File created: 29-Jul-88 15:58:41 **{ICE}<KOOMEN>LISPUSERS>KOTO>SEDIT-PATCHES.;20**

changes to: (VARS SEDIT-PATCHESCOMS)
(FNS \\exit.from.keyboard \\edit.selection \\substitute.text)

previous date: 11-Jul-88 10:29:41 {ICE}<KOOMEN>LISPUSERS>KOTO>SEDIT-PATCHES.;17

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS


```
(RPAQQ SEDIT-PATCHESCOMS
       [(FILES (SYSLOAD)
               SEDIT)
        (DECLARE: DONTEVAL@LOAD DONTCOPY EVAL@COMPILE (LOCALVARS . T)
               (FILES (LOADCOMP)
                      SEDIT)
               (VARS (CompilingForKoto T)))
```

;;; Make SEdit able to edit something other than functions

```
        (FNS \\edit.selection \\editdef)
```

;;; Make SEdit show in the title bar (with *) that something has changed

```
        (FNS \\mark.context)
                                                                    ; new
        (FNS SEditTTYfn \\handle.completion \\note.change \\undo)
```

;;; Facility for adding and removing commands

```
        (FNS ADD.SEDIT.COMMAND REMOVE.SEDIT.COMMAND)
        (FNS \\flatten.command.table \\cmdorder)
```

;;; Facility for adding quote-type functiona

```
        (FNS ADD.SEDIT.QUOTE \\create.constant.strings)
        [INITVARS (\\quotestring.info (COPYALL (QUOTE ((QUOTE . "'")
                                                       (BQUOTE . "`")
                                                       (\, . ",")
                                                       (\,@ . ",@")
                                                       (CL:FUNCTION . "#'"]
```

;;; Patch to remove AddMenu command, as freemenu description and interface is bogus

```
        [DECLARE: DONTEVAL@LOAD DOCOPY (P (REMOVE.SEDIT.COMMAND (QUOTE \\add.menu]
```

;;; New functionality: DefineFunction using current selection, Text Substitution, and keyboard exit

```
        (FNS \\define.function \\exit.from.keyboard \\substitute.text)
        (INITVARS (\\substitute.text.old.candidate NIL)
               (\\substitute.text.new.candidate NIL))
        (DECLARE: DONTEVAL@LOAD DOCOPY (P (ADD.SEDIT.COMMAND (QUOTE ("1,d" "1,D" (DefineFunction)))
                                                 (FUNCTION \\define.function)
                                                 "Define Function       M-D" "Define function using current
                                                 selection and substitute call")
                                          (ADD.SEDIT.COMMAND (QUOTE ("1,^X" (ExitFromKeyboard)))
                                                 (FUNCTION \\exit.from.keyboard)
                                                 "Done & Close  M-^X" "Same as closing this SEdit window")
                                          (ADD.SEDIT.COMMAND (QUOTE ("1,t" "1,T" (SubstituteText)))
                                                 (FUNCTION \\substitute.text)
                                                 "Substitute Text       M-T" "Prompt for text patterns to
                                                 substitute in current selection (cf. ESUBST, IRM 16.73)")))
```

;;; Provide Lyric-style interface to edit window regions

```
        (FNS SEDIT.GET.WINDOW.REGION SEDIT.SAVE.WINDOW.REGION)
        (FNS \\build.window \\disintegrate.context \\expandfn \\shrinkfn)
        (VARS (:CREATE (QUOTE :CREATE))
              (:EXPAND (QUOTE :EXPAND))
              (:CLOSE (QUOTE :CLOSE))
              (:SHRINK (QUOTE :SHRINK)))
```

;;; Patch to circumvent bug in \\linearize.form

```
        (FNS \\linearize.form)
```

;;; Patch to fix deadly bug when typing non-list after PROG, LAMBDA, etc

```
        (FNS \\reparse.litatom)
```

;;; I/O Patch

```
        [DECLARE: DONTEVAL@LOAD DOCOPY (P (CHANGENAME (QUOTE \\sedit)
                                                      (QUOTE READP)
                                                      (QUOTE \SYSBUFP]
```

;;; Give Dorado a BQUOTE character

```
        (DECLARE: DONTEVAL@LOAD DOCOPY (P (COND ((EQ (MACHINETYPE)
                                                     (QUOTE DORADO))
                                                 (METASHIFT T)
```

;; Make BLANK-TOP key a BQUOTE

```
                                                 (KEYACTION (QUOTE BLANK-TOP)
                                                        (QUOTE ((96 96])

(FILESLOAD (SYSLOAD)
       SEDIT)

(DECLARE: DONTEVAL@LOAD DONTCOPY EVAL@COMPILE

(DECLARE: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(FILESLOAD (LOADCOMP)
       SEDIT)

(RPAQQ CompilingForKoto T)
)
```

;;; Make SEdit able to edit something other than functions

```
(DEFINEQ
```

(\\edit.selection

```
  [LAMBDA (context)                                          (* Koomen "22-Jun-88 07:26")
                                                             (* jow "18-Sep-86 12:50")
```

