmwp
1172  *        Called By:            WHEN Clause
1173  *        Object Type:          Field
1174  *        Snippet Number:       4
1175  *
1176  *
1177  *
1178  FUNCTION _qkf0vrq4t     && m.hmwp WHEN
1179  #REGION 1
1180  m.oldhmwp=m.hmwp
1181
1182
1183  *
1184  *
```

```
*
*       _QKF0VRQ72              m.hmwp VALID
*
*       Function Origin:
*
*       From Platform:          MS-DOS
*       From Screen:            HMTAB,      Record Number:   10
*       Variable:               m.hmwp
*       Called By:              VALID Clause
*       Object Type:            Field
*       Snippet Number:         5
*
*
FUNCTION _qkf0vrq72     && m.hmwp VALID
#REGION 1
IF m.oldhmwp <> m.hmwp
  IF NOT EMPTY(m.hmwp)
     m.hmwp=ALLTRIM(m.hmwp)
     m.hmwp=IIF(m.hmwp="?","",UPPER(m.hmwp))
     m.hmwp=ALLTRIM(get_hmwp(m.hmwp))
     IF NOT EMPTY(m.hmwp)
        SELECT hmwp.hmwpid;
            FROM hmwp;
            WHERE hmwp.hmwp IN (m.hmwp);
            INTO ARRAY X
        m.hmwpid=X[1]
     ELSE
        m.hmwp = SPACE(80)
     ENDIF
  ENDIF
  SHOW GETS
ENDIF


*
*       _QKF0VRQAL              m.hmet WHEN
*
*       Function Origin:
*
*       From Platform:          MS-DOS
*       From Screen:            HMTAB,      Record Number:   12
*       Variable:               m.hmet
*       Called By:              WHEN Clause
*       Object Type:            Field
*       Snippet Number:         6
*
*
FUNCTION _qkf0vrqal     && m.hmet WHEN
#REGION 1
m.oldhmet=m.hmet


*
*       _QKF0VRQC2              m.hmet VALID
*
*       Function Origin:
*
*       From Platform:          MS-DOS
*       From Screen:            HMTAB,      Record Number:   12
*       Variable:               m.hmet
*       Called By:              VALID Clause
*       Object Type:            Field
*       Snippet Number:         7
*
*
FUNCTION _qkf0vrqc2     && m.hmet VALID
#REGION 1
IF m.oldhmet <> m.hmet
  IF NOT EMPTY(m.hmet)
     m.hmet=ALLTRIM(m.hmet)
     m.hmet=IIF(m.hmet="?","",UPPER(m.hmet))
     m.hmet=ALLTRIM(get_hmet(m.hmet))
     IF NOT EMPTY(m.hmet)

        SELECT DISTINCT hmet.hmetid;
            FROM hmet;
            WHERE hmet.hmet IN (m.hmet);
            INTO ARRAY X
        m.hmetid=X[1]

     ENDIF
  ENDIF
  SHOW GETS
ENDIF


*
*       _QKF0VRQFR              m.hmcf WHEN
*
*       Function Origin:
*
*       From Platform:          MS-DOS
*       From Screen:            HMTAB,      Record Number:   15
*       Variable:               m.hmcf
*       Called By:              WHEN Clause
*       Object Type:            Field
*       Snippet Number:         8
*
*
FUNCTION _qkf0vrqfr     && m.hmcf WHEN
#REGION 1
m.oldhmcf=m.hmcf


*
*       _QKF0VRQH7              m.hmcf VALID
*
*       Function Origin:
*
*       From Platform:          MS-DOS
*       From Screen:            HMTAB,      Record Number:   15
*       Variable:               m.hmcf
*       Called By:              VALID Clause
*       Object Type:            Field
*       Snippet Number:         9
*
*
FUNCTION _qkf0vrqh7     && m.hmcf VALID
#REGION 1
IF m.oldhmcf <> m.hmcf
  IF NOT EMPTY(m.hmcf)
     m.hmcf=ALLTRIM(m.hmcf)
     m.hmcf=IIF(m.hmcf="?","",UPPER(m.hmcf))
     m.hmcf=ALLTRIM(get_cf(m.hmcf))
```

```
1317        IF NOT EMPTY(m.hmcf)
1318
1319            SELECT DISTINCT hmcf.hmcfid;
1320                FROM hmcf;
1321                WHERE hmcf.hmcf IN (m.hmcf);
1322                INTO ARRAY X
1323
1324            m.hmcfid=X[1]
1325            m.hmcfeid=0
1326            m.hmcfe=""
1327            m.hmcfeiid=0
1328            m.hmcfei=""
1329            m.hmcfecost=0
1330            *                   m.hmunit=""
1331            m.prob=0
1332        ENDIF
1333      ENDIF
1334      SHOW GETS
1335    ENDIF
1336
1337
1338  *
1339  *
1340  *           _QKF0VRQKZ            m.element VALID
1341  *
1342  *           Function Origin:
1343  *
1344  *           From Platform:        MS-DOS
1345  *           From Screen:          HMTAB,     Record Number:   16
1346  *           Variable:             m.element
1347  *           Called By:            VALID Clause
1348  *           Snippet Number:       10
1349  *
1350  *
1351  *
1352  FUNCTION _qkf0vrqkz     &&  m.element VALID
1353  #REGION 1
1354  RELEASE cost
1355  DECLARE cost[5]
1356  cost[2]=""
1357  =rel(m.hmcfid)
1358  IF !EMPTY(cost[2])
1359      m.hmcf=cost[1]
1360      m.hmcfid=get_hmcfid(m.hmcf)
1361      m.hmcfe=cost[2]
1362      m.hmcfeid=get_eid(m.hmcfe)
1363      m.hmcfecost=cost[4]
1364      * m.hmunit=COST[5]
1365    IF !EMPTY(cost[3])
1366        m.hmcfei=cost[3]
1367        m.hmcfeiid=get_eiid(m.hmcfei)
1368    ELSE m.hmcfei=""
1369        m.hmcfeiid=0
1370    ENDIF
1371      SHOW GETS
1372  ENDIF
1373
1374
1375  *
1376  *
1377  *           _QKF0VRQRG            m.Save VALID
1378  *
1379  *           Function Origin:
1380  *
1381  *           From Platform:        MS-DOS
1382  *           From Screen:          HMTAB,     Record Number:   26
```



