

File created: 4-Dec-2020 21:30:35 {DSK}<Users>arunwelch>SKYDRIVE>DOCUMENTS>UNIX>LISP>CLOS>CURRENT>WEB-EDITOR.;2

changes to: (FNS TREE-ROOTS)

previous date: 17-May-93 11:16:38 {DSK}<Users>arunwelch>SKYDRIVE>DOCUMENTS>UNIX>LISP>CLOS>BROWSER>clt12>WEB-EDITOR.;10

Read Table: XCL

Package: WEB

Format: XCCS

; Copyright (c) 1987, 1988, 1989, 1991, 1993, 2020 by Xerox Corporation. All rights reserved.

(IL:RPAQQ IL:WEB-EDITORCOMS

((COMS IL:\* FILE-HEADER-COMS)

::

::: WEB EDITOR

::

::: Package Setup

(IL:DECLARE\ IL:DONTCOPY (IL:PROPS (IL:WEB-EDITOR IL:MAKEFILE-ENVIRONMENT)  
(IL:WEB-EDITOR IL:FILETYPE)))

::: Global Variables

::: global for cross-editor boxed node destination (like caret for current tty process copy and move destinations)

(VARIABLES DESTINATION-BROWSER)

(COMS ; Client Interface

::: Web Node Class

(CLASSES WEB-NODE)

::: Web Editor Class

(CLASSES WEB-EDITOR)

::: Top Level

(FUNCTIONS MAKE-WEB-EDITOR)  
(METHODS (INITIALIZE-EDITOR (WEB-EDITOR))  
(DESTROY (WEB-EDITOR))  
(BROWSE (WEB-EDITOR)))  
(METHODS

::: For Subclassing

(GET-LABEL (WEB-EDITOR WEB-NODE))  
(GET-SUBS (WEB-EDITOR WEB-NODE))  
(ICON-TITLE (WEB-EDITOR))

::: Adding, Removing, Hiding Nodes.

(ADD-NODE (WEB-EDITOR WEB-NODE))  
(NOTICE-NODE (WEB-EDITOR WEB-NODE WEB-NODE))  
(REMOVE-NODE (WEB-EDITOR WEB-NODE))  
(DELETE-FROM-BROWSER (WEB-EDITOR))  
(REMOVE-FROM-BAD-LIST (WEB-EDITOR))

:::

(RENAME-NODE (WEB-EDITOR WEB-NODE)))

) ; Window System Interface  
(COMS

(METHODS (UPDATE (WEB-EDITOR))  
(CREATE-WINDOW (WEB-EDITOR))  
(SETUP-WINDOW (WEB-EDITOR))  
(DETACH-LISP-WINDOW (WEB-EDITOR))  
(SHRINK (WEB-EDITOR))  
(SET-OUTER-REGION (WEB-EDITOR))  
(SET-REGION (WEB-EDITOR))  
(MOVE (WEB-EDITOR))  
(MOVE1 (WEB-EDITOR))  
(AFTER-MOVE (WEB-EDITOR))  
(AFTER-RESHAPE (WEB-EDITOR))  
(SCROLL-WINDOW (WEB-EDITOR))  
(CLEAR (WEB-EDITOR))

::: Prompt Window Interactions

(GET-PROMPT-WINDOW (WEB-EDITOR))  
(REMOVE-PROMPT-WINDOW (WEB-EDITOR))  
(PROMPT-PRINT (WEB-EDITOR))  
(PROMPT-READ (WEB-EDITOR))  
(PROMPT-FOR-LIST (WEB-EDITOR))  
(PROMPT-FOR-STRING (WEB-EDITOR))  
(PROMPT-FOR-WORD (WEB-EDITOR)))

```

(FUNCTIONS MOVE-DOWN-P)
(FNS WEB-WINDOW-AFTER-MOVE-FN WEB-WINDOW-BUTTON-EVENT-FN WEB-WINDOW-RESHAPE-FN
  WEB-WINDOW-CLOSE-FN IL:|PromptRead|)
(FNS WEB-WINDOW-EXPAND-FN)
(FUNCTIONS WEB-WINDOW-ICON-FN)
(BITMAPS *WEB-EDITOR-ICON-BM* *WEB-EDITOR-ICON-MASK*)
(VARIABLES *WEB-EDITOR-TEMPLATE*)
(VARS (IL:*D-WINDOW-DEFAULT-STREAM* IL:PROMPTWINDOW)
  (WEB-STREAM IL:PROMPTWINDOW))

(COMS ; Layout and Display Engine
(VARS IL:|BrowserMargin| IL:|MaxLatticeHeight| IL:|MaxLatticeWidth|)
(IL:SPECVARS IL:|MaxLatticeHeight| IL:|MaxLatticeWidth|)
(VARS IL:GRAYSHADE1 IL:GRAYSHADE2 IL:GRAYSHADE3 IL:GRAYSHADE4)
(FNS TREE-ROOTS CHILD-NODES REACHABLE-NODES!)
(METHODS (DISPLAY-BROWSER (WEB-EDITOR))
  (BROWSER-OBJECTS (WEB-EDITOR))
  (GET-NODE-LIST (WEB-EDITOR))
  (OBJ-NAME-PAIR (WEB-EDITOR))
  (GRAPH-FITS (WEB-EDITOR))
  (NODE-REGION (WEB-EDITOR))
  ;
  (RECOMPUTE (WEB-EDITOR))
  (RECOMPUTE-IN-PLACE (WEB-EDITOR))
  (RECOMPUTE-LABELS (WEB-EDITOR))
  (RECOMPUTE-IF-OPEN (WEB-EDITOR))
  (CLEAR-LABEL-CACHE (WEB-EDITOR))
  (OBJECT-FROM-LABEL (WEB-EDITOR))
  (CHANGE-FONT-SIZE (WEB-EDITOR))
  (CHANGE-FORMAT (WEB-EDITOR))
  (CHANGE-MAX-LABEL-SIZE (WEB-EDITOR))
  (SHAPE-TO-HOLD (WEB-EDITOR))
  ;
  ; Node Marking and Selecting
  (GET-DISPLAY-LABEL (WEB-EDITOR))
  (BOX-NODE (WEB-EDITOR))
  (UNMARK-NODES (WEB-EDITOR))
  (HIGHLIGHT-NODE (WEB-EDITOR))
  (SHADE-NODE (WEB-EDITOR))
  (DISPLAY-NODE-HIGHTLIGHTS (WEB-EDITOR))
  (DISPLAY-NODE-SHADING (WEB-EDITOR))
  (REMOVE-HIGHLIGHTS (WEB-EDITOR))
  (REMOVE-SHADING (WEB-EDITOR))
  (FLASH-NODE (WEB-EDITOR))
  (FLIP-NODE (WEB-EDITOR))
  (POSITION-NODE (WEB-EDITOR)))
(FNS BOX-PRINT-STRING BREAK-STRING-FOR-BOXING BOX-WINDOW-NODE))
(COMS ; Button Events
(FNS FIND-SELECTED-NODE)
(METHODS (BUTTON-EVENT-FN (WEB-EDITOR))
  (LEFT-SELECTION (WEB-EDITOR))
  (MIDDLE-SELECTION (WEB-EDITOR))
  (RIGHT-SELECTION (WEB-EDITOR))
  (TITLE-SELECTION (WEB-EDITOR))
  (NODE-SELECTION (WEB-EDITOR))
  (NODE-ACTION (WEB-EDITOR))
  (NODE-MENU-ITEMS (WEB-NODE))
  ;;
  (CHOICE-MENU (WEB-EDITOR))
  (DO-SELECTED-COMMAND (WEB-EDITOR))
  (WHEN-MENU-ITEM-HELD (WEB-EDITOR))
  (ITEM-MENU (WEB-EDITOR))
  (GET-MENU-ITEMS (WEB-EDITOR))
  (CLEAR-MENU-CACHE (WEB-EDITOR)))
(FNS WEB-MENU-WHENSELECTEDFN WINDOW-WHEN-HELD-FN)
(FNS SUB-ITEM-SELECTION DUAL-SUB-ITEMS WINDOW-WHEN-HELD-FN DO-MENU-METHOD DUAL-MENU
  DUAL-SELECTION)
  ; Node Moving Protocol
  (METHODS (NODE-MOVE (WEB-EDITOR))
  (NODE-MOVE-SHALLOW (WEB-EDITOR))
  (SCIONS (WEB-NODE))
  (MAKE-REG-ASSOC (WEB-EDITOR))
  (REORDER-TREE (WEB-EDITOR))
  (MOVE-NODE (WEB-NODE)))
  ;
  ;;
(IL:DECLARE\ IL:DONTEVAL@LOAD IL:DOEVAL@COMPILE IL:DONTCOPY IL:COMPILERVERS
  (IL:ADDVARS (IL:NLAMA)
  (IL:NLAML)
  (IL:LAMA WINDOW-WHEN-HELD-FN WINDOW-WHEN-HELD-FN WEB-WINDOW-EXPAND-FN
  WEB-WINDOW-RESHAPE-FN WEB-WINDOW-BUTTON-EVENT-FN WEB-WINDOW-AFTER-MOVE-FN))))
(IL:RPAQQ FILE-HEADER-COMS ((IL:P (FORMAT T "~&;WEB-EDITOR Copyright (c) 1987, VENUE Corporation. All rights
  reserved.~%"
  (PROVIDE "WEB-EDITOR"))))

```

(FORMAT T "~&;WEB-EDITOR Copyright (c) 1987, VENUE Corporation. All rights reserved.~%")

(PROVIDE "WEB-EDITOR")

::

::: WEB EDITOR

::

:: Package Setup

(IL:DECLARE\ : IL:DONTCOPY

(IL:PUTPROPS **IL:WEB-EDITOR IL:MAKEFILE-ENVIRONMENT**

(:PACKAGE (LET ((\*PACKAGE\*))

:: Put IN Seven EXtremely Random USEr Interface COmmands

(PROVIDE "WEB-EDITOR")

(IN-PACKAGE "WEB" :NICKNAMES ' ("WEB-EDITOR"))

:: EXPORT

(FLET ((XCL-USER::EXPORT-FROM-WEB (&REST XCL-USER::SYMBOL-NAMES)
(LET ((XCL-USER::PKG (FIND-PACKAGE "WEB"))))
(DOLIST (XCL-USER::NAME XCL-USER::SYMBOL-NAMES)
(EXPORT (INTERN XCL-USER::NAME XCL-USER::PKG)
XCL-USER::PKG))))))

:: Class Definitions and Slot Access

(XCL-USER::EXPORT-FROM-WEB "WEB-EDITOR" "WEB-NODE" "NODE-NAME" "NODE-LINKS"
"NODE-BACK-LINKS" "LOCAL-COMMANDS" "NODE-MOVER-P" "TITLE-ITEMS"
"LEFT-BUTTON-ITEMS" "MIDDLE-BUTTON-ITEMS" "RIGHT-BUTTON-ITEMS"
"BROWSE-FONT")

:: For Subclassing

(XCL-USER::EXPORT-FROM-WEB "GET-LABEL" "GET-SUBS" "ICON-TITLE"
"NODE-MENU-ITEMS" "REORDER-TREE" "MOVE-NODE")

:: Top Level

(XCL-USER::EXPORT-FROM-WEB "MAKE-WEB-EDITOR" "INITIALIZE-EDITOR" "BROWSE"
"DISPLAY-BROWSER" "DESTROY" "ADD-NODE" "NOTICE-NODE" "REMOVE-NODE"
"RENAME-NODE")

:: Window Operations

(XCL-USER::EXPORT-FROM-WEB "SHRINK" "MOVE" "CLEAR" "PROMPT-PRINT"
"PROMPT-READ" "PROMPT-FOR-LIST" "PROMPT-FOR-STRING" "PROMPT-FOR-WORD")

:: Recomputing and Changing parameters

(XCL-USER::EXPORT-FROM-WEB "RECOMPUTE" "RECOMPUTE-IN-PLACE"
"RECOMPUTE-LABELS" "RECOMPUTE-IF-OPEN" "CLEAR-LABEL-CACHE"
"CHANGE-FONT-SIZE" "CHANGE-FORMAT" "SHAPE-TO-HOLD")

:: For CLOS-BROWSER???

(XCL-USER::EXPORT-FROM-WEB "BOXED-NODE" "BOX-NODE")

:: USE

(USE-PACKAGE ' ("CLOS" "LISP" "XCL")
"WEB")

:: IMPORT

(FLET ((XCL-USER::IMPORT-FROM-PACKAGE
(XCL-USER::NAMES XCL-USER::FROM &OPTIONAL XCL-USER::SHADOW-P)
(LET ((XCL-USER::FROM-PACKAGE (FIND-PACKAGE XCL-USER::FROM))
(FUNCALL (IF XCL-USER::SHADOW-P
#' SHADOWING-IMPORT
#' IMPORT)
(MAPCAR #' (LAMBDA (XCL-USER::NAME)
(INTERN XCL-USER::NAME XCL-USER::FROM-PACKAGE))
XCL-USER::NAMES))))))
(XCL-USER::IMPORT-FROM-PACKAGE ' ("CLASSES" "METHODS")
"CLOS")
(XCL-USER::IMPORT-FROM-PACKAGE ' ("FALSE")
"XCL")
(XCL-USER::IMPORT-FROM-PACKAGE ' ("FUNCTIONS" "FNS" "VARIABLES" "VARS"
"BITMAPS" "COMS")
"IL"))
(FIND-PACKAGE "WEB"))
:READTABLE "XCL" :BASE 10))

(IL:PUTPROPS **IL:WEB-EDITOR IL:FILETYPE** :COMPILE-FILE)

)

:: Global Variables

:: global for cross-editor boxed node destination (like caret for current tty process copy and move destinations)

(DEFGLOBALPARAMETER **DESTINATION-BROWSER** NIL

"global for cross-editor boxed node destination (like caret for current tty process copy and move destinations)"

:: Client Interface
:: Web Node Class

```
(DEFCLASS WEB-NODE ()
  (NAME :INITFORM NIL ; Name of Node
   :ACCESSOR NODE-NAME)
  (TO-LINKS :INITFORM NIL ; Nodes that this Node has Links TO
   :ACCESSOR GET-TO-LINKS :ACCESSOR NODE-LINKS)
  (PARENT :INITFORM NIL :ACCESSOR NODE-BACK-LINKS))
```

:: Web Editor Class

```
(DEFCLASS WEB-EDITOR ()
  ;; NODES
  (STARTING-LIST :INITFORM NIL ; list of objects used to compute this browser
   )
  (GOOD-LIST :INITFORM NIL ; limit choices to this set
   )
  (BAD-LIST :INITFORM NIL ; Don't put in any items on this set
   )
  ;; GRAPHER FORMAT
  (TOP-ALIGN :INITFORM NIL)
  (BROWSE-FONT :INITFORM (IL:FONTCREATE ' (IL:HELVETICA 10 IL:BOLD)))
  (BROWSE-FONT-FAMILY :INITFORM ' IL:HELVETICA)
  (BROWSE-FONT-FACE :INITFORM ' IL:BOLD)
  (GRAPH-FORMAT :INITFORM ' (IL:LATTICE))
  (GRAPH-FORMAT-CHOICES :ALLOCATION :CLASS :INITFORM ' ((IL:HORIZONTAL/LATTICE ' (IL:LATTICE))
   (IL:VERTICAL/LATTICE ' (IL:VERTICAL IL:LATTICE))
   (IL:HORIZONTAL/TREE ' (IL:COPIES/ONLY))
   (IL:VERTICAL/TREE ' (IL:VERTICAL IL:COPIES/ONLY))))
```