;; (Koomen -- 22-Jun-88) Replaced EDITDEF by \\editdef to allow editing of things other than functions

```
    [LET ((structure (\\selected.fn.name context)))
         (if structure
             then (\\set.selection.nowhere (fetch (EditContext Selection) of context))
                  (\\set.point.nowhere (fetch (EditContext CaretPoint) of context))
                  (\\update context)
                  (if [NULL (NLSETQ (if (CompilingForKoto)
                                        then (\\editdef structure)
                                      else (ED structure T]
                      then (printout (GETPROMPTWINDOW (fetch DisplayWindow of context))
                                 T structure " not editable."]
    T])
```

(\\editdef

```
  [LAMBDA (structure)                                        (* Koomen "13-Nov-87 14:26")
```

;; Patch to get around the problem of SEdit insisting on editing FNS on META-O, instead of checking if more than one type is defined (as the ;; documentation claims).

```
    (PROG* [(STYPES (TYPESOF structure NIL NIL (QUOTE ?)))
            (DEFTYPE (if (OR (NULL STYPES)
                             (CDR STYPES))
                         then (MENU (create MENU
                                            TITLE _ "Select type: "
                                            ITEMS _ [OR STYPES (QUOTE (("  New Function " (QUOTE NEWFN)
                                                                              "Create and edit a new function")
                                                                       ("  New Macro " (QUOTE NEWMACRO)
                                                                              "Create and edit a new macro"]
                                            CENTERFLG _ T))
                       else (CAR STYPES]
           (RETURN (if DEFTYPE
                       then (if (NULL STYPES)
                                then (SELECTQ DEFTYPE
                                              (NEWFN (PUTDEF structure (SETQ DEFTYPE (QUOTE FNS))
                                                             (LIST (QUOTE LAMBDA)
                                                                   \\args.gap \\body.gap)))
                                              (NEWMACRO [PUTDEF structure (SETQ DEFTYPE (QUOTE MACROS))
```

```
                                                    (BQUOTE (PUTPROPS (\, structure) MACRO ((\, \\args.gap)
                                                                                            (\, \\body.gap)))
                                                            ])
                                        (SHOULDNT)))
                                (EDITDEF structure DEFTYPE])

)
```

;;; Make SEdit show in the title bar (with *) that something has changed

```
(DEFINEQ
```

(\\mark.context

```
  [LAMBDA (context changed?)                                        (* Koomen "15-Jun-88 13:18")
```

;; This sets the ChangedStructure? flag according to the argument changed? This used to be done inline. Seperated so that this function will also ;; chnage the window title to make this flag visible

```
    (PROG (waschanged? dspwin title oldtitle)
          (SETQ waschanged? (fetch ChangedStructure? of context))
          (replace ChangedStructure? of context with changed?)
          (SETQ dspwin (fetch DisplayWindow of context))
          (if (AND dspwin (NEQ changed? waschanged?)
                   (OPENWP dspwin))
              then (SETQ oldtitle (WINDOWPROP dspwin (QUOTE TITLE)))
                   (SETQ title (OR (WINDOWPROP dspwin (QUOTE OLDTITLE))
                                   (CONCAT "* " oldtitle)))
                   (WINDOWPROP dspwin (QUOTE TITLE)
                          title)
                   (WINDOWPROP dspwin (QUOTE OLDTITLE)
                          oldtitle))
          (RETURN waschanged?])

)
```

;; new

```
(DEFINEQ
```

(SEditTTYfn

```
  [LAMBDA (ATM TYPE)                                                (* Koomen "15-Jun-88 13:17")
                                                                    (* mdd "24-Oct-86 15:26")
    (DECLARE (USEDFREE L))
```

;; this is a replacement for the TTY editor's TTY: command, which starts and SEdit process to do interactive editing for a while. it uses the TTY ;; editor's edit chain to determine the initial selection in the structure, and scrolls the window to make sure the selection's visible. it then waits until ;; the user signals that they've done enough editing (usually by closing or shrinking the window)

;; Koomen 15-Jun-88 Replaced (replace ChangedStructure? ...) with \\mark.context

```
    (LET ((context (SEdit (CAR (LAST L))
                          (LIST (QUOTE NAME)
                                ATM
                                (QUOTE TYPE)
                                TYPE
                                (QUOTE DONTWAIT)
                                T)))
          node)
         (DECLARE (USEDFREE EDITCHANGES))
         (WITH.MONITOR (fetch ContextLock of context)
             (if (SETQ node (\\locate.node.from.editchain L (fetch Root of context)))
                 then (\\selection.down context)
                      (\\select.node context node)
                      (\\compute.selection.position (fetch Selection of context)
                             context)
                      (\\set.point.nowhere (fetch CaretPoint of context))
                      (\\normalize.selection context)
                      (if (NOT (fetch SelectionDisplayed? of context))
                          then (\\display.selection (fetch Selection of context)
                                       (fetch DisplayWindow of context))
                               (replace SelectionDisplayed? of context with T))))
                                                                    ; let the user do their editing before we return
         (AWAIT.EVENT (fetch CompletionEvent of context))
         (if (\\mark.context context NIL)
             then (RPLACA (CDR EDITCHANGES)
                          T])
```

(\\handle.completion

```
  [LAMBDA (context)                                                 (* Koomen "15-Jun-88 13:17")
                                                                    (* mdd " 1-Aug-86 11:54")
```