```
1383  *           Variable:             m.Save
1384  *           Called By:            VALID Clause
1385  *           Object Type:          Push Button
1386  *           Snippet Number:       11
1387  *
1388  *
1389  *
1390  FUNCTION _qkf0vrqrg     &&  m.Save VALID
1391  #REGION 1
1392  m.wtaverage = m.hmcfecost * m.prob
1393  IF m.save = 1 && Selected Save Button
1394    IF m.adding && Adding a new record
1395        APPEND BLANK
1396        GATHER MEMVAR
1397    ELSE
1398        DO CHANGE
1399      IF m.change && Changing an old record
1400          GATHER MEMVAR
1401      ENDIF
1402    ENDIF
1403      m.oldunit = m.hmunit
1404  ENDIF
1405
1406    SCATTER MEMVAR
1407
1408  IF m.hmlcid>0
1409      m.hmlc=get_hmlcn(m.hmlcid)
1410      m.oldhmlc=m.hmlc
1411  ELSE
1412      m.hmlc=""
1413      m.oldhmlc=""
1414  ENDIF
1415
1416  IF m.hmwpid>0
1417      m.hmwp=get_hmwpn(m.hmwpid)
1418      m.oldhmwp=m.hmwp
1419  ELSE
1420      m.hmwp=""
1421      m.oldhmwp=""
1422  ENDIF
1423
1424  IF hmetid>0
1425      m.hmet=get_hmetn(m.hmetid)
1426      m.oldhmet=m.hmet
1427  ELSE
1428      m.hmet=""
1429      m.oldhmet=""
1430  ENDIF
1431
1432  IF m.hmcfid>0
1433      m.hmcf=get_cfar(m.hmcfid)
1434      m.oldhmcf=m.hmcf
1435  ELSE
1436      m.hmcf=""
1437      m.oldhmcf=""
1438  ENDIF
1439
1440  IF m.hmcfeid>0
1441      m.hmcfe=get_cfar1(m.hmcfeid)
1442  ELSE
1443      m.hmcfe=""
1444  ENDIF
1445
1446  IF m.hmcfeiid>0
1447      m.hmcfei=get_ei(m.hmcfeiid)
1448  ELSE
```

```
1449        m.hmcfei=""
1450      ENDIF
1451      SHOW GETS
1452      SHOW GET action enabled
1453      SHOW GET m.element disabled
1454      SHOW GET SAVE disabled
1455      SHOW GET m.hmlc disabled
1456      SHOW GET m.hmmp disabled
1457      SHOW GET m.hmet disabled
1458      SHOW GET m.hmetprob disabled
1459      SHOW GET m.hmcf disabled
1460      SHOW GET m.hmcfecost disabled
1461      SHOW GET m.prob disabled
1462      SHOW GET m.hmunit disabled
1463      SHOW GET m.perp disabled
1464      SHOW GET m.perd disabled
1465      SHOW GET m.perq disabled
1466      m.adding=.F.
1467      m.change=.F.
1468      *: EOF: HMTAB.AC1
```

```
*
*       10/27/93                HMWP.SPR                    14:48:59
*
*       Author's Name
*
*       Copyright (c) 1993 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENSCRN.
*
*

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*                   MS-DOS Window definitions
*

IF NOT WEXIST("hmwp") ;
      OR UPPER(WTITLE("HMWP")) == "HMWP.PJX" ;
      OR UPPER(WTITLE("HMWP")) == "HMWP.SCX" ;
      OR UPPER(WTITLE("HMWP")) == "HMWP.MNX" ;
      OR UPPER(WTITLE("HMWP")) == "HMWP.PRG" ;
      OR UPPER(WTITLE("HMWP")) == "HMWP.FRX" ;
      OR UPPER(WTITLE("HMWP")) == "HMWP.QPR"
   DEFINE WINDOW hmwp ;
      FROM INT((SROW()-15)/2),INT((SCOL()-64)/2) ;
      TO INT((SROW()-15)/2)+14,INT((SCOL()-64)/2)+63 ;
      NOFLOAT ;
      NOCLOSE ;
      SHADOW ;
      nominimize ;
      DOUBLE ;
      COLOR SCHEME 1
ENDIF

*
*                HMWP/MS-DOS Setup Code - SECTION 2
*

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
m.oldescape = SET( "ESCAPE" )
SET ESCAPE OFF
m.adding = .F.
m.change = .F.

IF !USED('HMWP')
   SELECT 0
   USE hmwp
   SET ORDER TO TAG hmwp OF hmwp.cdx
ELSE
   SELECT hmwp
   SET ORDER TO TAG hmwp
ENDIF

************************************************
* Check see if the last record is defined
IF TYPE( "m.LastRec" ) = "U"

   * Start with the first record
   GO TOP
   m.lastrec = RECNO()

ELSE
   * Start on the last record used
   GO m.lastrec
ENDIF
************************************************
SCATTER MEMVAR