:: WINDOW Interface

```
(WINDOW :INITFORM NIL)
(TITLE :INITFORM "Web Editor" ; If not NIL will be put in title of window
 )
(LEFT :INITFORM 0 ; left position of window
 )
(BOTTOM :INITFORM 0 ; bottom position of window
 )
(WIDTH :INITFORM 64)
(HEIGHT :INITFORM 32)
```

:: NODE Labels

```
(LABEL-CACHE :INITFORM NIL)
(LABEL-MAX-LINES :INITFORM NIL
 ;; the maximum number of lines to use in 'boxed' labels -- note that if the label wont fit within the LabelMaxLines and
 ;; LabelMaxCharsWidth restrictions, it will be truncated
 )
(LABEL-MAX-CHARS-WIDTH :INITFORM NIL
 ;; the maximum width for labels -- if label is too big, it will be 'boxed'
 )
```

:: NODE Operations

```
(LAST-SELECTED-OBJECT :INITFORM NIL ; last object selected
 )
(BOXED-NODE :INITFORM NIL ; last item Boxed, if any
 )
(BOX-LINE-WIDTH :ALLOCATION :CLASS
 ;; width to make box for BoxNode
 :INITFORM 1)
(NODE-MOVER-P :ALLOCATION :CLASS :INITFORM NIL)
```

:: MENUS

```
(CACHE-MENU-P :INITFORM T)
(MENU-CACHE :INITFORM NIL ; Will Cache Menus only if CACHE-MENU-P is T
 )
(LOCAL-COMMANDS :ALLOCATION :CLASS
 ;; messages that should be sent to browser when item seleted in menu, even if object does understand them
 :INITFORM
 ' (BOX-NODE RECOMPUTE ADD-ROOT))
(TITLE-ITEMS :ALLOCATION :CLASS
 ;; Items for menu of selections in title of window
 :INITFORM
 ' ("Recompute" RECOMPUTE " " (IL:SUBITEMS ("Recompute" RECOMPUTE "Recompute lattice from starting
 objects"))
```

```

("Recompute Labels" RECOMPUTE-LABELS "Recomputes the labels")
("Recompute In Place" RECOMPUTE-IN-PLACE "Recompute keeping
  current view in window"))
("Shape To Hold" SHAPE-TO-HOLD "Make window large or small enough to just hold graph")
("Change Font Size" CHANGE-FONT-SIZE "Choose a new size Font")
("Change Format" CHANGE-FORMAT "Change format between lattice and tree")))
(LEFT-BUTTON-ITEMS :ALLOCATION :CLASS
  ;; Menu items for LeftButton selection -- Value sent as message to object or browser -- see LocalCommands
  :INITFORM
  ' ("Box Node" BOX-NODE "Draw box around selected node.\
Unboxed by another BoxNode")
  ("Pretty Print" PP "Prettyprint selected item")))
(MIDDLE-BUTTON-ITEMS :ALLOCATION :CLASS
  ;; Menu items for MiddleButton selection -- Value sent as message to object or browser -- see LocalCommands
  :INITFORM
  ' ("Inspect" IL:|Inspect| INSPECT "Inspect selected item")
  ("Edit" EDIT-OBJECT "Edit selected item")
  ("Delete From Browser" DELETE-FROM-BROWSER "Do not show item or its subs")))
(RIGHT-BUTTON-ITEMS :ALLOCATION :CLASS :INITFORM ' ("Close" (CLOSEW ("Close" CLOSEW
  ("Destroy" DESTROY))))
  ("Snap" SNAP)
  ("Paint" PAINT)
  ("Clear" CLEAR)
  ("Bury" BURY)
  ("Repaint" REPAINT)
  ("Hardcopy" (HARDCOPY ("Hardcopy to File"
    HARDCOPY-TO-FILE)
    ("Hardcopy to Printer"
    HARDCOPY-TO-PRINTER))))
  ("Move" MOVE)
  ("Shape" SHAPE)
  ("Shrink" SHRINK))
  ; Items to be done if Right button is selected
)))

```

;; Top Level

```

(DEFUN MAKE-WEB-EDITOR ()
  (LET ((EDITOR (MAKE-INSTANCE 'WEB-EDITOR))
        (INITIALIZE-EDITOR EDITOR)))
  )

```

```

(DEFMETHOD INITIALIZE-EDITOR ((SELF WEB-EDITOR))
  (LET NIL (CREATE-WINDOW SELF)
    SELF))

```

```

(DEFMETHOD DESTROY ((SELF WEB-EDITOR))
  (LET* ((WINDOW (SLOT-VALUE SELF 'WINDOW))
         (ICON-WINDOW (IL:WINDOWPROP WINDOW 'IL:ICONWINDOW))
         (IL:CLOSEW WINDOW)
         (IF ICON-WINDOW (IL:CLOSEW ICON-WINDOW))
         (DETACH-LISP-WINDOW SELF)))
  )

```

```

(DEFMETHOD BROWSE ((SELF WEB-EDITOR)
  &OPTIONAL BROWSE-LIST WINDOW-OR-TITLE GOOD-LIST POSITION)
  ; 11-Sep-84 07:24
  ; Call Show and then shape to hold and move for first time

```

```

(COND
  ((IL:WINDOWP WINDOW-OR-TITLE)
   (SETF (SLOT-VALUE SELF 'WINDOW)
         WINDOW-OR-TITLE))
  (WINDOW-OR-TITLE (SETF (SLOT-VALUE SELF 'TITLE)
                        WINDOW-OR-TITLE)))
(COND
  ((AND BROWSE-LIST (IL:NLISTP BROWSE-LIST))
   (IL:SETQ BROWSE-LIST (LIST BROWSE-LIST))))
(SETF (SLOT-VALUE SELF 'STARTING-LIST)
      BROWSE-LIST)
(SETF (SLOT-VALUE SELF 'GOOD-LIST)
      GOOD-LIST)
(DISPLAY-BROWSER SELF)
(SHAPE-TO-HOLD SELF)
(MOVE SELF POSITION)
SELF)

```

```

(DEFMETHOD GET-LABEL ((WEB-EDITOR WEB-EDITOR)
  (NODE WEB-NODE))
  ; Get a label for an object to be displayed in the browser.
  (NODE-NAME NODE))

```

```
(DEFMETHOD GET-SUBS ((EDITOR WEB-EDITOR)
                     (NODE WEB-NODE))
  (NODE-LINKS NODE)) ; Gets a set of subs from an object for browsing
```

```
(DEFMETHOD ICON-TITLE ((SELF WEB-EDITOR)) ; 18-Jan-85 15:35
  ;; Compute the icon title for this browser
  '|Web Editor|)
```

```
(DEFMETHOD ADD-NODE ((WEB-EDITOR WEB-EDITOR)
                     (NEW-NODE WEB-NODE)) ; 11-Dec-86 10:23
  ;; Add a new node to the browser.
  (PUSHNEW NEW-NODE (SLOT-VALUE WEB-EDITOR 'STARTING-LIST))
  (IF (SLOT-VALUE WEB-EDITOR 'GOOD-LIST)
      (PUSHNEW NEW-NODE (SLOT-VALUE WEB-EDITOR 'GOOD-LIST))))
```

```
(DEFMETHOD NOTICE-NODE ((WEB-EDITOR WEB-EDITOR)
                        (WEB-NODE WEB-NODE)
                        (PARENT-NODE WEB-NODE))
  (PUSH WEB-NODE (SLOT-VALUE PARENT-NODE 'TO-LINKS))
  (ADD-NODE WEB-EDITOR WEB-NODE))
```

```
(DEFMETHOD REMOVE-NODE ((WEB-EDITOR WEB-EDITOR)
                        (BYE-NODE WEB-NODE))
  (WITH-SLOTS (STARTING-LIST GOOD-LIST BAD-LIST)
              WEB-EDITOR
  ;;
  (SETF STARTING-LIST (DELETE BYE-NODE STARTING-LIST))
  (IF GOOD-LIST
      (SETF GOOD-LIST (DELETE BYE-NODE GOOD-LIST)))
  (IF BAD-LIST
      (SETF BAD-LIST (DELETE BYE-NODE BAD-LIST)))
  (SETF (NODE-LINKS (NODE-BACK-LINKS BYE-NODE))
        (DELETE BYE-NODE (NODE-LINKS (NODE-BACK-LINKS BYE-NODE))))))
```

```
(DEFMETHOD DELETE-FROM-BROWSER ((SELF WEB-EDITOR)
                                (OBJ OBJ-NAME)) ; 5-Aug-86 16:50
  ;; Place on badList for Browser
  (PUSHNEW OBJ (SLOT-VALUE SELF 'BAD-LIST))
  (RECOMPUTE SELF))
```

```
(DEFMETHOD REMOVE-FROM-BAD-LIST ((SELF WEB-EDITOR)) ; 28-Dec-85 10:04
  ;; Remove an item from BadList to allow it to be displayed once
  ;; again
  (COND
   ((NULL (SLOT-VALUE SELF 'BAD-LIST))
    (IL:CLRSPROMPT)
    (IL:PROMPTPRINT "No BadList items.")(
    (T (PROG ((IL:|item| (IL:MENU (IL:|create| IL:MENU
                                         IL:TITLE IL:_ "BadList Items"
                                         IL:ITEMS IL:_ (SLOT-VALUE SELF 'BAD-LIST))))
            (COND
             (IL:|item| (SETF (SLOT-VALUE SELF 'BAD-LIST)
                             '(IL:DREMOVE IL:|item| (SLOT-VALUE SELF 'BAD-LIST)))
              (RECOMPUTE SELF))
             (T (IL:CLRSPROMPT)
                (IL:PROMPTPRINT "Nothing Selected"))))))))
```

```
(DEFMETHOD RENAME-NODE ((WEB-EDITOR WEB-EDITOR)
                        (WEB-NODE WEB-NODE)
                        (NEW-NAME))
  (SETF (NODE-NAME WEB-NODE)
        NEW-NAME)
  (CLEAR-LABEL-CACHE WEB-EDITOR WEB-NODE))
```

```
;;
;; Window System Interface
```

```
(DEFMETHOD UPDATE ((SELF WEB-EDITOR)) ; 29-Sep-86 11:56
  ;; make the Lisp window be consistent with ivs
  (LET* ((WINDOW (SLOT-VALUE SELF 'WINDOW))
         (REGION (AND (SLOT-VALUE SELF 'WIDTH)
                      (SLOT-VALUE SELF 'HEIGHT)
                      (IL:|create| IL:REGION
```

```

IL:LEFT IL:_ (OR (SLOT-VALUE SELF 'LEFT)
                (SETF (SLOT-VALUE SELF 'LEFT)
                     IL:LASTMOUSEX))
IL:BOTTOM IL:_ (OR (SLOT-VALUE SELF 'BOTTOM)
                  (SETF (SLOT-VALUE SELF 'BOTTOM)
                       IL:LASTMOUSEY))
IL:WIDTH IL:_ (SLOT-VALUE SELF 'WIDTH)
IL:HEIGHT IL:_ (SLOT-VALUE SELF 'HEIGHT))))
(COND
  ((AND REGION (NOT (IL:EQUAL REGION (IL:WINDOWPROP WINDOW 'IL:REGION))))
    ; The shape has changed. --- This is complicated because of
    ; ATTACHEDWINDOWS.
    (LET* ((ATTACHED-WINDOWS (IL:WINDOWPROP WINDOW 'IL:ATTACHEDWINDOWS))
           (ATTACHMENT-SPECS (IL:|for| IL:\w IL:|in| ATTACHED-WINDOWS
                              IL:|collect| (LIST (IL:WINDOWPROP IL:\w 'IL:DOWINDOWCOMFN)
                                                (IL:WINDOWPROP IL:\w 'IL:WHEREATTACHED)
                                                (IL:WINDOWPROP IL:\w 'IL:PASSTOMAINCOMS))))))
      ((IL:|for| IL:\w IL:|in| ATTACHED-WINDOWS IL:|do| (IL:DETACHWINDOW IL:\w))
       (IL:SHAPEW WINDOW REGION)
       ((IL:|for| IL:\w IL:|in| ATTACHED-WINDOWS IL:|as| IL:|spec| IL:|in| ATTACHMENT-SPECS
          IL:|do| (IL:ATTACHWINDOW IL:\w WINDOW (CAADR IL:|spec|)
                                (CDADR IL:|spec|))
                (IL:WINDOWPROP IL:\w 'IL:DOWINDOWCOMFN (CAR IL:|spec|))
                (IL:WINDOWPROP IL:\w 'IL:PASSTOMAINCOMS (CADDR IL:|spec|))))))
      (AND (NOT (IL:EQUAL (SLOT-VALUE SELF 'TITLE)
                          (IL:WINDOWPROP WINDOW 'IL:TITLE)))
           (IL:WINDOWPROP WINDOW 'IL:TITLE (SLOT-VALUE SELF 'TITLE))))))

```

```

(DEFMETHOD CREATE-WINDOW ((SELF WEB-EDITOR) ; 10-Apr-86 14:32
                          ; Create the Lisp window for this window but don't open it.
  (LET ((WINDOW (IL:CREATEW (IL:CREATEREGION IL:LASTMOUSEX IL:LASTMOUSEY 25 25)
                             (SLOT-VALUE SELF 'TITLE)
                             NIL T)))
    (SETF (SLOT-VALUE SELF 'WINDOW)
          WINDOW)
    (SETUP-WINDOW SELF)
    WINDOW))

```

```

(DEFMETHOD SETUP-WINDOW ((SELF WEB-EDITOR) ; 10-Apr-86 14:32
                          ; Create the Lisp window for this window but don't open it.
  (LET ((WINDOW (SLOT-VALUE SELF 'WINDOW)))
    (IL:WINDOWPROP WINDOW 'WEB-EDITOR SELF)
    (IL:WINDOWPROP WINDOW 'IL:ICONFN 'WEB-WINDOW-ICON-FN)
    (IL:WINDOWPROP WINDOW 'IL:BUTTONEVENTFN 'WEB-WINDOW-BUTTON-EVENT-FN)
    (IL:WINDOWADDPROP WINDOW 'IL:AFTERMOVEFN 'WEB-WINDOW-AFTER-MOVE-FN)
    (IL:WINDOWADDPROP WINDOW 'IL:RESHAPEFN 'WEB-WINDOW-RESHAPE-FN)
    (IL:WINDOWADDPROP WINDOW 'IL:CLOSEFN 'WEB-WINDOW-CLOSE-FN)
    (IL:WINDOWPROP WINDOW 'IL:ICONFN 'WEB-WINDOW-ICON-FN) ; window should be invert so that links etc. can be erased
    (IL:DSOPERATION 'IL:INVERT WINDOW) ; kludge: because GRAPHER adds its own
    ; COPYBUTTONEVENTFN
    (IL:WINDOWPROP WINDOW 'IL:COPYBUTTONEVENTFN NIL)
    (IL:WINDOWPROP WINDOW 'IL:TITLE (SLOT-VALUE SELF 'TITLE))
    WINDOW))

```

```

(DEFMETHOD DETACH-LISP-WINDOW ((SELF WEB-EDITOR) ; 8-Apr-87 17:25

```

;;; Forget about the current lisp window

```

(LET ((VAL (SLOT-VALUE SELF 'WINDOW)))
  (IL:|if| (IL:WINDOWP VAL)
    IL:|then| (SETF (SLOT-VALUE SELF 'WINDOW)
                   NIL)
              (IL:WINDOWPROP VAL 'WEB-EDITOR NIL)
              (IL:WINDOWPROP VAL 'IL:RIGHTBUTTONFN NIL)
              (IL:WINDOWPROP VAL 'IL:BUTTONEVENTFN NIL)
              NIL)
  IL:|else| NIL))