;; Koomen 15-Jun-88 Replaced (replace ChangedStructure? ...) with \\mark.context

```
    (NOTIFY.EVENT (fetch CompletionEvent of context))
    (replace UndoList of context with NIL)
    (replace UndoUndoList of context with NIL)
    (replace AtomStarted of context with NIL)
    (replace AtomStartedUndoPointer of context with NIL)
```

```
    (if (\\mark.context context NIL)
        then (LET ((fn (fetch CompletionFn of context)))
                  (if fn
                      then (APPLY (if (LISTP fn)
                                      then (CAR fn)
                                    else fn)
                                  (LIST* context (fetch Structure of (\\subnode 1 (fetch Root of context)))
                                         (CDR (LISTP fn])
```

(\\note.change

```
  [LAMBDA (node context)                                                (* Koomen "15-Jun-88 13:17")
                                                                        (* mdd "20-Jun-86 18:11")
```

(* we've made some change to this node. clobber any clisp translation, and insert it into the ChangedNodes list (which is sorted by depth))

;; Koomen 15-Jun-88 Replaced (replace ChangedStructure? ...) with \\mark.context

```
    (if (NOT (fetch Changed? of node))
        then (for (super _ node) by (fetch SuperNode of super) while super when (LISTP (fetch Structure of super))
                do (\\zap.clisp.translation (fetch Structure of super)))
             (replace Changed? of node with T)
             (bind next (last _ (fetch ChangedNodes of context)) while (AND (SETQ next (CDR last))
                                                                            (IGREATERP (fetch Depth
                                                                                              of (CAR next))
                                                                                       (fetch Depth of node)))
                do (SETQ last next) finally (RPLACD last (CONS node next)))
             (\\mark.context context T])
```

(\\undo

```
  [LAMBDA (context)                                                     (* Koomen "15-Jun-88 13:17")
                                                                        (* jow " 4-Sep-86 14:38")
```

;; Koomen 15-Jun-88 Replaced (replace ChangedStructure? ...) with \\mark.context

```
    (\\close.open.node context)
    [LET [(undo.list (fetch UndoList of context))
          (promptwindow (GETPROMPTWINDOW (fetch DisplayWindow of context]
         (if undo.list
             then (replace UndoList of context with (fetch UndoUndoList of context))
                  (\\set.selection.nowhere (fetch Selection of context))
                  (\\set.point.nowhere (fetch CaretPoint of context))
                  (\\undo.event (CAR undo.list)
                                context)
                  (replace UndoUndoList of context with (fetch UndoList of context))
                  (if (NULL (replace UndoList of context with (CDR undo.list)))
                      then (\\mark.context context NIL))
           else (printout promptwindow T (if (fetch UndoUndoList of context)
                                             then "Nothing else to Undo"
                                           else "Nothing to Undo"]
    T])

)

;;; Facility for adding and removing commands

(DEFINEQ

## (ADD.SEDIT.COMMAND

  [LAMBDA (keyspec commandfn menulabel menucomment)                     (* Koomen "22-Jun-88 08:56")
    (DECLARE (GLOBALVARS \\SEdit.Contexts \\command.table.spec))
    (PROG* [(cmdtbl (\\flatten.command.table))
            (cmdname commandfn)
            [keyspec2 (for key inside keyspec collect (if (LISTP key)
                                                          then (SETQ cmdname (CAR key))
                                                               key
                                                        else (\\charcode key]
            (menuspec (if menulabel
                          then (LIST (CONS menulabel (MKLIST menucomment]
           (for key in keyspec2 bind cmd
              do (if [SETQ cmd (if (FIXP key)
                                   then (ASSOC key cmdtbl)
                                 else (for c in cmdtbl thereis (AND (LISTP (CAR c))
                                                                    (EQUAL key (CAAR c]
                     then (if (NOT (EQUAL (CADR cmd)
                                          commandfn))
                              then (PROMPTPRINT "Rebinding " (if (FIXP (CAR cmd))
                                                                 then (CHARACTER (CAR cmd))
                                                               else (CAR cmd))
                                                " from "
                                                (CADR cmd)
                                                " to " cmdname))
                          (RPLACA (CDR cmd)
                                  commandfn)
                          (RPLACD (CDR cmd)
                                  menuspec)
```

```
                        (SETQ keyspec2 (REMOVE key keyspec2))
                        (SETQ menuspec NIL)))
          (for key in keyspec2 do (push cmdtbl (LIST* (if (FIXP key)
                                                          then key
                                                        else (LIST key))
                                                      commandfn menuspec))
                                  (SETQ menuspec NIL))
          (SETQ \\command.table.spec (SORT cmdtbl (FUNCTION \\cmdorder)))
          (if (NULL \\SEdit.Contexts)
              then (SEDIT.RESET)
            else (PROMPTPRINT "Close SEdit windows, then (SEDIT.RESET) to enable " cmdname))
          (RETURN cmdname])
```

(REMOVE.SEDIT.COMMAND

```
  [LAMBDA (commandfn)                                                    (* Koomen "22-Jun-88 08:04")
    (DECLARE (GLOBALVARS \\SEdit.Contexts \\command.table.spec))
    (SETQ \\command.table.spec (SORT (for cmd in (\\flatten.command.table)
                                       unless (EQ commandfn (if (LISTP (CADR cmd))
                                                                then (CAR (CADR cmd))
                                                              else (CADR cmd)))
                                       collect cmd)
                                     (FUNCTION \\cmdorder)))
    (if (NULL \\SEdit.Contexts)
        then (SEDIT.RESET)
      else (PROMPTPRINT "Close SEdit windows, then (SEDIT.RESET) to disable " commandfn))
    commandfn])

)

(DEFINEQ
```