*
*                HMWP/MS-DOS Screen Layout
*

#REGION 1
IF WVISIBLE("hmwp")
   ACTIVATE WINDOW hmwp SAME
ELSE
   ACTIVATE WINDOW hmwp NOSHOW
ENDIF
@ 2,50 GET m.action ;
      PICTURE "@*VN \<Add;\<Edit;\!\<Next;\<Previous;\?E\<xit" ;
      SIZE 1,10,1 ;
      DEFAULT 1 ;
      VALID _qkf0vr92i()
@ 2,11 GET m.hmwpid ;
      SIZE 1,10 ;
      DEFAULT " " ;
      DISABLE
@ 4,11 GET m.hmwp ;
      SIZE 6,34 ;
      DEFAULT " " ;
      PICTURE "@!" ;
      VALID _qkf0vr99i() ;
      DISABLE
@ 11,14 GET m.save ;
      PICTURE "@*HN \<Save;\<Cancel" ;
      SIZE 1,8,1 ;
      DEFAULT 1 ;
      VALID _qkf0vr9c5() ;
      DISABLE
@ 1,0 TO 12,46
@ 2,3 SAY "ID NUM:" ;
```

```
133         SIZE 1,7, 0
134      @ 4,2 SAY "PROCESS:" ;
135         SIZE 1,8, 0
136      @ 0,16 SAY "HM PROCESSES" ;
137         SIZE 1,12, 0
138
139   IF NOT WVISIBLE("hmwp")
140      ACTIVATE WINDOW hmwp
141   ENDIF
142
143   READ CYCLE
144
145   RELEASE WINDOW hmwp
146
147   #REGION 0
148   IF m.talkstat = "ON"
149      SET TALK ON
150   ENDIF
151   IF m.compstat = "ON"
152      SET COMPATIBLE ON
153   ENDIF
154
155
156   *
157   *
158   *              HMWP/MS-DOS Cleanup Code
159   *
160   *
161   *
162
163   #REGION 1
164   IF USED('HMWP')
165      USE
166   ENDIF
167   POP KEY ALL
168   SET ESCAPE &oldescape
169   ********* End of Main Body - Entry Cleanup
170
171   ******************
172
173
174   *
175   *
176   *          HMWP/MS-DOS Supporting Procedures and Functions
177   *
178   *
179   *
180
181   #REGION 1
182   PROCEDURE CHANGE
183   ******************
184   m.oldexact = SET( "EXACT" )
185   SET EXACT ON
186   m.change =TRIM(hmwp.hmwp) <> TRIM( m.hmwp)
187   SET EXACT &oldexact
188   RETURN m.change
189
190
191   *
192   *
193   *        _QKF0VR92I          m.Action VALID
194   *
195   *        Function Origin:
196   *
197   *        From Platform:      MS-DOS
198   *        From Screen:        HMWP,     Record Number:    2
199   *        Variable:           m.Action
200   *        Called By:          VALID Clause
201   *        Object Type:        Push Button
202   *        Snippet Number:     1
203   *
204   *
205   *
206   FUNCTION _qkf0vr92i     &&  m.Action VALID
207   #REGION 1
208   IF m.action = 1
209      SCATTER MEMVAR BLANK
210      m.hmwpid=RECCOUNT()+1
211      SHOW GETS
212      SHOW GET m.hmwp enabled
213      SHOW GET action disabled
214      SHOW GET SAVE enabled
215      m.adding = .T.
216
217   ELSE
218
219      DO CASE
220
221      CASE m.action = 2
222         SHOW GETS
223         SHOW GET m.hmwp enabled
224         SHOW GET action disabled
225         SHOW GET SAVE enabled
226
227      CASE m.action = 3
228         SKIP
229         IF EOF()
230            ?? CHR( 7 )
231            WAIT "Last record" WINDOW NOWAIT
232            SKIP -1
233         ELSE
234            SCATTER MEMVAR
235            SHOW GETS
236         ENDIF
237
238      CASE m.action = 4
239         SKIP -1
240         IF BOF()
241            ?? CHR( 7 )
242            WAIT "First record" WINDOW NOWAIT
243            SKIP
244         ELSE
245            SCATTER MEMVAR
246            SHOW GETS
247         ENDIF
248
249      CASE m.action = 5
250         CLEAR READ
251      ENDCASE
252   ENDIF
253
254
255   *
256   *
257   *        _QKF0VR99I          m.hmwp VALID
258   *
259   *        Function Origin:
260   *
261   *        From Platform:      MS-DOS
262   *        From Screen:        HMWP,     Record Number:    4
263   *        Variable:           m.hmwp
264   *        Called By:          VALID Clause
```

```
265     *         | Object Type:         Field
266     *         | Snippet Number:      2
267     *         |
268     *         |
269     *
270     FUNCTION _qkf0vr99i     &&  m.hmwp VALID
271     #REGION 1
272     IF m.adding
273        m.oldrec = RECNO()
274        SEEK m.hmwp
275        IF FOUND()
276           =errmsg("Record already exists",1)
277           SCATTER FIELD hmwp MEMVAR BLANK
278           _CUROBJ = OBJNUM(m.hmwp)
279        ENDIF
280        GO m.oldrec
281     ENDIF
282     SHOW GETS
283
284
285     *
286     *
287     *         _QKF0VR9C5           m.Save VALID
288     *
289     *         Function Origin:
290     *
291     *         From Platform:       MS-DOS
292     *         From Screen:         HMWP,     Record Number:    5
293     *         Variable:            m.Save
294     *         Called By:           VALID Clause
295     *         Object Type:         Push Button
296     *         Snippet Number:      3
297     *
298     *
299     *
300     FUNCTION _qkf0vr9c5     &&  m.Save VALID
301     #REGION 1
302     m.notsave = .F.
303     IF m.save = 1 && Selected Save Button
304        IF EMPTY(m.hmwp)
305           =errmsg("Data empty, could not save!!",2)
306           _CUROBJ = OBJNUM(m.hmwp)
307           m.notsave = .T.
308        ELSE
309           IF m.adding && Adding a new record
310              APPEND BLANK
311              GATHER MEMVAR
312           ELSE
313
314              IF m.change && Changing an old record
315                 GATHER MEMVAR
316              ENDIF
317           ENDIF
318        ENDIF
319     ELSE
320        SCATTER MEMVAR
321     ENDIF
322
323     SHOW GETS
324     IF m.notsave
325        SHOW GET m.hmwp enabled
326        SHOW GET action disabled
327        SHOW GET SAVE enabled
328     ELSE
329        SHOW GET m.hmwp disabled
330        SHOW GET action enabled
331        SHOW GET SAVE disabled
332        m.adding = .F.
333        m.change = .F.
334     ENDIF
335
336     *: EOF: HMWP.AC1
```

