File created: 11-Apr-2024 08:27:34 **{WMEDLEY}<lispusers>DINFO.;13**

edit by: rmk

changes to: (FNS DINFO.OPENTEXTSTREAM)

previous date: 10-Mar-2024 15:38:36 {WMEDLEY}<lispusers>DINFO.;12

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ DINFOCOMS
       ((FILES TEDIT GRAPHER)
        (DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS DINFOGRAPH DINFONODE)
               (FUNCTIONS DINFOGRAPHPROP))
        (INITRECORDS DINFOGRAPH)
        (FNS                                                    ; Primary functions
             DINFO DINFO.UPDATE DINFOGRAPH DINFO.SPECIAL.UPDATE DINFO.READ.GRAPH DINFO.WRITE.GRAPH
             DINFO.SELECT.GRAPH DINFO.DEFAULT.MENU DINFO.FIND DINFO.LOOKUP)
        (FNS                                                    ; Koto compatability
             DINFO.READ.KOTO.GRAPH)
        (FNS                                                    ; Window functions
             DINFO.SETUP.WINDOW DINFO.CLOSEFN DINFO.SHRINKFN DINFO.EXPANDFN DINFO.ICONFN)
        (FNS                                                    ; FreeMenu functions
             DINFO.ADD.FMENU DINFO.CREATE.FMENU DINFO.FMW.CLOSEFN DINFO.FMENU.HANDLER DINFO.UPDATE.FMENU
             DINFO.TOGGLE.MENU DINFO.TOGGLE.GRAPH DINFO.TOGGLE.HISTORY DINFO.TOGGLE.TEXT)
        (FNS                                                    ; Other menu functions
             DINFO.UPDATE.MENU.DISPLAY DINFO.UPDATE.FROM.MENU DINFO.UPDATE.HISTORY DINFO.HISTORIC.UPDATE)
        (FNS                                                    ; Interface to GRAPHER
             DINFO.UPDATE.GRAPH.DISPLAY DINFO.UPDATE.FROM.GRAPH DINFO.GET.GRAPH.WINDOW DINFO.CREATE.GRAPH.WINDOW
             DINFO.SHOWGRAPH DINFO.INVERT.NODE DINFO.LAYOUTGRAPH)
        (FNS                                                    ; Interface to TEdit
             DINFO.UPDATE.TEXT.DISPLAY DINFO.TITLEMENUFN DINFO.OPENTEXTSTREAM DINFO.SHOWSEL DINFO.GET.FILENAME)
        (ADDVARS (BackgroundMenuCommands (DInfo (DINFO.SELECT.GRAPH)
                                                "Open a DInfo window for browsing documentation.")))
        (VARS (BackgroundMenu))
        (INITVARS (DINFO.GRAPHS)
               (DINFOMODES '(TEXT GRAPH))
               (DINFO.HISTORY.LENGTH 20)
               (\DINFO.MAX.MENU.LEN 10))
        (GLOBALVARS DINFO.GRAPH.FILES DINFOMODES DINFO.HISTORY.LENGTH \DINFO.MAX.MENU.LEN)
        (PROP (FILETYPE)
              DINFO)
        (DECLARE%: DONTCOPY (TEMPLATES DINFOGRAPHPROP))))

(FILESLOAD TEDIT GRAPHER)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(DATATYPE DINFOGRAPH
       (NAME NODELST TOPNODEID CURRENTNODE USERDATA TEXTPROPS FREEMENUITEMS LOOKUPFN MENUFN DEFAULTHOST
             DEFAULTDEVICE DEFAULTDIR MONITORLOCK DINFO.MENU WINDOW MENUFONT FMENU.WINDOW GRAPH.WINDOW
             HISTORY.MENU.WINDOW SUBNODE.MENU.WINDOW LAST.TEXT LAST.INVERTED.NODE LAST.GRAPH.LOCATION
             HISTORY.ITEMS FIND.STRING LOOKUP.STRING)
       (SYSTEM))

(RECORD DINFONODE (ID LABEL FILE FROMBYTE TOBYTE PARENT CHILDREN NEXTNODE PREVIOUSNODE USERDATA)
       (SYSTEM))
)

(/DECLAREDATATYPE 'DINFOGRAPH
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER)
       ;; ---field descriptor list elided by lister---
       '52)


(DEFMACRO DINFOGRAPHPROP (GRAPH PROP &OPTIONAL (NEW-VALUE NIL NEW-VALUE-SUPPLIED))
   [LET [(REAL-FIELD (AND (LISTP PROP)
                          (EQ (CAR PROP)
                              'QUOTE)
                          (FMEMB (CADR PROP)
                                 (RECORDFIELDNAMES 'DINFOGRAPH T))
                          (CADR PROP]
        (IF NEW-VALUE-SUPPLIED
            THEN [IF REAL-FIELD
                     THEN `(REPLACE (DINFOGRAPH ,REAL-FIELD) OF ,GRAPH WITH ,NEW-VALUE)
                   ELSE `(LET* ((SI::$GRAPH$ ,GRAPH)
                                (SI::$USERDATA$ (FETCH (DINFOGRAPH USERDATA) OF SI::$GRAPH$))
                                (SI::$PROP$ ,PROP)
                                (SI::$NEW-VALUE$ ,NEW-VALUE))
```

```
                                (IF (LISTP SI::$USERDATA$)
                                    THEN (LISTPUT SI::$USERDATA$ SI::$PROP$ SI::$NEW-VALUE$)
                                  ELSE (REPLACE (DINFOGRAPH USERDATA) OF SI::$GRAPH$ WITH (LIST SI::$PROP$
                                                                                                SI::$NEW-VALUE$
                                                                                                ))
                                                SI::$NEW-VALUE$]
          ELSE (IF REAL-FIELD
                   THEN `(FETCH (DINFOGRAPH ,REAL-FIELD) OF ,GRAPH)
                 ELSE `(LISTGET (FETCH (DINFOGRAPH USERDATA) OF ,GRAPH)
                                ,PROP])
)

(/DECLAREDATATYPE 'DINFOGRAPH
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                 POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                 POINTER)
       ;; ---field descriptor list elided by lister---
       '52)

(DEFINEQ
```

(DINFO

```
  [LAMBDA (GRAPH.OR.FILE WINDOW.OR.REGION SETUP.ONLY? NO.FREEMENU?)
                                                                  (* drc%: "25-Jan-86 18:23")
                                                                  (* Starts a DInfo browser.)
    (RESETLST
        (LET ((W (OR (WINDOWP WINDOW.OR.REGION)
                     (AND (REGIONP WINDOW.OR.REGION)
                          (CREATEW WINDOW.OR.REGION "DInfo" NIL T))
                     (AND (type? DINFOGRAPH GRAPH.OR.FILE)
                          (WINDOWP (fetch (DINFOGRAPH WINDOW) of GRAPH.OR.FILE)))
                     (CREATEW NIL "DInfo")))
              GRAPH MONITORLOCK)
             (OPENW W)
             [SETQ GRAPH (if (type? DINFOGRAPH GRAPH.OR.FILE)
                             then GRAPH.OR.FILE
                           else (RESETFORM (TTYDISPLAYSTREAM (GETPROMPTWINDOW W))
                                           (DINFO.READ.GRAPH GRAPH.OR.FILE]
             [SETQ MONITORLOCK (OR (fetch (DINFOGRAPH MONITORLOCK) of GRAPH)
                                   (replace (DINFOGRAPH MONITORLOCK) of GRAPH with (CREATE.MONITORLOCK "DInfo"]
             (RESETSAVE NIL (LIST 'RELEASE.MONITORLOCK MONITORLOCK))
             (OBTAIN.MONITORLOCK MONITORLOCK)
             (DINFO.SETUP.WINDOW GRAPH W NO.FREEMENU?)
             (OR SETUP.ONLY? (DINFO.UPDATE GRAPH NIL NIL T))
             GRAPH))])
```

(DINFO.UPDATE

```
  [LAMBDA (GRAPH NEW.NODE SEL FORCE?)                             (* jow "20-May-86 15:14")

          (* * Called to visit a NEW.NODE in GRAPH, or to just make sure that the display of GRAPH is current.)