(\\flatten.command.table

```
  [LAMBDA NIL                                                            (* Koomen "22-Jun-88 07:45")
    ;; Normalize table, so we can redefine things more easily
    (DECLARE (GLOBALVARS \\command.table.spec))
    (for spec in \\command.table.spec
       join (if (FIXP (CAR spec))
                then (LIST spec)
              else (for key inside (CAR spec) bind (entry _ (CADDR spec))
                      collect (LIST* (if (LISTP key)
                                         then (LIST key)
                                       else (\\charcode key))
                                     (CADR spec)
                                     (if entry
                                         then (PROG1 (LIST entry)
                                                     (SETQ entry])
```

(\\cmdorder

```
  [LAMBDA (cmd.x cmd.y)                                                  (* Koomen "22-Jun-88 07:57")
    (DECLARE (GLOBALVARS UPPERCASEARRAY))
    (LET ((key.x (CAR cmd.x))
          (key.y (CAR cmd.y)))
         (if (FIXP key.x)
             then (if (FIXP key.y)
                      then (LEQ key.x key.y)
                    else T)
           elseif [AND (LISTP (CAR (LISTP key.x)))
                       (LISTP (CAR (LISTP key.y]
             then (ALPHORDER (CAAR key.x)
                             (CAAR key.y)
                             UPPERCASEARRAY])

)

;;; Facility for adding quote-type functiona

(DEFINEQ
```

(ADD.SEDIT.QUOTE

```
  [LAMBDA (QUOTESTR QUOTEFN)                                             (* Koomen "22-Jun-88 09:09")
    (LISTPUT \\list.ParseInfo QUOTEFN (CONS (QUOTE \\parse..quote)
                                            (QUOTE \\reparse.list.to.quote)))
    (PUTASSOC QUOTEFN QUOTESTR \\quotestring.info)
    (ADD.SEDIT.COMMAND QUOTESTR (LIST (QUOTE \\insert.quoted.gap)
                                      QUOTEFN])
```

(\\create.constant.strings

```
  [LAMBDA (env)                                                          (* Koomen "22-Jun-88 09:03")
                                                                         (* jow "17-Oct-86 16:39")
    ;; [Koomen 22-Jun-88]  replace inline list with variable \\quotestring.info
    (LET ((font (fetch DefaultFont of env)))
```

```
        (replace LParenString of env with (\\create.string.item "(" font))
        (replace RParenString of env with (\\create.string.item ")" font))
        (replace DotString of env with (\\create.string.item "." font))
        [replace QuoteString of env with (for prefix in \\quotestring.info join (LIST (CAR prefix)
                                                                                       (\\create.string.item
                                                                                        (CDR prefix)
                                                                                        font]
        (replace CommentString of env with (for prefix in (QUOTE ((1 . "; ")
                                                                  (2 . ";; ")
                                                                  (3 . ";;; ")))
                                           join (LIST (CAR prefix)
                                                      (\\create.string.item (CDR prefix)
                                                             (fetch KeywordFont of env])

)

(RPAQ? \\quotestring.info [COPYALL (QUOTE ((QUOTE . "'")
                                           (BQUOTE . "`")
                                           (\, . ",")
                                           (\,@ . ",@")
                                           (CL:FUNCTION . "#'"])


;;; Patch to remove AddMenu command, as freemenu description and interface is bogus

(DECLARE: DONTEVAL@LOAD DOCOPY

(REMOVE.SEDIT.COMMAND (QUOTE \\add.menu))
)


;;; New functionality:  DefineFunction using current selection, Text Substitution, and keyboard exit

(DEFINEQ
```

(\\define.function

```
  [LAMBDA (context)                                                    (* Koomen "22-Jun-88 09:14")
    ;; This command prompts for the name of a new function, and defines the function using the current selection (which may be extended over several
    ;; forms) as the body.  The current selection is then replaced by a call to this new function.
    (PROG (promptwindow fnname selection node)
          (SETQ promptwindow (GETPROMPTWINDOW (fetch DisplayWindow of context)))
          (TERPRI promptwindow)
          (SETQ fnname (PROMPTFORWORD "Define function: " NIL NIL promptwindow))
          (if (NOT (STRINGP fnname))
              then (TERPRI promptwindow)
                   (RETURN))
          (SETQ fnname (MKATOM fnname))
          (if (AND (DEFINEDP fnname)
                   (NOT (MOUSECONFIRM (CONCAT "Redefine " fnname " ? ")
                              NIL promptwindow)))
              then (TERPRI promptwindow)
                   (RETURN))
          (\\parenthesize.current.selection context)
          (SETQ selection (fetch Selection of context))
          (SETQ node (fetch SelectNode of selection))
          (if (AND node (fetch Structure of node)
                   (EQ (fetch SelectType of selection)
                       (QUOTE Structure))
                   (NULL (fetch SelectStart of selection)))
              then (PUTDEF fnname (QUOTE FNS)
                           (LIST* (QUOTE LAMBDA)
                                  NIL
                                  (fetch Structure of node)))
                   (\\replace.node context node (\\parse.new (LIST fnname)
                                                             context))
            else (printout promptwindow T "Nothing appropriate selected.")))
    T])
```