```
  1  *:*************************************************************************
=> ******
  2  *:
  3  *: Procedure file: C:\HAZMAT\GHM\WORK\MEMOWIN.SPR
  4  *:
  5  *:         System: Hazardous Material Life Cycle Cost Model
  6  *:         Author: Naval Health Research Center
  7  *:      Copyright (c) 1993, Naval Health Research Center
  8  *:  Last modified: 09/10/93      8:12
  9  *:
 10  *:  Procs & Fncts: _QIGOIXOXD()
 11  *:                : _QIGOIXPJO()
 12  *:
 13  *:          Calls: _QIGOIXOXD()       (function  in MEMOWIN.SPR)
 14  *:                : _QIGOIXPJO()      (function  in MEMOWIN.SPR)
 15  *:
 16  *:      Documented 09/14/93 at 08:20              FoxDoc  version 2.10f
 17  *:*************************************************************************
=> ******
 18        *       ************************************************************
 19        *       *
 20        *       * 08/17/93             MEMOWIN.SPR                08:50:04
 21        *       *
 22        *       ************************************************************
 23        *       *
 24        *       * G. Pang
 25        *       *
 26        *       * Copyright (c) 1993 Company Name
 27        *       * Address
 28        *       * City,     Zip
 29        *       *
 30        *       * Description:
 31        *       * This program was automatically generated by GENSCRN.
 32        *       *
 33        *       ************************************************************
 34
 35        PARAMETERS TEXT, TITLE
 36        DO CASE
 37        CASE _dos
 38
 39
 40           *       ************************************************************
 41           *       *
 42           *       *          MEMOWIN/MS-DOS Setup Code - SECTION 1
 43           *       *
 44           *       ************************************************************
 45           *
 46
 47           #REGION 1
 48           PRIVATE ALL
 49           DO CASE
 50              CLEAR
 51           CASE PARAMETER() = 0
 52              m.title = ""
 53              m.text = ""
 54           CASE PARAMETER() = 1
 55              m.title = ""
 56           ENDCASE
 57
 58
 59
 60           #REGION 0
 61           REGIONAL m.currarea, m.talkstat, m.compstat
 62
 63           IF SET("TALK") = "ON"
 64              SET TALK OFF
```

```
 65              m.talkstat = "ON"
 66           ELSE
 67              m.talkstat = "OFF"
 68           ENDIF
 69           m.compstat = SET("COMPATIBLE")
 70           SET COMPATIBLE FOXPLUS
 71
 72           m.currarea = SELECT()
 73
 74
 75           *       ************************************************************
 76           *       *
 77           *       *                MS-DOS Window definitions
 78           *       *
 79           *       ************************************************************
 80           *
 81
 82           IF NOT WEXIST("memowin") ;
 83                 OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.PJX" ;
 84                 OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.SCX" ;
 85                 OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.MNX" ;
 86                 OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.PRG" ;
 87                 OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.FRX" ;
 88                 OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.QPR"
 89              DEFINE WINDOW memowin ;
 90                 FROM INT((SROW()-26)/2),INT((SCOL()-80)/2) ;
 91                 TO INT((SROW()-26)/2)+25,INT((SCOL()-80)/2)+79 ;
 92                 NOFLOAT ;
 93                 NOCLOSE ;
 94                 SHADOW ;
 95                 nominimize ;
 96                 NONE ;
 97                 COLOR SCHEME 1
 98           ENDIF
 99
100
101           *       ************************************************************
102           *       *
103           *       *                MEMOWIN/MS-DOS Screen Layout
104           *       *
105           *       ************************************************************
106           *
107
108           #REGION 1
109           IF WVISIBLE("memowin")
110              ACTIVATE WINDOW memowin SAME
111           ELSE
112              ACTIVATE WINDOW memowin NOSHOW
113           ENDIF
114           @ 1,2 GET m.title ;
115              SIZE 1,74 ;
116              DEFAULT " " ;
117              PICTURE "@!" ;
118              DISABLE
119           @ 2,2 EDIT m.text ;
120              SIZE 20,77,0 ;
121              DEFAULT " " ;
122              SCROLL
123           @ 23,26 GET m.save ;
124              PICTURE "@*HT \<Print;\<Another;\<Quit" ;
125              SIZE 1,9,1 ;
126              DEFAULT 1 ;
127              VALID _qig0ixoxd()
128
129           IF NOT WVISIBLE("memowin")
130              ACTIVATE WINDOW memowin
```