    (LET ([NODE (OR NEW.NODE (fetch (DINFOGRAPH CURRENTNODE) of GRAPH)
                    (FASSOC (fetch (DINFOGRAPH TOPNODEID) of GRAPH)
                            (fetch (DINFOGRAPH NODELST) of GRAPH]
          (PREVIOUS.NODE (fetch (DINFOGRAPH CURRENTNODE) of GRAPH))
          (WINDOW (fetch (DINFOGRAPH WINDOW) of GRAPH)))
         (OPENW WINDOW)
         (WINDOWPROP WINDOW 'DINFOGRAPH GRAPH)
         (OR (FMEMB NODE (fetch (DINFOGRAPH NODELST) of GRAPH))
             (ERROR NODE "NOT IN NODELST"))
         (LET ((FMENU.WINDOW (fetch (DINFOGRAPH FMENU.WINDOW) of GRAPH))
               (MONITORLOCK (fetch (DINFOGRAPH MONITORLOCK) of GRAPH)))
              (RESETLST
                  (RESETSAVE NIL (LIST 'RELEASE.MONITORLOCK MONITORLOCK))
                  [if (NOT (OBTAIN.MONITORLOCK MONITORLOCK T))
                      then                                        (* somebody else is messing with this graph.)
                           (FLASHWINDOW WINDOW)
                           (PROMPTPRINT "DInfo is busy")
                    elseif (NULL FMENU.WINDOW)
                      then (replace (DINFOGRAPH CURRENTNODE) of GRAPH with NODE)
                                                                  (* FreeMenu turned off, so just display text)
                           (DINFO.UPDATE.TEXT.DISPLAY GRAPH NODE SEL)
                      else                                        (* We've got a FreeMenu, so update away!)
                           (DINFO.UPDATE.FMENU GRAPH NODE)
                           (LET ((STATUS (FM.GETSTATE FMENU.WINDOW)))
                                (replace (DINFOGRAPH CURRENTNODE) of GRAPH with NODE)
                                (AND (LISTGET STATUS 'GRAPH)
                                     (DINFO.UPDATE.GRAPH.DISPLAY GRAPH NODE FORCE?))
                                (AND (LISTGET STATUS 'MENU)
                                     (DINFO.UPDATE.MENU.DISPLAY GRAPH NODE))
                                (AND (LISTGET STATUS 'TEXT)
                                     (DINFO.UPDATE.TEXT.DISPLAY GRAPH NODE SEL))
                                (DINFO.UPDATE.HISTORY GRAPH NODE SEL (LISTGET STATUS 'HISTORY])
              (CLEARW (GETPROMPTWINDOW WINDOW])
```

(DINFOGRAPH

```
[LAMBDA (X)                                                    (* drc%: " 8-Jan-86 11:12")
  (if (type? DINFOGRAPH X)
      then X
    elseif (AND (WINDOWP X)
                (WINDOWPROP X 'DINFOGRAPH))
    elseif (AND (WINDOWP X)
                (WINDOWPROP X 'MAINWINDOW))
      then (WINDOWPROP (WINDOWPROP X 'MAINWINDOW)
                  'DINFOGRAPH])
```

(DINFO.SPECIAL.UPDATE

```
[LAMBDA (TYPE GRAPH)                                           (* drc%: "25-Jan-86 18:26")

        (* * Do a TYPE update of Graph, where TYPE is one of Top, Parent, Previous or Next.)

  (LET* [(DINFOW (fetch (DINFOGRAPH WINDOW) of GRAPH))
         (CURRENT.NODE (fetch (DINFOGRAPH CURRENTNODE) of GRAPH))
         (NEW.NODE (FASSOC (SELECTQ TYPE
                                    (Top (fetch (DINFOGRAPH TOPNODEID) of GRAPH))
                                    (Parent (fetch (DINFONODE PARENT) of CURRENT.NODE))
                                    (Next (fetch (DINFONODE NEXTNODE) of CURRENT.NODE))
                                    (Previous (fetch (DINFONODE PREVIOUSNODE) of CURRENT.NODE))
                                    NIL)
                           (fetch (DINFOGRAPH NODELST) of GRAPH]
        (if (OBTAIN.MONITORLOCK (fetch (DINFOGRAPH MONITORLOCK) of GRAPH)
                  T)
            then (if NEW.NODE
                     then (PROCESSPROP (THIS.PROCESS)
                                       'NAME
                                       (CONCAT "DInfo " TYPE))
                          (DINFO.UPDATE GRAPH NEW.NODE)
                 else                                          (* TYPE of Top! or Node! will sound silly here, but should never
                      happen.)
                      (printout (GETPROMPTWINDOW (fetch (DINFOGRAPH WINDOW) of GRAPH))
                                T "This node has no " TYPE))
          else (FLASHWINDOW DINFOW)
               (PROMPTPRINT "DInfo is busy"])
```

(DINFO.READ.GRAPH

```
[LAMBDA (FILE QUIETFLG)                                        (* drc%: "25-Jan-86 18:17")
                                                               (* Reads a file written by DINFO.WRITE.GRAPH.
                                                               Returns the DInfo graph stored on FILE.)
  (OR QUIETFLG (printout T T "Reading " (FILENAMEFIELD FILE 'NAME)
                     " graph..."))
  (LET* ((FULLFILENAME (INFILEP FILE))
         [DATA (CDR (READFILE (OR FULLFILENAME (ERROR "FILE NOT FOUND" FILE]
         (GRAPH (create DINFOGRAPH)))                          (* fields stored on file)
        (replace (DINFOGRAPH TOPNODEID) of GRAPH with (LISTGET DATA 'TOPNODEID))
        (replace (DINFOGRAPH TEXTPROPS) of GRAPH with (LISTGET DATA 'TEXTPROPS))
        (replace (DINFOGRAPH LOOKUPFN) of GRAPH with (LISTGET DATA 'LOOKUPFN))
        (replace (DINFOGRAPH MENUFN) of GRAPH with (LISTGET DATA 'MENUFN))
        (replace (DINFOGRAPH FREEMENUITEMS) of GRAPH with (LISTGET DATA 'FREEMENUITEMS))
        (replace (DINFOGRAPH NODELST) of GRAPH with (LISTGET DATA 'NODELST))
        (replace (DINFOGRAPH USERDATA) of GRAPH with (LISTGET DATA 'USERDATA))
                                                               (* fields filled in at read time)
        (replace (DINFOGRAPH NAME) of GRAPH with (FILENAMEFIELD FULLFILENAME 'NAME))
        (replace (DINFOGRAPH DEFAULTHOST) of GRAPH with (FILENAMEFIELD FULLFILENAME 'HOST))
        (replace (DINFOGRAPH DEFAULTDEVICE) of GRAPH with (FILENAMEFIELD FULLFILENAME 'DEVICE))
        (replace (DINFOGRAPH DEFAULTDIR) of GRAPH with (FILENAMEFIELD FULLFILENAME 'DIRECTORY))
        (OR QUIETFLG (printout T "OK."))
        GRAPH])
```

(DINFO.WRITE.GRAPH

```
[LAMBDA (GRAPH FILE)                                           (* drc%: "25-Jan-86 18:16")

        (* Writes a DInfo graph to a file for reading by DINFO.READ.GRAPH.
        Returns the full file name of the file.)
                                                               (* dump it out as a props list)
  (WRITEFILE (LIST 'TOPNODEID (fetch (DINFOGRAPH TOPNODEID) of GRAPH)
                   'TEXTPROPS
                   (fetch (DINFOGRAPH TEXTPROPS) of GRAPH)
                   'LOOKUPFN
                   (fetch (DINFOGRAPH LOOKUPFN) of GRAPH)
                   'MENUFN
                   (fetch (DINFOGRAPH MENUFN) of GRAPH)
                   'FREEMENUITEMS
                   (fetch (DINFOGRAPH FREEMENUITEMS) of GRAPH)
                   'NODELST
                   (fetch (DINFOGRAPH NODELST) of GRAPH)
                   'USERDATA
                   (fetch (DINFOGRAPH USERDATA) of GRAPH))
         FILE])
```

(DINFO.SELECT.GRAPH

```
[LAMBDA NIL                                                    (* drc%: "24-Jan-86 13:25")

        (* * This is called when DInfo is selected from the Background Menu.)