(\\exit.from.keyboard

```
  [LAMBDA (context)                                                    (* Koomen "29-Jul-88 12:47")
    (CLOSEW (fetch DisplayWindow of context))
    T])
```

(\\substitute.text

```
  [LAMBDA (context)                                                    (* Koomen "29-Jul-88 11:26")
    (DECLARE (GLOBALVARS \\substitute.text.new.candidate \\substitute.text.old.candidate))
    (PROG (promptwindow selection node struct parenthesized old new rplstruct rplnode)
          (SETQ promptwindow (GETPROMPTWINDOW (fetch DisplayWindow of context)))
          (SETQ selection (fetch Selection of context))
          (if [OR (NEQ (fetch SelectType of selection)
                       (QUOTE Structure))
                  (NULL (SETQ node (fetch SelectNode of selection)))
                  (NULL (SETQ struct (fetch Structure of node]
              then (printout promptwindow T "Nothing appropriate selected.")
```

```
                (RETURN))
            (TERPRI promptwindow)
            [SETQ old (PROMPTFORWORD "Replace old text: " \\substitute.text.old.candidate NIL promptwindow NIL NIL
                             (CHARCODE (EOL LF]
            (if (NULL old)
                then (printout promptwindow T "Text substitution aborted.")
                     (RETURN))
            (SETQ \\substitute.text.old.candidate (SETQ old (MKATOM old)))
            (TERPRI promptwindow)
            (SETQ new (OR (PROMPTFORWORD "with new text: " \\substitute.text.new.candidate NIL promptwindow NIL
                                    NIL (CHARCODE (EOL LF)))
                          ""))
            (SETQ \\substitute.text.new.candidate (SETQ new (MKATOM new)))
            (if (fetch SelectStart of selection)
                then (\\parenthesize.current.selection context)
                     (SETQ selection (fetch Selection of context))
                     (SETQ node (fetch SelectNode of selection))
                     (SETQ struct (fetch Structure of node))
                     (SETQ parenthesized T))
            (SETQ struct (COPYALL struct))
            (SETQ rplstruct (NLSETQ (ESUBST new old struct NIL T)))
            (if (NULL rplstruct)
                then (printout promptwindow T "No text substitutions made.")
              else (SETQ rplnode (\\parse.new (CAR rplstruct)
                                        context))
                   (if (type? EditNode rplnode)
                       then (\\replace.node context node rplnode)
                            (printout promptwindow T "Done.")
                     else (printout promptwindow T "Oops! Returned ESUBST value unparsable!!!")))
            (if parenthesized
                then (\\extract.current.selection context)))
    T])

)

(RPAQ? \\substitute.text.old.candidate NIL)

(RPAQ? \\substitute.text.new.candidate NIL)

(DECLARE: DONTEVAL@LOAD DOCOPY

(ADD.SEDIT.COMMAND (QUOTE ("1,d" "1,D" (DefineFunction)))
       (FUNCTION \\define.function)
       "Define Function M-D" "Define function using current selection and substitute call")

(ADD.SEDIT.COMMAND (QUOTE ("1,^X" (ExitFromKeyboard)))
       (FUNCTION \\exit.from.keyboard)
       "Done & Close    M-^X" "Same as closing this SEdit window")

(ADD.SEDIT.COMMAND (QUOTE ("1,t" "1,T" (SubstituteText)))
       (FUNCTION \\substitute.text)
       "Substitute Text M-T" "Prompt for text patterns to substitute in current selection (cf. ESUBST, IRM
       16.73)")
)


;;; Provide Lyric-style interface to edit window regions

(DEFINEQ

(SEDIT.GET.WINDOW.REGION
  [LAMBDA (CONTEXT REASON)                                          (* Koomen "11-Jul-88 10:27")
    ;; Reason ignored
    (OR (pop \\SEdit.Regions)
        (GETREGION MINWIDTH MINHEIGHT])

(SEDIT.SAVE.WINDOW.REGION
  [LAMBDA (CONTEXT REASON)                                          (* Koomen "11-Jul-88 10:27")
    ;; REASON ignored
    (push \\SEdit.Regions (LET [(REG (WINDOWPROP (fetch (EditContext DisplayWindow) of CONTEXT)
                                         (QUOTE REGION]
                                ;; Make a copy, with HEIGHT extended by a one-line promptwindow, because SEdit destructively modifies
                                ;; regions by subtracting from HEIGHT a one-line promptwindow!
                                (CREATEREGION (fetch (REGION LEFT) of REG)
                                       (fetch (REGION BOTTOM) of REG)
                                       (fetch (REGION WIDTH) of REG)
                                       (IPLUS (fetch (REGION HEIGHT) of REG)
                                              (HEIGHTIFWINDOW (FONTPROP NIL (QUOTE HEIGHT])

)

(DEFINEQ
```

```
(\\build.window
  [LAMBDA (context)                                          (* Koomen "22-Jun-88 08:21")
                                                             (* mdd "17-Sep-86 20:45")
                                                             ; this is a new context.  fill in all the important fields and set
                                                             ; things up for editing

    ;; [Koomen 13-Nov-87]  Save initial region if necessary, obtain one using SEDIT.GET.WINDOW.REGION instead of GETREGION

    (DECLARE (GLOBALVARS \\type.root))
    (LET* [(environment (fetch Environment of context))
           (structure (fetch Root of context))
           (root (create EditNode
                         NodeType _ \\type.root
                         Depth _ 1
                         SubNodes _ (LIST 0)
                         LinearForm _ (CONS)