```
131     ENDIF
132
133     READ CYCLE
134
135     RELEASE WINDOW memowin
136     SELECT (m.currarea)
137
138
139     #REGION 0
140     IF m.talkstat = "ON"
141        SET TALK ON
142     ENDIF
143     IF m.compstat = "ON"
144        SET COMPATIBLE ON
145     ENDIF
146
147
148     *       ************************************************************
149     *       *
150     *       *                MEMOWIN/MS-DOS Cleanup Code
151     *       *
152     *       ************************************************************
153     *
154
155     #REGION 1
156     RETURN
157
158  CASE _windows
159
160
161     *       ************************************************************
162     *       *
163     *       *          MEMOWIN/Windows Setup Code - SECTION 1
164     *       *
165     *       ************************************************************
166     *
167
168     #REGION 1
169     PRIVATE ALL
170     DO CASE
171        CLEAR
172     CASE PARAMETER() = 0
173        m.title = ""
174        m.text = ""
175     CASE PARAMETER() = 1
176        m.title = ""
177     ENDCASE
178
179
180
181     #REGION 0
182     REGIONAL m.currarea, m.talkstat, m.compstat
183
184     IF SET("TALK") = "ON"
185        SET TALK OFF
186        m.talkstat = "ON"
187     ELSE
188        m.talkstat = "OFF"
189     ENDIF
190     m.compstat = SET("COMPATIBLE")
191     SET COMPATIBLE FOXPLUS
192
193     m.currarea = SELECT()
194
195
196     *       ************************************************************
197     *       *
198     *       *                Windows Window definitions
199     *       *
200     *       ************************************************************
201     *
202
203  IF NOT WEXIST("memowin") ;
204        OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.PJX" ;
205        OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.SCX" ;
206        OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.MNX" ;
207        OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.PRG" ;
208        OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.FRX" ;
209        OR UPPER(WTITLE("MEMOWIN")) == "MEMOWIN.QPR"
210     DEFINE WINDOW memowin ;
211        AT  0.000, 0.000  ;
212        SIZE 26.769,80.000 ;
213        font "MS Sans Serif", 8 ;
214        STYLE "B" ;
215        NOFLOAT ;
216        NOCLOSE ;
217        SHADOW ;
218        nominimize ;
219        NONE
220     MOVE WINDOW memowin CENTER
221  ENDIF
222
223
224     *       ************************************************************
225     *       *
226     *       *              MEMOWIN/Windows Screen Layout
227     *       *
228     *       ************************************************************
229     *
230
231  #REGION 1
232  IF WVISIBLE("memowin")
233     ACTIVATE WINDOW memowin SAME
234  ELSE
235     ACTIVATE WINDOW memowin NOSHOW
236  ENDIF
237   @ 1.000,4.667 GET m.title ;
238        SIZE 1.154,77.000 ;
239        DEFAULT " " ;
240        font "MS Sans Serif", 8 ;
241        PICTURE "@I" ;
242        DISABLE
243   @ 3.692,5.000 EDIT m.text ;
244        SIZE 13.769,75.800,0.000 ;
245        DEFAULT " " ;
246        font "MS Sans Serif", 8 ;
247        SCROLL
248   @ 21.077,23.667 GET m.save ;
249        PICTURE "@*HT \<Print;\<Another;\<Quit" ;
250        SIZE 1.769,8.500,1.000 ;
251        DEFAULT 1 ;
252        font "MS Sans Serif", 8 ;
253        STYLE "B" ;
254        VALID _qig0ixpjo()
255
256  IF NOT WVISIBLE("memowin")
257     ACTIVATE WINDOW memowin
258  ENDIF
259
260   READ CYCLE
261
262   RELEASE WINDOW memowin
```

```
263         SELECT (m.currarea)
264
265
266         #REGION 0
267         IF m.talkstat = "ON"
268            SET TALK ON
269         ENDIF
270         IF m.compstat = "ON"
271            SET COMPATIBLE ON
272         ENDIF
273
274
275         *       ************************************************************
276         *       *
277         *       *               MEMOWIN/Windows Cleanup Code
278         *       *
279         *       ************************************************************
280         *
281
282         #REGION 1
283   <-----RETURN
284      |
285      ENDCASE
286
287
288         *       ************************************************************
289         *       *
290         *       * _QIGOIXOXD           m.save VALID
291         *       *
292         *       * Function Origin:
293         *       *
294         *       * From Platform:       MS-DOS
295         *       * From Screen:         MEMOWIN,     Record Number:    4
296         *       * Variable:            m.save
297         *       * Called By:           VALID Clause
298         *       * Snippet Number:      1
299         *       *
300         *       ************************************************************
301         *
312   *!*************************************************************************
=> ******
313   *!
314   *!       Function: _QIGOIXOXD()
315   *!
316   *!      Called by: MEMOWIN.SPR
317   *!               : MEMOWIN.PRG
318   *!
319   *!          Calls: W_PRINT.SPR
320   *!
321   *!*************************************************************************
=> ******
312      FUNCTION _qigOixoxd     &&  m.save VALID
313      #REGION 1
314      IF m.save = 1                && Print
315         DO w_print.spr WITH m.text
316   <-----RETURN m.save
317      ENDIF
318      IF m.save = 2                && Another
319   <-----RETURN m.save
320      ENDIF
321      IF m.save = 3                && Quit
322   <-----RETURN m.save
323      ENDIF
324
325
326         *       ************************************************************
```

```
327         *       *
328         *       * _QIGOIXPJO           m.save VALID
329         *       *
330         *       * Function Origin:
331         *       *
332         *       * From Platform:       Windows
333         *       * From Screen:         MEMOWIN,     Record Number:    8
334         *       * Variable:            m.save
335         *       * Called By:           VALID Clause
336         *       * Snippet Number:      2
337         *       *
338         *       ************************************************************
339         *
350   *!*************************************************************************
=> ******
351   *!
352   *!       Function: _QIGOIXPJO()
353   *!
354   *!      Called by: MEMOWIN.SPR
355   *!               : MEMOWIN.PRG
356   *!
357   *!          Calls: W_PRINT.SPR
358   *!
359   *!*************************************************************************
=> ******
350      FUNCTION _qigOixpjo     &&  m.save VALID
351      #REGION 1
352      IF m.save = 1                && Print
353         DO w_print.spr WITH m.text
354         m.quit = .F.
355      ENDIF
356      IF m.save = 2                && Another
357         m.quit = .F.
358      ENDIF
359      IF m.save = 3                && Quit
360         m.quit = .T.
361      ENDIF
362      *: EOF: MEMOWIN.AC1
```

```
*
*    10/27/93          W_PRINT.SPR          14:49:39
*
*    Author's Name
*
*    Copyright (c) 1993 Company Name
*    Address
*    City,     Zip
*
*    Description:
*    This program was automatically generated by GENSCRN.
*