```

```

(DEFMETHOD SHRINK ((SELF WEB-EDITOR)
                   &OPTIONAL TOWHAT POS EXPANDFN)
  (LET* ((WINDOW (SLOT-VALUE SELF 'WINDOW)))
    (IF (IL:WINDOWP WINDOW)
        (IL:SHRINKW WINDOW TOWHAT POS EXPANDFN))))

```

```

(DEFMETHOD SET-OUTER-REGION ((SELF WEB-EDITOR)
                              REGION NO-UPDATE-FLG) ; 16-Apr-86 13:21

```

;;; Make Loops Window have region parameters

```

(SETF (SLOT-VALUE SELF 'LEFT)
      (IL:|fetch| IL:LEFT IL:|of| REGION))
(SETF (SLOT-VALUE SELF 'BOTTOM)

```

```

      (IL:|fetch| IL:BOTTOM IL:|of| REGION))
    (SETF (SLOT-VALUE SELF 'WIDTH)
      (IL:|fetch| IL:WIDTH IL:|of| REGION))
    (SETF (SLOT-VALUE SELF 'HEIGHT)
      (IL:|fetch| IL:HEIGHT IL:|of| REGION))
    (IL:|if| (NOT NO-UPDATE-FLG)
      IL:|then| (UPDATE SELF))
    REGION)

```

```

(DEFMETHOD SET-REGION ((SELF WEB-EDITOR)
  REGION &OPTIONAL NO-UPDATE-FLG)
  ; 16-Apr-86 13:22

```

;;; Make Loops Window have region parameters

```

  (SET-OUTER-REGION SELF (IL:CREATEREGION (IL:|fetch| IL:LEFT IL:|of| REGION)
    (IL:|fetch| IL:BOTTOM IL:|of| REGION)
    (IL:WIDTHIFWINDOW (IL:|fetch| IL:WIDTH IL:|of| REGION)
      (IL:WINDOWPROP (SLOT-VALUE SELF 'WINDOW)
        'IL:BORDER))
    (IL:HEIGHTIFWINDOW (IL:|fetch| IL:HEIGHT IL:|of| REGION)
      (SLOT-VALUE SELF 'TITLE)
      (IL:WINDOWPROP (SLOT-VALUE SELF 'WINDOW)
        'IL:BORDER)))
    NO-UPDATE-FLG))

```

```

(DEFMETHOD MOVE ((SELF WEB-EDITOR)
  X-OR-POS &OPTIONAL Y)
  ; 11-Sep-86 13:24

```

;;; Move the window

```

  (MOVE1 SELF (OR X-OR-POS (LET* ((ENTIRE-REGION (IL:WINDOWREGION (SLOT-VALUE SELF 'WINDOW)))
    (POS (IL:GETBOXPOSITION (IL:|fetch| IL:WIDTH IL:|of| ENTIRE-REGION)
      (IL:|fetch| IL:HEIGHT IL:|of| ENTIRE-REGION)
      (IL:|fetch| IL:LEFT IL:|of| ENTIRE-REGION)
      (IL:|fetch| IL:BOTTOM IL:|of| ENTIRE-REGION))))
    (IL:|create| IL:POSITION
      IL:XCOORD IL:_ (IL:PLUS (IL:|fetch| IL:XCOORD IL:|of| POS)
        (IL:DIFFERENCE (SLOT-VALUE SELF 'LEFT)
          (IL:|fetch| IL:LEFT IL:|of| ENTIRE-REGION)))
      IL:YCOORD IL:_ (IL:PLUS (IL:|fetch| IL:YCOORD IL:|of| POS)
        (IL:DIFFERENCE (SLOT-VALUE SELF 'BOTTOM)
          (IL:|fetch| IL:BOTTOM IL:|of| ENTIRE-REGION))))
    Y))

```

```

(DEFMETHOD MOVE1 ((SELF WEB-EDITOR)
  X-OR-POS &OPTIONAL Y)
  ; 13-Aug-86 19:10

```

;; Move the window

```

  (LET ((NEEDS-UPDATE? (NOT (IL:SUBREGIONP (IL:CONSTANT (IL:CREATEREGION 0 0 IL:SCREENWIDTH IL:SCREENHEIGHT)
    (IL:WINDOWPROP (SLOT-VALUE SELF 'WINDOW)
      'IL:REGION))))))
    (PROG1 (IL:MOVEW (SLOT-VALUE SELF 'WINDOW)
      X-OR-POS Y)
      ; The left and right IVs are updated by the message AfterMove
      (COND
        (NEEDS-UPDATE? (UPDATE SELF)))))

```

```

(DEFMETHOD AFTER-MOVE ((SELF WEB-EDITOR)
  ; 10-Apr-86 16:10

```

;;; The window has been moved. Update the left and bottom

```

  (LET ((REGION (IL:WINDOWPROP (SLOT-VALUE SELF 'WINDOW)
    'IL:REGION)))
    (SETF (SLOT-VALUE SELF 'LEFT)
      (IL:|fetch| IL:LEFT IL:|of| REGION))
    (SETF (SLOT-VALUE SELF 'BOTTOM)
      (IL:|fetch| IL:BOTTOM IL:|of| REGION)))

```

```

(DEFMETHOD AFTER-RESHAPE ((SELF WEB-EDITOR)
  OLD-BITMAP-IMAGE OLD-REGION OLD-SCREEN-REGION)
  ; 10-Apr-86 16:12

```

;;; The window has been reshaped

```

  (LET ((REGION (IL:WINDOWPROP (SLOT-VALUE SELF 'WINDOW)
    'IL:REGION)))
    (SETF (SLOT-VALUE SELF 'LEFT)
      (IL:|fetch| IL:LEFT IL:|of| REGION))
    (SETF (SLOT-VALUE SELF 'BOTTOM)
      (IL:|fetch| IL:BOTTOM IL:|of| REGION))
    (SETF (SLOT-VALUE SELF 'WIDTH)
      (IL:|fetch| IL:WIDTH IL:|of| REGION))

```



```
(SETF (SLOT-VALUE SELF 'HEIGHT)
      (IL:|fetch| IL:HEIGHT IL:|of| REGION))
(IL:RESHAPEBYREPAINTFN (SLOT-VALUE SELF 'WINDOW)
                      OLD-BITMAP-IMAGE OLD-REGION OLD-SCREEN-REGION)))
```

```
(DEFMETHOD SCROLL-WINDOW ((SELF WEB-EDITOR)
                          DSP-X DSP-Y WINDOW-X WINDOW-Y) ; 10-Apr-86 14:58
```

;; scroll the window to set the point dspX,dspY in the given window position -- default is the lower left corner. If any x or y is a FIXP, it is treated as a  
 ;; absolute position. If FLOATP, it is treated as a relative position. Return the position of the new lower left corner.

```
(LET* ((WINDOW (SLOT-VALUE SELF 'WINDOW))
       (VISIBLE-REGION (IL:DSPCLIPPINGREGION NIL WINDOW))
       (EXTENT (IL:WINDOWPROP WINDOW 'IL:EXTENT))) ; figure out what to do with default and relative offsets
      (IL:SETQ WINDOW-X (IL:|if| (NULL WINDOW-X)
                                IL:|then| 0
                                IL:|elseif| (IL:FLOATP WINDOW-X)
                                IL:|then| (IL:FIX (IL:TIMES WINDOW-X (IL:WINDOWPROP WINDOW 'IL:WIDTH)))
                                IL:|else| WINDOW-X))
      (IL:SETQ WINDOW-Y (IL:|if| (NULL WINDOW-Y)
                                IL:|then| 0
                                IL:|elseif| (IL:FLOATP WINDOW-Y)
                                IL:|then| (IL:FIX (IL:TIMES WINDOW-Y (IL:WINDOWPROP WINDOW 'IL:HEIGHT)))
                                IL:|else| WINDOW-Y))
      (IL:SETQ DSP-X (IL:|if| (NULL DSP-X)
                             IL:|then| (IL:|fetch| IL:LEFT IL:|of| VISIBLE-REGION)
                             IL:|elseif| (IL:FLOATP DSP-X)
                             IL:|then| (IL:FIX (IL:TIMES DSP-X (IL:|fetch| IL:WIDTH IL:|of| EXTENT)))
                             IL:|else| DSP-X))
      (IL:SETQ DSP-Y (IL:|if| (NULL DSP-Y)
                             IL:|then| (IL:IMINUS (IL:|fetch| IL:BOTTOM IL:|of| VISIBLE-REGION))
                             IL:|elseif| (IL:FLOATP DSP-Y)
                             IL:|then| (IL:FIX (IL:TIMES DSP-Y (IL:|fetch| IL:HEIGHT IL:|of| EXTENT)))
                             IL:|else| DSP-Y))
      (IL:SCROLLW WINDOW (IL:IPLUS WINDOW-X (IL:IDIFFERENCE (IL:|fetch| IL:LEFT IL:|of| VISIBLE-REGION)
                                                            DSP-X))
                    (IL:IPLUS WINDOW-Y (IL:IDIFFERENCE (IL:|fetch| IL:BOTTOM IL:|of| VISIBLE-REGION)
                                                            DSP-Y))) ; return the resulting position
      (IL:SETQ VISIBLE-REGION (IL:DSPCLIPPINGREGION NIL WINDOW))
      (IL:|create| IL:POSITION
                  IL:XCOORD IL:_ (IL:|fetch| IL:LEFT IL:|of| VISIBLE-REGION)
                  IL:YCOORD IL:_ (IL:|fetch| IL:BOTTOM IL:|of| VISIBLE-REGION))))
```

```
(DEFMETHOD CLEAR ((SELF WEB-EDITOR) ; empty the window of active regions, return the window
                  (LET ((WINDOW (SLOT-VALUE SELF 'WINDOW))
                        (IL:WINDOWPROP WINDOW 'IL:GRAPH NIL)
                        (IL:CLEARW WINDOW)
                        WINDOW))
```

```
(DEFMETHOD GET-PROMPT-WINDOW ((SELF WEB-EDITOR)
                               &OPTIONAL LINES FONT-DEF) ; 8-Apr-87 15:43
```

```
;; Return the current prompt window
(LET ((W (IL:GETPROMPTWINDOW (SLOT-VALUE SELF 'WINDOW)
                             (OR LINES 2)
                             (OR (IL:FONTCREATE FONT-DEF))))
      (IF FONT-DEF
          (IL:DSPFONT (IL:FONTCREATE FONT-DEF)
                      W))
      W))
```

```
(DEFMETHOD REMOVE-PROMPT-WINDOW ((SELF WEB-EDITOR) ; 8-Apr-87 15:43
                                   (IL:REMOVEPROMPTWINDOW (SLOT-VALUE SELF 'WINDOW)))
```

```
(DEFMETHOD PROMPT-PRINT ((SELF WEB-EDITOR)
                          PROMPT) ; 13-Aug-86 18:46
```

```
;; Prints out a prompt in an attached prompt window
(IL:PRIN1 PROMPT (GET-PROMPT-WINDOW SELF))
```

```
(DEFMETHOD PROMPT-READ ((SELF WEB-EDITOR)
                          MSG) ; 13-Aug-86 19:15
```

```
;; Prompt the user for some input, using an attached prompt window
(LET ((P-WINDOW (GET-PROMPT-WINDOW SELF))
      (IL:CLEARW P-WINDOW)
      (PROG1 (IL:|PromptRead| MSG P-WINDOW T)
              (IL:CLEARW P-WINDOW)
              (IL:DETACHWINDOW P-WINDOW)
              (IL:CLOSEW P-WINDOW))))
```

(DEFMETHOD **PROMPT-FOR-LIST** ((SELF WEB-EDITOR) PROMPT-STR INITIAL-STRING) ; 8-Apr-87 16:44

;;; Prompt user in prompt window for a list of words.

```
(LET ((P-WINDOW (GET-PROMPT-WINDOW SELF)))
      (IL:RESETFORM (IL:TTYDISPLAYSTREAM P-WINDOW)
                    (IL:CLEARW P-WINDOW)
                    (IL:TTYIN PROMPT-STR NIL NIL ' (IL:NORAISE)
                    NIL NIL INITIAL-STRING))))
```

(DEFMETHOD **PROMPT-FOR-STRING** ((SELF WEB-EDITOR) PROMPT-STR INITIAL-STR) ; 13-Aug-86 18:42

;;; Prompt user in prompt window for a string.

```
(LET ((P-WINDOW (GET-PROMPT-WINDOW SELF))
      VALUE)
      (IL:RESETFORM (IL:TTYDISPLAYSTREAM P-WINDOW)
                    (IL:CLEARW P-WINDOW)
                    (SETQ VALUE (IL:TTYIN PROMPT-STR NIL NIL ' (STRING IL:NORAISE)
                    NIL NIL INITIAL-STR))
                    (IL:CLEARW P-WINDOW))
      (REMOVE-PROMPT-WINDOW SELF)
      VALUE))
```

(DEFMETHOD **PROMPT-FOR-WORD** ((SELF WEB-EDITOR) &OPTIONAL PROMPT-STR INITIAL-WORD) ; 8-Apr-87 16:43

;;; Prompt user in prompt window for a word.

```
(CAR (PROMPT-FOR-LIST SELF PROMPT-STR INITIAL-WORD))
```

(DEFMACRO **MOVE-DOWN-P** () ' (OR (IL:KEYDOWNP 'IL:MOVE) (IL:SHIFTDOWNP 'IL:CTRL)))

(IL:DEFINEQ

**(WEB-WINDOW-AFTER-MOVE-FN** (LAMBDA (WINDOW)

; Edited 13-Jul-87 15:59 by Rao  
; 10-Apr-86 16:16

;;; The SimpleWindow AFTERMOVEFN

```
(LET ((W (IL:WINDOWPROP WINDOW 'WEB-EDITOR))
      (AND W (AFTER-MOVE W))))
```

**(WEB-WINDOW-BUTTON-EVENT-FN** (LAMBDA (WINDOW)

; Edited 13-Jul-87 13:38 by Rao  
; 11-Sep-86 13:50

```
(LET ((WINDOW-FOR-MENU (IL:WINDOWPROP WINDOW 'WEB-EDITOR))
      (DECLARE (IL:SPECVARS WINDOW-FOR-MENU))
      (IL:TOTOPW WINDOW)
      (BUTTON-EVENT-FN WINDOW-FOR-MENU))))
```

**(WEB-WINDOW-RESHAPE-FN**

(LAMBDA (WINDOW IL:|oldBitmapImage| IL:|oldRegion| IL:|oldScreenRegion|)

; Edited 12-Jun-87 15:56 by Rao  
; 9-May-86 10:07

;;; The RESHAPEFN for a Window

```
(LET ((IL:\w (IL:WINDOWPROP WINDOW 'WEB-EDITOR))
      (AND IL:\w (AFTER-RESHAPE IL:\w IL:|oldBitmapImage| IL:|oldRegion| IL:|oldScreenRegion|))))
```

**(WEB-WINDOW-CLOSE-FN**

(IL:LAMBDA (WINDOW)

; Edited 12-Jun-87 11:42 by Rao  
; Remove link back to LoopsWindow

```
(IL:WINDOWPROP WINDOW 'WEB-EDITOR NIL))
```

**(IL:|PromptRead|**

(IL:LAMBDA (PROMPT-STRING WINDOW SAME-LINE?)