                         StartX _ 1
                         ActualWidth _ 0))
           [package (AND (CompilingPostKoto)
                         (PACKAGE-NAME (fetch Package of context]
           (display.window (CREATEW (LET (region)
                                         (if (REGIONP (fetch DisplayWindow of context))
                                             then (push \\SEdit.Regions (fetch DisplayWindow of context)))
                                         (replace (EditContext DisplayWindow) of context with NIL)
                                         (SETQ region (SEDIT.GET.WINDOW.REGION context :CREATE))
                                                             ; this will subtract the height of a one line prompt window in the
                                                             ; defualt font
                                         [replace (REGION HEIGHT) of region
                                            with (IDIFFERENCE (fetch (REGION HEIGHT) of region)
                                                              (HEIGHTIFWINDOW (FONTPROP NIL (QUOTE HEIGHT]
                                         region)
                                    (if package
                                        then (CONCAT (PROCESSPROP (THIS.PROCESS)
                                                                  (QUOTE NAME))
                                                     " in " package)
                                      else (PROCESSPROP (THIS.PROCESS)
                                                        (QUOTE NAME]
          (GETPROMPTWINDOW display.window 1)
          (if SEDIT.WANT.MENU
              then (\\add.menu display.window))
          (replace CommentWidth of context with 200)
          (replace CommentSeparation of context with 30)
          (replace Environment of context with environment)
          (replace Root of context with root)
          (replace DisplayWindow of context with display.window)
          (replace CaretPoint of context with (create EditPoint))
          (replace Selection of context with (create EditSelection))
          (replace SelfLink of root with (create WeakLink
                                                 Destination _ root))
          (WYOFFSET (SUB1 (WINDOWPROP display.window (QUOTE HEIGHT)))
                 display.window)
          (replace WindowLeft of context with (fetch LEFT of (DSPCLIPPINGREGION NIL display.window)))
          (replace WindowBottom of context with (fetch BOTTOM of (DSPCLIPPINGREGION NIL display.window)))
          (replace WindowRight of context with (fetch RIGHT of (DSPCLIPPINGREGION NIL display.window)))
          (replace WindowTop of context with (fetch TOP of (DSPCLIPPINGREGION NIL display.window)))
          (DSPLINEFEED (IMINUS (IPLUS (FONTPROP (fetch DefaultFont of environment)
                                                (QUOTE HEIGHT))
                                      (fetch DefaultLineSkip of environment)))
                 display.window)

    ;; set the window's right margin big enough that things won't be wrapped on us.  this is sort of gross -- there should be a way to completely
    ;; disable wrap

          (DSPRIGHTMARGIN 64000 display.window)
          (WINDOWPROP display.window (QUOTE EditContext)
                 context)
          (replace CurrentNode of context with root)
          (replace \X of context with NIL)
          (replace OpenNode of context with NIL)
          [LET ((string (ALLOCSTRING 512 NIL NIL T)))
               (replace OpenNodeInfo of context with (create OpenString
                                                             BufferString _ string
                                                             Substring _ (SUBSTRING string 1 1]
          (\\parse structure context)
          (\\build.linear.form context)
          (LET [(height (IDIFFERENCE (fetch LineHeight of (fetch LastLine of root))
                                     (fetch YCoord of (fetch LastLine of root]
               (WINDOWPROP display.window (QUOTE EXTENT)
                      (create REGION
                              LEFT _ 0
                              BOTTOM _ (IDIFFERENCE 1 height)
                              WIDTH _ (fetch ActualWidth of root)
                              HEIGHT _ height)))
          (WINDOWPROP display.window (QUOTE SCROLLEXTENTUSE)
                 (QUOTE (- . +)))
          (WINDOWPROP display.window (QUOTE REPAINTFN)
                 (FUNCTION \\repaintfn))
          (WINDOWPROP display.window (QUOTE SCROLLFN)
                 (FUNCTION SCROLLBYREPAINTFN))
```

```
          (WINDOWPROP display.window (QUOTE WINDOWENTRYFN)
                 (FUNCTION \\new.buttoneventfn))
          (WINDOWPROP display.window (QUOTE BUTTONEVENTFN)
                 (FUNCTION \\new.buttoneventfn))
          (WINDOWPROP display.window (QUOTE RIGHTBUTTONFN)
                 (FUNCTION \\new.buttoneventfn))                ; use windowaddprop to preserve attached window functions
                                                                ; already there
          (WINDOWADDPROP display.window (QUOTE CLOSEFN)
                 (FUNCTION \\closefn))
          (WINDOWADDPROP display.window (QUOTE SHRINKFN)
                 (FUNCTION \\shrinkfn))
          (WINDOWADDPROP display.window (QUOTE EXPANDFN)
                 (FUNCTION \\expandfn))
          (WINDOWADDPROP display.window (QUOTE RESHAPEFN)
                 (FUNCTION \\reshapefn])
```