PARAMETERS TEXT

*
*
*        W_PRINT/MS-DOS Setup Code - SECTION 1
*
*
*

#REGION 1
IF PARAMETER() = 0
    RETURN
ENDIF


#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

*
*
*        MS-DOS Window definitions
*
*
*

IF NOT WEXIST("w_prn") ;
        OR UPPER(WTITLE("W_PRN")) == "W_PRN.PJX" ;
        OR UPPER(WTITLE("W_PRN")) == "W_PRN.SCX" ;
        OR UPPER(WTITLE("W_PRN")) == "W_PRN.MNX" ;
        OR UPPER(WTITLE("W_PRN")) == "W_PRN.PRG" ;
        OR UPPER(WTITLE("W_PRN")) == "W_PRN.FRX" ;
        OR UPPER(WTITLE("W_PRN")) == "W_PRN.QPR"
    DEFINE WINDOW w_prn ;
        FROM INT((SROW()-13)/2),INT((SCOL()-43)/2) ;
        TO INT((SROW()-13)/2)+12,INT((SCOL()-43)/2)+42 ;
        TITLE "Print Option" ;
        FLOAT ;
        NOCLOSE ;
        SHADOW ;
        nominimize ;
        DOUBLE ;
        COLOR SCHEME 5
ENDIF


*
*
*        W_PRINT/MS-DOS Setup Code - SECTION 2
*
*
*

#REGION 1
PRIVATE mchoice, mprintfile, mfilename, i
mchoice = 1
mfilename = "hmlccm.txt"
mprintfile = mfilename


*
*
*        W_PRINT/MS-DOS Screen Layout
*
*
*

#REGION 1
IF WVISIBLE("w_prn")
    ACTIVATE WINDOW w_prn SAME
ELSE
    ACTIVATE WINDOW w_prn NOSHOW
ENDIF
@ 0,0 TO 3,40
@ 5,0 TO 8,40
@ 6,1 SAY "Filename:" ;
    SIZE 1,9, 0
@ 6,11 GET mprintfile ;
    SIZE 1,29 ;
    DEFAULT " " ;
    VALID _qkf0vs4nc()
@ 10,9 GET mbuttons ;
    PICTURE "@*HT OK;\?Cancel" ;
    SIZE 1,8,3 ;
    DEFAULT 1 ;
    VALID _qkf0vs4ph()
@ 1,17 GET mchoice ;
    PICTURE "@*RVN File;Printer" ;
    SIZE 1,11,0 ;
    DEFAULT 1 ;
    VALID _qkf0vs4rm()
@ 1,1 SAY "Print to: " ;
    SIZE 1,10, 0

IF NOT WVISIBLE("w_prn")
    ACTIVATE WINDOW w_prn
ENDIF

READ CYCLE

RELEASE WINDOW w_prn

#REGION 0
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
```

```
    SET COMPATIBLE ON
ENDIF


*
*           W_PRINT/MS-DOS Cleanup Code
*
*

#REGION 1
* print encounter
**************************


*
*           W_PRINT/MS-DOS Supporting Procedures and Functions
*
*

#REGION 1
FUNCTION printfile
**************************
mprintfile = ALLTRIM( mprintfile )
IF !EMPTY( mprintfile )
   * list to printer or file
   IF mprintfile  != "PRINT"
      * list to file
      SET PRINTER TO (mprintfile)
   ENDIF
   SET CONSOLE OFF
   SET PRINTER ON

   ?  m.text

   IF mprintfile = "PRINT"
      EJECT
   ENDIF
   SET PRINTER OFF
   SET PRINTER TO
   SET CONSOLE ON
ENDIF
RETURN


*
*           _QKFOVS4NC            mprintfile VALID
*
*           Function Origin:
*
*           From Platform:        MS-DOS
*           From Screen:          W_PRINT,     Record Number:    5
*           Variable:             mprintfile
*           Called By:            VALID Clause
*           Object Type:          Field
*           Snippet Number:       1
*
*

FUNCTION _qkf0vs4nc     &&  mprintfile VALID
#REGION 1
mfilename = mprintfile
RETURN .T.


*
*           _QKFOVS4PH            mbuttons VALID
*
*           Function Origin:
*
*           From Platform:        MS-DOS
*           From Screen:          W_PRINT,     Record Number:    6
*           Variable:             mbuttons
*           Called By:            VALID Clause
*           Object Type:          Push Button
*           Snippet Number:       2
*
*

FUNCTION _qkf0vs4ph     &&  mbuttons VALID
#REGION 1
IF mbuttons = 1
   = printfile()
ENDIF


*
*           _QKFOVS4RM            mchoice VALID
*
*           Function Origin:
*
*           From Platform:        MS-DOS
*           From Screen:          W_PRINT,     Record Number:    7
*           Variable:             mchoice
*           Called By:            VALID Clause
*           Object Type:          Radio Button
*           Snippet Number:       3
*
*

FUNCTION _qkf0vs4rm     &&  mchoice VALID
#REGION 1
IF mchoice = 1
   mprintfile = mfilename
   _CUROBJ = OBJNUM(mprintfile)
ELSE
   mprintfile = "PRINT"
ENDIF
SHOW GETS
RETURN .T.
*: EOF: W_PRINT.AC1
```