; Edited 20-Jul-87 16:20 by Rao  
; Printout promptString in promptwindow and return value of  
; expression read there

```
(PROG (NEWVALUE)
      (IL:RESETLST
        (IL:RESETSAVE (IL:TTYDISPLAYSTREAM (OR WINDOW IL:PROMPTWINDOW))
```

```
(IL:RESETSAVE (IL:TTY.PROCESS (IL:THIS.PROCESS)))
(IL:CLRSPROMPT)
(IL:RESETSAVE (IL:PRINTLEVEL 4 3))
(IL:|printout| T PROMPT-STRING)
(IL:|if| SAME-LINE?
  IL:|then| (IL:|printout| T "> ")
  IL:|else| (IL:|printout| T T "> "))
(IL:CLEARBUF T T) ; clear tty buffer because it sometimes has stuff left.
(IL:ALLOW.BUTTON.EVENTS)
(IL:SETQ NEWVALUE (CAR (IL:ERSETQ (IL:TTYINREAD T T))))
(RETURN NEWVALUE)))
```

)

(IL:DEFINEQ

**(WEB-WINDOW-EXPAND-FN**

(LAMBDA (WINDOW)

; Edited 13-Nov-87 12:58 by Rao  
; 19-Feb-85 13:58

;; When a browser window is expanded, it should be recomputed

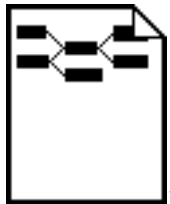
```
(LET ((SELF (IL:WINDOWPROP WINDOW 'WEB-EDITOR)))
  (RECOMPUTE-IN-PLACE SELF)))
```

)

**(DEFUN WEB-WINDOW-ICON-FN (WINDOW ICON DUMMY)**

```
(LET NIL (OR ICON (IL:TITLEDICONW *WEB-EDITOR-TEMPLATE* (ICON-TITLE (IL:WINDOWPROP WINDOW 'WEB-EDITOR))
  NIL
  '(0 . 0)
  T
  'IL:BOTTOM
  (IL:CONSTANT (LIST (IL:CHARCODE "-")
    (IL:CHARCODE IL:SPACE)
    (IL:CHARCODE IL:EOL)))))))
```

(IL:RPAQQ \*WEB-EDITOR-ICON-BM\*



(IL:RPAQQ \*WEB-EDITOR-ICON-MASK\*



```
(DEFVAR *WEB-EDITOR-TEMPLATE* (IL:|create| IL:TITLEDICON
  IL:ICON IL:_ *WEB-EDITOR-ICON-BM*
  IL:MASK IL:_ *WEB-EDITOR-ICON-MASK*
  IL:TITLEREG IL:_ (IL:CREATEREGION 5 2 50 30)))
```

(IL:RPAQ IL:\*D-WINDOW-DEFAULT-STREAM\* IL:PROMPTWINDOW)

(IL:RPAQ WEB-STREAM IL:PROMPTWINDOW)

;; Layout and Display Engine

(IL:RPAQQ IL:|BrowserMargin| 0)

(IL:RPAQQ IL:|MaxLatticeHeight| 750)

(IL:RPAQQ IL:|MaxLatticeWidth| 900)

(IL:DECLARE\ : IL:DOEVAL@COMPILE IL:DONTCOPY

(IL:SPECVARS IL:|MaxLatticeHeight| IL:|MaxLatticeWidth|)

(IL:RPAQQ IL:GRAYSHADE1 1)

(IL:RPAQQ IL:GRAYSHADE2 1025)

(IL:RPAQQ IL:GRAYSHADE3 64510)

(IL:RPAQQ IL:GRAYSHADE4 65534)

(IL:DEFINEQ

(TREE-ROOTS

(IL:LAMBDA (NODE-LST)

; Edited 10-Jul-87 19:22 by Rao  
; 29-Sep-86 19:46

:: Computes a minimal set of root nodes for a lattice --- those with no connections TO them in list of nodes, or a single node from a cycle of nodes.

(PROG ((ROOT-NODES (IL:LDIFFERENCE NODE-LST (IL:|for| IL:|node| IL:|in| NODE-LST IL:|join| (CHILD-NODES IL:|node|  
NODE-LST))))  
REACHABLE-NODES NOT-REACHABLE-NODES)  
(SETQ REACHABLE-NODES (IL:COPY ROOT-NODES))  
(SETQ NOT-REACHABLE-NODES (IL:LDIFFERENCE NODE-LST REACHABLE-NODES))

; recompute the nodes that can't be reached from the current  
; rootNodes

IL:|RecomputeReachableNodes|

:: Compute the transitive closure of the set of reachableNodes --- updating the notReachableNodes at the same time

(IL:|for| IL:|node| IL:|in| REACHABLE-NODES IL:|do| (IL:|for| IL:|childNode| IL:|in| (CHILD-NODES IL:|node|  
NODE-LST)  
IL:|when| (IL:MEMB IL:|childNode| NOT-REACHABLE-NODES)  
IL:|do|  
; put the newly found reachable node at the end of the list, so we will find it  
; later on during this iteration  
(IL:NCONC1 REACHABLE-NODES IL:|childNode|)  
(SETQ NOT-REACHABLE-NODES (IL:DREMOVE  
IL:|childNode|  
NOT-REACHABLE-NODES  
))))  
; if we can reach all the nodes, fine...

(IL:|if| (NULL NOT-REACHABLE-NODES)  
IL:|then|  
; Now need to prune down to a minimal set  
(IL:|bind| (IL:|stable?| IL:|\_NIL| IL:|until| IL:|stable?|  
IL:|do| (SETQ IL:|stable?| T)  
(IL:|for| IL:|node| IL:|in| ROOT-NODES IL:|bind| IL:|extraRoots|  
IL:|do| (SETQ IL:|extraRoots| (IL:DREMOVE IL:|node| (IL:INTERSECTION ROOT-NODES  
(REACHABLE-NODES!  
IL:|node| NODE-LST))))  
(IL:|if| IL:|extraRoots|  
IL:|then| (SETQ IL:|stable?| NIL)  
(SETQ ROOT-NODES (IL:LDIFFERENCE ROOT-NODES IL:|extraRoots|))  
(RETURN T))  
IL:|finally| (RETURN NIL)))  
(RETURN (IL:|for| IL:|node| IL:|in| ROOT-NODES IL:|collect| (IL:|fetch| IL:NODEID IL:|of| IL:|node|)))  
; return the node ids, not the GRAPHNODES  
; must be a cycle. Select the least prolific node in the cycle as  
; the a new root node.  
(IL:|push| ROOT-NODES (LET ((PROLIFIC-NODE (IL:|for| IL:|node| IL:|in| NOT-REACHABLE-NODES  
IL:|smallest| (IL:LENGTH (IL:|fetch| IL:TONODES  
IL:|of| IL:|node|))))  
(SETQ NOT-REACHABLE-NODES (IL:DREMOVE PROLIFIC-NODE  
NOT-REACHABLE-NODES)  
PROLIFIC-NODE))  
(GO IL:|RecomputeReachableNodes|))))))

(CHILD-NODES

(IL:LAMBDA (PARENT-NODE NODE-LIST)

; Edited 10-Jul-87 19:23 by Rao  
; 8-Oct-85 14:15  
; Find all GRAPHNODES that are immediatly reachable from this  
; node

(IL:|for| IL:|label| IL:|in| (IL:|fetch| IL:TONODES IL:|of| PARENT-NODE)  
IL:|collect| (IL:|for| IL:|node| IL:|in| NODE-LIST IL:|thereis| (EQ IL:|label| (IL:|fetch| IL:NODEID IL:|of| IL:|node|))))))

(REACHABLE-NODES!

(IL:LAMBDA (IL:|root| IL:|nodeList|)

; 30-Sep-86 10:22  
(IL:\* IL:\; IL:|Return| IL:\a IL:\list| IL:\of| IL:\all| IL:\nodes| IL:\that|

IL:|are| IL:|reachable| IL:|from| IL:|the| IL:|root|)  
(LET ((IL:|reachableNodes| (LIST IL:|root|)))  
(IL:|for| IL:|node| IL:|in| IL:|reachableNodes| IL:|do| (IL:|for| IL:|childNode| IL:|in| (CHILD-NODES IL:|node|  
IL:|nodeList|)  
IL:|when| (NOT (IL:MEMB IL:|childNode|  
IL:|reachableNodes|))  
IL:|do|

(IL:\* IL:\; IL:|put| IL:|the| IL:|newly| IL:|found| IL:|reachable| IL:|node| IL:|at| IL:|the| IL:|end| IL:|of| IL:|the| IL:|list| IL:|so| IL:|we|  
IL:|will| IL:|find| IL:|it| IL:|later| IL:|on| IL:|during| IL:|this| IL:|iteration|)

(IL:NCONC1 IL:|reachableNodes| IL:|childNode|  
))

IL:|reachableNodes|))

)

(DEFMETHOD **DISPLAY-BROWSER** ((SELF WEB-EDITOR)) ; 29-Sep-86 12:15  
; New method template

```
(LET ((NODELIST (AND (SLOT-VALUE SELF 'STARTING-LIST)
                    (GET-NODE-LIST SELF (SLOT-VALUE SELF 'STARTING-LIST)
                                        (SLOT-VALUE SELF 'GOOD-LIST))))))
  (COND
   (NODELIST (IL:SHOWGRAPH (IL:LAYOUTGRAPH NODELIST (TREE-ROOTS NODELIST)
                                                (SLOT-VALUE SELF 'GRAPH-FORMAT)
                                                (SLOT-VALUE SELF 'BROWSE-FONT))
                           (SLOT-VALUE SELF 'WINDOW)
                           NIL NIL (SLOT-VALUE SELF 'TOP-ALIGN)) ; kludge to reset the window props
             (SETUP-WINDOW SELF))
    (T (CLEAR SELF))))))
```

(DEFMETHOD **BROWSER-OBJECTS** ((IL:|self| WEB-EDITOR)) ; 28-May-84 12:58  
; Return a list of all the objects shown in the browser

```
(IL:|for| IL:|node| IL:|in| (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE IL:|self| 'WINDOW)
                                                                              'IL:GRAPH))
  IL:|when| (IL:NLISTP (CAR IL:|node|)) IL:|collect| (CAR IL:|node|)))
```

(DEFMETHOD **GET-NODE-LIST** ((SELF WEB-EDITOR) BROWSE-LIST GOOD-LIST) ; 21-Mar-85 14:09

;; Compute the node data structures of the tree starting at browseList. If goodList is given, only include elements of it. If goodList=T make it be  
;; browseList.

```
(DECLARE (IL:GLOBALVARS IL:WHITESHAD))
(COND
 (EQ GOOD-LIST T)
 (IL:SETQ GOOD-LIST BROWSE-LIST))
(PROG (SUBS PAIR NODE (OLD-NODES (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE SELF 'WINDOW)
                                                                              'IL:GRAPH))
                                (OBJ-LIST (CONS))))
```

;; first make objList which is a list of pairs (object . objName). objName will be used as a title for a node in the browser. This structure will be  
;; replaced by a graphNode when it is processed. The nodeID of the graphNode will be the object, and the label will be the name.

```
(IL:|for| IL:|objOrName| IL:|in| BROWSE-LIST IL:|do| (AND (IL:SETQ PAIR (OBJ-NAME-PAIR SELF IL:|objOrName|)
                                                         (NOT (IL:FASSOC (CAR PAIR)
                                                         (CAR OBJ-LIST))))
                                                         (IL:TCONC OBJ-LIST PAIR)))
```

;; Now MAP ON list so pair can be replaced by graphNode

```
(IL:|for| PAIR IL:|name| IL:|obj| IL:|subObjs| IL:|on| (CAR OBJ-LIST) IL:|when| (IL:NLISTP (IL:SETQ IL:|name|
                                                                                          (CDAR PAIR)))
  IL:|do| (IL:SETQ IL:|subObjs| (CONS))
  (IL:|for| IL:|sub| IL:|objPair| IL:|obj1| IL:|in| (GET-SUBS SELF (IL:SETQ IL:|obj| (CAAR PAIR)))
    IL:|do|
      ;; ObjNamePair returns NIL for destroyed objects. include only members of goodList in subs if given. Add to objList
      ;; only once
      (IL:SETQ IL:|obj1| (COND
                        ((EQ (CAR IL:|sub|)
                            'IL:|Link Parameters|)
                         (CADR IL:|sub|))
                        (T IL:|sub|)))
        (COND
         ((IL:SETQ IL:|objPair| (OBJ-NAME-PAIR SELF IL:|obj1|))
          (COND
           ((NOT (IL:FASSOC IL:|obj1| (CAR OBJ-LIST))
                (IL:TCONC OBJ-LIST IL:|objPair|)))
            (IL:TCONC IL:|subObjs| IL:|sub|)))
          (RPLACA PAIR (IL:SETQ NODE (OR (IL:FASSOC IL:|obj| OLD-NODES)
                                       (IL:|create| IL:GRAPHNODE
                                                    IL:NODEID IL:_ IL:|obj|
                                                    IL:NODEBORDER IL:_ (LIST (IL:ADD1 (SLOT-VALUE
                                                                                          SELF
                                                                                          'BOX-LINE-WIDTH))
                                                                                          IL:WHITESHAD))))))
          (IL:|replace| IL:TONODES IL:|of| NODE IL:|with| (CAR IL:|subObjs|))
          (IL:|replace| IL:NODELABEL IL:|of| NODE IL:|with| IL:|name|)
          (IL:|replace| IL:NODEFONT IL:|of| NODE IL:|with| (SLOT-VALUE SELF 'BROWSE-FONT))
          (IL:|replace| IL:NODEWIDTH IL:|of| NODE IL:|with| NIL)
          (IL:|replace| IL:NODEHEIGHT IL:|of| NODE IL:|with| NIL))
        (RETURN (CAR OBJ-LIST))))))
```

(DEFMETHOD **OBJ-NAME-PAIR** ((IL:|self| WEB-EDITOR) IL:|obj|)

;; Make a pair (object . objName) where objName is label to be used in browser

```
(LET NIL (IL:|if| (NULL IL:|obj|)
  IL:|then| NIL
  IL:|elseif| (AND (SLOT-VALUE IL:|self| 'GOOD-LIST)
```

```
(NOT (IL:FMEMB IL:|obj| (SLOT-VALUE IL:|self| 'GOOD-LIST)))
  IL:|then| NIL
  IL:|elseif| (IL:FMEMB IL:|obj| (SLOT-VALUE IL:|self| 'BAD-LIST))
  IL:|then| NIL
  IL:|else| (CONS IL:|obj| (GET-DISPLAY-LABEL IL:|self| IL:|obj|))))
```

(DEFMETHOD **GRAPH-FITS** ((|self| WEB-EDITOR)) ;24-Apr-86 15:00

;;; Tests if graph fits in region

```
(LET ((|window| (SLOT-VALUE |self| 'WINDOW))
      (LET ((|width| 0)
            (|height| 0)
            (|region| (IL:WINDOWPROP |window| 'IL:REGION))
            (|nodes| (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP |window| 'IL:GRAPH))))
      (COND
        (|nodes| (IL:SETQ |width| (IL:WIDTHIFWINDOW (IL:IDIFFERENCE (IL:MAX/RIGHT |nodes|
                                                                    (IL:MIN/LEFT |nodes|)
                                                                    (IL:WINDOWPROP |window| 'IL:BORDER)))
                                                    (IL:WINDOWPROP |window| 'IL:MAX/TOP |nodes|)
                                                    (IL:MIN/BOTTOM |nodes|)
                                                    (IL:WINDOWPROP |window| 'IL:TITLE)
                                                    (IL:WINDOWPROP |window| 'IL:BORDER))))))
        (NOT (OR (IL:IGREATERP |width| (IL:|fetch| IL:WIDTH IL:|of| |region|))
                 (IL:IGREATERP |height| (IL:|fetch| IL:HEIGHT IL:|of| |region|)))))))
```

