

File created: 26-Jan-90 10:14:54 {DSK}/users/welch/migration/TRANSLATOR-ASSISTANT.;5

changes to: (IL:VARIABLES *DEFAULT-OPTIONS-FREEMENU-DESCRIPTION* *OPTIONS-FREEMENU-DESCRIPTION*)

previous date: 25-Jan-90 14:46:05 {DSK}/users/welch/migration/TRANSLATOR-ASSISTANT.;4

Read Table: XCL

Package: IL-CONVERT

Format: XCCS

; Copyright (c) 1989, 1990 by ENVOS Corporation. All rights reserved.

(IL:RPAQQ **IL:TRANSLATOR-ASSISTANTCOMS**

((IL:DECLARE\ : IL:EVAL@LOAD IL:EVAL@COMPILE IL:DONTCOPY

;; Sure wish FILESLOAD did what I want...

(IL:P (MAPC #' (LAMBDA (X)

(IL:LOAD X 'IL:PROP))

' (IL:SEDI-DECLS IL:FILEPKGRECORDS IL:TABLEBROWSERDECLS))))

(IL:VARIABLES *DEF-EDITOR-ADDED-SEDI-COMMANDS* *DEF-EDITOR-SEDI-COMTAB-SPEC*

DEF-EDITOR-SEDI-COMTAB *DEF-EDITOR-SEDI-ENVIRONMENT*

DEFAULT-OPTIONS-FREEMENU-DESCRIPTION *ITERATION-OPERS* *OPTIONS-FREEMENU-DESCRIPTION*

TRANSLATOR-CONTROL-FREEMENU-DESCRIPTION *TRANSLATOR-OPTIONS* *USER-ADDED-TEMPLATES*

USER-ADDED-TRANSLATOR-MACROS *WARN-FOR-COMPLEX-I.S.-FORMS*

FILE-CONTEXT-FREEMENU-DESCRIPTION *FILE-CONTEXTS*)

(IL:STRUCTURES FILE-CONTEXT WARNING-ENTRY WARNING-HEADER)