(**disintegrate.context**

```
  [LAMBDA (CONTEXT)                                             (* Koomen " 6-Nov-87 17:13")
    ;; Replaces inline window region saving with call to SEDIT.SAVE.WINDOW.REGION
    (DECLARE (GLOBALVARS \\SEdit.Contexts))
    (if CONTEXT
        then (replace (EditContext ContextLock) of CONTEXT with (QUOTE Dead))
             (SEDIT.SAVE.WINDOW.REGION CONTEXT (QUOTE CLOSED))
             (WINDOWPROP (fetch (EditContext DisplayWindow) of CONTEXT)
                    (QUOTE EditContext)
                    NIL)
             (replace (EditContext DisplayWindow) of CONTEXT with NIL)
             (SETQ \\SEdit.Contexts (DREMOVE CONTEXT \\SEdit.Contexts])
```

(**expandfn**

```
  [LAMBDA (window)                                              (* Koomen "22-Jun-88 08:26")
                                                                (* jow "21-Aug-86 14:45")

          (* called by the window system when SEdit window icons are expanded.
          start a new command process for the window)
    ;; [Koomen 13-Nov-87]  Get a (possibly new) region through interface
    (LET [(context (WINDOWPROP window (QUOTE EditContext]
         (SHAPEW window (SEDIT.GET.WINDOW.REGION context :EXPAND))
         (if (NOT (WINDOWPROP window (QUOTE PROCESS)))
             then (replace EvalInProcess of context with (\\eval.in.process))
                  (ADD.PROCESS (LIST (QUOTE \\sedit)
                                     (KWOTE context))
                               (QUOTE NAME)
                               (CONCAT "SEdit " (fetch IconTitle of context])
```

(**shrinkfn**

```
  [LAMBDA (window)                                              (* Koomen "22-Jun-88 08:34")
                                                                (* jow "13-Aug-86 11:40")

          (* called by the window system when an SEdit window is shrunk.
          if it doesn't already, have one, give it a pretty icon with an appropriate title.
          also make sure the command process notices that it should die)
    ;; [Koomen 13-Nov-87]  Save the region through interface
    (LET [(context (WINDOWPROP window (QUOTE EditContext]
         (SEDIT.SAVE.WINDOW.REGION context :SHRINK)
         (if (NOT (WINDOWPROP window (QUOTE ICON)))
             then (WINDOWPROP window (QUOTE ICON)
                              (TITLEDICONW \\titled.icon (fetch IconTitle of context)
                                           NIL T)))
         (\\awake.command.process context NIL])

)

(RPAQQ :CREATE :CREATE)

(RPAQQ :EXPAND :EXPAND)

(RPAQQ :CLOSE :CLOSE)

(RPAQQ :SHRINK :SHRINK)

;;; Patch to circumvent bug in \\linearize.form

(DEFINEQ
```