```
*
*       12/01/93              HMENU.MPR                14:37:41
*
*       Author's Name
*
*       Copyright (c) 1993 Company Name
*       Address
*       City,     Zip
*
*       Description:
*       This program was automatically generated by GENMENU.
*
*
*                          Menu Definition
*
*

set sysmenu to

set sysmenu automatic

define pad _msm_systm of _msysmenu prompt "\<System" color scheme 3 ;
    key alt+s, ""
define pad _msm_file of _msysmenu prompt "\<File" color scheme 3 ;
    key alt+f, ""
define pad _msm_edit of _msysmenu prompt "\<Edit" color scheme 3 ;
    key alt+e, ""
define pad _msm_data of _msysmenu prompt "\<Database" color scheme 3
=> ;
    key alt+d, ""
define pad _msm_recrd of _msysmenu prompt "\<Record" color scheme 3 ;
    key alt+r, ""
define pad _msm_prog of _msysmenu prompt "\<Program" color scheme 3 ;
    key alt+p, ""
define pad _msm_windo of _msysmenu prompt "\<Window" color scheme 3 ;
    key alt+w, ""
on pad _msm_systm of _msysmenu activate popup _msystem
on pad _msm_file of _msysmenu activate popup _mfile
on pad _msm_edit of _msysmenu activate popup _medit
on pad _msm_data of _msysmenu activate popup _mdata
on pad _msm_recrd of _msysmenu activate popup _mrecord
on pad _msm_prog of _msysmenu activate popup _mprog
on pad _msm_windo of _msysmenu activate popup _mwindow

define popup _msystem margin relative shadow color scheme 4
define bar _mst_help of _msystem prompt "\<Help..." ;
    key f1, "F1"
define bar _mst_sp100 of _msystem prompt "\-"
define bar _mst_calcu of _msystem prompt "\<Calculator"
define bar 4 of _msystem prompt "\-"
define bar 5 of _msystem prompt "\<HMLCCM"
on bar 5 of _msystem activate popup hmlccm

define popup hmlccm margin relative shadow color scheme 4
define bar 1 of hmlccm prompt "\<Cost Analysis"
define bar 2 of hmlccm prompt "\<OSHA Z-Table"
define bar 3 of hmlccm prompt "\<Reference Material"
define bar 4 of hmlccm prompt "System \<Maintenance"
define bar 5 of hmlccm prompt "\<Set Parameters" ;
    skip for (m.pass) != "SYSHAZ"
on bar 1 of hmlccm activate popup costanalys
on selection bar 2 of hmlccm do hmztab
on selection bar 3 of hmlccm do hmmsds
on bar 4 of hmlccm activate popup systemmain
on bar 5 of hmlccm activate popup setparamet

define popup costanalys margin relative shadow color scheme 4
define bar 1 of costanalys prompt "\<Build Hazmat Scenario"
on selection bar 1 of costanalys do hmsc

define popup systemmain margin relative shadow color scheme 4
define bar 1 of systemmain prompt "\<Back-Up (floppy)"
define bar 2 of systemmain prompt "\<Up-Load Data"
on selection bar 1 of systemmain ;
    do _qle0vcqdc ;
    in locfile("WORK\HMENU" ,"MPX;MPR|FXP;PRG" ,"Where is HMENU?")
on selection bar 2 of systemmain ;
    do _qle0vcqgq ;
    in locfile("WORK\HMENU" ,"MPX;MPR|FXP;PRG" ,"Where is HMENU?")

define popup setparamet margin relative shadow color scheme 4
define bar 1 of setparamet prompt "\<Materials" ;
    key ctrl+m, "CTRL+M"
define bar 2 of setparamet prompt "\<Life Cycle Phase" ;
    key ctrl+l, "CTRL+L"
define bar 3 of setparamet prompt "\<Process" ;
    key ctrl+w, "CTRL+W"
define bar 4 of setparamet prompt "E\<Xposure Type" ;
    key ctrl+x, "CTRL+X"
define bar 5 of setparamet prompt "Cost \<Factors " ;
    key ctrl+f, "CTRL+F"
define bar 6 of setparamet prompt "Cost Factor \<Elements " ;
    key ctrl+e, "CTRL+E"
define bar 7 of setparamet prompt "Cost Factor Element \<Items" ;
    key ctrl+i, "CTRL+I"
define bar 8 of setparamet prompt "\<Build Hazmat Table" ;
    key ctrl+b, "CTRL+B"
on selection bar 1 of setparamet do hmat.spr
on selection bar 2 of setparamet do hmlc.spr
on selection bar 3 of setparamet do hmwp.spr
on selection bar 4 of setparamet do hmet.spr
on selection bar 5 of setparamet do hmcf.spr
on selection bar 6 of setparamet do hmcfe.spr
on selection bar 7 of setparamet do hmcfei.spr
on selection bar 8 of setparamet do hmtab.spr

define popup _mfile margin relative shadow color scheme 4
define bar _mfi_setup of _mfile prompt "Pr\<inter Setup..."
define bar _mfi_print of _mfile prompt "\<Print..."
define bar _mfi_sp300 of _mfile prompt "\-"
define bar 4 of _mfile prompt "\<Quit"
on selection bar 4 of _mfile do _quit in hminit

define popup _medit margin relative shadow color scheme 4
define bar _med_undo of _medit prompt "\<Undo" ;
    key ctrl+u, "^U"
define bar _med_redo of _medit prompt "\<Redo" ;
    key ctrl+r, "^R"
define bar _med_sp100 of _medit prompt "\-"
define bar _med_cut of _medit prompt "Cu\<t" ;
    key ctrl+x, "^X"
define bar _med_copy of _medit prompt "\<Copy" ;
    key ctrl+c, "^C"
define bar _med_paste of _medit prompt "\<Paste" ;
```