(IL:FUNCTIONS ADD\ 'L-FILE-SELECTEDFN COMPARE-PARAMETER-LISTS CONVERT-FILE-FOR-CONTEXT

CONVERT-ONE-FILECOM-FOR-CONTEXT CONVERT-SETFS-FILECOM CONVERT-UNKNOWN-COM

DEF-EDITOR-BROWSE-WARNINGS DEF-EDITOR-CONVERT-STRUCTURE DEF-WARNING-BROWSER-DELETEDFN

DEF-WARNING-BROWSER-SELECTEDFN ENTER-NEW-MACRO-TEMPLATE FILE-WARNING-BROWSER-DELETEDFN

FILE-WARNING-BROWSER-SELECTEDFN FS-WINDOW-ADD\ 'L-FILE-BROWSER FS-WINDOW-BROWSE-TRANSLATIONS

FS-WINDOW-BROWSE-WARNINGS FS-WINDOW-DO-TRANSLATE FS-WINDOW-EDIT-OPTIONS FS-WINDOW-ICONFN

FS-WINDOW-SAVE-CONTEXT FS-WINDOW-SELECT-NEXT-DEFINITION FS-WINDOW-WRITE-TRANSLATION

FS-WINDOW-WRITE-TRANSLATION-INTERNAL GET-FILE-CONTEXT-OPTION INVERT-CALLER-LIST

LOCATE-SUBEXPRESSION MAKE-WARNINGS-BROWSER MAP-INTO-CONTEXT MAPCAN-INTO-CONTEXT MENU-CHOOSE

NEW-FILE-CONTEXT NTH-SUBEXPRESSION OPTIONS-EDITOR-ACCEPT OPTIONS-EDITOR-BROWSE-TEMPLATES

OPTIONS-EDITOR-BROWSE-TRANSLATOR-MACROS OPTIONS-EDITOR-CLOSEW OPTIONS-EDITOR-NEW-TEMPLATE

OPTIONS-EDITOR-NEW-TRANSLATOR-MACRO OPTIONS-EDITOR-REVERT PACKAGE-VALIDATIONFN

PARSE-FILENAME-FOR-FILE-CONTEXT PROMPT-FOR-FILENAME RECORD-WARNING TEMPLATE-BROWSER-CLOSEFN

TEMPLATE-BROWSER-SELECTEDFN TRANSLATION-BROWSER-SELECTEDFN TRANSLATOR-EDIT-DEFAULT-OPTIONS

TRANSLATOR-NEW-FILE-CONTEXT TRANSLATOR-NEW-FILE-CONTEXT-INTERNAL

TRANSLATOR-NOTE-ADDITIONAL-FILE TRANSLATOR-READ-SAVE-CONTEXT TRANSMACRO-BROWSER-CLOSEFN

TRANSMACRO-BROWSER-SELECTEDFN UNKNOWN-MACRO-FORM WALKER-FIND-PARAMETER-LIST

WARN-FOR-PARM-CHANGES WARNING-DEFINITIONS-BROWSER-SELECTEDFN WB.BUTTONEVENTFN WB.CLOSEFN

WB.DO.ITEM.SELECTION WITH-FILE-CONTEXT-OPTIONS YESNOCHANGESTATE)

(IL:VARIABLES *TRANSLATOR-MAIN-MENU*)

(IL:P (IL:TOTOPW *TRANSLATOR-MAIN-MENU*)))

(IL:DECLARE\ : IL:EVAL@LOAD IL:EVAL@COMPILE IL:DONTCOPY

(MAPC #' (LAMBDA (X)

(IL:LOAD X 'IL:PROP))

' (IL:SEDI-DECLS IL:FILEPKGRECORDS IL:TABLEBROWSERDECLS))

(DEFPARAMETER ***DEF-EDITOR-ADDED-SEDI-COMMANDS*** ' ((DEF-EDITOR-CONVERT-STRUCTURE NIL T "1,=")))

(DEFPARAMETER ***DEF-EDITOR-SEDI-COMTAB-SPEC*** (APPEND *DEF-EDITOR-ADDED-SEDI-COMMANDS*
SEDI::COMMAND-TABLE-SPEC))

(DEFPARAMETER ***DEF-EDITOR-SEDI-COMTAB*** (SEDI::CREATE-COMMAND-TABLE *DEF-EDITOR-SEDI-COMTAB-SPEC*))

(DEFPARAMETER ***DEF-EDITOR-SEDI-ENVIRONMENT*** (IL:CREATE SEDI::EDIT-ENV IL:USING

SEDI::LISP-EDIT-ENVIRONMENT

SEDI::COMMAND-TABLE IL:_)

(FIRST

DEF-EDITOR-SEDI-COMTAB

)

SEDI::HELP-MENU IL:_)

(SECOND

DEF-EDITOR-SEDI-COMTAB

)))

(DEFPARAMETER ***DEFAULT-OPTIONS-FREEMENU-DESCRIPTION***

' ((IL:PROPS IL:FONT (IL:TIMESROMAN 10))

((TYPE IL:DISPLAY IL:LABEL "Warn for complex I.S. forms")

(TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *WARN-FOR-COMPLEX-I.S.-FORMS* IL:FONT

(IL:GACHA 10 IL:BOLD)))

((TYPE IL:DISPLAY IL:LABEL "Unknown macros: ")