   (DECLARE (GLOBALVARS DINFO.GRAPHS))
   (ALLOW.BUTTON.EVENTS)
   (RESETFORM (TTY.PROCESS (THIS.PROCESS))
          (LET [(GRAPH (if (NULL DINFO.GRAPHS)
                           then (PROMPTPRINT "No Graphs installed -- load HelpSys or DInfoEdit")
                       elseif (NULL (CDR DINFO.GRAPHS))
                         then (EVAL (CADAR DINFO.GRAPHS))
                       else (MENU (create MENU
                                          CENTERFLG _ T
                                          TITLE _ "Select Graph"
                                          ITEMS _ DINFO.GRAPHS]
               (AND GRAPH (DINFO GRAPH])
```

(DINFO.DEFAULT.MENU

```
[LAMBDA (GRAPH)                                                (* jow "15-Jul-86 17:36")

        (* * This is the default MENUFN for DInfo graphs.)

   (LET ((DINFOW (fetch (DINFOGRAPH WINDOW) of GRAPH)))
        (CLEARW (GETPROMPTWINDOW DINFOW))
        (LET [(TYPE (MENU (OR (fetch (DINFOGRAPH DINFO.MENU) of GRAPH)
                              (replace (DINFOGRAPH DINFO.MENU) of GRAPH
                                 with (create MENU
                                              ITEMS _ '(("Top" 'Top "Visit the top node in the graph")
                                                        ("Parent" 'Parent "Visit the parent of the current node")
                                                        ("Previous" 'Previous "Visit the node before this node")
                                                        ("Next " 'Next "Visit the node following this node")
                                                        ("Find" 'Find "Search the text of this node")
                                                        ("Lookup" 'Lookup "Lookup a new term in this graph")
                                                        ("Expanded Menu" 'FreeMenu "Add an expanded options
                                                              menu."))
                                              CENTERFLG _ T
                                              MENUFONT _ (FONTCREATE 'HELVETICA 10 'BOLD]
            (if TYPE
                then (PROCESSPROP (THIS.PROCESS)
                                  'NAME
                                  (CONCAT "DInfo " TYPE))
                     (SELECTQ TYPE
                          ((Top Parent Previous Next)
                              (DINFO.SPECIAL.UPDATE TYPE GRAPH))
                          (Find (DINFO.FIND GRAPH))
                          (Lookup (DINFO.LOOKUP GRAPH '(LEFT)))
                          (FreeMenu (DINFO.ADD.FMENU GRAPH)
                                    (DINFO.UPDATE GRAPH))
                          NIL])
```

(DINFO.FIND

```
[LAMBDA (GRAPH BUTTONS)                                        ; Edited 21-Jan-2022 23:15 by rmk
                                                               (* drc%: "25-Jan-86 18:23")
   (LET ((DINFOW (fetch (DINFOGRAPH WINDOW) of GRAPH)))
        (if (NOT (OBTAIN.MONITORLOCK (fetch (DINFOGRAPH MONITORLOCK) of GRAPH)
                                     T))
            then (FLASHWINDOW DINFOW)
                 (PROMPTPRINT "DInfo is busy")
          else (RESETFORM (TTYDISPLAYSTREAM (GETPROMPTWINDOW DINFOW))
                          (TERPRI T)
                          (LET ([STRING (if (AND (FMEMB 'MIDDLE BUTTONS)
                                                 (fetch (DINFOGRAPH FIND.STRING) of GRAPH))
                                            else (TTYINPROMPTFORWORD "Find: " (fetch (DINFOGRAPH FIND.STRING)
                                                                                 of GRAPH)
                                                                      NIL NIL NIL 'TTY (CONSTANT (CHARCODE (EOL ESCAPE LF]
                                (TEXTSTREAM (WINDOWPROP DINFOW 'TEXTSTREAM))
                                PAIR)
                               (replace (DINFOGRAPH FIND.STRING) of GRAPH with STRING)
                               (if STRING
                                   then (PRINTOUT T " Searching...")
                                        (if (SETQ PAIR (TEDIT.FIND TEXTSTREAM STRING NIL NIL T))
                                            then (printout T "OK.")
                                                 (TEDIT.NORMALIZECARET TEXTSTREAM (TEDIT.SHOWSEL TEXTSTREAM T
                                                                                                 (TEDIT.SETSEL TEXTSTREAM
                                                                                                        (CAR PAIR)
                                                                                                        (NCHARS STRING)
                                                                                                        'RIGHT T)))
                                          else (printout T "not found.")
                                               (TEDIT.NORMALIZECARET TEXTSTREAM (TEDIT.SETSEL TEXTSTREAM 0 0])
```

(DINFO.LOOKUP

```
  [LAMBDA (GRAPH BUTTONS)                                    ; Edited 21-Jan-2022 23:15 by rmk
                                                             (* drc%: "25-Jan-86 18:22")
    (LET ((DINFOW (fetch (DINFOGRAPH WINDOW) of GRAPH)))
         (if (OBTAIN.MONITORLOCK (fetch (DINFOGRAPH MONITORLOCK) of GRAPH)
                   T)
             then [RESETFORM (TTYDISPLAYSTREAM (GETPROMPTWINDOW DINFOW))
                         (LET ((LOOKUPFN (fetch (DINFOGRAPH LOOKUPFN) of GRAPH)))
                              (if LOOKUPFN
                                  then (CLEARW T)
                                       (LET* [(OLD.STRING (fetch (DINFOGRAPH LOOKUP.STRING) of GRAPH))
                                              (STRING (if (AND OLD.STRING (FMEMB 'MIDDLE BUTTONS))
                                                          then OLD.STRING
                                                        else (TTYINPROMPTFORWORD "Lookup: " OLD.STRING NIL NIL NIL
                                                                                 'TTY
                                                                                 (CONSTANT (CHARCODE (EOL ESCAPE LF]
                                                     (replace (DINFOGRAPH LOOKUP.STRING) of GRAPH with STRING)
                                                     (AND STRING (APPLY* LOOKUPFN STRING GRAPH)))
                                  else (PRINTOUT T T "The " (fetch (DINFOGRAPH NAME) of GRAPH)
                                               " graph has no LOOKUPFN."]
              else (FLASHWINDOW DINFOW)
                   (PROMPTPRINT "DInfo is busy"])

)

(DEFINEQ
```

(DINFO.READ.KOTO.GRAPH

```
  [LAMBDA (FILE QUIETFLG)                                    (* drc%: " 4-Feb-86 11:27")

          (* Reads a file written by Koto DINFO.WRITE.GRAPH and returns a Lute DINFOGRAPH.
          Thus, (DINFO.WRITE.GRAPH (DINFO.READ.KOTO.GRAPH <file1>) <file2>) will convert the Koto format graph in <file1>
          to a Lute format graph in <file2>.)

    (OR QUIETFLG (printout T T "Reading " (FILENAMEFIELD FILE 'NAME)
                       " graph..."))
    (LET* ((FULLFILENAME (INFILEP FILE))
           [DATA (CDR (READFILE (OR FULLFILENAME (ERROR "FILE NOT FOUND" FILE]
           (GRAPH (create DINFOGRAPH)))                      (* in Koto we just wrote out the DINFOGRAPH record)
          (for FIELD in DATA as N from 1 to 11 do            (* fields stored on file)
                                                  (SELECTQ N
                                                       ((1 4 5 6 10 11))
                                                       (2 (replace (DINFOGRAPH NODELST) of GRAPH with FIELD))
                                                       (3 (replace (DINFOGRAPH TOPNODEID) of GRAPH with FIELD))
                                                       (7 (replace (DINFOGRAPH TEXTPROPS) of GRAPH with FIELD))
                                                       (8 (replace (DINFOGRAPH LOOKUPFN) of GRAPH with FIELD))
                                                       (9 (replace (DINFOGRAPH MENUFN) of GRAPH with FIELD))
                                                       (SHOULDNT)))(* fields filled in at read time)
          (replace (DINFOGRAPH NAME) of GRAPH with (FILENAMEFIELD FULLFILENAME 'NAME))
          (replace (DINFOGRAPH DEFAULTHOST) of GRAPH with (FILENAMEFIELD FULLFILENAME 'HOST))
          (replace (DINFOGRAPH DEFAULTDEVICE) of GRAPH with (FILENAMEFIELD FULLFILENAME 'DEVICE))
          (replace (DINFOGRAPH DEFAULTDIR) of GRAPH with (FILENAMEFIELD FULLFILENAME 'DIRECTORY))
          (OR QUIETFLG (printout T "OK."))
          GRAPH])

)

(DEFINEQ
```

(DINFO.SETUP.WINDOW

```
  [LAMBDA (GRAPH WINDOW NO.FREEMENU?)                        (* jow "10-Jun-86 15:29")
    (replace (DINFOGRAPH WINDOW) of GRAPH with WINDOW)
    (WINDOWPROP WINDOW 'DINFOGRAPH GRAPH)
    (DETACHALLWINDOWS WINDOW)
    (if (NOT NO.FREEMENU?)
        then (DINFO.ADD.FMENU GRAPH))
    (DINFO.UPDATE.TEXT.DISPLAY GRAPH NIL NIL T)
    (WINDOWPROP WINDOW 'TITLE (CONCAT (fetch (DINFOGRAPH NAME) of GRAPH)
                                      " DInfo"))
    (WINDOWADDPROP WINDOW 'CLOSEFN 'DINFO.CLOSEFN)
    (WINDOWADDPROP WINDOW 'SHRINKFN 'DINFO.SHRINKFN)
    (WINDOWADDPROP WINDOW 'EXPANDFN 'DINFO.EXPANDFN])
```

(DINFO.CLOSEFN

```
  [LAMBDA (W)                                                (* drc%: "25-Jan-86 18:26")
    (LET [(GRAPH (WINDOWPROP W 'DINFOGRAPH]
         (if (type? DINFOGRAPH GRAPH)
             then (CLOSEW (fetch (DINFOGRAPH GRAPH.WINDOW) of GRAPH))
                                                             (* remove circularity...)
                  (WINDOWPROP W 'DINFOGRAPH NIL])
```

(DINFO.SHRINKFN

```
  [LAMBDA (W)                                                (* drc%: "25-Jan-86 18:26")
    (CLOSEW (fetch (DINFOGRAPH GRAPH.WINDOW) of (DINFOGRAPH W])
```

(DINFO.EXPANDFN

```
  [LAMBDA (W)                                                    (* jow "15-Jul-86 17:00")
    (LET* ((GRAPH (DINFOGRAPH W))
           (FMENU (fetch (DINFOGRAPH FMENU.WINDOW) of GRAPH)))
          (if (AND FMENU (LISTGET (FM.GETSTATE FMENU)
                                  'GRAPH))
              then (LET ((GRAPHW (fetch (DINFOGRAPH GRAPH.WINDOW) of GRAPH)))
                        (OPENW GRAPHW)
                        (TOTOPW W)
                        (WINDOWPROP GRAPHW 'DINFOGRAPH GRAPH])
```

(DINFO.ICONFN

```
  [LAMBDA (W)                                                    (* drc%: "25-Jan-86 16:33")
    (OR (WINDOWPROP WINDOW 'ICON)
        (WINDOWPROP WINDOW 'ICON (TITLEDICONW TEDIT.TITLED.ICON.TEMPLATE (WINDOWPROP WINDOW 'TITLE)
                                              TEDIT.ICON.FONT NIL T))
        (WINDOWPROP WINDOW 'ICON])

)

(DEFINEQ
```

(DINFO.ADD.FMENU

```
  [LAMBDA (GRAPH)                                                (* jow "20-May-86 15:41")

          (* * Add a DInfo FreeMenu to WINDOW. then update the FreeMenu's display.)

    (LET ((WINDOW (fetch (DINFOGRAPH WINDOW) of GRAPH))
          (FM.WINDOW (fetch (DINFOGRAPH FMENU.WINDOW) of GRAPH)))
         (if [AND (WINDOWP FM.WINDOW)
                  (FMEMB FM.WINDOW (WINDOWPROP WINDOW 'ATTACHEDWINDOWS]
             then (OPENW FM.WINDOW)
          else (REMOVEPROMPTWINDOW WINDOW)
               (SETQ FM.WINDOW (OR (WINDOWP FM.WINDOW)
                                   (DINFO.CREATE.FMENU GRAPH)))
               (replace (DINFOGRAPH FMENU.WINDOW) of GRAPH with FM.WINDOW)
               (ATTACHWINDOW FM.WINDOW WINDOW)
               (WINDOWPROP FM.WINDOW 'FM.PROMPTWINDOW (GETPROMPTWINDOW WINDOW))
               (WINDOWDELPROP FM.WINDOW 'PASSTOMAINCOMS 'CLOSEW)
               (WINDOWADDPROP FM.WINDOW 'CLOSEFN 'DINFO.FMW.CLOSEFN T)
               (DINFO.UPDATE.FMENU GRAPH])
```

(DINFO.CREATE.FMENU

```
  [LAMBDA (GRAPH)                                                ; Edited  9-Mar-2024 14:20 by rmk
                                                                 ; Edited 25-Oct-2021 23:23 by rmk:
                                                                 (* jow "15-Jul-86 17:39")
```

;;; Makes a DInfo FreeMenu for GRAPH

;; RMK: Added MINSIZE and MAXSIZE so that the menu doesn't get distorted during reshaping

```
    (LET* [(ADD.ITEMS (fetch (DINFOGRAPH FREEMENUITEMS) of GRAPH))
           (FONT (OR (FONTP (fetch (DINFOGRAPH MENUFONT) of GRAPH))
                     MENUFONT))
           [FM (FREEMENU `((PROPS FONT ,FONT)
                           ((LABEL Node%: TYPE DISPLAY FONT (HELVETICA 10))
                            (ID NODE LABEL "" TYPE DISPLAY))
                           ((LABEL Top! SELECTEDFN DINFO.FMENU.HANDLER FONT (HELVETICA 10 BOLD)
                                   MESSAGE "Visit the top node")
                            (ID TOP LABEL "" TYPE DISPLAY))
                           ((LABEL Parent! SELECTEDFN DINFO.FMENU.HANDLER FONT (HELVETICA 10 BOLD)
                                   MESSAGE "Visit the parent of the current node")
                            (ID PARENT LABEL "" TYPE DISPLAY))
                           ((LABEL Previous! SELECTEDFN DINFO.FMENU.HANDLER FONT (HELVETICA 10 BOLD)
                                   MESSAGE "Visit the node previous to the current node")
                            (ID PREVIOUS LABEL "" TYPE DISPLAY))
                           ((LABEL Next! SELECTEDFN DINFO.FMENU.HANDLER FONT (HELVETICA 10 BOLD)
                                   MESSAGE "Visit the node after the current node")
                            (ID NEXT LABEL "" TYPE DISPLAY))
                           ((LABEL Display%: TYPE DISPLAY FONT (HELVETICA 10))
                            (LABEL Graph ID GRAPH INITSTATE ,(MEMB 'GRAPH DINFOMODES)
                                   TYPE TOGGLE SELECTEDFN DINFO.TOGGLE.GRAPH FONT (HELVETICA 10 BOLD)
                                   MESSAGE "Toggle display of the graph")
                            (LABEL Menu ID MENU INITSTATE ,(MEMB 'MENU DINFOMODES)
                                   TYPE TOGGLE SELECTEDFN DINFO.TOGGLE.MENU FONT (HELVETICA 10 BOLD)
                                   MESSAGE "Toggle display of the subnode menu")
                            (LABEL Text ID TEXT INITSTATE ,(MEMB 'TEXT DINFOMODES)
                                   TYPE TOGGLE SELECTEDFN DINFO.TOGGLE.TEXT FONT (HELVETICA 10 BOLD)
                                   MESSAGE "Toggle display of the text of the current node")
                            (LABEL History ID HISTORY INITSTATE ,(MEMB 'HISTORY DINFOMODES)
                                   TYPE TOGGLE FONT (HELVETICA 10 BOLD)
                                   SELECTEDFN DINFO.TOGGLE.HISTORY MESSAGE "Toggle the display of the History
                                   Menu"))
                          ,(APPEND '((LABEL Find! SELECTEDFN DINFO.FMENU.HANDLER FONT (HELVETICA 10 BOLD)
                                           MESSAGE "Perform a string search in the selected text of the
                                           current node")
```

```
                                    (LABEL Lookup! SELECTEDFN DINFO.FMENU.HANDLER FONT (HELVETICA 10 BOLD)
                                           MESSAGE "Lookup a term in this graph.  LEFT for new term, MIDDLE to
                                           repeat last."))
                                ADD.ITEMS]
          (HEIGHT (FETCH (REGION HEIGHT) OF (WINDOWPROP FM 'REGION]
         (WINDOWPROP FM 'FM.DONTRESHAPE T)
         (WINDOWPROP FM 'MINSIZE (CONS 0 HEIGHT))
         (WINDOWPROP FM 'MAXSIZE (CONS 64000 HEIGHT))
        FM])
```

(DINFO.FMW.CLOSEFN

```
[LAMBDA (W)                                                          (* drc%: "25-Jan-86 18:19")
```

(* * CLOSEFN for a DInfo FreeMenu window.)

```
  (LET* ((DINFOW (WINDOWPROP W 'MAINWINDOW))
         (GRAPH (DINFOGRAPH DINFOW)))
        (if GRAPH
            then (DETACHWINDOW W)
                 (replace (DINFOGRAPH FMENU.WINDOW) of GRAPH with NIL)
                 (DETACHWINDOW (fetch (DINFOGRAPH SUBNODE.MENU.WINDOW) of GRAPH))
                 (CLOSEW (fetch (DINFOGRAPH SUBNODE.MENU.WINDOW) of GRAPH))
                 (DETACHWINDOW (fetch (DINFOGRAPH HISTORY.MENU.WINDOW) of GRAPH))
                 (CLOSEW (fetch (DINFOGRAPH GRAPH.WINDOW) of GRAPH))
                 (REMOVEPROMPTWINDOW DINFOW])
```

(DINFO.FMENU.HANDLER

```
[LAMBDA (ITEM WINDOW BUTTONS)                                        (* drc%: "16-Jan-86 11:42")
```

(* * Handle a command from the FreeMenu.)

```
  (LET [(GRAPH (WINDOWPROP (WINDOWPROP WINDOW 'MAINWINDOW)
                           'DINFOGRAPH))
        (TYPE (MKATOM (SUBSTRING (FM.ITEMPROP ITEM 'LABEL)
                                 1 -2]
       (SELECTQ TYPE
           ((Top Parent Previous Next)
                (DINFO.SPECIAL.UPDATE TYPE GRAPH))
           (Find (DINFO.FIND GRAPH BUTTONS))
           (Lookup (DINFO.LOOKUP GRAPH BUTTONS))
           (SHOULDNT])
```

(DINFO.UPDATE.FMENU

```
[LAMBDA (GRAPH NEW.NODE)                                             (* jow "20-May-86 15:13")
```

(* * Update the display of GRAPH's FreeMenu. If NEW.NODE is not specified, use Top node of GRAPH, and change Top node title.)

```
  (LET* [(W (fetch (DINFOGRAPH FMENU.WINDOW) of GRAPH))
         (NODELST (fetch (DINFOGRAPH NODELST) of GRAPH))
         (NODE (OR NEW.NODE (FASSOC (fetch (DINFONODE ID) of (fetch (DINFOGRAPH CURRENTNODE) of GRAPH))
                                    NODELST)
                   (FASSOC (fetch (DINFOGRAPH TOPNODEID) of GRAPH)
                           NODELST]
        (OR NEW.NODE (FM.CHANGELABEL (FM.GETITEM 'TOP NIL W)
                                     (fetch (DINFONODE LABEL) of (FASSOC (fetch (DINFOGRAPH TOPNODEID) of GRAPH)
                                                                         (fetch (DINFOGRAPH NODELST) of GRAPH)))
                                     W))
        (FM.CHANGELABEL (FM.GETITEM 'NODE NIL W)
               (fetch (DINFONODE LABEL) of NODE)
               W)
        (FM.CHANGELABEL (FM.GETITEM 'PARENT NIL W)
               (fetch (DINFONODE LABEL) of NODE (FASSOC (fetch (DINFONODE PARENT) of NODE)
                                                        NODELST))
               W)
        (FM.CHANGELABEL (FM.GETITEM 'NEXT NIL W)
               (fetch (DINFONODE LABEL) of NODE (FASSOC (fetch (DINFONODE NEXTNODE) of NODE)
                                                        NODELST))
               W)
        (FM.CHANGELABEL (FM.GETITEM 'PREVIOUS NIL W)
               (fetch (DINFONODE LABEL) of NODE (FASSOC (fetch (DINFONODE PREVIOUSNODE) of NODE)
                                                        NODELST))
               W])
```

(DINFO.TOGGLE.MENU

```
[LAMBDA (ITEM WINDOW)                                                (* jow "10-Jun-86 14:15")
  (LET [(GRAPH (WINDOWPROP (WINDOWPROP WINDOW 'MAINWINDOW)
                           'DINFOGRAPH]
       (if (FM.ITEMPROP ITEM 'STATE)
           then (DINFO.UPDATE.MENU.DISPLAY GRAPH (fetch (DINFOGRAPH CURRENTNODE) of GRAPH))
         else (LET ((SUBNODE.MENU.WINDOW (fetch (DINFOGRAPH SUBNODE.MENU.WINDOW) of GRAPH)))
                   (DETACHWINDOW SUBNODE.MENU.WINDOW)
                   (CLOSEW SUBNODE.MENU.WINDOW])
```

(DINFO.TOGGLE.GRAPH

```
  [LAMBDA (ITEM WINDOW)                                    ; Edited  1-Oct-87 09:56 by drc:
    (LET [(GRAPH (WINDOWPROP (WINDOWPROP WINDOW 'MAINWINDOW)
                             'DINFOGRAPH]
          (if (FM.ITEMPROP ITEM 'STATE)
              then (DINFO.UPDATE.GRAPH.DISPLAY GRAPH (fetch CURRENTNODE of GRAPH)
                           T)
            else (CLOSEW (fetch (DINFOGRAPH GRAPH.WINDOW) of GRAPH)))
          ITEM])
```

(DINFO.TOGGLE.HISTORY

```
  [LAMBDA (ITEM WINDOW)                                    (* jow "10-Jun-86 14:22")
    (LET [(GRAPH (WINDOWPROP (WINDOWPROP WINDOW 'MAINWINDOW)
                             'DINFOGRAPH]
          (if (FM.ITEMPROP ITEM 'STATE)
              then (DINFO.UPDATE.HISTORY GRAPH NIL NIL T)
            else (LET ((HISTORY.MENU.WINDOW (fetch (DINFOGRAPH HISTORY.MENU.WINDOW) of GRAPH)))
                      (DETACHWINDOW HISTORY.MENU.WINDOW)
                      (CLOSEW HISTORY.MENU.WINDOW])
```

(DINFO.TOGGLE.TEXT

```
  [LAMBDA (ITEM WINDOW)                                    (* drc%: "25-Jan-86 18:26")
    (LET* ((DINFOW (WINDOWPROP WINDOW 'MAINWINDOW))
           (GRAPH (WINDOWPROP DINFOW 'DINFOGRAPH))
           (MONITORLOCK (fetch (DINFOGRAPH MONITORLOCK) of GRAPH)))
          (if (NOT (OBTAIN.MONITORLOCK MONITORLOCK T))
              then (FLASHWINDOW DINFOW)
                   (PROMPTPRINT "DInfo is busy")
            elseif (FM.ITEMPROP ITEM 'STATE)
              then (DINFO.UPDATE.TEXT.DISPLAY GRAPH (fetch (DINFOGRAPH CURRENTNODE) of GRAPH))
                   (RELEASE.MONITORLOCK MONITORLOCK)
            else (DINFO.UPDATE.TEXT.DISPLAY GRAPH (fetch (DINFOGRAPH CURRENTNODE) of GRAPH)
                        NIL T)
                 (RELEASE.MONITORLOCK MONITORLOCK])

)

(DEFINEQ
```

(DINFO.UPDATE.MENU.DISPLAY

```
  [LAMBDA (GRAPH NODE)                                     (* drc%: "25-Jan-86 18:20")
    (LET* [(DINFOW (fetch (DINFOGRAPH WINDOW) of GRAPH))
           (WINDOW (fetch (DINFOGRAPH SUBNODE.MENU.WINDOW) of GRAPH))
           [CHILDREN (DREVERSE (for ID in (fetch (DINFONODE CHILDREN) of NODE)
                                  bind (NODELST _ (fetch (DINFOGRAPH NODELST) of GRAPH))
                                  collect (FASSOC ID NODELST]
           (LENGTH (FLENGTH CHILDREN))
           (SCROLLABLE (GREATERP LENGTH \DINFO.MAX.MENU.LEN))
           (MENU (create MENU
                         MENUFONT _ (OR (FONTP (fetch (DINFOGRAPH MENUFONT) of GRAPH))
                                        MENUFONT)
                         ITEMWIDTH _ (WINDOWPROP DINFOW 'WIDTH)
                         CENTERFLG _ T
                         MENUCOLUMNS _ 1
                         MENUOUTLINESIZE _ 0
                         ITEMS _ (for CHILD in CHILDREN collect (LIST (fetch (DINFONODE LABEL) of CHILD)
                                                                      CHILD "Will visit this node if selected."))
                         WHENSELECTEDFN _ (FUNCTION DINFO.UPDATE.FROM.MENU]
          (AND WINDOW (PROGN (DETACHWINDOW WINDOW)
                             (CLOSEW WINDOW)))
          (if CHILDREN
              then (UPDATE/MENU/IMAGE MENU)
                   (SETQ WINDOW (CREATEW (create REGION
                                                 LEFT _ 0
                                                 BOTTOM _ 0
                                                 WIDTH _ (WINDOWPROP DINFOW 'WIDTH)
                                                 HEIGHT _ (HEIGHTIFWINDOW (if SCROLLABLE
                                                                              then (TIMES \DINFO.MAX.MENU.LEN
                                                                                          (fetch (MENU ITEMHEIGHT)
                                                                                             of MENU))
                                                                            else (fetch (MENU IMAGEHEIGHT)
                                                                                    of MENU))
                                                                          T))
                                         "Subnodes" NIL T))
                   (ADDMENU MENU WINDOW (create POSITION
                                                XCOORD _ 0
                                                YCOORD _ (if SCROLLABLE
                                                             then (TIMES (DIFFERENCE \DINFO.MAX.MENU.LEN LENGTH)
                                                                         (fetch (MENU ITEMHEIGHT) of MENU))
                                                           else 0))
                         T)
                   (ATTACHWINDOW WINDOW DINFOW 'BOTTOM)
                   (REDISPLAYW WINDOW)
```

```
                    (replace (DINFOGRAPH SUBNODE.MENU.WINDOW) of GRAPH with WINDOW)
                    (LET [(BITS (fetch (REGION BOTTOM) of (WINDOWPROP WINDOW 'REGION]
                                                                  (* Slide DINFOW up if our new menu is off the screen)
                         (AND (ILESSP BITS 0)
                              (RELMOVEW DINFOW (create POSITION
                                                       XCOORD _ 0
                                                       YCOORD _ (IDIFFERENCE 0 BITS])
```

(DINFO.UPDATE.FROM.MENU

```
  [LAMBDA (ITEM MENU BUTTONS)                                  (* drc%: "12-Dec-85 14:49")
    (DINFO.UPDATE (WINDOWPROP (WINDOWPROP (WFROMMENU MENU)
                                          'MAINWINDOW)
                              'DINFOGRAPH)
          (CADR ITEM])
```

(DINFO.UPDATE.HISTORY

```
  [LAMBDA (GRAPH NODE SEL DISPLAY?)                            (* drc%: "25-Jan-86 18:21")
    (LET* ((DINFOW (fetch (DINFOGRAPH WINDOW) of GRAPH))
           (OLDWINDOW (fetch (DINFOGRAPH HISTORY.MENU.WINDOW) of GRAPH))
           (OLDITEMS (fetch (DINFOGRAPH HISTORY.ITEMS) of GRAPH))
           (NEWITEM (if SEL
                        then (LIST (if (LISTP SEL)
                                       then (CAR SEL)
                                     else SEL)
                                   (LIST (fetch (DINFONODE ID) of NODE)
                                         SEL)
                                   "Will re-lookup this term")
                    elseif NODE
                      then (LIST (fetch (DINFONODE LABEL) of NODE)
                                 (LIST (fetch (DINFONODE ID) of NODE)
                                       SEL)
                                 "Will re-visit this node")))
           (ITEMS (if [AND NEWITEM (NOT (EQUAL NEWITEM (CAR OLDITEMS]
                      then (CONS NEWITEM (for ITEM in OLDITEMS as I from 2 to DINFO.HISTORY.LENGTH collect ITEM))
                    else OLDITEMS)))
          (replace (DINFOGRAPH HISTORY.ITEMS) of GRAPH with ITEMS)
          (AND OLDWINDOW (PROGN (DETACHWINDOW OLDWINDOW)
                                (CLOSEW OLDWINDOW)))
          (AND DISPLAY? ITEMS
               (LET [(HISTORYW (ATTACHMENU (create MENU
                                                   MENUFONT _ (OR (FONTP (fetch (DINFOGRAPH MENUFONT)
                                                                            of GRAPH))
                                                                  MENUFONT)
                                                   TITLE _ "History"
                                                   CENTERFLG _ T
                                                   MENUCOLUMNS _ 1
                                                   ITEMS _ ITEMS
                                                   WHENSELECTEDFN _ (FUNCTION DINFO.HISTORIC.UPDATE))
                                           DINFOW
                                           'LEFT
                                           'TOP]
                    (replace (DINFOGRAPH HISTORY.MENU.WINDOW) of GRAPH with HISTORYW])
```

(DINFO.HISTORIC.UPDATE

```
  [LAMBDA (ITEM MENU BUTTONS)                                  (* drc%: "25-Jan-86 18:24")
    (LET* [(ID (CAADR ITEM))
           (SEL (CADADR ITEM))
           (WINDOW (WINDOWPROP (WFROMMENU MENU)
                               'MAINWINDOW))
           (GRAPH (WINDOWPROP WINDOW 'DINFOGRAPH))
           (NODE (FASSOC ID (fetch (DINFOGRAPH NODELST) of GRAPH]
          (if (NOT (OBTAIN.MONITORLOCK (fetch (DINFOGRAPH MONITORLOCK) of GRAPH)
                                       T))
              then (FLASHWINDOW WINDOW)
                   (PROMPTPRINT "DInfo is busy")
            elseif (NULL NODE)
              then (PRINTOUT (GETPROMPTWINDOW WINDOW)
                             T "This node no longer exists")
            else (DINFO.UPDATE GRAPH NODE SEL])

)

(DEFINEQ
```

(DINFO.UPDATE.GRAPH.DISPLAY

```
  [LAMBDA (DINFO.GRAPH NODE FORCE?)                            (* drc%: "27-Jan-86 16:19")
    (LET [(DINFOW (fetch (DINFOGRAPH WINDOW) of DINFO.GRAPH))
          (LOCATION (CONS (fetch (DINFONODE PARENT) of NODE)
                          (fetch (DINFONODE CHILDREN) of NODE]
         (if (AND (NOT FORCE?)
                  (EQUAL LOCATION (fetch (DINFOGRAPH LAST.GRAPH.LOCATION) of DINFO.GRAPH)))
             then                                              (* don't need to relayout grapher display --
                                                               just change which node is inverted.)
                  (DINFO.INVERT.NODE (fetch (DINFOGRAPH GRAPH.WINDOW) of DINFO.GRAPH)
```

```
                        NODE DINFO.GRAPH)
          else (DINFO.SHOWGRAPH (DINFO.LAYOUTGRAPH DINFO.GRAPH NODE)
                       DINFO.GRAPH))
        (replace (DINFOGRAPH LAST.GRAPH.LOCATION) of DINFO.GRAPH with LOCATION)
        (WINDOWPROP (fetch (DINFOGRAPH GRAPH.WINDOW) of DINFO.GRAPH)
               'TITLE
               (CONCAT (fetch (DINFOGRAPH NAME) of DINFO.GRAPH)
                      " - "
                      (fetch (DINFONODE LABEL) of (fetch (DINFOGRAPH CURRENTNODE) of DINFO.GRAPH])
```

(DINFO.UPDATE.FROM.GRAPH

```
[LAMBDA (GRAPHER.NODE GRAPH.WINDOW)                          ; Edited 9-Mar-2024 14:21 by rmk
                                                             (* drc%: "12-Dec-85 18:34")
  (AND GRAPHER.NODE (ADD.PROCESS `[DINFO.UPDATE ',(WINDOWPROP GRAPH.WINDOW 'DINFOGRAPH)
                                        ',(fetch (GRAPHNODE NODEID) of GRAPHER.NODE]
                           'NAME "DInfo From Graph"])
```

(DINFO.GET.GRAPH.WINDOW

```
[LAMBDA (GRAPH REGION)                                       (* drc%: "25-Jan-86 18:05")
  (LET ((W (fetch (DINFOGRAPH GRAPH.WINDOW) of GRAPH)))
       (COND
          ((WINDOWP W))
          (T (SETQ W (DINFO.CREATE.GRAPH.WINDOW GRAPH REGION))
             [WINDOWPROP W 'CLOSEFN (FUNCTION (LAMBDA (W)
                                                (WINDOWPROP W 'DINFOGRAPH NIL]
          (replace (DINFOGRAPH GRAPH.WINDOW) of GRAPH with W)))
       (WINDOWPROP W 'DINFOGRAPH GRAPH)
       W])
```

(DINFO.CREATE.GRAPH.WINDOW

```
[LAMBDA (GRAPH REGION)                                       (* drc%: "25-Jan-86 17:49")
  (LET* ((DINFOW (fetch (DINFOGRAPH WINDOW) of GRAPH))
         (DINFOREGION (WINDOWPROP DINFOW 'REGION))
         (LEFT (DIFFERENCE (DIFFERENCE (fetch (REGION LEFT) of DINFOREGION)
                                       (fetch (REGION WIDTH) of REGION))
                           10))
         (BOTTOM (DIFFERENCE (DIFFERENCE (fetch (REGION BOTTOM) of DINFOREGION)
                                         (fetch (REGION HEIGHT) of REGION))
                             50)))
        (CREATEW (CREATEREGION (if (GEQ LEFT 0)
                                   then LEFT
                                 else (RAND 0 10))
                               (if (GEQ BOTTOM 0)
                                   then BOTTOM
                                 else (RAND 0 10))
                               (fetch (REGION WIDTH) of REGION)
                               (fetch (REGION HEIGHT) of REGION))
                 NIL NIL T])
```

(DINFO.SHOWGRAPH

```
[LAMBDA (GRAPHER.GRAPH DINFO.GRAPH)                          (* drc%: "27-Jan-86 16:15")
  (LET* [(GRAPH.REGION (GRAPHREGION GRAPHER.GRAPH))
         (GRAPH.WINDOW (DINFO.GET.GRAPH.WINDOW DINFO.GRAPH GRAPH.REGION))
         (WINDOW.REGION (WINDOWPROP GRAPH.WINDOW 'REGION]
        [SHAPEW GRAPH.WINDOW (LET [(LEFT (fetch (REGION LEFT) of WINDOW.REGION))
                                   (BOTTOM (fetch (REGION BOTTOM) of WINDOW.REGION))
                                   (HEIGHT (HEIGHTIFWINDOW (fetch (REGION HEIGHT) of GRAPH.REGION)
                                                           T))
                                   (WIDTH (WIDTHIFWINDOW (fetch (REGION WIDTH) of GRAPH.REGION]
                                  (create REGION
                                          LEFT _ LEFT
                                          BOTTOM _ BOTTOM
                                          HEIGHT _ (if (GEQ (IPLUS BOTTOM HEIGHT)
                                                            SCREENHEIGHT)
                                                       then (IDIFFERENCE SCREENHEIGHT BOTTOM)
                                                     else HEIGHT)
                                          WIDTH _ (if (GEQ (IPLUS LEFT WIDTH)
                                                           SCREENWIDTH)
                                                      then (IDIFFERENCE SCREENWIDTH LEFT)
                                                    else WIDTH]
        (SHOWGRAPH GRAPHER.GRAPH GRAPH.WINDOW (FUNCTION DINFO.UPDATE.FROM.GRAPH)
               (FUNCTION DINFO.UPDATE.FROM.GRAPH])
```

(DINFO.INVERT.NODE

```
[LAMBDA (WINDOW DINFO.NODE DINFO.GRAPH)                      (* drc%: "25-Jan-86 18:24")
  (LET* ((NODE (for NODE in (fetch (GRAPH GRAPHNODES) of (WINDOWPROP WINDOW 'GRAPH))
                  thereis (EQ (fetch (GRAPHNODE NODEID) of NODE)
                              DINFO.NODE)))
         (LAST.NODE (fetch (DINFOGRAPH LAST.INVERTED.NODE) of DINFO.GRAPH)))
        (replace (DINFOGRAPH LAST.INVERTED.NODE) of DINFO.GRAPH with NODE)
        (if (NEQ NODE LAST.NODE)
            then (replace (GRAPHNODE NODELABELSHADE) of NODE with BLACKSHADE)
```

```
                                                                    (* (PRINTDISPLAYNODE NODE
                                                                    (create POSITION XCOORD _ 0 YCOORD _ 0) WINDOW))
                (replace (GRAPHNODE NODELABELSHADE) of LAST.NODE with WHITESHADE)
                                                                    (* (PRINTDISPLAYNODE LAST.NODE
                                                                    (create POSITION XCOORD _ 0 YCOORD _ 0) WINDOW))
                (REDISPLAYW WINDOW)
          else (OPENW WINDOW])
```

(DINFO.LAYOUTGRAPH

```
  [LAMBDA (DINFO.GRAPH NODE)                                        (* drc%: "25-Jan-86 18:20")
    (LET* [(WINDOW (fetch (DINFOGRAPH WINDOW) of DINFO.GRAPH))
           (FONT (OR (FONTP (fetch (DINFOGRAPH MENUFONT) of DINFO.GRAPH))
                     MENUFONT))
           (NODELST (fetch (DINFOGRAPH NODELST) of DINFO.GRAPH))
           (CHILDREN (for ID in (fetch (DINFONODE CHILDREN) of NODE) collect (FASSOC ID NODELST)))
           [CHILD.GRAPHER.NODES (for CHILD in CHILDREN collect (create GRAPHNODE
                                                                      NODEID _ CHILD
                                                                      NODELABEL _ (fetch (DINFONODE LABEL)
                                                                                    of CHILD]
           (GRAPHER.NODE (create GRAPHNODE
                                 NODELABELSHADE _ BLACKSHADE
                                 NODEID _ NODE
                                 TONODES _ CHILDREN
                                 NODELABEL _ (fetch (DINFONODE LABEL) of NODE]
          (replace (DINFOGRAPH LAST.INVERTED.NODE) of DINFO.GRAPH with GRAPHER.NODE)
                                                                    (* so DINFO.INVERT.NODE will work right)
          (if (fetch (DINFONODE PARENT) of NODE)
              then (LET* ((PARENT (FASSOC (fetch (DINFONODE PARENT) of NODE)
                                          NODELST))
                          (SIBLINGS (for ID in (fetch (DINFONODE CHILDREN) of PARENT) collect (FASSOC ID NODELST)))
                          [SIBLING.GRAPHER.NODES (for SIBLING in SIBLINGS
                                                    collect (if (EQ (fetch (DINFONODE ID) of SIBLING)
                                                                    (fetch (DINFONODE ID) of NODE))
                                                                then GRAPHER.NODE
                                                              else (create GRAPHNODE
                                                                           NODEID _ SIBLING
                                                                           NODELABEL _ (fetch (DINFONODE LABEL)
                                                                                         of SIBLING]
                          (PARENT.GRAPHER.NODE (create GRAPHNODE
                                                       NODEID _ PARENT
                                                       NODELABEL _ (fetch (DINFONODE LABEL) of PARENT)
                                                       TONODES _ SIBLINGS)))
                         (LAYOUTGRAPH (CONS PARENT.GRAPHER.NODE (NCONC SIBLING.GRAPHER.NODES CHILD.GRAPHER.NODES
                                                                       ))
                                      (LIST PARENT)
                                      NIL FONT))
            else (LAYOUTGRAPH (CONS GRAPHER.NODE CHILD.GRAPHER.NODES)
                              (LIST NODE)
                              NIL FONT])