```
132         key ctrl+v, "^V"
133     define bar _med_clear of _medit prompt "Clear"
134     define bar _med_sp200 of _medit prompt "\-"
135     define bar _med_slcta of _medit prompt "Select \<All" ;
136         key ctrl+a, "^A"
137     define bar _med_sp300 of _medit prompt "\-"
138     define bar _med_goto of _medit prompt "Goto \<Line..."
139     define bar _med_find of _medit prompt "\<Find..." ;
140         key ctrl+f, "^F"
141     define bar _med_finda of _medit prompt "Find A\<gain" ;
142         key ctrl+g, "^G"
143     define bar _med_repl of _medit prompt "R\<eplace And Find Again" ;
144         key ctrl+e, "^E"
145     define bar _med_repla of _medit prompt "Replace All"
146     define bar _med_sp400 of _medit prompt "\-"
147     define bar _med_pref of _medit prompt "Prefere\<nces..."
148
149     define popup _mdata margin relative shadow color scheme 4
150     define bar _mda_brow of _mdata prompt "\<Browse"
151     define bar _mda_sp100 of _mdata prompt "\-"
152     define bar _mda_sort of _mdata prompt "\<Sort..."
153     define bar _mda_total of _mdata prompt "\<Total..."
154     define bar _mda_sp200 of _mdata prompt "\-"
155     define bar _mda_avg of _mdata prompt "A\<verage..."
156     define bar _mda_count of _mdata prompt "C\<ount..."
157     define bar _mda_sum of _mdata prompt "Su\<m..."
158     define bar _mda_calc of _mdata prompt "Calculat\<e..."
159     define bar _mda_reprt of _mdata prompt "\<Report..."
160
161     define popup _mrecord margin relative shadow color scheme 4
162     define bar _mrc_goto of _mrecord prompt "\<Goto..."
163     define bar _mrc_locat of _mrecord prompt "\<Locate..."
164     define bar _mrc_cont of _mrecord prompt "\<Continue" ;
165         key ctrl+k, "^K"
166     define bar _mrc_seek of _mrecord prompt "\<Seek..."
167     define bar _mrc_sp200 of _mrecord prompt "\-"
168     define bar _mrc_repl of _mrecord prompt "Re\<place..."
169     define bar _mrc_delet of _mrecord prompt "\<Delete..."
170     define bar _mrc_recal of _mrecord prompt "\<Recall..."
171
172     define popup _mprog margin relative shadow color scheme 4
173     define bar _mpr_cancl of _mprog prompt "\<Cancel"
174     define bar _mpr_resum of _mprog prompt "\<Resume" ;
175         key ctrl+m, "^M"
176
177     define popup _mwindow margin relative shadow color scheme 4
178     define bar _mwi_hide of _mwindow prompt "\<Hide"
179     define bar _mwi_hidea of _mwindow prompt "\<Hide All"
180     define bar _mwi_showa of _mwindow prompt "Sh\<ow All"
181     define bar _mwi_clear of _mwindow prompt "Clea\<r"
182     define bar _mwi_sp100 of _mwindow prompt "\-"
183     define bar _mwi_move of _mwindow prompt "\<Move" ;
184         key ctrl+f7, "^F7"
185     define bar _mwi_size of _mwindow prompt "\<Size" ;
186         key ctrl+f8, "^F8"
187     define bar _mwi_zoom of _mwindow prompt "\<Zoom ↑" ;
188         key ctrl+f10, "^F10"
189     define bar _mwi_min of _mwindow prompt "Z\<oom ↓" ;
190         key ctrl+f9, "^F9"
191     define bar _mwi_rotat of _mwindow prompt "\<Cycle" ;
192         key ctrl+f1, "^F1"
193     define bar _mwi_color of _mwindow prompt "Co\<lor..."
194     define bar _mwi_sp200 of _mwindow prompt "\-"
195     define bar _mwi_cmd of _mwindow prompt "Command" ;
196         key ctrl+f2, "^F2"
197     define bar _mwi_debug of _mwindow prompt "\<Debug"
198     define bar _mwi_trace of _mwindow prompt "\<Trace"
199     define bar _mwi_view of _mwindow prompt "\<View"
200
201
202     *
203     *                  Cleanup Code & Procedures
204     *
205     *
206     *
207     *
208
209     dropdead = .f.
210
211
212     *
213     *   _QLE0VCQDC  ON SELECTION BAR 1 OF POPUP systemmain
214     *
215     *   Procedure Origin:
216     *
217     *   From Menu:  HMENU.MPR,              Record:   18
218     *   Called By:  ON SELECTION BAR 1 OF POPUP systemmain
219     *   Prompt:     Back-Up (floppy)
220     *   Snippet.    1
221     *
222     *
223     *
224     *
225     procedure _qle0vcqdc
226     * Do back-up on floppy
227     =backup()
228     return
229     * EOF Back-Up
230
231
232
233     *
234     *   _QLE0VCQGQ  ON SELECTION BAR 2 OF POPUP systemmain
235     *
236     *   Procedure Origin:
237     *
238     *   From Menu:  HMENU.MPR,              Record:   19
239     *   Called By:  ON SELECTION BAR 2 OF POPUP systemmain
240     *   Prompt:     Up-Load Data
241     *   Snippet:    2
242     *
243     *
244     *
245     *
246     procedure _qle0vcqgq
247     * BACK TO VAX FROM REMOTE SITE
248     =errmsg("This option is not available in this version.")
249     *=errmsg("Uploading from remote site to VAX")
250     *=vax_back()
251     return
252
253     *: EOF: HMENU.AC2
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATE COVERED
	OCT 1993	FINAL OCT 92 - SEPT 93

4. TITLE AND SUBTITLE
HAZARDOUS MATERIAL LIFE-CYCLE COST MODEL
TECHNICAL MANUAL, VERSION 1.2

5. FUNDING NUMBERS
Program Element: REIMB
Work Unit Number:
NAVFAC.WR.1082W

6. AUTHOR(S)
H. LY AND G. PANG

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Health Research Center
P. O. Box 85122
San Diego, CA 92186-5122

8. PERFORMING ORGANIZATION
Technical
Document No. 93-3C

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Medical Research and Development Command
National Naval Medical Center
Building 1, Tower 2
Bethesda, MD 20889-5044

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This technical manual contains the information on the program code, data elements, and file structures needed to maintain the Hazardous Material Life-Cycle Cost Model. This documentation was created using the FoxDoc Version 2.5a program.

14. SUBJECT TERMS
LIFE-CYCLE COST
MODEL
COST-BENEFIT ANALYSIS
TECHNICAL
HAZARDOUS MATERIALS
HAZARDOUS WASTE

15. NUMBER OF PAGES
158

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited