File created:	26-Jan-88 15:45:41	**{ERINYES}<LISPUSERS>LYRIC>DIRGRAPHER.;2**
changes to:	(VARS DIRGRAPHERCOMS)	
	(FNS DG.ADVISE-GENERATE-FILE)	
previous date:	20-Aug-87 19:28:44	{ERINYES}<LISPUSERS>LYRIC>DIRGRAPHER.;1
Read Table:	XCL	
Package:	INTERLISP	
Format:	XCCS	

```
(RPAQQ DIRGRAPHERCOMS
       ((FILES GRAPHER FILEBROWSER)
        (INITVARS

;;; VARIABLES TO BE SET BY THE USER

               (DG.PROMPTWINDOWFONT (FONTCREATE '(GACHA 8)))
               (DG.GRAPH-LABEL-FONT (FONTCREATE '(GACHA 8)))
               (DG.VERTICAL-HORIZONTAL-OPTION 'VERTICAL)
               (DG.COPY-OVER NIL)
               (DG.BOX-ALL T)
               (DG.DEFAULT-DIR '{DSK}<LISPFILES>)
               (DG.CONCURRENT NIL)
               (DG.CONCURRENT-ALL T)
               (DG.MAX-WIDTH 900)
               (DG.MAX-HEIGHT 700)
               (DG.MIN-WIDTH 200)
               (DG.MIN-HEIGHT 100)
               (DG.MENU-EDGE 'RIGHT)
               (DG.MENU-FONT LITTLEFONT)
               (DG.FILE-INFO-ATTRIBUTES '(SIZE CREATIONDATE WRITEDATE AUTHOR))
               (DG.DEFAULT-BACKUP-DIRECTORY '{FLOPPY})
               (DG.STAND-ALONE-HOSTS '(DSK FLOPPY CORE))
               (DG.BACKGROUND-DIRECTORIES NIL))
        (VARS DG.CANCEL-BUTTON-BM DG.OK-BUTTON-BM DG.RESET-BUTTON-BM)

;;; PROGRAM VARIABLES

               (DG.ICONFONT (FONTCREATE '(GACHA 8)))
               (DG.DIRECTORY-WAS-SELECTED-EVENT (CREATE.EVENT))
               (DG.MONITOR-LOCK (CREATE.MONITORLOCK "DG.MONITOR"))
               (DG.LAST-DIRECTORY-SELECTED NIL)
               (DG.WINDOW-OF-LAST-DIRECTORY NIL))
        (FNS DG.ADD-ITEM-TO-BACKGROUND-MENU DG.ADVISE-GENERATE-FILE DG.APPLY-DIRGRAPHER-COMMAND
             DG.APPLY-FB-ON-DIR-AND-SUBDIRS-COMMAND DG.APPLY-FILEBROWSER-COMMAND DG.ASK-FOR-BACKUP-TYPE
             DG.ATTACH-DIRECTORY-FILES-MENU DG.BACKUP-COMMAND DG.FILE-EXISTS DG.FILE-INFO-COMMAND
             DG.FILE-INFO-SELECTION-FN DG.FILES-HARDCOPY-COMMAND DG.LOAD-FILES-COMMAND DG.NEWER-FILE
             DG.BACKUP-ON-DEFAULT-COMMAND DG.BACKUP-ON-SELECTED-COMMAND DG.CONNECT-DIR-COMMAND
             DG.COPY-DIRECTORY-COMMAND DG.COPY-FILE-COMMAND DG.COPYBUTTONEVENTFN DG.CREATE-BACKUP-FREE-MENU
             DG.CREATE-BACKUP-NAME DG.CREATE-DIRECTORY-CHAIN DG.CREATE-DIRECTORY-SPEC DG.CREATE-DIRECTORY-TREE
             DG.CREATE-GRAPH-FROM-TREE DG.CREATE-ICON-TITLE DG.CREATE-MENU DG.CREATE-PATH-FROM-FILE-NAME
             DG.CREATE-PREFIX-FROM-PATH DG.CREATE-QUIT-MENU DG.CREATE-SUBDIR-COMMAND DG.CREATE-TREE-FROM-LISTS
             DG.CREATE-TREE-NODES DG.CREATEICONW DG.CURRENT-CONNECTED-DIRECTORY DG.DELETE-DIRECTORY-COMMAND
             DG.DELETE-FROM-TREE DG.DELETE-OLD-VERSIONS-COMMAND DG.DELETE-SELECTED-FILES-COMMAND
             DG.DIRECTORY-MINUS-PREFIX DG.DIRECTORY-SELECTION-FUNCTION DG.DISPLAY-DIRECTORY-TREE DG.DOCOMMAND
             DG.DOCOMMAND-WITH-MONITOR DG.DOCOMMAND-WITHOUT-MONITOR DG.EXCLUSIVE-DIRECTORY
             DG.EXCLUSIVE-NEW-VERSIONS DG.EXCLUSIVE-OLD-VERSIONS DG.FBICONFN DG.FILE-SELECTION-FN DG.FILECOPY
             DG.FIND-DIRECTORY-SUBTREE DG.FIND-MENU-ITEM DG.FIND-PARENT-DIERCTORY DG.FLASHALLWINDOWS
             DG.GET-FB-PATTERN DG.GET-FILE-LIST DG.GET-INITIAL-REGION DG.KILL-PROCESS DG.MOVE-DIR-COMMAND
             DG.MOVE-FILE-COMMAND DG.MOVECOPY-COMMAND DG.MOVECOPY-DIR-COMMAND DG.PACK-NAME-VER-EXT
             DG.PROMPTWINDOW DG.REDISPLAYFN DG.RESIDUAL-PATH DG.RESTORE-COMMAND DG.RESTORE-FROM-DEFAULT-COMMAND
             DG.RESTORE-FROM-SELECTED-COMMAND DG.SETIFY DG.SHADE-CURRENT-DIRECTORY DG.TRIM DG.UNADVISE
             DG.UNADVISE-GENERATE-FILE DG.UNPACK-DIRECTORY-NAME DG.UPDATE-COMMAND DG.UPDATE-DIRECTORY-TREE
             DG.UPDATE-DIRECTROY-TREE DG.WAIT-FOR-DIR-SELECTION DIRGRAPHER)
        (BITMAPS DG.ICON DG.MASK DG.FILL-WINDOW-TEXTURE DG.WHITEBM)
        (P (DG.ADVISE-GENERATE-FILE)
           (SETQ DG.BACKGROUND-DIRECTORIES (UNION DG.BACKGROUND-DIRECTORIES
                                                  (UNION DIRECTORIES (CONS '{FLOPPY}
                                                                           (FOR D IN (VOLUMES)
                                                                              WHEN
                                                                              (LISPDIRECTORYP D)
                                                                              COLLECT
                                                                              (PACK* '{DSK}< D '>))))))
           (DG.ADD-ITEM-TO-BACKGROUND-MENU
            '|DirGrapher|
            '(DIRGRAPHER)
            "Will initiate dirgrapher process on the current directory"
            (CONS 'SUBITEMS (CONS '("DG Windows to top" (|for| W |in| (OPENWINDOWS)
                                                           |when|
                                                           (OR (WINDOWPROP W 'DG.PATH-TO-ROOT)
                                                               (AND (WINDOWPROP W 'ICONFOR)
                                                                    (WINDOWPROP (WINDOWPROP W 'ICONFOR)
                                                                           'DG.PATH-TO-ROOT)))
```

```
                                                    │do│
                                                    (TOTOPW W)))
                        (FOR D IN DG.BACKGROUND-DIRECTORIES COLLECT (LIST D (LIST 'DIRGRAPHER
                                                                              (KWOTE D))
                                                              (CONCAT "WILL APPLY
                                                                      DIRGRAPHER ON " D))))
            )))
```

;;; (declare\: dontcopy (prop makefile-environment dirgrapher))
))

(FILESLOAD GRAPHER FILEBROWSER)

(RPAQ? **DG.PROMPTWINDOWFONT** (FONTCREATE '(GACHA 8)))

(RPAQ? **DG.GRAPH-LABEL-FONT** (FONTCREATE '(GACHA 8)))

(RPAQ? **DG.VERTICAL-HORIZONTAL-OPTION** 'VERTICAL)

(RPAQ? **DG.COPY-OVER** NIL)

(RPAQ? **DG.BOX-ALL** T)

(RPAQ? **DG.DEFAULT-DIR** '{DSK}<LISPFILES>)

(RPAQ? **DG.CONCURRENT** NIL)

(RPAQ? **DG.CONCURRENT-ALL** T)

(RPAQ? **DG.MAX-WIDTH** 900)

(RPAQ? **DG.MAX-HEIGHT** 700)

(RPAQ? **DG.MIN-WIDTH** 200)

(RPAQ? **DG.MIN-HEIGHT** 100)

(RPAQ? **DG.MENU-EDGE** 'RIGHT)

(RPAQ? **DG.MENU-FONT** LITTLEFONT)

(RPAQ? **DG.FILE-INFO-ATTRIBUTES** '(SIZE CREATIONDATE WRITEDATE AUTHOR))

(RPAQ? **DG.DEFAULT-BACKUP-DIRECTORY** '{FLOPPY})

(RPAQ? **DG.STAND-ALONE-HOSTS** '(DSK FLOPPY CORE))

(RPAQ? **DG.BACKGROUND-DIRECTORIES** NIL)

(RPAQQ **DG.CANCEL-BUTTON-BM**

)

(RPAQQ **DG.OK-BUTTON-BM**

)

(RPAQQ **DG.RESET-BUTTON-BM**

)

(RPAQ **DG.ICONFONT** (FONTCREATE '(GACHA 8)))

(RPAQ **DG.DIRECTORY-WAS-SELECTED-EVENT** (CREATE.EVENT))

(RPAQ **DG.MONITOR-LOCK** (CREATE.MONITORLOCK "DG.MONITOR"))

(RPAQQ **DG.LAST-DIRECTORY-SELECTED** NIL)

(RPAQQ **DG.WINDOW-OF-LAST-DIRECTORY** NIL)

(DEFINEQ

(**DG.ADD-ITEM-TO-BACKGROUND-MENU**
  (LAMBDA (LABEL COMMAND MESSAGE SUBITEMLIST)   (* |edited:| "13-May-86 14:03")
    (DECLARE (GLOBALVARS │BackgroundMenuCommands│ │BackgroundMenu│))
    (SETQ │BackgroundMenuCommands│ (REMOVE (FASSOC LABEL │BackgroundMenuCommands│)
                                          │BackgroundMenuCommands│))
    (NCONC1 │BackgroundMenuCommands│ (LIST LABEL COMMAND MESSAGE SUBITEMLIST))
    (SETQ │BackgroundMenu│ NIL)))

(**DG.ADVISE-GENERATE-FILE**
  (LAMBDA NIL   ; Edited 26-Jan-88 15:42 by smL
    (MOVD '\\GENERATENEXTFILE 'DG.GENERATE-NEXT-FILE T)
    (UNADVISE (\\GENERATENEXTFILE IN FB.UPDATEBROWSERITEMS))
    (ADVISE '(\\GENERATENEXTFILE IN FB.UPDATEBROWSERITEMS)
```