(TYPE IL:NWAY IL:COLLECTION *UNKNOWN-MACRO-ACTION* IL:ID :UM-SILENT IL:LABEL "ignore" IL:NWAYPROPS

(IL:INITSTATE (IL:GROUP :UM-WARN)))

```
(TYPE IL:NWAY IL:COLLECTION *UNKNOWN-MACRO-ACTION* IL:ID :UM-WARN IL:LABEL "warn")
(TYPE IL:NWAY IL:COLLECTION *UNKNOWN-MACRO-ACTION* IL:ID :UM-PROMPT IL:LABEL "prompt")
((TYPE IL:EDITSTART IL:LINKS (IL:EDIT (IL:GROUP *PACKAGE-FOR-IL-SYMBOLS*))
  IL:LABEL "Package for IL symbols:")
 (TYPE IL:EDIT IL:LABEL "" IL:ID *PACKAGE-FOR-IL-SYMBOLS* IL:FONT (IL:GACHA 10 IL:BOLD)
  IL:MAXWIDTH 100))
((TYPE IL:EDITSTART IL:LINKS (IL:EDIT (IL:GROUP *PACKAGE-FOR-RESULT-FILE*))
  IL:LABEL "Package of result file:")
 (TYPE IL:EDIT IL:LABEL "" IL:ID *PACKAGE-FOR-RESULT-FILE* IL:FONT (IL:GACHA 10 IL:BOLD)
  IL:MAXWIDTH 100 VALIDATION-FN PACKAGE-VALIDATIONFN IL:LIMITCHARS OPTIONS-LIMITCHARFN))
((TYPE IL:DISPLAY IL:LABEL "All parameters optional")
 (TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *PARAMETERS-ALWAYS-OPTIONAL* IL:FONT
  (IL:GACHA 10 IL:BOLD)))
((TYPE IL:DISPLAY IL:LABEL "Extra arguments ignored")
 (TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *ADD-REST-ARG* IL:FONT (IL:GACHA 10
  IL:BOLD)))
)
((TYPE IL:DISPLAY IL:LABEL "Warn for all IL symbols")
 (TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *WARN-FOR-ALL-IL-SYMBOLS* IL:FONT
  (IL:GACHA 10 IL:BOLD)))
((TYPE IL:EDITSTART IL:LINKS (IL:EDIT (IL:GROUP *WARN-FOR-IL-SYMBOLS-LOWER-THAN-THIS*))
  IL:LABEL "Warn for IL symbols below: ")
 (TYPE IL:EDIT IL:LABEL "NIL" IL:ID *WARN-FOR-IL-SYMBOLS-LOWER-THAN-THIS* IL:FONT (IL:GACHA 10 IL:BOLD)
  IL:MAXWIDTH 100))
((TYPE IL:DISPLAY IL:LABEL "GetValue forms: ")
 (IL:PROPS IL:LEFT 10))
(TYPE IL:NWAY IL:COLLECTION *GETVALUE-TRANSLATION* IL:ID :SLOT-VALUE IL:LABEL "slot-value" IL:NWAYPROPS
  (IL:INITSTATE (IL:GROUP :SLOT-VALUE)))
(TYPE IL:NWAY IL:COLLECTION *GETVALUE-TRANSLATION* IL:ID :ACCESSOR IL:LABEL "accessor")
(TYPE IL:NWAY IL:COLLECTION *GETVALUE-TRANSLATION* IL:ID :ACTIVE-VALUE IL:LABEL "active value system"))
((TYPE IL:DISPLAY IL:LABEL "Slot for IV Props in every class")
 (TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *ALWAYS-INCLUDE-PROPS* IL:FONT
  (IL:GACHA 10 IL:BOLD)))
((TYPE IL:DISPLAY IL:LABEL "CL Macro Templates:")
 (IL:LABEL "Browse" IL:SELECTEDFN OPTIONS-EDITOR-BROWSE-TEMPLATES IL:BOX 1 IL:FONT (IL:GACHA 10))
 (IL:LABEL "New" IL:SELECTEDFN OPTIONS-EDITOR-NEW-TEMPLATE IL:BOX 1 IL:FONT (IL:GACHA 10)))
((TYPE IL:DISPLAY IL:LABEL "IL Translator Macros:")
 (IL:LABEL "Browse" IL:SELECTEDFN OPTIONS-EDITOR-BROWSE-TRANSLATOR-MACROS IL:BOX 1 IL:FONT (IL:GACHA 10))
 (IL:LABEL "New" IL:SELECTEDFN OPTIONS-EDITOR-NEW-TRANSLATOR-MACRO IL:BOX 1 IL:FONT (IL:GACHA 10)))
((IL:LABEL "Accept" IL:SELECTEDFN OPTIONS-EDITOR-ACCEPT IL:BOX 1 IL:FONT (IL:GACHA 10))
 (IL:LABEL "Abort" IL:SELECTEDFN OPTIONS-EDITOR-CLOSEW IL:BOX 1 IL:FONT (IL:GACHA 10)
  IL:HJUSTIFY IL:CENTER)
 (IL:LABEL "Revert" IL:SELECTEDFN OPTIONS-EDITOR-REVERT IL:BOX 1 IL:FONT (IL:GACHA 10)
  IL:HJUSTIFY IL:RIGHT)))
```

(DEFPARAMETER *ITERATION-OPERS* NIL)

(DEFPARAMETER *OPTIONS-FREEMENU-DESCRIPTION*

```
' (IL:PROPS IL:FONT (IL:TIMESROMAN 10))
((TYPE IL:DISPLAY IL:LABEL "Warn for complex I.S. forms")
 (TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *WARN-FOR-COMPLEX-I.S.-FORMS* IL:FONT
  (IL:GACHA 10 IL:BOLD)))
((TYPE IL:DISPLAY IL:LABEL "Unknown macros: ")
 (TYPE IL:NWAY IL:COLLECTION *UNKNOWN-MACRO-ACTION* IL:ID :UM-SILENT IL:LABEL "ignore" IL:NWAYPROPS
  (IL:INITSTATE (IL:GROUP :UM-WARN)))
 (TYPE IL:NWAY IL:COLLECTION *UNKNOWN-MACRO-ACTION* IL:ID :UM-WARN IL:LABEL "warn")
 (TYPE IL:NWAY IL:COLLECTION *UNKNOWN-MACRO-ACTION* IL:ID :UM-PROMPT IL:LABEL "prompt"))
((TYPE IL:EDITSTART IL:LINKS (IL:EDIT (IL:GROUP *PACKAGE-FOR-IL-SYMBOLS*))
  IL:LABEL "Package for IL symbols:")
 (TYPE IL:EDIT IL:LABEL "" IL:ID *PACKAGE-FOR-IL-SYMBOLS* IL:FONT (IL:GACHA 10 IL:BOLD)
  IL:MAXWIDTH 100 VALIDATION-FN PACKAGE-VALIDATIONFN))
((TYPE IL:EDITSTART IL:LINKS (IL:EDIT (IL:GROUP *PACKAGE-FOR-RESULT-FILE*))
  IL:LABEL "Package of result file:")
 (TYPE IL:EDIT IL:LABEL "" IL:ID *PACKAGE-FOR-RESULT-FILE* IL:FONT (IL:GACHA 10 IL:BOLD)
  IL:MAXWIDTH 100 VALIDATION-FN PACKAGE-VALIDATIONFN))
((TYPE IL:DISPLAY IL:LABEL "All parameters optional")
 (TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *PARAMETERS-ALWAYS-OPTIONAL* IL:FONT
  (IL:GACHA 10 IL:BOLD)))
((TYPE IL:DISPLAY IL:LABEL "Extra arguments ignored")
 (TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *ADD-REST-ARG* IL:FONT (IL:GACHA 10
  IL:BOLD)))
)
((TYPE IL:DISPLAY IL:LABEL "Warn for all IL symbols")
 (TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *WARN-FOR-ALL-IL-SYMBOLS* IL:FONT
  (IL:GACHA 10 IL:BOLD)))
((TYPE IL:DISPLAY IL:LABEL "GetValue forms: ")
 (TYPE IL:NWAY IL:COLLECTION *GETVALUE-TRANSLATION* IL:ID :SLOT-VALUE IL:LABEL "slot-value" IL:NWAYPROPS
  (IL:INITSTATE (IL:GROUP :SLOT-VALUE)))
 (TYPE IL:NWAY IL:COLLECTION *GETVALUE-TRANSLATION* IL:ID :ACCESSOR IL:LABEL "accessor")
 (TYPE IL:NWAY IL:COLLECTION *GETVALUE-TRANSLATION* IL:ID :ACTIVE-VALUE IL:LABEL "active value system"))
((TYPE IL:DISPLAY IL:LABEL "Slot for IV Props in every class")
 (TYPE IL:STATE IL:CHANGESTATE YESNOCHANGESTATE IL:LABEL "No" IL:ID *ALWAYS-INCLUDE-PROPS* IL:FONT
  (IL:GACHA 10 IL:BOLD)))
((TYPE IL:EDITSTART IL:LINKS (IL:EDIT (IL:GROUP *WARN-FOR-IL-SYMBOLS-LOWER-THAN-THIS*))
  IL:LABEL "Warn for IL symbols below: ")
```

```
(TYPE IL:EDIT IL:LABEL "NIL" IL:ID *WARN-FOR-IL-SYMBOLS-LOWER-THAN-THIS* IL:FONT (IL:GACHA 10 IL:BOLD)
IL:MAXWIDTH 100))
((IL:LABEL "Accept" IL:BOX 1 IL:FONT (IL:GACHA 10)
IL:SELECTEDFN OPTIONS-EDITOR-ACCEPT)
(IL:LABEL "Abort" IL:SELECTEDFN OPTIONS-EDITOR-CLOSEW IL:BOX 1 IL:FONT (IL:GACHA 10)
IL:HJUSTIFY IL:CENTER)
(IL:LABEL "Revert" IL:HJUSTIFY IL:RIGHT IL:BOX 1 IL:FONT (IL:GACHA 10)
IL:SELECTEDFN OPTIONS-EDITOR-REVERT)))
```

(DEFPARAMETER ***TRANSLATOR-CONTROL-FREEMENU-DESCRIPTION***

```
'(((IL:LABEL "Code Migration Tool" IL:FONT (IL:MODERN 18 IL:BOLD)
IL:HJUSTIFY IL:CENTER TYPE IL:DISPLAY))
((IL:PROPS IL:FONT (IL:MODERN 12))
(IL:LABEL "New File" IL:BOX 2 IL:SELECTEDFN TRANSLATOR-NEW-FILE-CONTEXT IL:MESSAGE "Open a new File
Window, prompting for file name.")
(IL:LABEL "Get State" IL:BOX 2 IL:SELECTEDFN TRANSLATOR-READ-SAVED-CONTEXT IL:MESSAGE "Restore a saved
File Context to a window")
(IL:LABEL "Edit Default Options" IL:BOX 2 IL:SELECTEDFN TRANSLATOR-EDIT-DEFAULT-OPTIONS IL:MESSAGE
"Open an editor for the default translation options"))))
```

(DEFPARAMETER ***TRANSLATOR-OPTIONS***

```
'(*WARN-FOR-COMPLEX-I.S.-FORMS* *UNKNOWN-MACRO-ACTION*
*PACKAGE-FOR-IL-SYMBOLS* *PACKAGE-FOR-RESULT-FILE*
*PARAMETERS-ALWAYS-OPTIONAL* *ADD-REST-ARG*
*WARN-FOR-ALL-IL-SYMBOLS* *WARN-FOR-IL-SYMBOLS-LOWER-THAN-THIS*
*GETVALUE-TRANSLATION*)
"A list of the names of special variables that are translator
options."
```

(DEFVAR ***USER-ADDED-TEMPLATES*** NIL)

(DEFVAR ***USER-ADDED-TRANSLATOR-MACROS*** NIL)

(DEFVAR ***WARN-FOR-COMPLEX-I.S.-FORMS*** NIL)

(DEFPARAMETER ***FILE-CONTEXT-FREEMENU-DESCRIPTION***

```
'(((IL:LABEL "File:" TYPE IL:DISPLAY IL:FONT (IL:MODERN 12))
(IL:ID FILENAME TYPE IL:DISPLAY IL:LABEL "")
(IL:LABEL "Put State" IL:BOX 2 IL:FONT (IL:MODERN 12)
IL:HJUSTIFY IL:RIGHT IL:SELECTEDFN FS-WINDOW-SAVE-CONTEXT))
((IL:LABEL "Directory:" TYPE IL:DISPLAY IL:FONT (IL:MODERN 12))
(IL:ID DIRNAME TYPE IL:DISPLAY IL:LABEL ""))
(IL:PROPS IL:FONT (IL:MODERN 12))
(IL:LABEL "Translate" IL:BOX 2 IL:SELECTEDFN FS-WINDOW-DO-TRANSLATE IL:MESSAGE "Translate from source
code, discarding any current warnings.")
(IL:LABEL "Options" IL:BOX 2 IL:SELECTEDFN FS-WINDOW-EDIT-OPTIONS IL:MESSAGE "Edit translation options for
this file")
(IL:LABEL "Write Translation" IL:BOX 2 IL:SELECTEDFN FS-WINDOW-WRITE-TRANSLATION IL:MESSAGE "Write the
translated file to disk")
(IL:LABEL "Next Definition" IL:BOX 2 IL:SELECTEDFN FS-WINDOW-SELECT-NEXT-DEFINITION IL:MESSAGE "Edit the
next definition in the warnings browser"))))
```

(DEFVAR ***FILE-CONTEXTS*** NIL)

(DEFSTRUCT **FILE-CONTEXT**

```
(:PRINT-FUNCTION (LAMBDA (SELF STREAM DEPTH)
(DECLARE (IGNORE DEPTH))
(FORMAT STREAM "#<~a for ~a @ ~o,~o>" (TYPE-OF SELF)
(FILE-CONTEXT-FILENAME SELF)
(IL:\\HILOC SELF)
(IL:\\LOLOC SELF))))
```

- (FILENAME)
- (DIRNAME)
- (DEFINITIONS)
- (DEFINITIONS-LAST-CONS)
- (WARNINGS)
- (FUNCTION-CALLS)
- (FREE-REFERENCES)
- (OPTIONS)
- (MENU)
- (EXPORTED-SYMS)

(DEFSTRUCT **WARNING-ENTRY**

```
(:CONC-NAME "WE-")
(:TYPE LIST))
STRING
EXPRESSION
DELETED)
```

(DEFSTRUCT **WARNING-HEADER**

```
(:CONC-NAME "WH-")
```



```

(DEFUN CONVERT-UNKNOWN-COM (COM)
  (MAP-INTO-CONTEXT #'(LAMBDA (C)
    (WARN "Unable to translate a ~a filecom." (CAR C))
    C)
    (LIST COM)))

(DEFUN DEF-EDITOR-BROWSE-WARNINGS (WINDOW WARNINGS &OPTIONAL FILE-CONTEXT)
  (WHEN WARNINGS
    (LET ((WB (MAKE-WARNINGS-BROWSER (MAPCAR #'(LAMBDA (W)
      (IL:CREATE IL:TABLEITEM
        IL:TIDATA IL:_ W
        IL:TIDELETED IL:_ (THIRD W))))
      WARNINGS)
      "Warnings for definition" WINDOW 'IL:TOP
      #'(LAMBDA (TB ITEM WINDOW)
        (DECLARE (IGNORE TB))
        (PRINC (FIRST (IL:|fetch| IL:TIDATA IL:|of| ITEM))
          WINDOW))
      :SELECTEDFN
      'DEF-WARNING-BROWSER-SELECTEDFN :DELETEDFN 'DEF-WARNING-BROWSER-DELETEDFN
      :MAINWINDOWPROP 'WARNINGS-BROWSER)))
    (IL:WINDOWPROP WB 'FILE-CONTEXT FILE-CONTEXT)
    (IL:WINDOWADDPROP WB 'IL:CLOSEFN #'(LAMBDA (W)
      (IL:WINDOWPROP W 'FILE-CONTEXT NIL)))
    WB)))

```

```

(DEFUN DEF-EDITOR-CONVERT-STRUCTURE (SEDT::CONTEXT)
  ;; This code all stolen from SEDIT::MUTATE.
  (LET* ((SEDT::PROMPTWINDOW (SEDT:GET-PROMPT-WINDOW SEDT::CONTEXT))
    (SEDT::SELECTION (IL:|fetch| SEDT::SELECTION IL:|of| SEDT::CONTEXT))
    (SEDT::NODE (IL:|fetch| SEDT::SELECT-NODE IL:|of| SEDT::SELECTION)))
    (COND
      ((AND SEDT::NODE (EQ (IL:|fetch| SEDT::SELECT-TYPE IL:|of| SEDT::SELECTION)
        'STRUCTURE)
        (NULL (IL:|fetch| SEDT::SELECT-START IL:|of| SEDT::SELECTION)))
        (UNLESS (SEDT::DO-MUTATION SEDT::CONTEXT SEDT::NODE 'CONVERT)
          (IL:|printout| SEDT::PROMPTWINDOW T "Error during mutation. No changes made.")))
        (T (IL:|printout| SEDT::PROMPTWINDOW T "Select whole structure to mutate."))))
    T)