(DEFMETHOD **NODE-REGION** ((IL:|self| WEB-EDITOR) IL:|object|) ;10-Dec-84 18:26

;; what region does the object occupy in the display stream?

```
(LET ((IL:|node| (IL:FASSOC (COND
                            ((IL:LITATOM IL:|object|)
                             (IL:SETQ IL:|object| (IL:|GetObjectRec| IL:|object|)))
                            (T IL:|object|)))
      (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE IL:|self| 'WINDOW)
                                                         'IL:GRAPH))))
      (IL:|if| IL:|node|
        IL:|then| (IL:|create| IL:REGION
                             IL:LEFT IL:_ (IL:IDIFFERENCE (IL:|fetch| IL:XCOORD IL:|of| (IL:|fetch| IL:NODEPOSITION
                                                                 IL:|of| IL:|node|))
                                                         (IL:IQUOTIENT (IL:|fetch| IL:NODEWIDTH IL:|of| IL:|node|)
                                                         2))
                             IL:BOTTOM IL:_ (IL:IDIFFERENCE (IL:|fetch| IL:YCOORD IL:|of| (IL:|fetch| IL:NODEPOSITION
                                                                 IL:|of| IL:|node|))
                                                         (IL:IQUOTIENT (IL:|fetch| IL:NODEHEIGHT IL:|of| IL:|node|)
                                                         2))
                             IL:WIDTH IL:_ (IL:|fetch| IL:NODEWIDTH IL:|of| IL:|node|)
                             IL:HEIGHT IL:_ (IL:|fetch| IL:NODEHEIGHT IL:|of| IL:|node|))))))
```

(DEFMETHOD **RECOMPUTE** ((SELF WEB-EDITOR) &OPTIONAL DONT-RESHAPE-FLG) ; 8-Apr-87 14:42  
; Recompute the browseGraph in the same window

```
(PROG ((GRAPH-FITS (GRAPH-FITS SELF))
      (DISPLAY-BROWSER SELF)
      (COND
        ((OR DONT-RESHAPE-FLG (NULL GRAPH-FITS)) ; Dont Reshape or rescroll. Assume window wants to stay the
        ; same size
        )
      (T (SHAPE-TO-HOLD SELF))))
  SELF)
```

(DEFMETHOD **RECOMPUTE-IN-PLACE** ((IL:|self| WEB-EDITOR)) ;10-Dec-84 18:27

;;; recompute the graph, maintaining the current position

```
(LET* ((IL:|visibleRegion| (IL:DSPCLIPPINGREGION NIL (SLOT-VALUE IL:|self| 'WINDOW))
       (IL:\x (IL:|fetch| IL:LEFT IL:|of| IL:|visibleRegion|))
       (IL:\y (IL:|fetch| IL:BOTTOM IL:|of| IL:|visibleRegion|))) ; if we want to RecomputeInPlace, we must want the window to
                                                                ; be kept the same
                                                                ; we had to save x and y because visibleRegion gets clobbered
                                                                ; by Recompute! Suprise!
      (RECOMPUTE IL:|self| T)
      (SCROLL-WINDOW IL:|self| IL:\x IL:\y)))
```

(DEFMETHOD **RECOMPUTE-LABELS** ((|self| WEB-EDITOR)) ;27-Feb-85 11:27  
; recompute the graph, including the labels

```
(CLEAR-LABEL-CACHE |self| T)
(RECOMPUTE |self|)
```

(DEFMETHOD **RECOMPUTE-IF-OPEN** ((WEB-EDITOR WEB-EDITOR)) ;27-Aug-86 12:37

```
(IF (IL:OPENWP (SLOT-VALUE WEB-EDITOR 'WINDOW))
```

(RECOMPUTE WEB-EDITOR))

(DEFMETHOD **CLEAR-LABEL-CACHE** ((WEB-EDITOR WEB-EDITOR) OBJECTS) ; 5-Dec-85 12:02

(LET (CACHED-LABEL)
;; Delete the cached label for these items
(COND
((EQ OBJECTS T)
(SETF (SLOT-VALUE WEB-EDITOR 'LABEL-CACHE)
NIL))
(T (IF (ATOM OBJECTS)
(SETQ OBJECTS (CONS OBJECTS)))
(DOLIST (OBJ OBJECTS)
(IF (SETQ CACHED-LABEL (IL:ASSOC OBJ (SLOT-VALUE WEB-EDITOR 'LABEL-CACHE)))
(SETF (SLOT-VALUE WEB-EDITOR 'LABEL-CACHE)
(IL:DREMOVE CACHED-LABEL (SLOT-VALUE WEB-EDITOR 'LABEL-CACHE))))))))))

(DEFMETHOD **OBJECT-FROM-LABEL** ((SELF WEB-EDITOR) LABEL) ; 4-Jan-85 18:20

;; What object has this label?
(LET ((OBJECT-NODE (IL:|for| IL:|node| IL:|in| (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE SELF 'WINDOW) 'IL:GRAPH))
IL:|thereis| (IL:EQUAL LABEL (IL:|fetch| IL:NODELABEL IL:|of| IL:|node|))))))
(IL:|if| (IL:NLISTP (CAR OBJECT-NODE))
IL:|then| (CAR OBJECT-NODE)
IL:|else| NIL)))

(DEFMETHOD **CHANGE-FONT-SIZE** ((WEB-EDITOR WEB-EDITOR) &OPTIONAL SIZE) ; 13-Dec-84 13:04
; Change the font size from whatever it is to size

(WHEN (OR SIZE (SETQ SIZE (IL:MENU (IL:|create| IL:MENU
IL:TITLE IL:\_ "Select Desired Size"
IL:CHANGEOFFSETFLG IL:\_ T
IL:ITEMS IL:\_ '("Abort" NIL)
8 10 12 16))))))
(SETF (SLOT-VALUE WEB-EDITOR 'BROWSE-FONT)
(IL:FONTCREATE '(, (SLOT-VALUE WEB-EDITOR 'BROWSE-FONT-FAMILY)
,SIZE
, (SLOT-VALUE WEB-EDITOR 'BROWSE-FONT-FACE))))
; clear out the label cache!
(RECOMPUTE-LABELS WEB-EDITOR))

(DEFMETHOD **CHANGE-FORMAT** ((|self| WEB-EDITOR) &OPTIONAL |format|) ; 21-Apr-84 19:52
; Change format between Lattice and Tree

(COND
((IL:LISTP |format|)
(SETF (SLOT-VALUE |self| 'GRAPH-FORMAT)
|format|))
((SETQ |format| (IL:MENU (IL:|create| IL:MENU
IL:ITEMS IL:\_ (SLOT-VALUE |self| 'GRAPH-FORMAT-CHOICES))))
(SETF (SLOT-VALUE |self| 'GRAPH-FORMAT)
|format|))
(RECOMPUTE |self|))

(DEFMETHOD **CHANGE-MAX-LABEL-SIZE** ((SELF WEB-EDITOR) NEW-MAX-WIDTH NEW-MAX-LINES) ; 13-Dec-84 13:05
; change the max label dimensions and redisplay the nodes -- if
; new size is NULL, don't change

(IL:|if| NEW-MAX-LINES
IL:|then| (SETF (SLOT-VALUE SELF 'LABEL-MAX-LINES)
NEW-MAX-LINES))
(IL:|if| NEW-MAX-WIDTH
IL:|then| (SETF (SLOT-VALUE SELF 'LABEL-MAX-CHARS-WIDTH)
NEW-MAX-WIDTH))
; clear out the label cache
(RECOMPUTE-LABELS SELF))

(DEFMETHOD **SHAPE-TO-HOLD** ((SELF WEB-EDITOR)) ; 13-Jan-87 16:52

;; Shape the browse window to just hold the nodes with BrowserMargin to spare
(LET\* ((WINDOW (SLOT-VALUE SELF 'WINDOW))
(REGION (IL:WINDOWPROP WINDOW 'IL:REGION))
(NODES (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP WINDOW 'IL:GRAPH)))
(MIN-WIDTH (IL:IPLUS 5 (IL:STRINGWIDTH (SLOT-VALUE SELF 'TITLE)
(IL:DSPFONT NIL IL:|WindowTitleDisplayStream|))))
(MIN-HEIGHT (IL:FONTHEIGHT (IL:DSPFONT NIL WINDOW)))
LEFT BOTTOM HEIGHT WIDTH RIGHT TOP)

```
(IF NODES
  (PROGN (SETQ LEFT (IL:MIN/LEFT NODES))
         (SETQ BOTTOM (IL:MIN/BOTTOM NODES))
         (SETQ RIGHT (IL:MAX/RIGHT NODES))
         (SETQ TOP (IL:MAX/TOP NODES))
         (SETQ WIDTH (IL:IMAX MIN-WIDTH (IL:IMIN IL:|MaxLatticeWidth| (IL:WIDTHIFWINDOW
                                                                           (IL:PLUS IL:|BrowserMargin|
                                                                           (IL:IDIFFERENCE RIGHT
                                                                           LEFT))
                                                                           (IL:WINDOWPROP WINDOW
                                                                           'IL:BORDER))))))
         (SETQ HEIGHT (IL:IMAX MIN-HEIGHT (IL:IMIN IL:|MaxLatticeHeight| (IL:PLUS IL:|BrowserMargin|
                                                                           (IL:IDIFFERENCE TOP
                                                                           BOTTOM))))))
         (UNLESS (AND (IL:EQP WIDTH (IL:|fetch| IL:WIDTH IL:|of| REGION))
                     (IL:EQP (IL:HEIGHTIFWINDOW HEIGHT (IL:WINDOWPROP WINDOW 'IL:TITLE)
                               (IL:WINDOWPROP WINDOW 'IL:BORDER))
                              (IL:|fetch| IL:HEIGHT IL:|of| REGION)))
                   (SET-REGION SELF (IL:CREATEREGION (IL:|fetch| IL:LEFT IL:|of| REGION)
                                                      (IL:|fetch| IL:BOTTOM IL:|of| REGION)
                                                      WIDTH HEIGHT)
                               NIL)))
  ;; ELSE
  (SET-REGION SELF (IL:CREATEREGION (IL:|fetch| IL:LEFT IL:|of| REGION)
                                     (IL:|fetch| IL:BOTTOM IL:|of| REGION)
                                     MIN-WIDTH MIN-HEIGHT))))
```

```
(DEFMETHOD GET-DISPLAY-LABEL ((SELF WEB-EDITOR)
                              OBJECT)
```

;;; get the display label. use the cache if it provides the answer; if not, and maxLabelWidth is set, use it to compute the appropriate bit map and then  
 ;;; cache the result.

```
(LET ((CACHED-LABEL (IL:ASSOC OBJECT (SLOT-VALUE SELF 'LABEL-CACHE)))
      (IF CACHED-LABEL
          (CDR CACHED-LABEL)
          (LET ((NEW-LABEL (BOX-PRINT-STRING (GET-LABEL SELF OBJECT)
                                             (SLOT-VALUE SELF 'LABEL-MAX-CHARS-WIDTH)
                                             (SLOT-VALUE SELF 'LABEL-MAX-LINES)
                                             (SLOT-VALUE SELF 'BROWSE-FONT))))
              (IL:|if| (IL:LISTP NEW-LABEL)
                      IL:|then| ; GRAPHER dies if the label is a list
                      (IL:SETQ NEW-LABEL (IL:MKSTRING NEW-LABEL)))
              (PUSH (CONS OBJECT NEW-LABEL)
                    (SLOT-VALUE SELF 'LABEL-CACHE))
              NEW-LABEL)))
```

```
(DEFMETHOD BOX-NODE ((SELF WEB-EDITOR)
                    OBJECT &OPTIONAL KEEP-PREVIOUS-BOX) ; 8-Apr-87 18:34
```

"Puts a box around the node in the graph representing the object"

;; If there was a previously boxed node, remove the box from around it and set it to nil

```
(WHEN (AND (NOT KEEP-PREVIOUS-BOX)
            DESTINATION-BROWSER
            (SLOT-VALUE DESTINATION-BROWSER 'BOXED-NODE))
  (HIGHLIGHT-NODE DESTINATION-BROWSER (SLOT-VALUE DESTINATION-BROWSER 'BOXED-NODE)
                 (SLOT-VALUE SELF 'BOX-LINE-WIDTH)
                 IL:WHITESHADE)
  (SETF (SLOT-VALUE DESTINATION-BROWSER 'BOXED-NODE)
        NIL))
(SETQ DESTINATION-BROWSER SELF) ; update the global
(HIGHLIGHT-NODE SELF OBJECT (SLOT-VALUE SELF 'BOX-LINE-WIDTH)
               IL:BLACKSHADE)
(SETF (SLOT-VALUE SELF 'BOXED-NODE)
      OBJECT))
```

```
(DEFMETHOD UNMARK-NODES ((IL:|self| WEB-EDITOR) ; 10-Dec-84 12:27
```

(REMOVE-HIGHLIGHTS IL:|self|) ; clear the graph nodes, removing all shading and highlighting  
 (REMOVE-SHADING IL:|self|))

```
(DEFMETHOD HIGHLIGHT-NODE ((SELF WEB-EDITOR)
                           OBJECT WIDTH SHADE) ; 13-Dec-85 15:16
```

;;; highlight a node by surrounding it with a shaded box

```
(LET ((NODE (IL:FASSOC OBJECT (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE SELF 'WINDOW)
                                                                           'IL:GRAPH))))
      (AND NODE (DISPLAY-NODE-HIGHTLIGHTS SELF NODE SHADE WIDTH))))
```

```
(DEFMETHOD SHADE-NODE ((IL:|self| WEB-EDITOR)
```



IL:|object| IL:|shade|) ; 15-Jan-87 18:34

:: shade the background of a node

```
(LET ((IL:|node| (IL:FASSOC (COND
  ((IL:LITATOM IL:|object|)
    (IL:SETQ IL:|object| (IL:|GetObjectRec| IL:|object|)))
  (T IL:|object|)))
  (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE IL:|self| 'WINDOW)
    'IL:GRAPH))))))
```

```
(IL:|if| IL:|node|
  IL:|then| (IL:|if| (IL:BITMAPP (IL:|fetch| IL:NODELABEL IL:|of| IL:|node|))
    IL:|then|
```

:: Need to forget the old bitmap, in case it already has a shade bl'ted into it. This will fail if the GetDisplayLabel  
:: msg returns something different from the previous value, but what can you do?

```
(CLEAR-LABEL-CACHE IL:|self| IL:|object|)
(LET ((IL:|newLabel| (GET-DISPLAY-LABEL IL:|self| IL:|object|)))
  (IL:|replace| IL:NODELABEL IL:|of| IL:|node| IL:|with| IL:|newLabel|)
  (IL:|if| (AND IL:|shade| (IL:BITMAPP IL:|newLabel|))
    IL:|then| (IL:BITBLT NIL NIL NIL IL:|newLabel| NIL NIL NIL NIL 'IL:TEXTURE
      'IL:PAINT IL:|shade|)))
(DISPLAY-NODE-SHADING IL:|self| IL:|node| IL:|shade|)))
```

```
(DEFMETHOD DISPLAY-NODE-HIGHTLIGHTS ((SELF WEB-EDITOR)
  NODE SHADE BOX-WIDTH)
```

```
(IL:RESET/NODE/BORDER NODE (COND
  (SHADE (LIST BOX-WIDTH SHADE))
  (T BOX-WIDTH))
(SLOT-VALUE SELF 'WINDOW))
```

```
(DEFMETHOD DISPLAY-NODE-SHADING ((SELF WEB-EDITOR)
  NODE SHADE) ; 13-Dec-85 15:13
  ; New method template
```

```
(IL:RESET/NODE/LABELSHADE NODE (OR SHADE IL:WHITESHAE)
(SLOT-VALUE SELF 'WINDOW))
```

```
(DEFMETHOD REMOVE-HIGHLIGHTS ((IL:|self| WEB-EDITOR)) ; 13-Dec-85 15:16
```

:: gets rid of all highlighting in the lattice

```
(IL:|for| IL:|node| IL:|in| (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE IL:|self| 'WINDOW)
  'IL:GRAPH))
  IL:|do| (DISPLAY-NODE-HIGHTLIGHTS IL:|self| IL:|node| NIL))
(SETF (SLOT-VALUE IL:|self| 'BOXED-NODE)
  NIL))
```

```
(DEFMETHOD REMOVE-SHADING ((IL:|self| WEB-EDITOR)) ; 13-Dec-85 15:14
```

:: gets rid of all shading in the lattice

```
(IL:|for| IL:|node| IL:|in| (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE IL:|self| 'WINDOW)
  'IL:GRAPH))
  IL:|do| (DISPLAY-NODE-SHADING IL:|self| IL:|node| IL:WHITESHAE)))
```

```
(DEFMETHOD FLASH-NODE ((IL:|self| WEB-EDITOR)
  IL:|node| IL:N IL:|flashTime| IL:|leaveFlipped?|)
  ; 12-Dec-84 16:09
  ; Flip node N times
```

```
(IL:SETQ IL:|node| (IL:FASSOC (COND
  ((IL:LITATOM IL:|node|)
    (IL:SETQ IL:|node| (IL:|GetObjectRec| IL:|node|)))
  (T IL:|node|)))
  (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE IL:|self| 'WINDOW)
    'IL:GRAPH))))
```

```
(IL:|if| IL:|node|
  IL:|then| (IL:|for| IL:\i IL:|from| 1 IL:|to| (OR IL:N 3) IL:|do| (IL:FLIPNODE IL:|node| (SLOT-VALUE IL:|self|
    'WINDOW))
    (IL:DISMISS (OR IL:|flashTime| 300))
    (IL:FLIPNODE IL:|node| (SLOT-VALUE IL:|self|
    'WINDOW))
    (IL:DISMISS (OR IL:|flashTime| 300)))
  (IL:|if| IL:|leaveFlipped?|
    IL:|then| (IL:FLIPNODE IL:|node| (SLOT-VALUE IL:|self| 'WINDOW))))
```

```
(DEFMETHOD FLIP-NODE ((SELF WEB-EDITOR)
  OBJECT) ; 13-Dec-85 15:18
  ; Inverts the video around the node in the graph representing the
  ; object
```

```
(LET ((NODE (IL:FASSOC OBJECT (IL:|fetch| IL:GRAPHNODES IL:|of| (IL:WINDOWPROP (SLOT-VALUE SELF 'WINDOW)
  'IL:GRAPH))))))
  (AND NODE (DISPLAY-NODE-SHADING SELF NODE (IL:INVERTED/SHADE/FOR/GRAPHER (IL:|fetch| IL:NODELABELSHADE
```

IL:|of| NODE)))))

(DEFMETHOD POSITION-NODE ((SELF WEB-EDITOR)
OBJECT WINDOW-X WINDOW-Y) ; 10-Dec-84 18:24

;;; scrolls the window so that the node is in the given position of the window. If windowX or windowY is a FLOATP, it taken to be a window-relative
;;; postion; if a FIXP, it is a window-absolute position.

(LET ((REGION (NODE-REGION SELF OBJECT)))
(IL:|if| REGION
IL:|then| (SCROLL-WINDOW SELF (IL:|fetch| IL:LEFT IL:|of| REGION)
(IL:|fetch| IL:BOTTOM IL:|of| REGION)
WINDOW-X WINDOW-Y)))

(IL:DEFINEQ

(BOX-PRINT-STRING

(IL:LAMBDA (STRING MAX-CHARS-WIDTH MAX-LINES FONT OLD-BITMAP) ; Edited 29-Jan-88 15:06 by Rao

;; return a bitmap containing the string, in the given font, with MAX-WIDTH at most width ; sizes of NULL or 0 mean no max size

(IL:SETQ MAX-CHARS-WIDTH (OR MAX-CHARS-WIDTH 0))
(IL:SETQ MAX-LINES (OR MAX-LINES 0))
(IL:|if| (IL:ZEROP MAX-CHARS-WIDTH) ; no max width, then just return the STRING
IL:|then|
STRING

IL:|else| (PROG ((MAX-WIDTH (IL:ITIMES MAX-CHARS-WIDTH (IL:STRINGWIDTH "A" FONT)))
(NCHARS (IL:NCHARS STRING))
(NLINES 0)
(SPOS 0)
(REGION (IL:CONSTANT (IL:|create| IL:REGION)))
(TRUE-MAX-WIDTH 0)
(NEXTPOS DSP SUBSTR)
(IL:SETQ STRING (IL:MKSTRING STRING)) ; we need to find the size of the resultant bitmap
IL:NEXTBREAK
(IL:|if| (IL:ILESSP SPOS NCHARS)
IL:|then| (IL:|add| NLINES 1) ; at least one character, even if exceed MAX-WIDTH
(IL:SETQ NEXTPOS (IL:IMAX 1 (CAR (BREAK-STRING-FOR-BOXING (IL:SUBSTRING
STRING
(IL:ADD1 SPOS)
-1)
MAX-WIDTH FONT))))
(IL:SETQ TRUE-MAX-WIDTH (IL:IMAX TRUE-MAX-WIDTH (IL:STRINGWIDTH (IL:SUBSTRING
STRING
(IL:ADD1 SPOS)
(IL:IPLUS SPOS
NEXTPOS))
FONT)))
(IL:|add| SPOS NEXTPOS)
(GO IL:NEXTBREAK))
(IL:|if| (NOT (IL:ZEROP MAX-LINES))
IL:|then| (IL:SETQ NLINES (IL:IMIN MAX-LINES NLINES))) ; that we have the size, lets build it
(IL:SETQ DSP (IL:DSPCREATE
(IL:|if| (AND OLD-BITMAP (NOT (OR (IL:GREATERP TRUE-MAX-WIDTH (IL:BITMAPWIDTH
OLD-BITMAP))
(IL:GREATERP (IL:ITIMES NLINES
(IL:FONTPROP FONT
'IL:HEIGHT))
(IL:BITMAPHEIGHT OLD-BITMAP))))))
IL:|then| OLD-BITMAP
IL:|else| (IL:BITMAPCREATE TRUE-MAX-WIDTH (IL:ITIMES NLINES (IL:FONTPROP
FONT
'IL:HEIGHT))))))
(IL:DSPFONT FONT DSP)
(IL:DSPRESET DSP)
(IL:SETQ SPOS 0)
(IL:|replace| IL:LEFT IL:|of| REGION IL:|with| 0)
(IL:|replace| IL:WIDTH IL:|of| REGION IL:|with| TRUE-MAX-WIDTH)
(IL:|replace| IL:HEIGHT IL:|of| REGION IL:|with| (IL:FONTPROP FONT 'IL:HEIGHT))
(IL:|replace| IL:BOTTOM IL:|of| REGION IL:|with| (IL:ITIMES NLINES (IL:FONTPROP FONT 'IL:HEIGHT)))
IL:NEXTPIECE
(IL:|add| NLINES -1)
(IL:|if| (IL:ILESSP SPOS NCHARS)
IL:|then| (IL:SETQ NEXTPOS (IL:IMAX 1 (CAR (BREAK-STRING-FOR-BOXING (IL:SUBSTRING
STRING
(IL:ADD1 SPOS)
-1)
TRUE-MAX-WIDTH FONT))))
(IL:SETQ SUBSTR (IL:SUBSTRING STRING (IL:ADD1 SPOS)
(IL:IPLUS NEXTPOS SPOS)))
(IL:|replace| IL:BOTTOM IL:|of| REGION IL:|with| (IL:IDIFFERENCE (IL:|fetch| IL:BOTTOM
IL:|of| REGION)
(IL:|fetch| IL:HEIGHT IL:|of| REGION)))
(IL:|if| (AND (IL:ZEROP NLINES)
(IL:ILESSP (IL:IPLUS NEXTPOS SPOS)
NCHARS))

```

IL:|then| ; we need to abbreviate!
(IL: CENTERPRINTINREGION (IL: CONCAT (IL: SUBSTRING SUBSTR 1 -3)
"..."
REGION DSP)
(GO IL: ALLDONE)
IL:|else| ; out this piece
(IL: CENTERPRINTINREGION SUBSTR REGION DSP)
(IL:|add| SPOS NEXTPOS)
(GO IL: NEXTPIECE))
IL: ALLDONE
(RETURN (IL: DSPDESTINATION NIL DSP))))))

```

**BREAK-STRING-FOR-BOXING**

(IL: LAMBDA (IL: MSG IL: WIDTH IL: FONT) ; 11-Dec-84 10:29

```

(IL: * IL: \; IL: |Stolen| IL: |from| IL: |the| IL: |function| IL: |CONW.FORMATLINE IL: --
IL: |modified| IL: |to| IL: |try| IL: |to| IL: |break| IL: |at| "word" IL: |boundaries,| IL: |whatever| IL: |they| IL: |are|)

(IL: * IL: \; IL: \a IL: |list| IL: |of| IL: |the| IL: |char#| IL: |relative| IL: |to| IL: |char| 1 IL: |of| IL: |where| IL: |to| IL: |break| IL: |next| IL: |line,|
IL: |and| IL: |how| IL: |much| IL: |space| IL: |was| LEFT IL: |over| (IL: |for| IL: |centering| IL: &c))

```

```

(COND
(IL: MSG
(IL: * IL: \; IL: |there| IL: |really| IL: |is| IL: \a IL: |title,| IL: |go|
IL: |ahead| IL: |and| IL: |format| IL: |the| IL: |next| IL: |line.))

(IL: |bind| (IL: TX IL: _ 0)
(IL: LASTB IL: _ 0)
(IL: CH IL: _ 0)
(IL: TMSG IL: _ (IL: OPENSTRINGSTREAM IL: MSG))
(IL: MSGLEN IL: _ (IL: NCHARS IL: MSG)) IL: |for| IL: I IL: |from| 1 IL: |by| 1
(IL: * IL: \; IL: |thru| IL: |the| IL: |characters| IL: |one| IL: |by| IL: |one.))

IL: |do|
(COND
((IL: IGREATERP IL: TX IL: WIDTH)
(IL: * IL: \; IL: |past| IL: |the| IL: |right| IL: |margin,|
IL: |Time| IL: |to| IL: |stop.))

(IL: CLOSEF? IL: TMSG)
(RETURN (COND
((IL: LISTP IL: LASTB)
(IL: * IL: \; IL: |is| IL: \a IL: |space| IL: |we| IL: |can| IL: |break| IL: |the|
IL: |line| IL: |at,| IL: |Break| IL: |there.))

IL: LASTB)
(T

```

```

(IL: * IL: \; IL: |were| IL: |no| IL: |spaces| IL: |on| IL: |this| IL: |line,| IL: |Break| IL: |after| IL: |the| IL: |last| IL: |character| IL: |that| IL: |did|
IL: |fit.))

```

```

(CONS (IL: IDIFFERENCE IL: I 2)
(IL: IDIFFERENCE IL: WIDTH (IL: IDIFFERENCE IL: TX (IL: CHARWIDTH
IL: CH IL: FONT))))
)))
((IL: EOFP IL: TMSG)
(IL: CLOSEF? IL: TMSG)
(RETURN (CONS (IL: SUB1 IL: I)
(IL: IDIFFERENCE IL: WIDTH IL: TX))))
(T
(IL: SETQ IL: CH (IL: BIN IL: TMSG))
(IL: SELCHARQ IL: CH
((IL: SPACE IL: \. IL: \: IL: \; IL: \, / IL: \ \ IL: * - IL: \#)

```

```

(IL: * IL: \; IL: |where| IL: |word| IL: |breaks| IL: |are,| IL: |so| IL: |we| IL: |can| IL: |back| IL: |up| IL: |and| IL: |split| IL: |lines| IL: |there|
IL: |if| IL: |possible.))

```

```

(IL: SETQ IL: LASTB (CONS IL: I (IL: IDIFFERENCE IL: WIDTH IL: TX))))
(IL: CR
(RETURN (CONS (IL: IMINUS IL: I)
(IL: IDIFFERENCE IL: WIDTH IL: TX))))
(IL: |if| (AND (NOT (IL: U-CASEP (IL: CHARACTER IL: CH)))
(NOT (IL: EOFP IL: TMSG))
(IL: U-CASEP (IL: PEEKC IL: TMSG)))
IL: |then|
(IL: * IL: \; IL: |from| IL: |upper| IL: |to| IL: |lower| IL: |case| IL: |is|
IL: |also| IL: \a IL: |word| IL: |break|)
(IL: SETQ IL: LASTB (CONS IL: I (IL: IDIFFERENCE IL: WIDTH IL: TX))))))
(T
(IL: SETQ IL: TX (IL: IPLUS IL: TX (IL: CHARWIDTH IL: CH IL: FONT))))))
(IL: * IL: \; IL: |isn't| IL: \a IL: |title;| IL: |return| IL: \a IL: |dummy|
IL: |entry| IL: |for| IL: |the| IL: |line| IL: |formatter.))
(CONS 0 IL: WIDTH))))))

```

**BOX-WINDOW-NODE**

(IL: LAMBDA (IL: |nodeLabel| WINDOW) ; Edited 29-Jan-88 11:31 by Rao ; 7-Sep-84 14:36

;; a box around the node with nodeLabel in the graph. A nodeLabel in browsers is an object. Does nothing if node not found.

```

(PROG (IL: |node| IL: |nodes|)
(COND
(AND (IL: WINDOWP WINDOW)
(IL: SETQ IL: |nodes| (IL: |fetch| IL: GRAPHNODES IL: |of| (IL: WINDOWPROP WINDOW 'IL: GRAPH)))
(IL: SETQ IL: |node| (IL: FASSOC IL: |nodeLabel| IL: |nodes|)))
(IL: DRAWAREABOX (IL: GN/LEFT IL: |node|)

```

```
(IL:GN/BOTTOM IL:|node|)
(IL:|fetch| IL:NODEWIDTH IL:|of| IL:|node|)
(IL:|fetch| IL:NODEHEIGHT IL:|of| IL:|node|)
1
' IL:INVERT WINDOW)))))
```

)

:: Button Events

(IL:DEFINEQ

**(FIND-SELECTED-NODE**

(IL:LAMBDA (WINDOW) ; Edited 12-Nov-87 01:30 by Rao  
; 10-Dec-84 17:53

:: Used in BUTTONEVENTFN and gets called whenever cursor moves or button is down. Adapted from APPLYTOSELECTEDNODE in GRAPHER package; returns the selected item rather than applying a function on the inside of the button event fn.

:: Also this was modified to pop up the middle button menu on button down rather than button up.

```
(PROG ((LOOPS-WINDOW (IL:WINDOWPROP WINDOW 'WEB-EDITOR))
(NODELST (IL:|fetch| (IL:GRAPH IL:GRAPHNODES) IL:|of| (IL:WINDOWPROP WINDOW 'IL:GRAPH)))
(DS (IL:WINDOWPROP WINDOW 'IL:DSP))
BUTTON OLDPOS REG NOW NEAR) ; note which button is down.
; get the region of this window.
(IL:SETQ REG (IL:WINDOWPROP WINDOW 'IL:REGION))
(IL:|until| (IL:LASTMOUSESTATE (OR IL:LEFT IL:MIDDLE)) IL:|do| (IL:GETMOUSESTATE))
(IL:SETQ NEAR (IL:NODELST/AS/MENU NODELST (IL:SETQ OLDPOS (IL:CORSORPOSITION NIL DS))))
IL:FLIP
```

:: This is kirk's quick hack to get middle button to bring up immediately.

```
(WHEN (IL:LASTMOUSESTATE IL:MIDDLE)
(RETURN (IL:|fetch| IL:NODEID IL:|of| NEAR)))
(AND NOW (IL:FLIPNODE NOW DS))
(AND NEAR (IL:FLIPNODE NEAR DS))
(IL:SETQ NOW NEAR)
IL:LP
; wait for a button up or move out of region
(IL:GETMOUSESTATE)
(COND
((IL:LASTMOUSESTATE (AND (NOT IL:LEFT)
(NOT IL:MIDDLE))) ; left button up, process it.
; NOW node has been selected.
(AND NOW (IL:FLIPNODE NOW DS))
(RETURN (IL:|fetch| IL:NODEID IL:|of| NOW)))
((NOT (IL:INSIDE? (IL:WINDOWPROP WINDOW 'IL:REGION)
IL:LASTMOUSEX IL:LASTMOUSEY)) ; outside of region, return
(AND NOW (IL:FLIPNODE NOW DS))
(RETURN))
((EQ NOW (IL:SETQ NEAR (IL:NODELST/AS/MENU NODELST (IL:CORSORPOSITION NIL DS OLDPOS))))
(GO IL:LP))
(T (GO IL:FLIP))))))
```

)

(DEFMETHOD **BUTTON-EVENT-FN** ((SELF WEB-EDITOR))

; 2-Jan-86 16:41  
; Called when there is a button event in a Loops Window

```
(LET ((WINDOW (SLOT-VALUE SELF 'WINDOW))
(OR (IL:ERSETQ (COND
((NULL (IL:INSIDEP (IL:DSPCLIPPINGREGION NIL WINDOW)
(IL:LASTMOUSEX WINDOW)
(IL:LASTMOUSEY WINDOW)))
(TITLE-SELECTION SELF))
((IL:MOUSESTATE IL:LEFT)
(LEFT-SELECTION SELF))
((IL:MOUSESTATE IL:MIDDLE)
(MIDDLE-SELECTION SELF))
((IL:MOUSESTATE IL:RIGHT)
(RIGHT-SELECTION SELF)))))))
```

(DEFMETHOD **LEFT-SELECTION** ((SELF WEB-EDITOR))

```
(IF (MOVE-DOWN-P)
(IF (SLOT-VALUE SELF 'NODE-MOVER-P)
(NODE-MOVE SELF)
(NODE-MOVE-SHALLOW SELF))
(NODE-SELECTION SELF 'IL:LEFT)))
```

(DEFMETHOD **MIDDLE-SELECTION** ((SELF WEB-EDITOR))

; 15-May-85 19:04

:: This function called from the GRAPHER package when a node is selected with the middle mouse button. If no node is selected then just returns.

```
(PROG (SELECTION OBJECT (WINDOW (SLOT-VALUE SELF 'WINDOW))
(WEB-EDITOR SELF))
(DECLARE (IL:SPECVARS OBJECT WEB-EDITOR))
(COND
```

```

((NULL (IL:SETQ OBJECT (FIND-SELECTED-NODE WINDOW)))
 (RETURN)))
(SETF (SLOT-VALUE WEB-EDITOR 'LAST-SELECTED-OBJECT)
      OBJECT)
(IL:GETMOUSESTATE)
(FLIP-NODE SELF OBJECT)
(IL:SETQ SELECTION (OR (NODE-ACTION SELF OBJECT 'IL:MIDDLE)
                       (PROGN (FLIP-NODE SELF OBJECT)
                              (RETURN NIL))))
(FLIP-NODE SELF OBJECT)
(DO-SELECTED-COMMAND WEB-EDITOR SELECTION OBJECT))

```

```

(DEFMETHOD RIGHT-SELECTION ((SELF WEB-EDITOR)) ; 17-Apr-84 15:46
; Do RightButtonItems on selection.
(LET* ((CHOICE (CHOICE-MENU SELF 'RIGHT-BUTTON-ITEMS)))
      (IF CHOICE (FUNCALL CHOICE SELF))))

```

```

(DEFMETHOD TITLE-SELECTION ((SELF WEB-EDITOR)) ; 17-Apr-84 15:35
;; Do TitleItems if selected in title area. Replaces TitleSelection in Window because this one does evaluation in TTY process, and saves events on
;; history
(LET* ((CHOICE (CHOICE-MENU SELF 'TITLE-ITEMS)))
      (IF CHOICE (FUNCALL CHOICE SELF))))

```

```

(DEFMETHOD NODE-SELECTION ((SELF WEB-EDITOR)
                           BUTTON)
(LET* ((WINDOW (SLOT-VALUE SELF 'WINDOW))
      (OBJECT (FIND-SELECTED-NODE WINDOW)))
      (DECLARE (IL:SPECVARS OBJECT)) ; SPECVARS for whenHeldFn
      (IF (LISTP OBJECT)
          (SETQ OBJECT (CAR OBJECT)))
      (COND
       ((NOT (NULL OBJECT))
        (SETF (SLOT-VALUE SELF 'LAST-SELECTED-OBJECT)
              OBJECT)))
      (IL:GETMOUSESTATE)
      (WHEN OBJECT
       (LET ((SELECTOR (NODE-ACTION SELF OBJECT BUTTON)))
           (COND
            (SELECTOR (DO-SELECTED-COMMAND SELF SELECTOR OBJECT)))))))

```

```

(DEFMETHOD NODE-ACTION ((SELF WEB-EDITOR)
                        NODE BUTTON) ; 8-Apr-87 17:11
(DECLARE (IL:SPECVARS WINDOW-FOR-MENU))
(LET ((WINDOW-FOR-MENU SELF)
      (IL:GETMOUSESTATE)
      (CHOICE-MENU SELF
                    ;; A Hook for letting nodes tailor menu items.
                    (NODE-MENU-ITEMS NODE BUTTON))))

```

```

(DEFMETHOD NODE-MENU-ITEMS ((NODE WEB-NODE)
                             BUTTON)
(CASE BUTTON
 (IL:LEFT 'LEFT-BUTTON-ITEMS)
 (IL:MIDDLE 'MIDDLE-BUTTON-ITEMS))

```

```

(DEFMETHOD CHOICE-MENU ((SELF WEB-EDITOR)
                        ITEM-CV) ; 29-Dec-85 13:54
;; Create a menu which allows subitems to be displayed. Cache it in the web-editor
(LET (ITEMS MENU)
      (SETQ MENU (REST (ASSOC ITEM-CV (SLOT-VALUE SELF 'MENU-CACHE))))
      (COND
       ((AND MENU (IL:TYPE? IL:MENU MENU))
        (IL:MENU MENU))
       ((NOT (LISTP (SETQ ITEMS (GET-MENU-ITEMS SELF ITEM-CV))))
        ITEMS)
       (T (IL:SETQ MENU (IL:CREATE IL:MENU
                                   IL:ITEMS IL:_ ITEMS
                                   IL:MENUOFFSET IL:_ (IL:CREATEPOSITION -1 0)
                                   IL:WHENSELECTEDFN IL:_ 'WEB-MENU-WHENSELECTEDFN
                                   IL:WHENHELDFN IL:_ 'WINDOW-WHEN-HELD-FN
                                   IL:CHANGEOFFSETFLG IL:_ T
                                   IL:CENTERFLG IL:_ T)) ; Cache menu if menus is T
        (IF (SLOT-VALUE SELF 'CACHE-MENU-P)
            (SETF (SLOT-VALUE SELF 'MENU-CACHE)
                  (ACONS ITEM-CV MENU (SLOT-VALUE SELF 'MENU-CACHE))))
        (IL:MENU MENU))))

```

(DEFMETHOD **DO-SELECTED-COMMAND** ((WEB-EDITOR WEB-EDITOR)  
COMMAND OBJ &OPTIONAL NODE)  
; 17-Sep-86 17:49

:: Do the selected command or forwards it to the object

```
(IF COMMAND
  ;; Take care of being passed in a dummy node from browser in Lattice mode. --- Dummy nodes are indicated by having the object in a list
  (LET ((ARGS (IF (IL:LISTP COMMAND)
                  (CDR COMMAND)
                  NIL))
        (COMMAND (IF (IL:LISTP COMMAND)
                     (CAR COMMAND)
                     COMMAND))
        (OBJ (IF (IL:LISTP OBJ)
                 (CAR OBJ)
                 OBJ)))
    (WHEN (IL:FMEMB COMMAND (SLOT-VALUE WEB-EDITOR 'LOCAL-COMMANDS))
      (SETQ ARGS (CONS OBJ ARGS))
      (SETQ OBJ WEB-EDITOR))
    ;; Grays out the node at the beginning of the command, and ungrays it when the command completes.
    (SETQ NODE OBJ)
    (IF NODE
      (PROGN (SHADE-NODE WEB-EDITOR NODE IL:GRAYSHADE2)
             (IL:BLOCK 500)
             (SHADE-NODE WEB-EDITOR NODE IL:WHITESHAE)
             (APPLY COMMAND OBJ ARGS))
            (APPLY COMMAND OBJ ARGS))))
```

(DEFMETHOD **WHEN-MENU-ITEM-HELD** ((SELF WEB-EDITOR)  
ITEM MENU KEY)  
; 8-Apr-87 17:13

::: What to do when the menu item is held

```
(IL:PROMPTPRINT (OR (COND
  ((IL:NLISTP ITEM)
   NIL)
  (T (CADDR ITEM))))
  "When released this item will be selected"))
```

(DEFMETHOD **ITEM-MENU** ((SELF WEB-EDITOR)  
ITEMS TITLE)  
; 21-Apr-84 09:31  
; Create a simple (one level) menu which will not overflow  
; height of screen

```
(IL:|create| IL:MENU
  IL:ITEMS IL:_ ITEMS
  IL:MENU COLUMNS IL:_ (IL:ADD1 (IL:IQUOTIENT (IL:ITIMES (IL:FONTHEIGHT IL:MENUFONT)
                                                (IL:LENGTH ITEMS))
                                         750))
  IL:TITLE IL:_ TITLE
  IL:CHANGE OFFSET FLG IL:_ T))
```

(DEFMETHOD **GET-MENU-ITEMS** ((SELF WEB-EDITOR)  
ITEM-CV)  
; 23-Oct-84 12:36  
; Get item list for menu

```
(SLOT-VALUE SELF ITEM-CV))
```

(DEFMETHOD **CLEAR-MENU-CACHE** ((SELF WEB-EDITOR))  
; 11-Apr-86 14:46  
; Delete Menus saved on menus

```
(SETF (SLOT-VALUE SELF 'MENU-CACHE)
      NIL)
SELF)
```

(IL:DEFINEQ

(**WEB-MENU-WHENSELECTEDFN**  
(IL:LAMBDA (ITEM MENU BUTTON)  
; Edited 14-Jul-87 17:43 by Rao  
; 13-DEC-83 21:03

```
(PROG (SECOND-ELEMENT)
  (RETURN (COND
    ((IL:NLISTP ITEM)
     ITEM)
    ((IL:NLISTP (IL:SETQ SECOND-ELEMENT (CADR ITEM)))
     SECOND-ELEMENT)
    ((EQ (CAR SECOND-ELEMENT)
         'PROGN)
     (IL:EVAL SECOND-ELEMENT))
    (T SECOND-ELEMENT))))))
```

(**WINDOW-WHEN-HELD-FN**  
(LAMBDA (ITEM MENU KEY)  
; Edited 9-Jul-87 11:58 by Rao

; 29-Dec-85 15:28  
; Send to window the message to respond to time out on menu

(**DECLARE** (IL:SPECVARS WINDOW-FOR-MENU)  
(WHEN-MENU-ITEM-HELD WINDOW-FOR-MENU ITEM MENU KEY))

)

(IL:DEFINEQ

**(SUB-ITEM-SELECTION**

(IL:LAMBDA (ITEM MENU BUTTON)

; Edited 14-Jul-87 17:13 by Rao  
; 13-DEC-83 21:03

;; menu WHENSELECTEDFN which allows differential selection on LEFT and middle button. For such differential selection item should be of form  
;; --- (itemSeenInMenu (leftValue midValue)) --- where midValue can be an atom which is directly returned when item is selected with middle, or  
;; midValue can be an itemList, which will be displayed in a subselection menu

(PROG (IT IT1)  
  (RETURN (COND  
    ((IL:NLISTP ITEM)  
      ITEM)  
    ((IL:NLISTP (IL:SETQ IT (CADR ITEM)))  
      IT)  
    (EQ (IL:SETQ IT1 (CAR IT))  
       'QUOTE)  
      (CADR IT))  
    (EQ IT1 'PROGN)  
    (IL:EVAL IT))  
    ((IL:LISTP IT1)  
      (IL:EVAL IT1))  
    (T IT1))))))

**(DUAL-SUB-ITEMS**

(IL:LAMBDA (MENU ITEM)

; Edited 14-Jul-87 17:14 by Rao  
; 13-DEC-83 21:07

;; menu WHENSELECTEDFN which allows differential selection on LEFT and middle button. For such differential selection item should be of form  
;; --- (itemSeenInMenu (leftValue midValue)) --- where midValue can be an atom which is directly returned when item is selected with middle, or  
;; midValue can be an itemList, which will be displayed in a subselection menu

(PROG (IT IT1)  
  (RETURN (COND  
    ((OR (IL:NLISTP ITEM)  
          (IL:NLISTP (IL:SETQ IT (CADR ITEM)))  
          (EQ (IL:SETQ IT1 (CAR IT))  
              'QUOTE)  
          (EQ IT1 'PROGN)  
          (IL:NLISTP (IL:SETQ IT1 (CADR IT))))  
    NIL)  
    (T IT1))))))

**(WINDOW-WHEN-HELD-FN**

(LAMBDA (ITEM MENU KEY)

; Edited 9-Jul-87 11:58 by Rao  
; 29-Dec-85 15:28  
; Send to window the message to respond to time out on menu

(**DECLARE** (IL:SPECVARS WINDOW-FOR-MENU)  
(WHEN-MENU-ITEM-HELD WINDOW-FOR-MENU ITEM MENU KEY))

**(DO-MENU-METHOD**

(IL:LAMBDA (OBJECT ITEMS)

; Edited 14-Jul-87 17:15 by Rao  
; 13-NOV-83 16:20

(PROG ((SELECTOR (AND ITEMS (**DUAL-MENU** ITEMS))))  
  (AND SELECTOR (RETURN (FUNCALL SELECTOR OBJECT))))))

**(DUAL-MENU**

(IL:LAMBDA (ITEMS WHEN-HELD-FN)

; Edited 14-Jul-87 17:16 by Rao  
; 9-FEB-84 16:17  
; and pops up a menu which allows differential selection on LEFT  
; and a middle buttons

(IL:MENU (IL:**create**) IL:MENU  
  IL:ITEMS IL:\_ ITEMS  
  IL:WHENSELECTEDFN IL:\_ 'SUB-ITEM-SELECTION  
  IL:SUBITEMFN IL:\_ 'DUAL-SUB-ITEMS  
  IL:WHENHELD FN IL:\_ WHEN-HELD-FN  
  IL:CHANGEOFFSETFLG IL:\_ T)))

**(DUAL-SELECTION**

(IL:LAMBDA (ITEM MENU BUTTON)