)

(DEFINEQ
```

(DINFO.UPDATE.TEXT.DISPLAY

```
  [LAMBDA (GRAPH NODE SEL OFF?)                                     ; Edited 3-Feb-2022 11:50 by rmk
                                                                    (* drc%: "25-Jan-86 18:18")
    (LET ((WINDOW (fetch (DINFOGRAPH WINDOW) of GRAPH))
          (FILENAME (DINFO.GET.FILENAME GRAPH NODE))
          (FROM (fetch (DINFONODE FROMBYTE) of NODE))
          (TO (fetch (DINFONODE TOBYTE) of NODE))
          (PROPS (APPEND (LIST 'READONLY T 'NOTITLE T 'TITLEMENUFN 'DINFO.TITLEMENUFN)
                         (fetch (DINFOGRAPH TEXTPROPS) of GRAPH)))
          (OLD.TEXTSTREAM (WINDOWPROP (fetch (DINFOGRAPH WINDOW) of GRAPH)
                                      'TEXTSTREAM))
          TEXTSTREAM FULLFILENAME)                                  (* Default directory and host.)
         (if (OR OFF? (NULL FILENAME))
             then (OPENTEXTSTREAM (CL:UNLESS OFF? (OPENSTRINGSTREAM "This node has no text"))
                                  WINDOW NIL NIL PROPS)
                  (replace (DINFOGRAPH LAST.TEXT) of GRAPH with NIL)
           elseif (SETQ FULLFILENAME (MKATOM (INFILEP FILENAME)))
             then (SETQ TEXTSTREAM (DINFO.OPENTEXTSTREAM FULLFILENAME WINDOW FROM TO PROPS))
                  (DINFO.SHOWSEL TEXTSTREAM SEL)
           else (OPENTEXTSTREAM (OPENSTRINGSTREAM (CONCAT "Sorry, can't find the text for this node."
                                                          (MKSTRING (CHARACTER (CHARCODE CR)))
                                                          "Missing file is: " FILENAME))
                                WINDOW NIL NIL PROPS)
                (replace (DINFOGRAPH LAST.TEXT) of GRAPH with NIL))
         (CLOSEF? OLD.TEXTSTREAM)
         (WINDOWPROP WINDOW 'ICONFN 'DINFO.ICONFN)
         (WINDOWPROP WINDOW 'TEDIT.TITLEMENUFN 'DINFO.TITLEMENUFN])
```

(DINFO.TITLEMENUFN

```
  [LAMBDA (DINFOW)                                                  (* drc%: "25-Jan-86 18:19")
```

```
          (* * This is the TEdit TITLEMENUFN for a DInfo Window. Uses the MENUFN of graph, defaulting to
          DINFO.DEFAULT.MENU.)

    (LET [(GRAPH (WINDOWPROP DINFOW 'DINFOGRAPH]
         (if (OBTAIN.MONITORLOCK (fetch (DINFOGRAPH MONITORLOCK) of GRAPH)
                   T)
             then [LET ((MENUFN (fetch (DINFOGRAPH MENUFN) of GRAPH)))
                       (if (FGETD MENUFN)
                           then (OR (fetch (DINFOGRAPH FMENU.WINDOW) of GRAPH)
                                    (DINFO.ADD.FMENU GRAPH))
                                (RESETFORM (TTYDISPLAYSTREAM (GETPROMPTWINDOW DINFOW))
                                           (APPLY* MENUFN GRAPH))
                         else (RESETFORM (TTYDISPLAYSTREAM (GETPROMPTWINDOW DINFOW))
                                         (DINFO.DEFAULT.MENU GRAPH]
          else (FLASHWINDOW DINFOW)
               (PROMPTPRINT "DInfo is busy"])
```

(DINFO.OPENTEXTSTREAM

```
  [LAMBDA (FILE WINDOW FROM TO PROPS)                                    ; Edited 10-Apr-2024 23:46 by rmk
                                                                         ; Edited 10-Mar-2024 15:37 by rmk
                                                                         (* drc%: "25-Jan-86 18:24")
    (RESETFORM (TTYDISPLAYSTREAM (GETPROMPTWINDOW WINDOW))
           (LET ((TEXTSTREAM (WINDOWPROP WINDOW 'TEXTSTREAM))
                 (THIS.TEXT (LIST FILE FROM TO)))
                (if (AND TEXTSTREAM (EQUAL THIS.TEXT (fetch (DINFOGRAPH LAST.TEXT) of (DINFOGRAPH WINDOW)))
                         (\GETSTREAM TEXTSTREAM 'INPUT T))
                    then ;; Same text, and it's still there and open, so do nothing.
                         TEXTSTREAM
                  else (CL:WHEN TEXTSTREAM (TEDIT.KILL TEXTSTREAM))
                       (CLEARW T)
                       (CLEARW WINDOW)
                       [RESETSAVE NIL `(AND RESETSTATE (WINDOWPROP ,WINDOW 'LAST.TEXT NIL]
                       (PROG1 (OPENTEXTSTREAM FILE WINDOW FROM TO PROPS)
                              (replace (DINFOGRAPH LAST.TEXT) of (DINFOGRAPH WINDOW) with THIS.TEXT))])
```

(DINFO.SHOWSEL

```
  [LAMBDA (TEXTSTREAM SEL)                                               (* drc%: "16-Jan-86 21:30")
    (if (LISTP SEL)
        then (TEDIT.NORMALIZECARET TEXTSTREAM (TEDIT.SETSEL TEXTSTREAM (CADR SEL)
                                                            0))
      elseif (STRINGP SEL)
        then [LET ((CHAR# (TEDIT.FIND TEXTSTREAM SEL)))
                  (if CHAR#
                      then (TEDIT.NORMALIZECARET TEXTSTREAM (TEDIT.SETSEL TEXTSTREAM CHAR# (NCHARS SEL)
                                                                          NIL T]
      else (TEDIT.NORMALIZECARET TEXTSTREAM (TEDIT.SETSEL TEXTSTREAM 0 0])
```

(DINFO.GET.FILENAME

```
  [LAMBDA (GRAPH NODE)                                                   (* drc%: "10-Jan-86 14:47")

          (* * returns the filename of the documentation for NODE in GRAPH.
          Defaults HOST and DIRECTORY to that of graph)

    (LET ((FILE (fetch (DINFONODE FILE) of NODE)))
         (AND FILE (PACKFILENAME 'HOST (OR (FILENAMEFIELD FILE 'HOST)
                                           (fetch (DINFOGRAPH DEFAULTHOST) of GRAPH))
                                 'DEVICE
                                 (OR (FILENAMEFIELD FILE 'DEVICE)
                                     (fetch (DINFOGRAPH DEFAULTDEVICE) of GRAPH))
                                 'DIRECTORY
                                 (OR (FILENAMEFIELD FILE 'DIRECTORY)
                                     (fetch (DINFOGRAPH DEFAULTDIR) of GRAPH))
                                 'BODY FILE])

)

(ADDTOVAR BackgroundMenuCommands (DInfo (DINFO.SELECT.GRAPH)
                                        "Open a DInfo window for browsing documentation."))

(RPAQQ BackgroundMenu NIL)

(RPAQ? DINFO.GRAPHS )

(RPAQ? DINFOMODES '(TEXT GRAPH))

(RPAQ? DINFO.HISTORY.LENGTH 20)

(RPAQ? \DINFO.MAX.MENU.LEN 10)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DINFO.GRAPH.FILES DINFOMODES DINFO.HISTORY.LENGTH \DINFO.MAX.MENU.LEN)
)
```

```
(PUTPROPS DINFO FILETYPE :FAKE-COMPILE-FILE)

(DECLARE%: DONTCOPY

(SETTEMPLATE 'DINFOGRAPHPROP 'MACRO)
)
```

FUNCTION INDEX

VARIABLE INDEX

RECORD INDEX

TEMPLATE INDEX

PROPERTY INDEX

MACRO INDEX