(**linearize.form**

```
  [LAMBDA (node context index)                                  (* Koomen "16-Jun-88 16:01")
                                                                (* jow "26-Sep-86 12:10")
                                                                ; the linearize method for forms
    ;; Koomen 16-Jun-88  --  there was a reference to (fetch Unassigned of node) as one of the branches in the conditional under 'first' which maybe
```

```
;; NIL.  Hence, wrapped it in an OR to compute some alternate valid indent value (dunno if it's right) instead.
[if (NOT index)
    then (\\output.constant.string context (fetch LParenString of (fetch Environment of context]
[if (CDR (fetch SubNodes of node))
    then
    (bind (same.line? _ T)
          (space.width _ (fetch SpaceWidth of (fetch Environment of context)))
          [paren.width _ (fetch Width of (fetch LParenString of (fetch Environment of context]
          (first.subnode _ T)
          indent last.comment.level comment.start.x comment.indent line.skip
       first [SETQ indent (IPLUS (fetch StartX of node)
                                 (if [NOT (ATOM (fetch Structure of (CADR (fetch SubNodes of node]
                                     then
                                          ;; this will handle the case of comment first, too, like in COMS.  it will be ugly for comment at
                                          ;; beginning of function call, but who cares.
                                          paren.width
                                     elseif (ILEQ (IPLUS (fetch StartX of node)
                                                         (fetch PreferredWidth of node))
                                                  (fetch RightMargin of node))
                                     then [OR (fetch Unassigned of node)
                                              (MAX (fetch MinIndent of (fetch Environment of context))
                                                   (MIN (fetch MaxIndent of (fetch Environment of context))
                                                        (IPLUS paren.width (fetch (EditNode ActualWidth)
                                                                                  of (CADR (fetch SubNodes
                                                                                              of node)))
                                                               space.width]
                                     else (fetch MinIndent of (fetch Environment of context]
             (\\set.comment.positions comment.start.x comment.indent (IPLUS paren.width (fetch StartX
                                                                                            of node))
                    paren.width node context)
       for subnode in (CDR (fetch SubNodes of node))
       do
       (if (EQ (fetch NodeType of subnode)
               \\type.comment)
           then (if index
                    then (SETQ index (AND (NEQ index 1)
                                          (SUB1 index)))
                  else [if (EQ last.comment.level (fetch Unassigned of subnode))
                           then ;; we're following a comment of the same level.  force a cr and extra line space
                                (\\output.cr context (\\select.comment.indent (fetch Unassigned of subnode)
                                                                              comment.indent indent (fetch StartX
                                                                                                       of (fetch Root
                                                                                                             of context)))
                                        8)
                         elseif (AND first.subnode (NEQ (fetch Unassigned of subnode)
                                                        1))
                           then ;; dont' have to move at all
                         elseif (OR first.subnode (AND (EQ (fetch Unassigned of subnode)
                                                           1)
                                                       same.line?
                                                       (ILEQ (fetch CurrentX of context)
                                                             comment.start.x)))
                           then ;; just space if first subnode or its a single semi comment that will fit
                                (\\output.space context (IDIFFERENCE comment.indent (fetch CurrentX
                                                                                       of context)))
                         else (\\output.cr context (\\select.comment.indent (fetch Unassigned of subnode)
                                                                            comment.indent indent (fetch StartX
                                                                                                     of (fetch Root of context]
                              (\\linearize subnode context))
                (SETQ same.line? NIL)
                (SETQ last.comment.level (fetch Unassigned of subnode))
         else (if index
                  then NIL
                elseif first.subnode
                  then [if (NOT same.line?)
                           then (\\output.cr context (IPLUS paren.width (fetch StartX of node]
                       else (if (AND same.line? (NEQ same.line? (QUOTE paren))
                                     (LEQ (IPLUS (fetch CurrentX of context)
                                                 space.width)
                                          indent))
                                then                                    ; we're to the left of the indentation tab, so just space enough to
                                                                        ; get there
                                     (\\output.space context (IDIFFERENCE indent (fetch CurrentX of context)))
                              elseif (AND same.line? (NEQ same.line? (QUOTE paren))
                                          (fetch InlineWidth subnode)
                                          (LEQ (IPLUS (fetch CurrentX of context)
                                                      space.width
                                                      (fetch InlineWidth subnode)
                                                      (if (AND (CDR $$LST1)
                                                               (EQ (fetch NodeType of (CADR $$LST1))
                                                                   \\type.comment)
                                                               (EQ (fetch Unassigned of (CADR $$LST1))
                                                                   1))
```

```
                                                    then (fetch PreferredWidth of (CADR $$LST1))
                                                   else 0))
                                             (fetch RightMargin of node))
                                        (OR (EQ same.line? T)
                                            (ILESSP (CAR (fetch SubNodes of subnode))
                                                   2)))
                                   then                                          ; it will fit on this line
                                        (\\output.space context space.width)
                                  else (\\output.cr context indent)))
                        (SETQ same.line? (OR (AND (if index
                                                      then (SETQ index (AND (NEQ index 1)
                                                                            (SUB1 index)))
                                                           (fetch Inline? of subnode)
                                                     else (\\linearize subnode context))
                                                 (OR (ILESSP (CAR (fetch SubNodes of subnode))
                                                             2)
                                                     (QUOTE no.lists)))
                                             (QUOTE paren)))
                        (SETQ last.comment.level NIL))
                  (SETQ first.subnode NIL)
                  finally (if (NULL same.line?)
                              then (\\output.cr context (IPLUS paren.width (fetch StartX of node]
    (if index
        then (SHOULDNT "linearize index out of range"))
    (\\output.constant.string context (fetch RParenString of (fetch Environment of context])

)


;;; Patch to fix deadly bug when typing non-list after PROG, LAMBDA, etc

(DEFINEQ

(\\reparse.litatom
  [LAMBDA (node mode context)                                          (* Koomen "29-Apr-88 17:05")
                                                                       (* mdd " 5-Sep-86 12:23")
    ;; Koomen [4/29/88] Patched to avoid severe problem with SEdit attempts to reparse the CADR of a LAMBDA/LET/PROG list after typing a litatom
    ;; instead of a ParamList
          (* this atom is either (a) switching to or from a keyword, or (b) a NIL which is to be parsed as an empty list)

    (if (AND (PROGN

          (* * This doesn't work. Try leaving it alone.)

                 NIL)
           (NULL (fetch Structure of node))
           (EQ mode (QUOTE BindingList))
           (NEQ (fetch PointNode of (fetch CaretPoint of context))
                node))
       then (\\reparse.list node mode context)
      elseif (EQ mode (QUOTE BindingList))
       then NIL
      elseif (NEQ (SETQ mode (AND (EQ mode (QUOTE KeyWord))
                                  (QUOTE KeyWord)))
                  (fetch ParseMode of node))
       then (replace ParseMode of node with mode)
            (\\note.change node context])

)


;;; I/O Patch

(DECLARE: DONTEVAL@LOAD DOCOPY

(CHANGENAME (QUOTE \\sedit)
       (QUOTE READP)
       (QUOTE \SYSBUFP))
)


;;; Give Dorado a BQUOTE character

(DECLARE: DONTEVAL@LOAD DOCOPY

[COND
   ((EQ (MACHINETYPE)
        (QUOTE DORADO))
    (METASHIFT T)
    ;; Make BLANK-TOP key a BQUOTE
    (KEYACTION (QUOTE BLANK-TOP)
           (QUOTE ((96 96]
)

(PUTPROPS SEDIT-PATCHES COPYRIGHT ("Johannes A. G. M. Koomen" 1987 1988))
```

FUNCTION INDEX

VARIABLE INDEX