```
                    'AROUND
                    '(PROG* ((WINDOW (TB.WINDOW (FB.TABLEBROWSER BROWSER)))
                             (PATTERN (DG.GET-FB-PATTERN WINDOW))
                             (FBDIR (U-CASE (FILENAMEFIELD PATTERN 'DIRECTORY)))
                             (FBHOST (U-CASE (FILENAMEFIELD PATTERN 'HOST)))
                             NEXT-FILE TEMP-HOST)
                            (DECLARE (GLOBAL DG.STAND-ALONE-HOSTS))
                            (SETQ FBHOST (|if| (FMEMB FBHOST DG.STAND-ALONE-HOSTS)
                                               |then| FBHOST
                                               |else| (OR (CANONICAL.HOSTNAME FBHOST)
                                                          FBHOST)))
                       NEXT
                            (SETQ NEXT-FILE (DG.GENERATE-NEXT-FILE GENOBJ NAMEONLY))
                            (|if| (NULL NEXT-FILE)
                                  |then| (RETURN NIL))
                            (|if| (LISTP NEXT-FILE)
                                  |then| (SETQ NEXT-FILE (PACKC NEXT-FILE)))
                            (|if| (OR (NOT (WINDOWPROP (MAINWINDOW WINDOW)
                                                       'DG.DIRECTORY-ONLY))
                                      (AND (EQ (U-CASE (FILENAMEFIELD NEXT-FILE 'DIRECTORY))
                                               FBDIR)
                                           (EQ (FILENAMEFIELD NEXT-FILE 'HOST)
                                               FBHOST)))
                                  |then| (RETURN NEXT-FILE)
                                  |else| (GO NEXT))))))))
```

(**DG.APPLY-DIRGRAPHER-COMMAND**
```
  (LAMBDA (WINDOW)                                        (* |edited:| " 3-Apr-85 11:27")
    (PROG (CURRENT-PATH DIR-FILE-LIST FB-WINDOW TEMP-STRING)
          (SETQ CURRENT-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select directory on which to apply
                                       DirGrapher.")))
          (COND
              (CURRENT-PATH (INVERTW WINDOW)
                            (DIRGRAPHER (DG.CREATE-PREFIX-FROM-PATH CURRENT-PATH))
                            (INVERTW WINDOW))))))
```

(**DG.APPLY-FB-ON-DIR-AND-SUBDIRS-COMMAND**
```
  (LAMBDA (WINDOW)                                        (* |sm| "10-Jun-85 14:42")
    (DG.APPLY-FILEBROWSER-COMMAND WINDOW T)))
```

(**DG.APPLY-FILEBROWSER-COMMAND**
```
  (LAMBDA (WINDOW SUBDIRS)                                (* |edited:| " 7-May-86 13:09")
    (PROG (CURRENT-PATH DIR-FILE-LIST FB-WINDOW TEMP-STRING)
          (SETQ CURRENT-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select directory on which to apply
                                       FileBrowser.")))
          (COND
              (CURRENT-PATH (INVERTW WINDOW)
                            (SETQ FB-WINDOW (FILEBROWSER (DG.CREATE-PREFIX-FROM-PATH CURRENT-PATH)))
                            (|if| (NOT SUBDIRS)
                                  |then| (WINDOWPROP FB-WINDOW 'DG.DIRECTORY-ONLY T))
                            (WINDOWPROP FB-WINDOW 'ICONFN 'DG.FBICONFN)
                            (INVERTW WINDOW))))))
```

(**DG.ASK-FOR-BACKUP-TYPE**
```
  (LAMBDA (CURRENT-DIRECTORY)                             (* |sm| " 9-Jun-85 11:53")
    (PROG (BACKUP-TYPE)
          (COND
              ((NULL (CDR CURRENT-DIRECTORY))
               (SETQ BACKUP-TYPE 'FILES-ONLY))
              (T (SETQ BACKUP-TYPE (MENU (|create| MENU
                                                ITEMS _ (LIST (LIST "Back up files in this directory (but not in
                                                                 subdirectories)" ''FILES-ONLY)
                                                              (LIST "Backup files of this directory and
                                                                 subdirectories" ''FILES-AND-SUBS))
                                                MENUFONT _ BIGFONT)
                                           NIL T))))
          (RETURN BACKUP-TYPE))))
```

(**DG.ATTACH-DIRECTORY-FILES-MENU**
```
  (LAMBDA (PATH WINDOW MAIN-WINDOW FILE-SELECTION-FN INCLUDEFULLNAMES)
                                                          (* |edited:| "16-May-86 15:56")
    (DECLARE (GLOBALVARS LITTLEFONT SCREENHEIGHT))
    (PROG (MENU FILE-LIST NUMBER-OF-COLUMNS NUMBER-OF-ROWS LENGTH-FILE-LIST MAX-WIDTH MENU-WINDOW)
          (SETQ FILE-SELECTION-FN (OR FILE-SELECTION-FN 'DG.FILE-SELECTION-FN))
          (SETQ FILE-LIST (CONS '\ STOP (|for| F |in| (DG.EXCLUSIVE-DIRECTORY (DG.CREATE-PREFIX-FROM-PATH PATH))
                                               |collect| (|if| INCLUDEFULLNAMES
                                                               |then| (LIST (DG.PACK-NAME-VER-EXT F)
                                                                            F)
                                                               |else| (DG.PACK-NAME-VER-EXT F)))))
          (|if| (CDR FILE-LIST)
                |then| (SETQ LENGTH-FILE-LIST (LENGTH FILE-LIST))
                       (SETQ MAX-WIDTH 6)
```

```
                            (|for| F |in| FILE-LIST |bind| WIDTH |when| (GREATERP (SETQ WIDTH (STRINGWIDTH
                                                                                       (|if| INCLUDEFULLNAMES
                                                                                          |then| (CAR F)
                                                                                        |else| F)
                                                                                     LITTLEFONT))
                                                                              MAX-WIDTH)
                               |do| (SETQ MAX-WIDTH WIDTH))
                            (SETQ NUMBER-OF-COLUMNS (MAX 1 (IQUOTIENT (CAR (WINDOWSIZE MAIN-WINDOW))
                                                                      MAX-WIDTH)))
                            (SETQ NUMBER-OF-ROWS (IPLUS (IQUOTIENT LENGTH-FILE-LIST NUMBER-OF-COLUMNS)
                                                        (|if| (EQP (IREMAINDER LENGTH-FILE-LIST NUMBER-OF-COLUMNS)
                                                                   0)
                                                           |then| 0
                                                         |else| 1)))
                            (SETQ FILE-LIST (APPEND FILE-LIST (|for| I |from| (ADD1 LENGTH-FILE-LIST)
                                                                    |to| (ITIMES NUMBER-OF-COLUMNS NUMBER-OF-ROWS)
                                                                    |collect| " ")))
                            (SETQ FILE-LIST (|for| ROW |from| 1 |to| NUMBER-OF-ROWS
                                                 |join| (|for| I |in| (NTH FILE-LIST ROW) |by| (NTH I (ADD1 NUMBER-OF-ROWS))
                                                             |collect| I)))
                            (SETQ MENU-WINDOW
                             (ATTACHMENU (|create| MENU
                                                   ITEMS _ FILE-LIST
                                                   WHENSELECTEDFN _ FILE-SELECTION-FN
                                                   MENUCOLUMNS _ NUMBER-OF-COLUMNS
                                                   MENUFONT _ LITTLEFONT)
                                         MAIN-WINDOW
                                         (|if| (GREATERP (IPLUS (|fetch| BOTTOM |of| (WINDOWREGION MAIN-WINDOW))
                                                                (IQUOTIENT (CDR (WINDOWSIZE MAIN-WINDOW))
                                                                           2))
                                                         (IQUOTIENT SCREENHEIGHT 2))
                                            |then| 'BOTTOM
                                          |else| 'TOP)
                                         'JUSTIFY))
                            (RETURN MENU-WINDOW)
                      |else| (RETURN NIL)))))
```

(**DG.BACKUP-COMMAND**
```
  (LAMBDA (WINDOW BACKUP-DIR)                                        (* |edited:| "13-May-86 14:02")
    (DECLARE (GLOBALVARS DG.DIRECTORY-SUBTREE DG.FILES-BACKUP-NAMES COPYRIGHTFLG DG.COPY-OVER FILELST))
    (PROG (BACKUP-TYPE SOURCE-PATH SOURCE-DIRECTORY FILES-TO-BACKUP ROUND-OF-FILES FILES-BACKUP-NAMES
                 BACKUP-HOST OLDCOPYRIGHTFLG SOURCE-PATH-AND-WINDOW SOURCE-WINDOW)
          (SETQ SOURCE-PATH-AND-WINDOW (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select the directory that you want to
                                    back up." T))
          (SETQ SOURCE-PATH (CAR SOURCE-PATH-AND-WINDOW))
          (SETQ SOURCE-WINDOW (CADR SOURCE-PATH-AND-WINDOW))
          (SETQ SOURCE-DIRECTORY (DG.FIND-DIRECTORY-SUBTREE SOURCE-PATH SOURCE-WINDOW))
          (SETQ BACKUP-TYPE (DG.ASK-FOR-BACKUP-TYPE SOURCE-DIRECTORY))
          (|if| (NULL (FILENAMEFIELD BACKUP-DIR 'DIRECTORY))
             |then| (SETQ BACKUP-DIR (PACK* BACKUP-DIR '<)))
          (SETQ BACKUP-HOST (FILENAMEFIELD BACKUP-DIR 'HOST))
          (INVERTW WINDOW)
          (SETQ FILES-TO-BACKUP (|if| (EQ BACKUP-TYPE 'FILES-AND-SUBS)
                                    |then| (DIRECTORY (DG.CREATE-PREFIX-FROM-PATH SOURCE-PATH))
                                  |else| (DG.EXCLUSIVE-DIRECTORY (DG.CREATE-PREFIX-FROM-PATH SOURCE-PATH))))
          (SETQ FILES-BACKUP-NAMES (|for| F |in| FILES-TO-BACKUP |collect| (DG.CREATE-BACKUP-NAME F SOURCE-PATH
                                                                                SOURCE-DIRECTORY)))
          (SETQ DG.FILES-BACKUP-NAMES FILES-BACKUP-NAMES)
          (SETQ DG.DIRECTORY-SUBTREE SOURCE-DIRECTORY)
          (SET (FILECOMS 'BACKUPINFO)
               '((VARS DG.FILES-BACKUP-NAMES DG.DIRECTORY-SUBTREE)))
          (PUTPROP 'BACKUPINFO 'FILETYPE 'DON\'TLIST)
          (COND
            ((EQ BACKUP-TYPE 'FILES-AND-SUBS)
             (SETQ DG.DIRECTORY-SUBTREE SOURCE-DIRECTORY))
            (T (SETQ DG.DIRECTORY-SUBTREE NIL)))
          (|if| (EQ BACKUP-HOST 'FLOPPY)
             |then| (INVERTW WINDOW)
                    (|printout| (DG.PROMPTWINDOW WINDOW)
                            T "Insert floppy .")
                    (FLOPPY.WAIT.FOR.FLOPPY)
                    (INVERTW WINDOW))
          (SETQ OLDCOPYRIGHTFLG COPYRIGHTFLG)
          (SETQ COPYRIGHTFLG NIL)
          (MAKEFILE (PACK* BACKUP-DIR (CAR SOURCE-DIRECTORY)
                           '>
                           'BACKUPINFO)
                    'NEW)
          (SETQ COPYRIGHTFLG OLDCOPYRIGHTFLG)
          (|while| FILES-TO-BACKUP
             |do| (|for| F1 |in| FILES-TO-BACKUP |as| F2 |in| FILES-BACKUP-NAMES
                      |do| (|if| (AND (NOT DG.COPY-OVER)
                                      (DG.FILE-EXISTS (PACK* BACKUP-DIR F2))
                                      (NOT (DG.NEWER-FILE F1 (PACK* BACKUP-DIR F2))))
                               |then| (|printout| (DG.PROMPTWINDOW WINDOW)
                                              T "File " F1 " exists on " BACKUP-DIR " and was not copied.")
```

```
                                              (SETQ FILES-TO-BACKUP (REMOVE F1 FILES-TO-BACKUP))
                                              (SETQ FILES-BACKUP-NAMES (REMOVE F2 FILES-BACKUP-NAMES)))
                              |else| (|if| (AND DG.COPY-OVER (DG.FILE-EXISTS (PACK* BACKUP-DIR F2)))
                                       |then| (DELFILE (PACK* BACKUP-DIR F2)))
                                   (|if| (OR (NEQ BACKUP-HOST 'FLOPPY)
                                             (GREATERP (FLOPPY.FREE.PAGES)
                                                  (IPLUS (GETFILEINFO F1 'SIZE)
                                                         400)))
                                       |then| (|if| (DG.FILECOPY F1 (PACK* BACKUP-DIR F2)
                                                            DG.COPY-OVER)
                                                  |then| (|printout| (DG.PROMPTWINDOW WINDOW)
                                                            T "File " F1 " is backed up.")
                                                  |else| (|printout| (DG.PROMPTWINDOW WINDOW)
                                                            T "File " F1 " exists on " BACKUP-DIR " and was not
                                                              copied."))
                                             (SETQ FILES-TO-BACKUP (REMOVE F1 FILES-TO-BACKUP))
                                             (SETQ FILES-BACKUP-NAMES (REMOVE F2 FILES-BACKUP-NAMES)))))
                         (|if| FILES-TO-BACKUP
                             |then| (INVERTW WINDOW)
                                    (CLEARW (DG.PROMPTWINDOW WINDOW))
                                    (|printout| (DG.PROMPTWINDOW WINDOW)
                                       T "No more space on this floppy. Insert a new one. ")
                                    (FLASHWINDOW (DG.PROMPTWINDOW WINDOW)
                                       3)
                                    (|for| I |from| 100 |to| 1000 |by| 200 |do| (PLAYTUNE (LIST (CONS I 5000))))
                                    (FLOPPY.WAIT.FOR.FLOPPY T)
                                    (INVERTW WINDOW)))
                  (SETQ FILELST (REMOVE 'BACKUPINFO FILELST))
                  (INVERTW WINDOW))))
```

(**DG.FILE-EXISTS**
```
  (LAMBDA (F)                                                    (* |edited:| " 6-Mar-86 14:43")
    (INFILEP F)))
```

(**DG.FILE-INFO-COMMAND**
```
  (LAMBDA (WINDOW)                                               (* |edited:| "16-May-86 15:59")
    (DECLARE (GLOBALVARS DG.PROMPTWINDOWFONT LITTLEFONT DG.FILE-INFO-ATTRIBUTES))
    (PROG (SOURCE-PATH SELECTED-FILES SOURCE-PREFIX MENU-WINDOW SOURCE-PATH-WINDOW WREGION TITLE-WIDTH TITLE
                  COLUMNS DATEWIDTH PWINDOW)
          (SETQ PWINDOW (DG.PROMPTWINDOW WINDOW))
          (SETQ SOURCE-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select directory from which you want to
                                    files info.")))
          (COND
             (SOURCE-PATH (INVERTW WINDOW)
                  (SETQ SOURCE-PREFIX (DG.CREATE-PREFIX-FROM-PATH SOURCE-PATH))
                  (SETQ TITLE " ")
                  (|while| (GREATERP (STRINGWIDTH "MMMMMMMMMMMMMMMMMMMMM : " DG.PROMPTWINDOWFONT)
                                 (STRINGWIDTH TITLE LITTLEFONT))
                      |do| (SETQ TITLE (CONCAT TITLE " ")))
                  (SETQ DATEWIDTH (STRINGWIDTH (DATE)
                                        DG.PROMPTWINDOWFONT))
                  (|for| I |in| DG.FILE-INFO-ATTRIBUTES |bind| I-WITH-BLANKS
                     |do| (SETQ COLUMNS (NCONC1 COLUMNS (STRINGWIDTH TITLE LITTLEFONT)))
                          (SETQ I-WITH-BLANKS I)
                          (|if| (MEMBER I '(WRITEDATE READDATE CREATIONDATE))
                              |then| (|while| (GREATERP DATEWIDTH (STRINGWIDTH I-WITH-BLANKS LITTLEFONT))
                                         |do| (SETQ I-WITH-BLANKS (CONCAT I-WITH-BLANKS " "))))
                          (SETQ TITLE (CONCAT TITLE I-WITH-BLANKS "   ")))
                  (WINDOWPROP WINDOW 'DG.INFO-COLUMNS COLUMNS)
                  (SETQ TITLE-WIDTH (STRINGWIDTH TITLE LITTLEFONT))
                  (|if| (GREATERP TITLE-WIDTH (|fetch| WIDTH |of| (WINDOWPROP PWINDOW 'REGION)))
                      |then| (SETQ WREGION (WINDOWREGION WINDOW))
                             (|replace| WIDTH |of| WREGION |with| (IPLUS (|fetch| WIDTH |of| WREGION)
                                                                    (IDIFFERENCE TITLE-WIDTH
                                                                           (|fetch| WIDTH
                                                                               |of| (WINDOWPROP PWINDOW
                                                                                        'REGION)))))
                             (INVERTW WINDOW)
                             (RESHAPEALLWINDOWS WINDOW WREGION)
                             (INVERTW WINDOW))
                  (WINDOWPROP PWINDOW 'TITLE TITLE)
                  (SETQ MENU-WINDOW (DG.ATTACH-DIRECTORY-FILES-MENU SOURCE-PATH WINDOW WINDOW
                                        'DG.FILE-INFO-SELECTION-FN T))
                  (INVERTW WINDOW)
                  (|if| MENU-WINDOW
                      |then| (WINDOWPROP MENU-WINDOW 'DG.DETACH T)
                             (CLEARW (DG.PROMPTWINDOW WINDOW))
                             (AWAIT.EVENT (WINDOWPROP WINDOW 'DG.FILE-SELECTION-ENDED-EVENT)
                                    1000000)
                             (DETACHWINDOW MENU-WINDOW)
                             (CLOSEW MENU-WINDOW))
                  (WINDOWPROP PWINDOW 'TITLE NIL))))))
```

(**DG.FILE-INFO-SELECTION-FN**

```
(LAMBDA (ITEM MENU KEY)                                          (* |edited:| "13-May-86 14:17")
  (DECLARE (GLOBALVARS DG.FILE-INFO-ATTRIBUTES))
  (PROG (WINDOW)
        (SETQ WINDOW (MAINWINDOW (WFROMMENU MENU)))
        (COND
           ((EQ ITEM '\ STOP)
            (NOTIFY.EVENT (WINDOWPROP WINDOW 'DG.FILE-SELECTION-ENDED-EVENT)))
           ((EQUAL ITEM " "))
           (ITEM (PRINTOUT (DG.PROMPTWINDOW WINDOW)
                           T
                           (SUBSTRING (CAR ITEM)
                                      1
                                      (MIN (NCHARS (CAR ITEM)
                                                   20)))
                           " : ")
                 (|for| ATT |in| DG.FILE-INFO-ATTRIBUTES |as| COL |in| (WINDOWPROP WINDOW 'DG.INFO-COLUMNS)
                    |do| (DSPXPOSITION COL (DG.PROMPTWINDOW WINDOW))
                         (PRINTOUT (DG.PROMPTWINDOW WINDOW)
                                   (GETFILEINFO (CADR ITEM)
                                                ATT)))))))))
```

(**DG.FILES-HARDCOPY-COMMAND**
```
  (LAMBDA (WINDOW)                                               (* |edited:| "13-May-86 14:06")
    (PROG (SOURCE-PATH SELECTED-FILES SOURCE-PREFIX MENU-WINDOW SOURCE-PATH-WINDOW)
          (SETQ SOURCE-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select directory from which you want to
                                           hardcopy selected files.")))
          (COND
             (SOURCE-PATH (SETQ SOURCE-PREFIX (DG.CREATE-PREFIX-FROM-PATH SOURCE-PATH))
                          (SETQ MENU-WINDOW (DG.GET-FILE-LIST " The selected files will be hardcopied. " WINDOW
                                                    SOURCE-PATH WINDOW))
                          (SETQ SELECTED-FILES (WINDOWPROP MENU-WINDOW 'DG.SELECTED-FILES))
                          (|if| SELECTED-FILES
                              |then| (INVERTW WINDOW)
                                     (|for| FILE |in| SELECTED-FILES |do| (LISTFILES1 (PACK* SOURCE-PREFIX FILE))
                                                (SHADEITEM FILE (CAR (WINDOWPROP MENU-WINDOW
                                                                           'MENU))
                                                           HIGHLIGHTSHADE MENU-WINDOW)
                                                (BLOCK))
                                     (INVERTW WINDOW))
                          (DETACHWINDOW MENU-WINDOW)
                          (CLOSEW MENU-WINDOW))))))
```

(**DG.LOAD-FILES-COMMAND**
```
  (LAMBDA (WINDOW LOADTYPE LDFLG)                                (* |edited:| "13-May-86 16:31")
    (DECLARE (GLOBALVARS HIGHLIGHTSHADE))
    (PROG (SOURCE-PATH SELECTED-FILES SOURCE-PREFIX MENU-WINDOW SOURCE-PATH-WINDOW)
          (SETQ LOADTYPE (OR LOADTYPE 'LOAD))
          (SETQ SOURCE-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW (CONCAT "Select directory from which you want
                                           to " LOADTYPE " selected files."))))
          (COND
             (SOURCE-PATH (SETQ SOURCE-PREFIX (DG.CREATE-PREFIX-FROM-PATH SOURCE-PATH))
                          (SETQ MENU-WINDOW (DG.GET-FILE-LIST " The selected files will be LOADED. " WINDOW SOURCE-PATH
                                                    WINDOW))
                          (SETQ SELECTED-FILES (WINDOWPROP MENU-WINDOW 'DG.SELECTED-FILES))
                          (|if| SELECTED-FILES
                              |then| (INVERTW WINDOW)
                                     (|for| FILE |in| SELECTED-FILES |do| (APPLY* LOADTYPE (PACK* SOURCE-PREFIX FILE)
                                                       LDFLG)
                                                (SHADEITEM FILE (CAR (WINDOWPROP MENU-WINDOW
                                                                           'MENU))
                                                           HIGHLIGHTSHADE MENU-WINDOW)
                                                (BLOCK))
                                     (INVERTW WINDOW))
                          (DETACHWINDOW MENU-WINDOW)
                          (CLOSEW MENU-WINDOW))))))
```

(**DG.NEWER-FILE**
```
  (LAMBDA (F1 F2)                                                (* |edited:| " 7-Mar-86 15:31")
    (GREATERP (GETFILEINFO F1 'ICREATIONDATE)
              (GETFILEINFO F2 'ICREATIONDATE))))
```

(**DG.BACKUP-ON-DEFAULT-COMMAND**
```
  (LAMBDA (WINDOW)                                               (* |edited:| "13-May-86 13:56")
    (DECLARE (GLOBALVARS DG.DEFAULT-BACKUP-DIRECTORY))
    (|if| (NULL DG.DEFAULT-BACKUP-DIRECTORY)
        |then| (DG.FLASHALLWINDOWS WINDOW)
               (|printout| (DG.PROMPTWINDOW WINDOW)
                   T "NULL default directory.Backup aborted")
      |else| (DG.BACKUP-COMMAND WINDOW DG.DEFAULT-BACKUP-DIRECTORY)))
```

(**DG.BACKUP-ON-SELECTED-COMMAND**
```
  (LAMBDA (WINDOW)                                               (* |sm| " 8-Jun-85 12:49")
```

```
     (PROG (BACKUP-DIRECTORY)
           (SETQ BACKUP-DIRECTORY (DG.CREATE-PREFIX-FROM-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select the
                                                                                 directory that will contain the
                                                                                 backup." T))))
           (DG.BACKUP-COMMAND WINDOW BACKUP-DIRECTORY)))))
```

(DG.CONNECT-DIR-COMMAND
```
  (LAMBDA (WINDOW)                                              (* |edited:| "13-May-86 13:55")
    (DECLARE (GLOBALVARS DG.BOX-ALL))
    (PROG (CURRENT-PATH CURRENT-PREFIX)
          (SETQ CURRENT-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select the directory to which you want to
                                        be connected")))
          (COND
             (CURRENT-PATH (CNDIR (SETQ CURRENT-PREFIX (DG.CREATE-PREFIX-FROM-PATH CURRENT-PATH)))
                    (|if| DG.BOX-ALL
                        |then| (|for| W |in| (OPENWINDOWS) |when| (WINDOWPROP W 'DG.TREE) |do| (
                                                                      DG.SHADE-CURRENT-DIRECTORY
                                                                            W))
                      |else| (DG.SHADE-CURRENT-DIRECTORY WINDOW)))))))
```

(DG.COPY-DIRECTORY-COMMAND
```
  (LAMBDA (WINDOW)                                              (* |edited:| " 2-Apr-85 18:02")
    (DG.MOVECOPY-DIR-COMMAND WINDOW)))
```

(DG.COPY-FILE-COMMAND
```
  (LAMBDA (WINDOW)                                              (* |edited:| " 2-Apr-85 18:05")
    (DG.MOVECOPY-COMMAND WINDOW)))
```

(DG.COPYBUTTONEVENTFN
```
  (LAMBDA (WINDOW)                                              (* |sm| " 9-Jul-85 12:28")
    (PROG (CURSOR-POS SELECTED-NODE NODE-REGION RELEASED)
          (SETQ CURSOR-POS (CURSORPOSITION NIL WINDOW))
          (SETQ RELEASED (MOUSESTATE (AND (NOT LEFT)
                                          (NOT MIDDLE)
                                          (NOT RIGHT))))
          (|if| (SETQ SELECTED-NODE (|for| NODE |in| (|fetch| GRAPHNODES |of| (WINDOWPROP WINDOW 'GRAPH))
                                          |thereis| (INSIDEP (SETQ NODE-REGION (NODEREGION NODE))
                                                          CURSOR-POS)))
              |then| (FLIPNODE SELECTED-NODE WINDOW)
             |else| (GRAPHERCOPYBUTTONEVENTFN WINDOW))
          (|if| (AND RELEASED SELECTED-NODE)
              |then| (BKSYSBUF (DG.CREATE-PREFIX-FROM-PATH (APPEND (WINDOWPROP WINDOW 'DG.PATH-TO-ROOT)
                                                              (CDR (|fetch| NODEID |of| SELECTED-NODE)))))))))
```

(DG.CREATE-BACKUP-FREE-MENU
```
  (LAMBDA (LEFT BOTTOM)                                         (* |edited:| "16-May-86 17:50")
    (FM.FORMATMENU (LIST '((TYPE TITLE LABEL "Files to Backup  : " FONT (MODERN 12 BOLD))
                           (TYPE TOGGLE LABEL "Include SubDirectories" ID SUBDIRS)
                           (TYPE TOGGLE LABEL "No Old Versions" ID SUBDIRS))
                         (LIST `(TYPE NWAY LABEL ,DG.OK-BUTTON-BM ID BUTTONS)
                               `(TYPE NWAY LABEL ,\,DG.RESET-BUTTON-BM ID BUTTONS)
                               `(TYPE NWAY LABEL ,\,DG.CANCEL-BUTTON-BM ID BUTTONS))))))
```

(DG.CREATE-BACKUP-NAME
```
  (LAMBDA (F CURRENT-PATH CURRENT-DIRECTORY)                    (* |edited:| " 5-Apr-85 10:34")
    (PROG (SECOND-HALF FILE-BACKUP-NAME)
          (SETQ SECOND-HALF (SUBSTRING (FILENAMEFIELD F 'DIRECTORY)
                                   (IPLUS 2 (NCHARS (DG.CREATE-DIRECTORY-CHAIN (CDR CURRENT-PATH))))
                                   (NCHARS (FILENAMEFIELD F 'DIRECTORY))))
          (SETQ FILE-BACKUP-NAME (PACKFILENAME 'DIRECTORY (COND
                                                      (SECOND-HALF (DG.CREATE-DIRECTORY-CHAIN
                                                                      (LIST (CAR CURRENT-DIRECTORY)
                                                                            SECOND-HALF)))
                                                      (T (CAR CURRENT-DIRECTORY)))
                                          'NAME
                                          (FILENAMEFIELD F 'NAME)
                                          'EXTENSION
                                          (FILENAMEFIELD F 'EXTENSION)
                                          'VERSION
                                          (FILENAMEFIELD F 'VERSION)))
          (RETURN (PACK (CDR (UNPACK FILE-BACKUP-NAME)))))))
```

(DG.CREATE-DIRECTORY-CHAIN
```
  (LAMBDA (PATH)                                                (* |sm| " 9-Jun-85 11:49")
    (COND
       ((NULL PATH)
        '"")
       ((NULL (CDR PATH))
        (CAR PATH))
       (T (PACK* (CAR PATH)
```

```
                '>
                  (DG.CREATE-DIRECTORY-CHAIN (CDR PATH)))))))
```

(**DG.CREATE-DIRECTORY-SPEC**
```
  (LAMBDA (PATH WINDOW)                                        (* |edited:| "31-Dec-00 22:14")
    (PROG (CURRENT-DIRECTORY SPEC)
          (SETQ CURRENT-DIRECTORY (DG.FIND-DIRECTORY-SUBTREE PATH WINDOW))
          (SETQ SPEC (DG.CREATE-PREFIX-FROM-PATH PATH))
          (|for| SUB |in| (CDR CURRENT-DIRECTORY) |do| (SETQ SPEC (LIST SPEC '- (DG.CREATE-PREFIX-FROM-PATH
                                                                     (APPEND PATH (LIST (CAR SUB)))))))

          (RETURN SPEC))))
```

(**DG.CREATE-DIRECTORY-TREE**
```
  (LAMBDA (PATH)                                               (* |edited:| "13-May-86 12:34")
    (PROG (TREE-PATHS ROOT TREE)
          (SETQ TREE-PATHS (|for| F |in| (DIRECTORY (DG.CREATE-PREFIX-FROM-PATH PATH))
                                 |collect| (NTH (DG.CREATE-PATH-FROM-FILE-NAME F)
                                                (LENGTH PATH))))
          (SETQ TREE-PATHS (DG.SETIFY TREE-PATHS))
          (SETQ TREE (CAR (DG.CREATE-TREE-FROM-LISTS TREE-PATHS)))
          (|if| TREE
              |then| (RETURN TREE)
           |else| (RETURN (LAST PATH))))))
```

(**DG.CREATE-GRAPH-FROM-TREE**
```
  (LAMBDA (TREE)                                               (* |edited:| "13-May-86 14:09")
    (DECLARE (GLOBALVARS DG.VERTICAL-HORIZONTAL-OPTION DG.GRAPH-LABEL-FONT))
    (PROG (NODE-LIST)
          (SETQ NODE-LIST (DG.CREATE-TREE-NODES TREE NIL))
          (RETURN (LAYOUTGRAPH (CDR NODE-LIST)
                      (LIST (CAR NODE-LIST))
                      (LIST DG.VERTICAL-HORIZONTAL-OPTION)
                      DG.GRAPH-LABEL-FONT)))))
```

(**DG.CREATE-ICON-TITLE**
```
  (LAMBDA (L)                                                  (* |edited:| " 5-Apr-85 11:16")
    (COND
       (L (COND
            ((GREATERP (LENGTH L)
                    3)
             (SETQ L (CDR (LASTN L 3)))))
          (APPLY 'CONCAT (CONS (DG.TRIM (CAR L)
                                     11)
                             (|for| W |in| (CDR L) |collect| (CONCAT (CHARACTER 13)
                                                             (DG.TRIM W 11))))))
       (T " "))))
```

(**DG.CREATE-MENU**
```
  (LAMBDA (WINDOW)                                             (* |edited:| "16-May-86 16:19")
    (DECLARE (GLOBALVARS DG.MENU-EDGE DG.MENU-FONT))
    (PROG (MENUITEMS TEMP)
          (SETQ MENUITEMS '(("Create Dir" DG.CREATE-SUBDIR-COMMAND "Creates subdirectory. Will wait for parent
                             directory selection, and prompt for subdirectory name")
                          ("Delete Dir" DG.DELETE-DIRECTORY-COMMAND "Will delete entire directory including
                             all files in subdirectories. Will wait for directory selection and if the
                             directory is nonempty it will ask for confirmation")
                          ("Backup Dir" DG.BACKUP-ON-SELECTED-COMMAND "Will ask you to select the directory on
                             which the backup should be stored" (SUBITEMS ("On default"
                                                                         DG.BACKUP-ON-DEFAULT-COMMAND
                                                                         "Will backup the selected
                                                                         directory on the default
                                                                         backup directory (the
                                                                         value of
                                                                         DG.DEFAULT-BACKUP-DIRECTO
```
RY
```
                                                                         ) ")
                                                                        ("On selected"
                                                                         DG.BACKUP-ON-SELECTED-COMMAND
                                                                         "Will ask you to select the
                                                                         directory on which the
                                                                         backup should be stored")))
                          ("Restore Dir" DG.RESTORE-FROM-SELECTED-COMMAND "Will ask you to select the
                             directory from where the backup should be restored" (SUBITEMS ("From
                                                                                  default"
                                                                         DG.RESTORE-FROM-DEFAULT-COMMAND
                                                                                  "Will
                                                                                  restore
                                                                                  the
                                                                                  selected
                                                                                  director
```
Y
```
                                                                                  from the
```

```
                                                                          default
                                                                          backup
                                                                          director
y
                                                                          (the
                                                                          value of
                                                                          DG.DEFAU
LT-BACKUP-DIRECTORY
                                                                          ) ")
                                                          ("From selected"
                                             DG.RESTORE-FROM-SELECTED-COMMAND
                                                          "Will ask you to
                                                          select the
                                                          directory from
                                                          where the backup
                                                          should be
                                                          restored")))
                  ("Move Dir" DG.MOVE-DIR-COMMAND "Will move the selected directory (including
                          subdirectories) to a new parent directory.")
                  ("Copy Dir" DG.COPY-DIRECTORY-COMMAND "Will copy the selected directory (including
                          subdirectories) to a new parent directory.")
                  ("Connect Dir" DG.CONNECT-DIR-COMMAND "Changes the current directory")
                  ("Apply DG" DG.APPLY-DIRGRAPHER-COMMAND "Calls DirGrapher on the selected
                          directory")
                  ("Apply FB" DG.APPLY-FILEBROWSER-COMMAND "Calls FileBrowser on files in the selected
                          directory (but not  files in the subdirectories)." (SUBITEMS ("on directory
                                                          only"
                                             DG.APPLY-FILEBROWSER-COMMAND
                                                          "Calls
                                                          FileBrows
er
                                                          on files
                                                          in the
                                                          selected
                                                          directory
                                                          (but not
                                                          files in
                                                          the
                                                          subdirect
ories).")
                                                          ("On directory and
                                                          subdirectories"
                                             DG.APPLY-FB-ON-DIR-AND-SUBDIRS-COMMAND
                                                          "Calls FileBrowser
                                                          on files in the
                                                          selected directory
                                                          and
                                                          subdirectories."))
                  )
                  ("Files" NULL " The subitems of the <Files> item manipulate files within selected
                          directories" (SUBITEMS ("Move Files" DG.MOVE-FILE-COMMAND "Will ask you to
                                                          select: (1) source directory, (2) files to be
                                                          moved, and (3) target directory.")
                                             ("Copy Files" DG.COPY-FILE-COMMAND "Will ask you to
                                                          select: (1) source directory, (2) files to be
                                                          copied, and (3) target directory.")
                                             ("Load Files" (DG.LOAD-FILES-COMMAND LOAD NIL)
                                                          "Will ask you to select a directory and files to
                                                          be loaded"
                                                          (SUBITEMS ("LOAD" (DG.LOAD-FILES-COMMAND LOAD NIL
                                                                      )
                                                                      "Will ask you to select a
                                                                      directory and files to be
                                                                      loaded"
                                                                      (SUBITEMS ("LDFLG=NIL"
                                                                             (
                                                                      DG.LOAD-FILES-COMMAND
                                                                             LOAD NIL))
                                                                             ("LDFLG=SYSLOAD"
                                                                             (DG.LOAD-FILES-COMMAND
                                                                             LOAD SYSLOAD))))
                                                          ("LOAD?" (DG.LOAD-FILES-COMMAND LOAD? NIL)
                                                                      "Will ask you to select a directory
                                                                      and files to be loaded USING LOAD?
                                                                      function"
                                                                      (SUBITEMS ("LDFLG=NIL" (
                                                                             DG.LOAD-FILES-COMMAND
                                                                             LOAD? NIL))
                                                                             ("LDFLG=SYSLOAD" (
                                                                             DG.LOAD-FILES-COMMAND
                                                                             LOAD?
                                                                             SYSLOAD))))
                                                          ("LOADFROM" (DG.LOAD-FILES-COMMAND
                                                                      LOADFROM NIL)
                                                                      "Will ask you to select a directory
```

```
                                                              and files to be loaded USING
                                                              LOADFROM function"
                                                        (SUBITEMS ("LDFLG=NIL" (
                                                                        DG.LOAD-FILES-COMMAND
                                                                        LOADFROM NIL
                                                                        ))
                                                ("LDFLG=SYSLOAD" (
                                                                        DG.LOAD-FILES-COMMAND
                                                                        LOADFROM
                                                                        SYSLOAD))))
                                ))
                        ("Delete Files" DG.DELETE-SELECTED-FILES-COMMAND "Will
                                ask you to select a directory and files to be
                                deleted (EXPUNGED)." (SUBITEMS ("Selected Files"

                                                        DG.DELETE-SELECTED-FILES-COMMAND
                                                                "Will ask you to
                                                                select a directory
                                                                and files to be
                                                                deleted
                                                                (EXPUNGED).")
                                                        ("Old Versions"
                                                        DG.DELETE-OLD-VERSIONS-COMMAND
                                                                " Will ask you
                                                                to select a
                                                                directory, and
                                                                will delete all
                                                                old versions of
                                                                files in the
                                                                selected
                                                                directory. Only
                                                                DG.NUMBER-OF-VE
RSIONS

                                                                (1 by default)
                                                                last versions
                                                                will remain")))
                        ("Info" DG.FILE-INFO-COMMAND "Will ask you to select a
                                directory, and will display info on selected
                                files. ")
                        ("Hardcopy" DG.FILES-HARDCOPY-COMMAND "Will ask you to
                                select a directory and files to be hardcopied.")))
                ("Update" DG.UPDATE-COMMAND "Will update the tree structure according to the current
                                file system state")))
        (|if| (FMEMB DG.MENU-EDGE '(TOP BOTTOM))
                |then| (SETQ MENUITEMS (COPY MENUITEMS))
                        (SETQ TEMP (CADDR MENUITEMS))
                        (RPLACA (CDDR MENUITEMS)
                                (CAR (NTH MENUITEMS 8)))
                        (RPLACA (NTH MENUITEMS 8)
                                TEMP))
        (RETURN (|create| MENU
                        ITEMS _ MENUITEMS
                        MENUCOLUMNS _ (|if| (FMEMB DG.MENU-EDGE '(TOP BOTTOM))
                                        |then| 4
                                        |else| 1)
                        WHENSELECTEDFN _ 'DG.DOCOMMAND
                        MENUFONT _ DG.MENU-FONT)))))
```

(DG.CREATE-PATH-FROM-FILE-NAME

```
  (LAMBDA (FNAME)                                       (* |edited:| " 3-Apr-85 15:41")
    (PROG (DIR HOST)
            (SETQ DIR (FILENAMEFIELD FNAME 'DIRECTORY))
            (SETQ HOST (FILENAMEFIELD FNAME 'HOST))
            (RETURN (|if| HOST
                        |then| (CONS (PACKFILENAME 'HOST (FILENAMEFIELD FNAME 'HOST))
                                        (|if| (NULL DIR)
                                                |then| NIL
                                                |else| (DG.UNPACK-DIRECTORY-NAME (UNPACK DIR))))
                        |else| (|if| (NULL DIR)
                                        |then| NIL
                                        |else| (DG.UNPACK-DIRECTORY-NAME (UNPACK DIR))))))))
```

(DG.CREATE-PREFIX-FROM-PATH

```
  (LAMBDA (PATH)                                        (* |edited:| "17-Jul-84 14:58")
    (COND
        ((CDR PATH)
         (PACK* (CAR PATH)
                '<
                (PACK (|for| D |in| (CDR PATH) |join| (LIST D '>)))))
        (T (CAR PATH)))))
```

(DG.CREATE-QUIT-MENU

```
  (LAMBDA (WINDOW)                                      (* |edited:| "12-May-86 13:06")
    (|create| MENU
```

```
            ITEMS _ (SUBST WINDOW 'WINDOW '((" I  n  t  e  r  r  u  p  t / R e s e t " (DG.KILL-PROCESS WINDOW)
                                                          "Will kill the current DirGrapher process if active, and will
                                                          reset the DirGrapher display.")))
            CENTERFLG _ T)))
```

(DG.CREATE-SUBDIR-COMMAND
```
  (LAMBDA (WINDOW)                                            (* |edited:| "30-Sep-85 14:45")
    (PROG (CURRENT-PATH NEW-DIR-NAME DIRECTORY-SUBTREE)
          (SETQ CURRENT-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select parent directory.")))
          (COND
            (CURRENT-PATH (SETQ DIRECTORY-SUBTREE (DG.FIND-DIRECTORY-SUBTREE CURRENT-PATH WINDOW))
                  (TERPRI (DG.PROMPTWINDOW WINDOW))
                  (SETQ NEW-DIR-NAME (MKATOM (PROMPTFORWORD (CONCAT "Enter name for new subdirectory (of "
                                                      (CAR DIRECTORY-SUBTREE)
                                                      ") :")
                                          NIL NIL (DG.PROMPTWINDOW WINDOW)
                                          NIL
                                          'TTY)))
                  (COND
                    ((MEMBER NEW-DIR-NAME (CDR DIRECTORY-SUBTREE)))
                    (T (RPLACD DIRECTORY-SUBTREE (CONS (LIST NEW-DIR-NAME)
                                                    (CDR DIRECTORY-SUBTREE)))
                      (DG.DISPLAY-DIRECTORY-TREE (DG.CREATE-GRAPH-FROM-TREE (WINDOWPROP WINDOW
                                                                    'DG.TREE))
                              WINDOW)))))))))
```

(DG.CREATE-TREE-FROM-LISTS
```
  (LAMBDA (LISTS)                                            (* |edited:| "13-May-86 12:10")
    (PROG (TEMP L2)
          (|for| X |in| LISTS |when| X |do| (COND
                                          ((NOT (SETQ TEMP (FASSOC (CAR X)
                                                                L2)))
                                          (SETQ L2 (CONS (LIST (CAR X)
                                                              (CDR X))
                                                        L2)))
                                          (T (RPLACD TEMP (CONS (CDR X)
                                                              (CDR TEMP)))))
                                        (BLOCK))
          (RETURN (|for| X |in| L2 |collect| (CONS (CAR X)
                                          (DG.CREATE-TREE-FROM-LISTS (CDR X)))))))))
```

(DG.CREATE-TREE-NODES
```
  (LAMBDA (TREE FORMER-PATH)                                  (* |edited:| " 3-Apr-85 22:05")
    (PROG (CURRENT-ID TO-NODES CURRENT-NODE)
          (COND
            (TREE (RETURN (CONS (SETQ CURRENT-ID (APPEND FORMER-PATH (LIST (CAR TREE))))
                          (CONS (SETQ CURRENT-NODE (|create| GRAPHNODE
                                                      NODEID _ CURRENT-ID
                                                      NODELABEL _ (CAR TREE)
                                                      FROMNODES _ (LIST FORMER-PATH)
                                                      NODEBORDER _ -2))
                              (PROG (SON-LIST)
                                    (SETQ SON-LIST (|for| SON |in| (CDR TREE)
                                                  |join| (PROG (N-LIST)
                                                              (SETQ N-LIST (DG.CREATE-TREE-NODES
                                                                          SON CURRENT-ID))
                                                              (SETQ TO-NODES (CONS (CAR N-LIST)
                                                                              TO-NODES))
                                                              (RETURN (CDR N-LIST)))))
                                    (|replace| TONODES |of| CURRENT-NODE |with| TO-NODES)
                                    (RETURN SON-LIST)))))))))))
```

(DG.CREATEICONW
```
  (LAMBDA (WINDOW ICON)                                      (* |edited:| " 5-Apr-85 11:13")
    (COND
      ((NULL ICON)
        (SETQ ICON (TITLEDICONW (|create| TITLEDICON
                                      ICON _ DG.ICON
                                      MASK _ DG.MASK
                                      TITLEREG _ (CREATEREGION 5 5 65 60))
                              (WINDOWPROP WINDOW 'DG.ICONTITLE)
                              DG.ICONFONT))))
      ICON))
```

(DG.CURRENT-CONNECTED-DIRECTORY
```
  (LAMBDA NIL                                                (* |edited:| "22-May-85 13:55")
    (PROG (CURRENT-DIR)
          (SETQ CURRENT-DIR (DIRECTORYNAME T T))
          (|if| (NOT (FMEMB (CAR (LAST (UNPACK CURRENT-DIR)))
                          '({} >)))
            |then| (RETURN (PACK* CURRENT-DIR '>))
          |else| (RETURN CURRENT-DIR)))))
```

(**DG.DELETE-DIRECTORY-COMMAND**
```
  (LAMBDA (WINDOW)                                              (* |edited:| "19-Apr-85 14:23")
    (PROG (CURRENT-PATH CURRENT-DIRECTORY FATHER-DIRECTORY FILES-TO-BE-DELETED NUMBER-OF-FILES)
          (SETQ CURRENT-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select directory to be deleted")))
          (COND
             (CURRENT-PATH (INVERTW WINDOW)
                   (SETQ FILES-TO-BE-DELETED (DIRECTORY (DG.CREATE-PREFIX-FROM-PATH CURRENT-PATH)))
                   (INVERTW WINDOW)
                   (SETQ NUMBER-OF-FILES (LENGTH FILES-TO-BE-DELETED))
                   (COND
                      ((GREATERP NUMBER-OF-FILES 0)
                       (|printout| (DG.PROMPTWINDOW WINDOW)
                             T "CAUTION !!! " NUMBER-OF-FILES " files are going to be deleted !!" "Confirm
                             with left button .")
                       (FLASHWINDOW (DG.PROMPTWINDOW WINDOW)
                             2)
                       (COND
                          ((MOUSECONFIRM)
                           (INVERTW WINDOW)
                           (|for| F |in| FILES-TO-BE-DELETED |do| (DELFILE F)
                                                        (|printout| (DG.PROMPTWINDOW WINDOW)
                                                              T "File " F " was deleted."))
                           (INVERTW WINDOW))
                          (T (|printout| (DG.PROMPTWINDOW WINDOW)
                                    T "Delete directory aborted.")
                             (RETURN NIL)))))
                   (COND
                      ((EQUAL CURRENT-PATH (WINDOWPROP WINDOW 'DG.PATH-TO-ROOT))
                       (CLOSEW WINDOW)
                       (RETURN NIL)))
                   (DG.DELETE-FROM-TREE CURRENT-PATH WINDOW))))))
```

(**DG.DELETE-FROM-TREE**
```
  (LAMBDA (PATH WINDOW)                                         (* |edited:| "13-May-86 13:58")
    (PROG (PARENT-AND-SON FATHER-DIRECTORY)
          (SETQ PARENT-AND-SON (DG.FIND-PARENT-DIERCTORY PATH WINDOW))
          (SETQ FATHER-DIRECTORY (CAR PARENT-AND-SON))
          (RPLACD FATHER-DIRECTORY (REMOVE (CADR PARENT-AND-SON)
                                           (CDR FATHER-DIRECTORY)))
          (DG.DISPLAY-DIRECTORY-TREE (DG.CREATE-GRAPH-FROM-TREE (WINDOWPROP WINDOW 'DG.TREE))
                 WINDOW))))
```

(**DG.DELETE-OLD-VERSIONS-COMMAND**
```
  (LAMBDA (WINDOW)                                              (* |edited:| "13-May-86 12:07")
    (PROG (SOURCE-PATH SELECTED-FILES SOURCE-PREFIX MENU-WINDOW SOURCE-PATH-WINDOW)
          (SETQ SOURCE-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select directory from which you want to
                                   delete old versions of files.")))
          (COND
             (SOURCE-PATH (SETQ SOURCE-PREFIX (DG.CREATE-PREFIX-FROM-PATH SOURCE-PATH))
                   (SETQ SELECTED-FILES (DG.EXCLUSIVE-OLD-VERSIONS SOURCE-PREFIX))
                   (|if| SELECTED-FILES
                       |then| (PRINTOUT (DG.PROMPTWINDOW WINDOW)
                                    T "  " (LENGTH SELECTED-FILES)
                                    "  files will be DELETED from " SOURCE-PREFIX " .  Approve with left button.
")
                              (|if| (MOUSECONFIRM)
                                  |then| (INVERTW WINDOW)
                                         (|for| FILE |in| SELECTED-FILES |do| (DELFILE FILE)
                                                           (PRINTOUT (DG.PROMPTWINDOW WINDOW)
                                                                 T FILE " was deleted. ")
                                                           (BLOCK))
                                         (INVERTW WINDOW)
                                  |else| (PRINTOUT (DG.PROMPTWINDOW WINDOW)
                                               T " Delete files aborted.")))
                   (DETACHWINDOW MENU-WINDOW)
                   (CLOSEW MENU-WINDOW))))))
```

(**DG.DELETE-SELECTED-FILES-COMMAND**
```
  (LAMBDA (WINDOW)                                              (* |edited:| "13-May-86 16:31")
    (DECLARE (GLOBALVARS HIGHLIGHTSHADE))
    (PROG (SOURCE-PATH SELECTED-FILES SOURCE-PREFIX MENU-WINDOW SOURCE-PATH-WINDOW)
          (SETQ SOURCE-PATH (CAR (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select directory from which you want to
                                   delete selected files.")))
          (COND
             (SOURCE-PATH (SETQ SOURCE-PREFIX (DG.CREATE-PREFIX-FROM-PATH SOURCE-PATH))
                   (SETQ MENU-WINDOW (DG.GET-FILE-LIST (CONCAT " The selected files will be " " DELETED
                                                        (EXPUNGED) " " from " SOURCE-PREFIX " .")
                                           WINDOW SOURCE-PATH WINDOW))
                   (SETQ SELECTED-FILES (WINDOWPROP MENU-WINDOW 'DG.SELECTED-FILES))
                   (|if| SELECTED-FILES
                       |then| (PRINTOUT (DG.PROMPTWINDOW WINDOW)
                                    T "  " (LENGTH SELECTED-FILES)
```

```
                                       " files will be DELETED from " SOURCE-PREFIX " .  Approve with left button.
")
                                  (|if| (MOUSECONFIRM)
                                     |then| (INVERTW WINDOW)
                                            (|for| FILE |in| SELECTED-FILES
                                                 |do| (DELFILE (PACK* SOURCE-PREFIX FILE))
                                                      (SHADEITEM FILE (CAR (WINDOWPROP MENU-WINDOW 'MENU))
                                                                 HIGHLIGHTSHADE MENU-WINDOW)
                                                      (BLOCK))
                                            (INVERTW WINDOW)
                                     |else| (PRINTOUT (DG.PROMPTWINDOW WINDOW)
                                                   T " Delete files aborted.")))
                        (DETACHWINDOW MENU-WINDOW)
                        (CLOSEW MENU-WINDOW))))))
```

₍**DG.DIRECTORY-MINUS-PREFIX**

```
  (LAMBDA (DIRECTORY PREFIX)                               (* |edited:| "27-Mar-85 23:50")
    (COND
       (PREFIX (SUBATOM DIRECTORY (IPLUS 2 (NCHARS PREFIX))
                        (NCHARS DIRECTORY)))
       (T DIRECTORY))))
```

₍**DG.DIRECTORY-SELECTION-FUNCTION**

```
  (LAMBDA (SELECTED-OBJ G-WINDOW)                          (* |edited:| "13-May-86 14:26")
     (DECLARE (GLOBALVARS DG.LAST-DIRECTORY-SELECTED DG.WINDOW-OF-LAST-DIRECTORY DG.DIRECTORY-WAS-SELECTED-EVENT
                  ))
     (PROG (CURRENT-NODE)
           (COND
              (SELECTED-OBJ (SETQ DG.LAST-DIRECTORY-SELECTED (APPEND (WINDOWPROP G-WINDOW 'DG.PATH-TO-ROOT)
                                                              (CDR (|fetch| NODEID |of| SELECTED-OBJ)))))
                    (SETQ DG.WINDOW-OF-LAST-DIRECTORY G-WINDOW)))
           (NOTIFY.EVENT DG.DIRECTORY-WAS-SELECTED-EVENT T))))
```

₍**DG.DISPLAY-DIRECTORY-TREE**

```
  (LAMBDA (GRAPH WINDOW)                                   (* |sm| " 8-Jul-85 17:27")
     (SHOWGRAPH GRAPH WINDOW 'DG.DIRECTORY-SELECTION-FUNCTION 'DG.DIRECTORY-SELECTION-FUNCTION)
     (WINDOWPROP WINDOW 'REPAINTFN (CONS 'DG.REDISPLAYFN (MKLIST (WINDOWPROP WINDOW 'REPAINTFN))))
     (WINDOWPROP WINDOW 'COPYBUTTONEVENTFN 'DG.COPYBUTTONEVENTFN)
     (DG.SHADE-CURRENT-DIRECTORY WINDOW)))
```

₍**DG.DOCOMMAND**

```
  (LAMBDA (ITEM MENU KEY)                                  (* |edited:| "13-May-86 14:27")
     (DECLARE (GLOBALVARS DG.CONCURRENT DG.CONCURRENT-ALL DG.MONITOR-LOCK))
     (PROG (WINDOW)
           (SETQ WINDOW (MAINWINDOW (WFROMMENU MENU)))
           (COND
              ((NOT DG.CONCURRENT-ALL)
               (DG.DOCOMMAND-WITH-MONITOR DG.MONITOR-LOCK WINDOW ITEM MENU))
              ((NOT DG.CONCURRENT)
               (DG.DOCOMMAND-WITH-MONITOR (WINDOWPROP WINDOW 'DG.LOCK)
                       WINDOW ITEM MENU))
              (T (DG.DOCOMMAND-WITHOUT-MONITOR WINDOW ITEM MENU))))))
```

₍**DG.DOCOMMAND-WITH-MONITOR**

```
  (LAMBDA (MONITOR-LOCK WINDOW ITEM MENU)                  (* |edited:| " 8-May-86 12:10")
     (PROG (PARENT-ITEM)
           (SETQ PARENT-ITEM (DG.FIND-MENU-ITEM ITEM (|fetch| ITEMS |of| MENU)))
           (WINDOWPROP WINDOW 'DG.PROCESS
                   (ADD.PROCESS (SUBPAIR '(MONITOR-LOCK WINDOW MENU ITEM PARENT-ITEM)
                                         (LIST MONITOR-LOCK WINDOW MENU ITEM PARENT-ITEM)
                                         '(PROGN (WITH.MONITOR MONITOR-LOCK
                                                        (TTYDISPLAYSTREAM (DG.PROMPTWINDOW WINDOW))
                                                        (SHADEITEM 'PARENT-ITEM MENU GRAYSHADE)
                                                        (WINDOWPROP WINDOW 'DG.UNSHADE-IF-SHADED
                                                               '(SHADEITEM 'PARENT-ITEM MENU WHITESHADE))
                                                        (CLEARW (DG.PROMPTWINDOW WINDOW))
                                                        (IF (ATOM (CADR 'ITEM))
                                                            THEN (APPLY* (CADR 'ITEM)
                                                                         WINDOW)
                                                          ELSE (APPLY (CAADR 'ITEM)
                                                                      (CONS WINDOW (CDADR 'ITEM))))
                                                        (PRINTOUT (DG.PROMPTWINDOW WINDOW)
                                                              T
                                                              (CAR 'PARENT-ITEM)
                                                              " Completed. ")
                                                        (WINDOWPROP WINDOW 'DG.UNSHADE-IF-SHADED NIL)
                                                        (SHADEITEM 'PARENT-ITEM MENU WHITESHADE)
                                                        (WINDOWPROP WINDOW 'DG.SHADED-ITEM NIL)
                                                        (WINDOWPROP WINDOW 'DG.PROCESS NIL)))
                           'WINDOW
                           (DG.PROMPTWINDOW WINDOW)
                           'NAME
```

```
                          (CAR PARENT-ITEM))))))
```

(**DG.DOCOMMAND-WITHOUT-MONITOR**
```
  (LAMBDA (WINDOW ITEM MENU)                                (* |edited:| " 8-May-86 12:10")
    (WINDOWPROP WINDOW 'DG.PROCESS (ADD.PROCESS
                                    (SUBPAIR '(WINDOW MENU ITEM)
                                            (LIST WINDOW MENU ITEM)
                                          '(PROGN (TTYDISPLAYSTREAM (DG.PROMPTWINDOW WINDOW))
                                                  (SHADEITEM 'ITEM MENU GRAYSHADE)
                                                  (WINDOWPROP WINDOW 'DG.SHADED-ITEM 'ITEM)
                                                  (CLEARW (DG.PROMPTWINDOW WINDOW))
                                                  (IF (ATOM (CADR 'ITEM))
                                                      THEN (APPLY* (CADR 'ITEM)
                                                                    WINDOW)
                                                     ELSE (APPLY (CAADR 'ITEM)
                                                                 (CONS WINDOW (CDADR 'ITEM))))
                                                  (PRINTOUT (DG.PROMPTWINDOW WINDOW)
                                                            T
                                                            (CAR 'ITEM)
                                                            " Completed. ")
                                                  (SHADEITEM 'ITEM MENU WHITESHADE)
                                                  (WINDOWPROP WINDOW 'DG.SHADED-ITEM NIL)
                                                  (WINDOWPROP WINDOW 'DG.PROCESS NIL)))
                                    'WINDOW
                                    (DG.PROMPTWINDOW WINDOW)
                                    'NAME
                                    (CAR ITEM)))))
```

(**DG.EXCLUSIVE-DIRECTORY**
```
  (LAMBDA (DIRECTORY-PATTERN)                                (* |edited:| "22-May-85 12:23")
    (PROG (CURRENT-DIRECTORY)
          (SETQ CURRENT-DIRECTORY (FILENAMEFIELD DIRECTORY-PATTERN 'DIRECTORY))
          (RETURN (|for| F |in| (DIRECTORY DIRECTORY-PATTERN) |when| (EQ (FILENAMEFIELD F 'DIRECTORY)
                                                                        CURRENT-DIRECTORY)
                  |collect| F)))))
```

(**DG.EXCLUSIVE-NEW-VERSIONS**
```
  (LAMBDA (DIRECTORY-PATTERN)                                (* |edited:| "16-May-86 16:24")
    (PROG (CURRENT-DIRECTORY)
          (SETQ CURRENT-DIRECTORY (FILENAMEFIELD DIRECTORY-PATTERN 'DIRECTORY))
          (RETURN (|for| FILE |in| (|for| F |in| (DIRECTORY DIRECTORY-PATTERN) |when| (EQ (FILENAMEFIELD F 'DIRECTORY)
                                                                                         CURRENT-DIRECTORY)
                                    |collect| F)
                  |when| (EQUAL FILE (INFILEP (PACKFILENAME.STRING 'VERSION NIL 'BODY FILE))) |collect| FILE)))))
```

(**DG.EXCLUSIVE-OLD-VERSIONS**
```
  (LAMBDA (DIRECTORY-PATTERN)                                (* |edited:| " 7-May-86 17:59")
    (PROG (CURRENT-DIRECTORY)
          (SETQ CURRENT-DIRECTORY (FILENAMEFIELD DIRECTORY-PATTERN 'DIRECTORY))
          (RETURN (|for| FILE |in| (|for| F |in| (DIRECTORY DIRECTORY-PATTERN) |when| (EQ (FILENAMEFIELD F 'DIRECTORY)
                                                                                         CURRENT-DIRECTORY)
                                    |collect| F)
                  |when| (NOT (EQUAL FILE (INFILEP (PACKFILENAME.STRING 'VERSION NIL 'BODY FILE))))
                  |collect| FILE)))))
```

(**DG.FBICONFN**
```
  (LAMBDA (WINDOW ICON)                                      (* |edited:| "13-May-86 14:25")
    (DECLARE (GLOBALVARS FILEDRAWER FILEDRAWERREGION DG.ICONFONT))
    (COND
      ((NULL ICON)
       (SETQ ICON (TITLEDICONW (|create| TITLEDICON
                                       ICON _ FILEDRAWER
                                       TITLEREG _ FILEDRAWERREGION)
                              (DG.CREATE-ICON-TITLE (DG.CREATE-PATH-FROM-FILE-NAME (DG.GET-FB-PATTERN WINDOW)))
                              DG.ICONFONT)))
      ((ICONW.TITLE ICON (DG.CREATE-ICON-TITLE (DG.CREATE-PATH-FROM-FILE-NAME (DG.GET-FB-PATTERN WINDOW))))))
    ICON))
```

(**DG.FILE-SELECTION-FN**
```
  (LAMBDA (ITEM MENU KEY)                                    (* |edited:| "13-May-86 14:08")
    (DECLARE (GLOBALVARS WHITESHADE BLACKSHADE))
    (PROG (WINDOW)
          (SETQ WINDOW (WFROMMENU MENU))
          (COND
            ((EQ ITEM '\ STOP)
             (NOTIFY.EVENT (WINDOWPROP (MAINWINDOW WINDOW)
                                       'DG.FILE-SELECTION-ENDED-EVENT)))
            ((EQUAL ITEM " "))
            (ITEM (COND
                    ((FMEMB ITEM (WINDOWPROP WINDOW 'DG.SELECTED-FILES))
                     (WINDOWDELPROP WINDOW 'DG.SELECTED-FILES ITEM)
```

```
                              (SHADEITEM ITEM MENU WHITESHADE WINDOW))
                        (T (WINDOWADDPROP WINDOW 'DG.SELECTED-FILES ITEM)
                           (SHADEITEM ITEM MENU BLACKSHADE WINDOW)))))))))
```

(**DG.FILECOPY**
```
  (LAMBDA (F1 F2 OVER-FLAG)                              (* |edited:| " 6-Mar-86 15:06")
    (COND
       ((DG.FILE-EXISTS F2)
        (COND
           ((OR OVER-FLAG (DG.NEWER-FILE F1 F2))
            (DELFILE F2)
            (COPYFILE F1 F2))
           (T NIL)))
       (T (COPYFILE F1 F2)
          T))))
```

(**DG.FIND-DIRECTORY-SUBTREE**
```
  (LAMBDA (PATH WINDOW)                                 (* |edited:| "31-Dec-00 22:10")
    (PROG (DIRECTORY-SUBTREE)
          (SETQ PATH (NTH PATH (LENGTH (WINDOWPROP WINDOW 'DG.PATH-TO-ROOT))))
          (SETQ DIRECTORY-SUBTREE (LIST (WINDOWPROP WINDOW 'DG.TREE)))
          (|for| X |in| PATH |do| (SETQ DIRECTORY-SUBTREE (FASSOC X DIRECTORY-SUBTREE)))
          (RETURN DIRECTORY-SUBTREE))))
```

(**DG.FIND-MENU-ITEM**
```
  (LAMBDA (ITEM ITEM-LIST)                              (* |edited:| "26-Sep-85 14:19")
    (COND
       ((NULL ITEM-LIST)
        NIL)
       ((ATOM ITEM-LIST)
        NIL)
       ((FMEMB ITEM ITEM-LIST)
        ITEM)
       ((AND (CDDDR (CAR ITEM-LIST))
             (EQ (CAR (CADDDR (CAR ITEM-LIST)))
                 'SUBITEMS)
             (DG.FIND-MENU-ITEM ITEM (CDR (CADDDR (CAR ITEM-LIST)))))
        (CAR ITEM-LIST))
       (T (DG.FIND-MENU-ITEM ITEM (CDR ITEM-LIST))))))
```

(**DG.FIND-PARENT-DIERCTORY**
```
  (LAMBDA (PATH WINDOW)                                 (* |edited:| "30-Mar-85 16:29")
    (PROG (FATHER-DIRECTORY CURRENT-DIRECTORY)
          (SETQ FATHER-DIRECTORY (WINDOWPROP WINDOW 'DG.TREE))
          (SETQ PATH (CDR (DG.RESIDUAL-PATH WINDOW PATH)))
          (|while| (AND (SETQ CURRENT-DIRECTORY (FASSOC (CAR PATH)
                                                        FATHER-DIRECTORY))
                        (CDR PATH))
             |do| (SETQ FATHER-DIRECTORY CURRENT-DIRECTORY)
                  (SETQ PATH (CDR PATH)))
          (|if| (NOT (LISTP CURRENT-DIRECTORY))
              |then| (RETURN (LIST NIL FATHER-DIRECTORY))
              |else| (RETURN (LIST FATHER-DIRECTORY CURRENT-DIRECTORY))))))
```

(**DG.FLASHALLWINDOWS**
```
  (LAMBDA (WINDOW)                                      (* |edited:| " 3-Apr-85 16:21")
    (|for| W |in| (CONS WINDOW (ATTACHEDWINDOWS WINDOW)) |do| (FLASHWINDOW W))))
```

(**DG.GET-FB-PATTERN**
```
  (LAMBDA (W)                                           (* |edited:| " 7-May-86 12:50")
    (FETCHFIELD '(FILEBROWSER 14 POINTER)
          (WINDOWPROP (MAINWINDOW W)
                'FILEBROWSER))))
```

(**DG.GET-FILE-LIST**
```
  (LAMBDA (MESSAGE WINDOW SOURCE-PATH SOURCE-WINDOW)    (* |edited:| "16-May-86 16:16")
    (PROG (MENU-WINDOW)
          (INVERTW WINDOW)
          (SETQ MENU-WINDOW (DG.ATTACH-DIRECTORY-FILES-MENU SOURCE-PATH SOURCE-WINDOW WINDOW))
          (INVERTW WINDOW)
          (|if| MENU-WINDOW
              |then| (WINDOWPROP MENU-WINDOW 'DG.DETACH T)
                     (CLEARW (DG.PROMPTWINDOW WINDOW))
                     (|printout| (DG.PROMPTWINDOW WINDOW)
                             "Select files from menu. When done select STOP." T MESSAGE)
                     (AWAIT.EVENT (WINDOWPROP WINDOW 'DG.FILE-SELECTION-ENDED-EVENT)
                             1000000))
          (RETURN MENU-WINDOW))))
```

(**DG.GET-INITIAL-REGION**
  (LAMBDA (GRAPH)   ; Edited 19-Aug-87 08:55 by smL
    (DECLARE (GLOBALVARS DG.MAX-WIDTH DG.MAX-HEIGHT DG.MIN-WIDTH DG.MIN-HEIGHT))

    ;; Edited by smL to use WIDTHIFWINDOW and HEIGHTIFWINDOW

    (LET ((G-REGION (GRAPHREGION GRAPH)))
         (GETMOUSESTATE)
         (GETREGION (MIN DG.MAX-WIDTH (WIDTHIFWINDOW (MAX (|fetch| WIDTH |of| G-REGION)
                                                         DG.MIN-WIDTH)))
                    (MIN DG.MAX-HEIGHT (HEIGHTIFWINDOW (MAX (|fetch| HEIGHT |of| G-REGION)
                                                           DG.MIN-HEIGHT)
                                       T)))))))

(**DG.KILL-PROCESS**
  (LAMBDA (WINDOW)   (* |edited:| "12-May-86 13:03")
    (COND
       ((WINDOWPROP WINDOW 'DG.PROCESS)
        (DEL.PROCESS (WINDOWPROP WINDOW 'DG.PROCESS))
        (**DG.FLASHALLWINDOWS** WINDOW)
        (CLEARW (**DG.PROMPTWINDOW** WINDOW))
        (|printout| (**DG.PROMPTWINDOW** WINDOW)
               "User interupt. Directory Grapher process aborted.")))
    (|for| W |in| (ATTACHEDWINDOWS WINDOW) |when| (WINDOWPROP W 'DG.DETACH) |do| (DETACHWINDOW W)
                                                                             (CLOSEW W))

    (WINDOWPROP (**DG.PROMPTWINDOW** WINDOW)
           'TITLE NIL)
    (REDISPLAYW WINDOW)
    (EVAL (WINDOWPROP WINDOW 'DG.UNSHADE-IF-SHADED))
    (WINDOWPROP WINDOW 'DG.PROCESS NIL)))

(**DG.MOVE-DIR-COMMAND**
  (LAMBDA (WINDOW)   (* |edited:| " 2-Apr-85 18:02")
    (**DG.MOVECOPY-DIR-COMMAND** WINDOW T)))

(**DG.MOVE-FILE-COMMAND**
  (LAMBDA (WINDOW)   (* |edited:| " 2-Apr-85 18:05")
    (**DG.MOVECOPY-COMMAND** WINDOW T)))

(**DG.MOVECOPY-COMMAND**
  (LAMBDA (WINDOW MOVE)   (* |edited:| "20-May-86 23:22")
    (DECLARE (GLOBALVARS DG.COPY-OVER HIGHLIGHTSHADE))
    (PROG (TARGET-PATH SOURCE-PATH SELECTED-FILES TARGET-PREFIX SOURCE-PREFIX MENU-WINDOW SOURCE-WINDOW
                SOURCE-PATH-WINDOW)
          (SETQ SOURCE-PATH-WINDOW (**DG.WAIT-FOR-DIR-SELECTION** WINDOW "Select source directory " T))
          (SETQ SOURCE-WINDOW (CADR SOURCE-PATH-WINDOW))
          (SETQ SOURCE-PATH (CAR SOURCE-PATH-WINDOW))
          (COND
             (SOURCE-PATH (SETQ SOURCE-PREFIX (**DG.CREATE-PREFIX-FROM-PATH** SOURCE-PATH))
                    (SETQ TARGET-PATH (CAR (**DG.WAIT-FOR-DIR-SELECTION** WINDOW "Select target directory " T)))
                    (|if| TARGET-PATH
                        |then| (SETQ TARGET-PREFIX (**DG.CREATE-PREFIX-FROM-PATH** TARGET-PATH))
                            (SETQ MENU-WINDOW (**DG.GET-FILE-LIST** (CONCAT " The selected files will be "
                                                                       (|if| MOVE
                                                                           |then| " MOVED "
                                                                         |else| " COPIED ")
                                                                       " from " SOURCE-PREFIX " to "
                                                                       TARGET-PREFIX " .")
                                                     WINDOW SOURCE-PATH SOURCE-WINDOW))
                            (SETQ SELECTED-FILES (WINDOWPROP MENU-WINDOW 'DG.SELECTED-FILES))
                            (|if| SELECTED-FILES
                                |then| (INVERTW WINDOW)
                                    (|for| FILE |in| SELECTED-FILES
                                       |do| (|if| (**DG.FILECOPY** (PACK* SOURCE-PREFIX FILE)
                                                          (PACK* TARGET-PREFIX FILE)
                                                          DG.COPY-OVER)
                                                |then| (|printout| (**DG.PROMPTWINDOW** WINDOW)
                                                               T FILE " : " SOURCE-PREFIX " --> " TARGET-PREFIX)
                                              |else| (|printout| (**DG.PROMPTWINDOW** WINDOW)
                                                             T FILE " already exists and was not copied. "))
                                           (|if| MOVE
                                               |then| (DELFILE (PACK* SOURCE-PREFIX FILE)))
                                           (SHADEITEM FILE (CAR (WINDOWPROP MENU-WINDOW 'MENU))
                                                  HIGHLIGHTSHADE MENU-WINDOW)
                                           (BLOCK))
                                    (INVERTW WINDOW))
                            (DETACHWINDOW MENU-WINDOW)
                            (CLOSEW MENU-WINDOW)))))))

(**DG.MOVECOPY-DIR-COMMAND**
  (LAMBDA (WINDOW MOVE)   (* |edited:| "13-May-86 14:13")
    (DECLARE (GLOBALVARS DG.COPY-OVER))
    (PROG (TARGET-PATH SOURCE-PATH SELECTED-FILES TARGET-PREFIX SOURCE-PREFIX SOURCE-WINDOW-PATH MENU-WINDOW

```
                   SOURCE-FATHER-SON TARGET-FATHER-SON TARGET-PATH-WINDOW SOURCE-WINDOW TARGET-WINDOW FILE-NAME
                   NEW-NAME NEW-PREFIX)
           (SETQ SOURCE-WINDOW-PATH (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select the directory that you want to
                                     transfer." T))
           (SETQ SOURCE-PATH (CAR SOURCE-WINDOW-PATH))
           (SETQ SOURCE-WINDOW (CADR SOURCE-WINDOW-PATH))
           (COND
              (SOURCE-PATH (INVERTW WINDOW)
                     (SETQ SOURCE-PREFIX (DG.CREATE-PREFIX-FROM-PATH SOURCE-PATH))
                     (SETQ SELECTED-FILES (DIRECTORY SOURCE-PREFIX))
                     (INVERTW WINDOW)
                     (SETQ TARGET-PATH-WINDOW (DG.WAIT-FOR-DIR-SELECTION WINDOW (CONCAT "Select new parent for
                                                                            directory "
                                                                        (CAR (LAST SOURCE-PATH)))
                                                T))
                     (SETQ TARGET-PATH (CAR TARGET-PATH-WINDOW))
                     (|if| (AND MOVE (GREATERP (LENGTH TARGET-PATH)
                                         (LENGTH SOURCE-PATH))
                                (|for| A1 |in| SOURCE-PATH |as| A2 |in| TARGET-PATH |always| (EQUAL A1 A2)))
                        |then| (DG.FLASHALLWINDOWS WINDOW)
                             (|printout| (DG.PROMPTWINDOW WINDOW)
                                    T "Can not move a directory to its descendants")
                             (RETURN NIL))
                     (SETQ TARGET-WINDOW (CADR TARGET-PATH-WINDOW))
                     (COND
                        (TARGET-PATH
                           (INVERTW WINDOW)
                           (SETQ TARGET-PREFIX (DG.CREATE-PREFIX-FROM-PATH TARGET-PATH))
                           (|for| FILE |in| SELECTED-FILES
                              |do| (|if| (DG.FILECOPY FILE (SETQ NEW-NAME (PACK* (SETQ NEW-PREFIX
                                                                          (DG.CREATE-PREFIX-FROM-PATH
                                                                           (APPEND TARGET-PATH
                                                                                 (NTH (
                                                                                DG.CREATE-PATH-FROM-FILE-NAME
                                                                                   FILE)
                                                                                  (LENGTH SOURCE-PATH))))
                                                                        )
                                                                   (SETQ FILE-NAME (
                                                                                DG.PACK-NAME-VER-EXT
                                                                                 FILE)))))
                                               DG.COPY-OVER)
                                      |then| (|printout| (DG.PROMPTWINDOW WINDOW)
                                                  T FILE-NAME " : " (PACKFILENAME 'HOST (FILENAMEFIELD FILE
                                                                                     'HOST)
                                                                         'DIRECTORY
                                                                         (FILENAMEFIELD FILE 'DIRECTORY))
                                                  " --> " NEW-PREFIX)
                                      |else| (|printout| (DG.PROMPTWINDOW WINDOW)
                                                  T NEW-NAME " already exists and was not copied. "))
                                 (|if| MOVE
                                    |then| (DELFILE FILE))
                                 (BLOCK))
                           (SETQ SOURCE-FATHER-SON (DG.FIND-PARENT-DIERCTORY SOURCE-PATH SOURCE-WINDOW))
                           (SETQ TARGET-FATHER-SON (DG.FIND-PARENT-DIERCTORY TARGET-PATH TARGET-WINDOW))
                           (RPLACD (CADR TARGET-FATHER-SON)
                                  (CONS (COPY (CADR SOURCE-FATHER-SON))
                                        (CDR (CADR TARGET-FATHER-SON))))
                           (|if| MOVE
                              |then| (RPLACD (CAR SOURCE-FATHER-SON)
                                            (REMOVE (CADR SOURCE-FATHER-SON)
                                                  (CDAR SOURCE-FATHER-SON))))
                           (DG.DISPLAY-DIRECTORY-TREE (DG.CREATE-GRAPH-FROM-TREE (WINDOWPROP SOURCE-WINDOW
                                                                             'DG.TREE))
                                  SOURCE-WINDOW)
                           (|if| (NEQ TARGET-WINDOW SOURCE-WINDOW)
                              |then| (DG.DISPLAY-DIRECTORY-TREE (DG.CREATE-GRAPH-FROM-TREE (WINDOWPROP
                                                                             TARGET-WINDOW
                                                                             'DG.TREE))
                                            TARGET-WINDOW)))))))))
```

(**DG.PACK-NAME-VER-EXT**
  (LAMBDA (F)   (* |edited:| " 3-Apr-85 11:31")
    (PACKFILENAME 'NAME (FILENAMEFIELD F 'NAME)
          'VERSION
          (FILENAMEFIELD F 'VERSION)
          'EXTENSION
          (FILENAMEFIELD F 'EXTENSION))))

(**DG.PROMPTWINDOW**
  (LAMBDA (WINDOW)   (* |edited:| "29-Mar-85 16:07")
    (CAR (WINDOWPROP WINDOW 'PROMPTWINDOW))))

(**DG.REDISPLAYFN**
  (LAMBDA (WINDOW)   (* |edited:| "26-Sep-85 15:05")

```
      (DG.SHADE-CURRENT-DIRECTORY WINDOW)))
```

(**DG.RESIDUAL-PATH**
```
  (LAMBDA (WINDOW PATH)                                          (* |edited:| "29-Mar-85 12:37")
    (NTH PATH (LENGTH (WINDOWPROP WINDOW 'DG.PATH-TO-ROOT)))))
```

(**DG.RESTORE-COMMAND**
```
  (LAMBDA (WINDOW BACKUP-DIR)                                    (* |edited:| "12-May-86 14:57")
    (DECLARE (GLOBALVARS DG.FILES-BACKUP-NAMES DG.DIRECTORY-SUBTREE DG.COPY-OVER))
    (PROG (TARGET-PATH TARGET-DIRECTORY FILES-TO-BACKUP N OVER-WRITE-FLAG FLOPPY-FILES DSK-FILES
                 DIRECTORY-INFO-FILE-NAME F-NAME NEW-NAME OVER-FLAG-ASKED UNFOUND-FILES INFO-NAME BACKUP-HOST
                 FILES-BACKUP-NAMES TARGET-PATH-AND-WINDOW TARGET-WINDOW)
          (SETQ TARGET-PATH-AND-WINDOW (DG.WAIT-FOR-DIR-SELECTION WINDOW "Select directory that you want to
                                  restore" T))
          (SETQ TARGET-PATH (CAR TARGET-PATH-AND-WINDOW))
          (SETQ TARGET-WINDOW (CADR TARGET-PATH-AND-WINDOW))
          (SETQ TARGET-DIRECTORY (DG.FIND-DIRECTORY-SUBTREE TARGET-PATH TARGET-WINDOW))
          (SETQ BACKUP-HOST (FILENAMEFIELD BACKUP-DIR 'HOST))
          (|if| (EQ BACKUP-HOST 'FLOPPY)
              |then| (|printout| (DG.PROMPTWINDOW WINDOW)
                           T "Insert Floppy")
                     (FLOPPY.WAIT.FOR.FLOPPY))
          (|if| (NULL (FILENAMEFIELD BACKUP-DIR 'DIRECTORY))
              |then| (SETQ BACKUP-DIR (PACK* BACKUP-DIR '<)))
          (INVERTW WINDOW)
          (SETQ INFO-NAME (PACK* BACKUP-DIR (CAR TARGET-DIRECTORY)
                           '>BACKUPINFO))
          (|if| (NULL (DIRECTORY INFO-NAME))
              |then| (CLEARW (DG.PROMPTWINDOW WINDOW))
                     (INVERTW WINDOW)
                     (DG.FLASHALLWINDOWS WINDOW)
                     (|printout| (DG.PROMPTWINDOW WINDOW)
                           T "Couldn't find  the file " INFO-NAME " . Resore aborted. ")
            |else| (PROG NIL
                       (LOAD INFO-NAME 'SYSLOAD)
                       (RPLACD TARGET-DIRECTORY (APPEND (CDR TARGET-DIRECTORY)
                                               (|for| SUB |in| (CDR DG.DIRECTORY-SUBTREE)
                                                  |when| (NOT (FASSOC (CAR SUB)
                                                                 (CDR TARGET-DIRECTORY)))
                                                  |collect| SUB)))
                       (DG.DISPLAY-DIRECTORY-TREE (DG.CREATE-GRAPH-FROM-TREE (WINDOWPROP TARGET-WINDOW
                                                                       'DG.TREE))
                              TARGET-WINDOW)
                       (|if| (NEQ TARGET-WINDOW WINDOW)
                           |then| (INVERTW WINDOW))
                       (SETQ FILES-BACKUP-NAMES DG.FILES-BACKUP-NAMES)
              NEWROUNDOFCOPY
                       (INVERTW WINDOW)
                       (|for| F |in| FILES-BACKUP-NAMES |bind| F-WITH-<
                          |do| (COND
                                  ((DG.FILE-EXISTS (SETQ F-NAME (PACK* BACKUP-DIR F)))
                                   (SETQ NEW-NAME (PACK* (DG.CREATE-PREFIX-FROM-PATH
                                                       (APPEND TARGET-PATH (CDR (
                                                                        DG.CREATE-PATH-FROM-FILE-NAME
                                                                     (SETQ F-WITH-<
                                                                        (PACK* '< F))))))
                                                   (DG.PACK-NAME-VER-EXT F-WITH-<)))
                                   (COND
                                     ((DG.FILECOPY F-NAME NEW-NAME DG.COPY-OVER)
                                      (|printout| (DG.PROMPTWINDOW WINDOW)
                                            T NEW-NAME " Restored."))
                                     (T (|printout| (DG.PROMPTWINDOW WINDOW)
                                              T NEW-NAME " exists and wasn't restored. "))))
                                  (T (COND
                                       ((NOT (FMEMB F UNFOUND-FILES))
                                        (SETQ UNFOUND-FILES (CONS F UNFOUND-FILES)))))))
                       (INVERTW WINDOW)
                       (COND
                         (UNFOUND-FILES (RINGBELLS)
                                (DG.FLASHALLWINDOWS WINDOW)
                                (CLEARW (DG.PROMPTWINDOW WINDOW))
                                (|if| (EQ BACKUP-HOST 'FLOPPY)
                                    |then| (|printout| (DG.PROMPTWINDOW WINDOW)
                                                 (LENGTH UNFOUND-FILES)
                                                 " Files were not found on this floppy !! please insert another
                                                 one. ")
                                           (SETQ FILES-BACKUP-NAMES UNFOUND-FILES)
                                           (SETQ UNFOUND-FILES NIL)
                                           (FLOPPY.WAIT.FOR.FLOPPY T)
                                           (GO NEWROUNDOFCOPY)
                                    |else| (|printout| (DG.PROMPTWINDOW WINDOW)
                                                 (LENGTH UNFOUND-FILES)
                                                 " Files wre not found on the backup directory and were not
                                                 restored")))))))))))
```

(**DG.RESTORE-FROM-DEFAULT-COMMAND**
  (LAMBDA (WINDOW)   (* |edited:| "13-May-86 13:57")
    (**DECLARE** (GLOBALVARS DG.DEFAULT-BACKUP-DIRECTORY))
    (|**if**| (NULL DG.DEFAULT-BACKUP-DIRECTORY)
        |**then**| (**DG.FLASHALLWINDOWS** WINDOW)
              (|printout| (**DG.PROMPTWINDOW** WINDOW)
                     T "NULL default directory.Backup aborted")
      |**else**| (**DG.RESTORE-COMMAND** WINDOW DG.DEFAULT-BACKUP-DIRECTORY)))))

(**DG.RESTORE-FROM-SELECTED-COMMAND**
  (LAMBDA (WINDOW)   (* |sm| " 8-Jun-85 13:05")
    (PROG (BACKUP-DIRECTORY)
          (SETQ BACKUP-DIRECTORY (**DG.CREATE-PREFIX-FROM-PATH** (CAR (**DG.WAIT-FOR-DIR-SELECTION** WINDOW "Select the
                                                                directory that contains the backup."
                                                                T))))
          (**DG.RESTORE-COMMAND** WINDOW BACKUP-DIRECTORY))))

(**DG.SETIFY**
  (LAMBDA (L)   (* |edited:| "13-May-86 12:10")
    (PROG (NEW-SET)
          (|**for**| ONE-ELEMENT |**in**| L |**when**| (NOT (MEMBER ONE-ELEMENT NEW-SET)) |**do**| (SETQ NEW-SET (CONS ONE-ELEMENT
                                                                           NEW-SET))
                                                      (BLOCK))
          (RETURN NEW-SET))))

(**DG.SHADE-CURRENT-DIRECTORY**
  (LAMBDA (WINDOW)   (* |edited:| "22-May-85 13:56")
    (PROG (CURRENT-PATH)
          (SETQ CURRENT-PATH (NTH (**DG.CREATE-PATH-FROM-FILE-NAME** (**DG.CURRENT-CONNECTED-DIRECTORY**))
                                  (LENGTH (WINDOWPROP WINDOW 'DG.PATH-TO-ROOT))))
          (|**for**| NODE |**in**| (|**fetch**| GRAPHNODES |**of**| (WINDOWPROP WINDOW 'GRAPH))
             |**do**| (COND
                    ((EQUAL CURRENT-PATH (|**fetch**| NODEID |**of**| NODE))
                     (RESET/NODE/BORDER NODE 2 WINDOW))
                    (T (COND
                         ((GREATERP (|**fetch**| NODEBORDER |**of**| NODE)
                                 0)
                          (RESET/NODE/BORDER NODE -2 WINDOW)))))))))

(**DG.TRIM**
  (LAMBDA (W N)   (* |edited:| " 5-Apr-85 11:08")
    (COND
      ((GREATERP (NCHARS W)
              N)
       (SUBATOM W 1 N))
      (T W))))

(**DG.UNADVISE**
  (LAMBDA NIL   (* |edited:| "31-Mar-85 23:26")
    (UNADVISE (\\GENERATENEXTFILE IN FB.UPDATEBROWSERITEMS))))

(**DG.UNADVISE-GENERATE-FILE**
  (LAMBDA NIL   (* |edited:| "31-Mar-85 23:26")
    (UNADVISE (\\GENERATENEXTFILE IN FB.UPDATEBROWSERITEMS))))

(**DG.UNPACK-DIRECTORY-NAME**
  (LAMBDA (NAME)   (* |edited:| " 3-Apr-85 17:01")
    (COND
      ((NULL NAME)
       NIL)
      (T (CONS (PACK (PROG (PACKED-NAME)
                        LOOP
                           (COND
                             ((OR (NULL NAME)
                                  (EQ (CAR NAME)
                                      '>))
                              (RETURN PACKED-NAME))
                             (T (SETQ PACKED-NAME (NCONC1 PACKED-NAME (CAR NAME)))
                                (SETQ NAME (CDR NAME))
                                (GO LOOP)))))
               (**DG.UNPACK-DIRECTORY-NAME** (CDR NAME)))))))

(**DG.UPDATE-COMMAND**
  (LAMBDA (WINDOW)   (* |edited:| " 3-Apr-85 13:38")
    (INVERTW WINDOW)
    (**DG.UPDATE-DIRECTORY-TREE** WINDOW)))

(**DG.UPDATE-DIRECTORY-TREE**
```
  (LAMBDA (WINDOW)                                              (* |edited:| " 3-Apr-85 22:20")
    (PROG (TREE)
          (SETQ TREE (DG.CREATE-DIRECTORY-TREE (WINDOWPROP WINDOW 'DG.PATH-TO-ROOT)))
          (WINDOWPROP WINDOW 'DG.TREE TREE)
          (DG.DISPLAY-DIRECTORY-TREE (DG.CREATE-GRAPH-FROM-TREE TREE)
                WINDOW))))
```


(**DG.UPDATE-DIRECTROY-TREE**
```
  (LAMBDA (WINDOW)                                              (* |edited:| "30-Mar-85 15:21")
    (PROG (TREE)
          (SETQ TREE (DG.CREATE-DIRECTORY-TREE (WINDOWPROP WINDOW \'DG.PATH-TO-ROOT)))
          (DG.DISPLAY-DIRECTORY-TREE TREE WINDOW))))
```


(**DG.WAIT-FOR-DIR-SELECTION**
```
  (LAMBDA (WINDOW MESSAGE ALLOW-OTHER-WINDOWS)                  (* |edited:| "13-May-86 14:31")
    (DECLARE (GLOBALVARS DG.LAST-DIRECTORY-SELECTED DG.DIRECTORY-WAS-SELECTED-EVENT DG.WINDOW-OF-LAST-DIRECTORY
                  ))
    (PROG (WAITS)
          (SETQ DG.LAST-DIRECTORY-SELECTED NIL)
          (SETQ WAITS 0)
          (|printout| (DG.PROMPTWINDOW WINDOW)
                T MESSAGE)
      WAIT-AGAIN
          (SETQ WAITS (ADD1 WAITS))
          (AWAIT.EVENT DG.DIRECTORY-WAS-SELECTED-EVENT 1000)
          (COND
             ((OR (NULL DG.LAST-DIRECTORY-SELECTED)
                  (AND (NOT ALLOW-OTHER-WINDOWS)
                       (NEQ DG.WINDOW-OF-LAST-DIRECTORY WINDOW)))
              (COND
                 ((GREATERP WAITS 50)
                  (RETURN NIL))
                 ((ZEROP (IMOD WAITS 10))
                  (FLASHWINDOW (DG.PROMPTWINDOW WINDOW))
                  (PLAYTUNE (LIST (CONS (ITIMES 100 (IQUOTIENT WAITS 10))
                                        10000)
                                  (CONS (ITIMES 100 (ADD1 (IQUOTIENT WAITS 10)))
                                        10000)
                                  (CONS (ITIMES 100 (IQUOTIENT WAITS 10))
                                        10000)))))
              (GO WAIT-AGAIN)))
          (RETURN (LIST DG.LAST-DIRECTORY-SELECTED DG.WINDOW-OF-LAST-DIRECTORY)))))
```


(**DIRGRAPHER**
```
  (LAMBDA ({DEV}<DIR> WINDOW-REGION PATH?)                      (* |edited:| "13-May-86 14:24")
    (DECLARE (GLOBALVARS DG.DEFAULT-DIR WAITINGCURSOR PROMPTWINDOW DG.MIN-WIDTH DG.MIN-HEIGHT
                  DG.PROMPTWINDOWFONT DG.MENU-EDGE))
    (PROG (TREE DG.WINDOW PATH OLD-CURSOR GRAPH)
          (COND
             ((NULL {DEV}<DIR>)
              (SETQ {DEV}<DIR> DG.DEFAULT-DIR)))
          (SETQ OLD-CURSOR (CURSOR WAITINGCURSOR))
          (PRINTOUT PROMPTWINDOW T "DirGrapher : Computing directory structure ")
          (|if| (LISTP {DEV}<DIR>)
              |then| (SETQ TREE {DEV}<DIR>)
                     (SETQ PATH (OR PATH? (LIST (CAR TREE))))
             |else| (SETQ PATH (DG.CREATE-PATH-FROM-FILE-NAME {DEV}<DIR>))
                    (SETQ TREE (DG.CREATE-DIRECTORY-TREE (COPY PATH))))
          (SETQ GRAPH (DG.CREATE-GRAPH-FROM-TREE TREE))
          (PRINTOUT PROMPTWINDOW T "DirGrapher : Done.")
          (CURSOR OLD-CURSOR)
          (|if| WINDOW-REGION
              |then| (SETQ DG.WINDOW (CREATEW WINDOW-REGION {DEV}<DIR>))
             |else| (SETQ DG.WINDOW (CREATEW (DG.GET-INITIAL-REGION GRAPH)
                                           {DEV}<DIR>))
                    (WINDOWPROP DG.WINDOW 'MINSIZE (CONS DG.MIN-WIDTH DG.MIN-HEIGHT)))
          (WINDOWPROP DG.WINDOW 'DG.PATH-TO-ROOT PATH)
          (DG.DISPLAY-DIRECTORY-TREE GRAPH DG.WINDOW)
          (WINDOWPROP DG.WINDOW 'DG.TREE TREE)
          (WINDOWADDPROP DG.WINDOW 'REPAINTFN 'DG.REDISPLAYFN)
          (WINDOWPROP DG.WINDOW 'ICONFN 'DG.CREATEICONW)
          (WINDOWPROP DG.WINDOW 'EXPANDFN '(DG.SHADE-CURRENT-DIRECTORY REDISPLAYGRAPH))
          (WINDOWPROP DG.WINDOW 'DG.ICONTITLE (DG.CREATE-ICON-TITLE PATH))
          (WINDOWPROP DG.WINDOW 'DG.PROCESS NIL)
          (WINDOWPROP DG.WINDOW 'DG.FILE-SELECTION-ENDED-EVENT (CREATE.EVENT "DGEVENT"))
          (WINDOWPROP DG.WINDOW 'DG.LOCK (CREATE.MONITORLOCK "DG.LOCK"))
          (GETPROMPTWINDOW DG.WINDOW 3 DG.PROMPTWINDOWFONT)
          (ATTACHMENU (DG.CREATE-MENU DG.WINDOW)
                  DG.WINDOW DG.MENU-EDGE 'JUSTIFY)
          (ATTACHMENU (DG.CREATE-QUIT-MENU DG.WINDOW)
                  DG.WINDOW
                  'BOTTOM
                  'JUSTIFY)
```

```
          (RETURN DG.WINDOW))))
)
(RPAQQ DG.ICON
```



```
)
(RPAQQ DG.MASK
```



```
)
(RPAQQ DG.FILL-WINDOW-TEXTURE
                              )
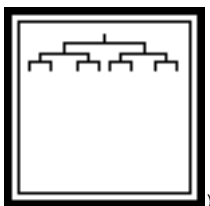
(RPAQQ DG.WHITEBM
                      )

(DG.ADVISE-GENERATE-FILE)

(SETQ DG.BACKGROUND-DIRECTORIES (UNION DG.BACKGROUND-DIRECTORIES (UNION DIRECTORIES
                                                  (CONS '{FLOPPY}
                                                     (FOR D IN (VOLUMES)
                                                         WHEN (LISPDIRECTORYP D)
                                                         COLLECT (PACK* '{DSK}< D
                                                                     '>))))))

(DG.ADD-ITEM-TO-BACKGROUND-MENU
 '|DirGrapher|
 '(DIRGRAPHER)
 "Will initiate dirgrapher process on the current directory"
 (CONS 'SUBITEMS (CONS '("DG Windows to top" (|for| W |in| (OPENWINDOWS)
                                              |when| (OR (WINDOWPROP W 'DG.PATH-TO-ROOT)
                                                         (AND (WINDOWPROP W 'ICONFOR)
                                                              (WINDOWPROP (WINDOWPROP W 'ICONFOR)
                                                                  'DG.PATH-TO-ROOT)))
                                              |do| (TOTOPW W)))
                         (FOR D IN DG.BACKGROUND-DIRECTORIES COLLECT (LIST D (LIST 'DIRGRAPHER (KWOTE D))
                                                       (CONCAT "WILL APPLY DIRGRAPHER ON " D)
                                                       )))))

;;; (declare\: dontcopy (prop makefile-environment dirgrapher))

(PUTPROPS DIRGRAPHER COPYRIGHT ("Shaul Markovitch" 1985 1986 1900 1987 1988))
```

FUNCTION INDEX

VARIABLE INDEX