; Edited 14-Jul-87 17:28 by Rao  
; 29-MAR-83 17:57

;; MENU WHENSELECTEDFN which allows differential selection on LEFT and middle button. For such differential selection ITEM should be of  
;; form --- (itemSeenInMenu (leftValue midValue)) --- where midValue can be an atom which is directly returned when ITEM is selected with middle,  
;; or midValue can be an itemList, which will be displayed in a subselection MENU

(PROG (IT IT1)

```
(RETURN (COND
  ((IL:NLISTP ITEM)
   ITEM)
  ((IL:NLISTP (IL:SETQ IT (CADR ITEM)))
   IT)
  ((EQ (IL:SETQ IT1 (CAR IT))
   'QUOTE)
   (CADR IT))
  ((EQ IT1 'PROGN)
   (IL:EVAL IT))
  ((EQ BUTTON 'IL:LEFT)
   (COND
    ((IL:LISTP IT1)
     (IL:EVAL IT1))
    (T IT1)))
  ((IL:NLISTP (IL:SETQ IT1 (CADR IT))
   IT1)
   (T (DUAL-MENU IT1))))))
```

)

:: Node Moving Protocol

```
(DEFMETHOD NODE-MOVE ((SELF WEB-EDITOR))
  (LET ((OLD-REGIONS (MAKE-REG-ASSOC SELF))
        NEW-REGIONS MOVED-PAIR NEW-FATHER CLOSEST-PAIR)
    (NODE-MOVE-SHALLOW SELF)
    (SETQ NEW-REGIONS (MAKE-REG-ASSOC SELF))
    (SETQ MOVED-PAIR (IL:|for| |npair| IL:|in| NEW-REGIONS IL:|as| |opair| IL:|in| OLD-REGIONS
                      IL:|thereis| (NOT (IL:EQUAL (CAR |opair|)
                                                (CAR |npair|)))))

    (WHEN (AND MOVED-PAIR
               ;; The moved guy has a parent
               (SLOT-VALUE (CDR MOVED-PAIR)
                           'PARENT))
      (IL:DREMOVE MOVED-PAIR NEW-REGIONS)
      (SETQ NEW-REGIONS
             ;; Collect the pairs that haven't changed.
             (IL:|bind| (SCIONS-OF-MOVED IL:_ (SCIONS (CDR MOVED-PAIR)) IL:|for| PAIR IL:|in| NEW-REGIONS
                       IL:|unless| (IL:MEMBER (CDR PAIR)
                                             SCIONS-OF-MOVED)
                       IL:|collect| PAIR))
              (SETQ CLOSEST-PAIR (IL:|bind| (\b IL:_ (IL:|fetch| IL:BOTTOM IL:|of| (CAR MOVED-PAIR)))
                                           (\l IL:_ (IL:|fetch| IL:LEFT IL:|of| (CAR MOVED-PAIR))) IL:|for| |pair|
                                           IL:|in| NEW-REGIONS IL:|smallest| (IL:PLUS (ABS (IL:IDIFFERENCE (IL:|fetch| IL:BOTTOM
                                                                                               IL:|of|
                                                                                               (CAR |pair|))
                                                                                               \b))
                                         (ABS (IL:IDIFFERENCE (IL:|fetch| IL:LEFT
                                                                                               IL:|of|
                                                                                               (CAR |pair|))
                                                                                               \l)))))

              ;; Either make moved node a sibling or a child of the node it is now closest to.
              (IL:|if| (IL:IGREATERP (IL:IDIFFERENCE (IL:|fetch| IL:LEFT IL:|of| (CAR MOVED-PAIR))
                                                  (IL:|fetch| IL:LEFT IL:|of| (CAR CLOSEST-PAIR)))
                          15)
                      IL:|then| (IL:SETQ NEW-FATHER (CDR CLOSEST-PAIR))
                      IL:|else| (IL:SETQ NEW-FATHER (OR (SLOT-VALUE (CDR CLOSEST-PAIR)
                                                                    'PARENT)
                                                         (CDR CLOSEST-PAIR))))

              (MOVE-NODE (CDR MOVED-PAIR)
                         NEW-FATHER)
              (REORDER-TREE SELF NEW-FATHER))
      (RECOMPUTE SELF)))

    (DEFMETHOD NODE-MOVE-SHALLOW ((SELF WEB-EDITOR))
      ;; Just moves the node graphically with no deep impact
      (LET ((WINDOW (SLOT-VALUE SELF 'WINDOW))
            (IL:RESETLST
             (IL:RESETSAVE NIL (LIST (IL:FUNCTION IL:DSOPERATION)
                                     (IL:DSOPERATION 'IL:INVERT WINDOW)
                                     WINDOW))
             (IL:GETMOUSESTATE) ; Here to move a node.
             (IL:DSOPERATION 'IL:INVERT WINDOW)
             (IL:EDITMOVENODE WINDOW))))
```

```
(DEFMETHOD SCIONS ((SELF WEB-NODE)) ; 14-Nov-86 03:01
                                         ; Used by the Node Mover
  (LET ((TO-LINKS (GET-TO-LINKS SELF))
        (APPEND TO-LINKS (IL:|for| IL:|child| IL:|in| TO-LINKS IL:|join| (SCIONS IL:|child|))))))
```



```
(DEFMETHOD MAKE-REG-ASSOC ((SELF WEB-EDITOR)) ; 14-Nov-86 02:08
; Ho hum
  (IL:|for| X IL:|in| (SLOT-VALUE SELF 'STARTING-LIST) IL:|collect| (CONS (NODE-REGION SELF X)
X)))
```

```
(DEFMETHOD REORDER-TREE ((SELF WEB-EDITOR) ; 14-Nov-86 02:35
  ROOT)
  (LET ((CHILDREN (GET-TO-LINKS ROOT)))
    (IF CHILDREN
      (IL:|SORT| CHILDREN #'(IL:LAMBDA (C1 C2)
        (LET ((R1 (NODE-REGION SELF C1))
              (R2 (NODE-REGION SELF C2)))
          (IL:LESSP (IL:|fetch| IL:BOTTOM IL:|of| R1)
                    (IL:|fetch| IL:BOTTOM IL:|of| R2))))))))))
```

```
(DEFMETHOD MOVE-NODE ((SELF WEB-NODE) ; 29-Jan-87 17:55
  NEW-PARENT)
  (LET ((OLD-PARENT (SLOT-VALUE SELF 'PARENT)))
    (UNLESS (EQ OLD-PARENT NEW-PARENT)
      (SETF (SLOT-VALUE SELF 'PARENT)
            NEW-PARENT)
      (SETF (SLOT-VALUE OLD-PARENT 'TO-LINKS)
            (IL:DREMOVE SELF (SLOT-VALUE OLD-PARENT 'TO-LINKS)))
      (SETF (SLOT-VALUE NEW-PARENT 'TO-LINKS)
            (IL:NCONC1 (SLOT-VALUE NEW-PARENT 'TO-LINKS)
                       SELF))
      T)))
```

::  
::

```
(IL:DECLARE\ : IL:DONTEVAL@LOAD IL:DOEVAL@COMPILE IL:DONTCOPY IL:COMPILERVARS
```

```
(IL:ADDTOVAR IL:NLAMA )
```

```
(IL:ADDTOVAR IL:NLAML )
```

```
(IL:ADDTOVAR IL:LAMA WINDOW-WHEN-HELD-FN WINDOW-WHEN-HELD-FN WEB-WINDOW-EXPAND-FN WEB-WINDOW-RESHAPE-FN
  WEB-WINDOW-BUTTON-EVENT-FN WEB-WINDOW-AFTER-MOVE-FN)
)
```

```
(IL:PUTPROPS IL:WEB-EDITOR IL:COPYRIGHT ("Xerox Corporation" 1987 1988 1989 1991 1993 2020))
```

---

### FUNCTION INDEX

BOX-PRINT-STRING .....	18	FIND-SELECTED-NODE .....	20	WEB-WINDOW-BUTTON-EVENT-FN .....	10
BOX-WINDOW-NODE .....	19	MAKE-WEB-EDITOR .....	5	WEB-WINDOW-CLOSE-FN .....	10
BREAK-STRING-FOR-BOXING .....	19	IL: PromptRead  .....	10	WEB-WINDOW-EXPAND-FN .....	11
CHILD-NODES .....	12	REACHABLE-NODES! .....	12	WEB-WINDOW-ICON-FN .....	11
DO-MENU-METHOD .....	23	SUB-ITEM-SELECTION .....	23	WEB-WINDOW-RESHAPE-FN .....	10
DUAL-MENU .....	23	TREE-ROOTS .....	12	WINDOW-WHEN-HELD-FN .....	22, 23
DUAL-SELECTION .....	23	WEB-MENU-WHENSELECTEDFN .....	22		
DUAL-SUB-ITEMS .....	23	WEB-WINDOW-AFTER-MOVE-FN .....	10		

---

### METHOD INDEX

(INITIALIZE-EDITOR (WEB-EDITOR)) .....	5	(RECOMPUTE-LABELS (WEB-EDITOR)) .....	14
(DESTROY (WEB-EDITOR)) .....	5	(RECOMPUTE-IF-OPEN (WEB-EDITOR)) .....	14
(BROWSE (WEB-EDITOR)) .....	5	(CLEAR-LABEL-CACHE (WEB-EDITOR T)) .....	15
(GET-LABEL (WEB-EDITOR WEB-NODE)) .....	5	(OBJECT-FROM-LABEL (WEB-EDITOR T)) .....	15
(GET-SUBS (WEB-EDITOR WEB-NODE)) .....	6	(CHANGE-FONT-SIZE (WEB-EDITOR)) .....	15
(ICON-TITLE (WEB-EDITOR)) .....	6	(CHANGE-FORMAT (WEB-EDITOR)) .....	15
(ADD-NODE (WEB-EDITOR WEB-NODE)) .....	6	(CHANGE-MAX-LABEL-SIZE (WEB-EDITOR T T)) .....	15
(NOTICE-NODE (WEB-EDITOR WEB-NODE WEB-NODE)) .....	6	(SHAPE-TO-HOLD (WEB-EDITOR)) .....	15
(REMOVE-NODE (WEB-EDITOR WEB-NODE)) .....	6	(GET-DISPLAY-LABEL (WEB-EDITOR T)) .....	16
(DELETE-FROM-BROWSER (WEB-EDITOR T T)) .....	6	(BOX-NODE (WEB-EDITOR T)) .....	16
(REMOVE-FROM-BAD-LIST (WEB-EDITOR)) .....	6	(UNMARK-NODES (WEB-EDITOR)) .....	16
(RENAME-NODE (WEB-EDITOR WEB-NODE T)) .....	6	(HIGHLIGHT-NODE (WEB-EDITOR T T T)) .....	16
(UPDATE (WEB-EDITOR)) .....	6	(SHADE-NODE (WEB-EDITOR T T)) .....	16
(CREATE-WINDOW (WEB-EDITOR)) .....	7	(DISPLAY-NODE-HIGHLIGHTS (WEB-EDITOR T T T)) .....	17
(SETUP-WINDOW (WEB-EDITOR)) .....	7	(DISPLAY-NODE-SHADING (WEB-EDITOR T T)) .....	17
(DETACH-LISP-WINDOW (WEB-EDITOR)) .....	7	(REMOVE-HIGHLIGHTS (WEB-EDITOR)) .....	17
(SHRINK (WEB-EDITOR)) .....	7	(REMOVE-SHADING (WEB-EDITOR)) .....	17
(SET-OUTER-REGION (WEB-EDITOR T T)) .....	7	(FLASH-NODE (WEB-EDITOR T T T)) .....	17
(SET-REGION (WEB-EDITOR T)) .....	8	(FLIP-NODE (WEB-EDITOR T)) .....	17
(MOVE (WEB-EDITOR T)) .....	8	(POSITION-NODE (WEB-EDITOR T T T)) .....	18
(MOVE1 (WEB-EDITOR T)) .....	8	(BUTTON-EVENT-FN (WEB-EDITOR)) .....	20
(AFTER-MOVE (WEB-EDITOR)) .....	8	(LEFT-SELECTION (WEB-EDITOR)) .....	20
(AFTER-RESHAPE (WEB-EDITOR T T T)) .....	8	(MIDDLE-SELECTION (WEB-EDITOR)) .....	20
(SCROLL-WINDOW (WEB-EDITOR T T T T)) .....	9	(RIGHT-SELECTION (WEB-EDITOR)) .....	21
(CLEAR (WEB-EDITOR)) .....	9	(TITLE-SELECTION (WEB-EDITOR)) .....	21
(GET-PROMPT-WINDOW (WEB-EDITOR)) .....	9	(NODE-SELECTION (WEB-EDITOR T)) .....	21
(REMOVE-PROMPT-WINDOW (WEB-EDITOR)) .....	9	(NODE-ACTION (WEB-EDITOR T T)) .....	21
(PROMPT-PRINT (WEB-EDITOR T)) .....	9	(NODE-MENU-ITEMS (WEB-NODE T)) .....	21
(PROMPT-READ (WEB-EDITOR T)) .....	9	(CHOICE-MENU (WEB-EDITOR T)) .....	21
(PROMPT-FOR-LIST (WEB-EDITOR T T)) .....	10	(DO-SELECTED-COMMAND (WEB-EDITOR T T)) .....	22
(PROMPT-FOR-STRING (WEB-EDITOR T T)) .....	10	(WHEN-MENU-ITEM-HELD (WEB-EDITOR T T T)) .....	22
(PROMPT-FOR-WORD (WEB-EDITOR)) .....	10	(ITEM-MENU (WEB-EDITOR T T)) .....	22
(DISPLAY-BROWSER (WEB-EDITOR)) .....	13	(GET-MENU-ITEMS (WEB-EDITOR T)) .....	22
(BROWSER-OBJECTS (WEB-EDITOR)) .....	13	(CLEAR-MENU-CACHE (WEB-EDITOR)) .....	22
(GET-NODE-LIST (WEB-EDITOR T T)) .....	13	(NODE-MOVE (WEB-EDITOR)) .....	24
(OBJ-NAME-PAIR (WEB-EDITOR T)) .....	13	(NODE-MOVE-SHALLOW (WEB-EDITOR)) .....	24
(GRAPH-FITS (WEB-EDITOR)) .....	14	(SCIONS (WEB-NODE)) .....	24
(NODE-REGION (WEB-EDITOR T)) .....	14	(MAKE-REG-ASSOC (WEB-EDITOR)) .....	25
(RECOMPUTE (WEB-EDITOR)) .....	14	(REORDER-TREE (WEB-EDITOR T)) .....	25
(RECOMPUTE-IN-PLACE (WEB-EDITOR)) .....	14	(MOVE-NODE (WEB-NODE T)) .....	25

---

### VARIABLE INDEX

IL:*D-WINDOW-DEFAULT-STREAM* .....	11	DESTINATION-BROWSER .....	3	IL:GRAYSHADE4 .....	12
*WEB-EDITOR-ICON-BM* .....	11	FILE-HEADER-COMS .....	2	IL: MaxLatticeHeight  .....	11
*WEB-EDITOR-ICON-MASK* .....	11	IL:GRAYSHADE1 .....	11	IL: MaxLatticeWidth  .....	11
*WEB-EDITOR-TEMPLATE* .....	11	IL:GRAYSHADE2 .....	11	WEB-STREAM .....	11
IL: BrowserMargin  .....	11	IL:GRAYSHADE3 .....	11		

---

### CLASS INDEX

WEB-EDITOR .....	4	WEB-NODE .....	4
------------------	---	----------------	---

---

### MACRO INDEX

MOVE-DOWN-P .....	10
-------------------	----

---

### PROPERTY INDEX

IL:WEB-EDITOR .....	3
---------------------	---

{MEDLEY}<clos>WEB-EDITOR.;1

---