```

```

(DEFUN DEF-WARNING-BROWSER-DELETEDFN (BROWSER ITEM WINDOW)
  (DECLARE (SPECIAL IL:PROMPTWINDOW))
  ;; Record deletedness in the warning-entry itself, so that it dumps out properly.
  (SETF (WE-DELETED (IL:FETCH IL:TIDATA IL:OF ITEM))
    (IL:FETCH IL:TIDELETED IL:OF ITEM))
  ;; Ripple the consequences back thru the browsers...
  (LET* ((WSTR (FIRST (IL:FETCH IL:TIDATA IL:OF ITEM)))
    (FWB (IL:WINDOWPROP (FILE-CONTEXT-MENU (IL:WINDOWPROP WINDOW 'FILE-CONTEXT))
      'WARNINGS-BROWSER)))
    ;; If all the instances of this message are deleted in this definition, line out the function name in the definitions browser.
    (BLOCK FOO
      (IL:TB.MAP.ITEMS BROWSER #'(LAMBDA (B I)
        (DECLARE (IGNORE B))
        (LET ((DATA (IL:FETCH IL:TIDATA IL:OF ITEM)))
          (WHEN (STRING= (FIRST DATA)
            WSTR)
            (UNLESS (WE-DELETED DATA)
              ;; (FORMAT IL:PROMPTWINDOW "~&Not all msg in def deleted.")
              (RETURN-FROM FOO NIL))))))
      ;; If we didn't throw out of the block above, it's OK.
      (LET* ((BS&IS (IL:WINDOWPROP WINDOW 'BROWSER-AND-ITEM))
        (WDB&I (CAR BS&IS))
        (FWB&I (CDR BS&IS))
        (B (CAR WDB&I))
        (I (CDR WDB&I)))
        (SETF (FOURTH (IL:FETCH IL:TIDATA IL:OF I))
          T) ; mark deleted.
        (IL:TB.DELETE.ITEM B I))
      ;; If all the items in that browser are deleted, knock off the message from the fwb.
      (IL:TB.MAP.ITEMS B #'(LAMBDA (B I)
        (DECLARE (IGNORE B))
        (UNLESS (IL:FETCH IL:TIDELETED IL:OF I)
          ;; (FORMAT IL:PROMPTWINDOW "~&Not all msg deleted.")
          (RETURN-FROM FOO NIL))))
        (SETF (SECOND (IL:FETCH IL:TIDATA IL:OF (CDR FWB&I)))
          T) ; mark deleted.

```

```
(IL:TB.DELETE.ITEM (CAR FWB&I)
(CDR FWB&I))))))
```

```
(DEFUN DEF-WARNING-BROWSER-SELECTEDFN (BROWSER ITEM WINDOW)
(LET* ((ECX (IL:WINDOWPROP (IL:WINDOWPROP WINDOW 'IL:MAINWINDOW)
'SEDIT::EDIT-CONTEXT))
(PROMPTWINDOW (SEdit:GET-PROMPT-WINDOW ECX))
(STRUC (SECOND (IL:FETCH IL:TIDATA IL:OF ITEM)))
(CONTEXT (IL:WINDOWPROP WINDOW 'FILE-CONTEXT))
(OPTS (FILE-CONTEXT-OPTIONS CONTEXT))
(*PACKAGE* (PROGV (CAR OPTS)
(CADR OPTS)
(FIND-PACKAGE *PACKAGE-FOR-RESULT-FILE*)))
(*READTABLE* (IL:\\GTREADTABLE "LISP" T)))
(SEDIT::SELECTION-DOWN ECX)
(LET ((TARGET (SEDIT::FIND-STRUCTURE STRUC (SEDIT::SUBNODE 1 (IL:FETCH SEDIT::ROOT IL:OF ECX))
NIL)))
(COND
(TARGET (IL:CLEARW PROMPTWINDOW)
(SEDIT::SELECT-NODE ECX TARGET T T)
(SEDIT::UPDATE ECX)
(SEDIT::NORMALIZE-SELECTION ECX))
(T (WRITE-STRING "Oops! Structure not found." PROMPTWINDOW))))))
```

```
(DEFUN ENTER-NEW-MACRO-TEMPLATE (FORM)
;; Edits a dummy template for the macro operator in FORM.
(LET ((TEMPLATE (LIST 'DEFTEMPLATE (OR (AND FORM (CAR FORM))
SEDIT::BASIC-GAP)
SEDIT::BODY-GAP))
(EDITOR-NAME (CONCATENATE 'STRING "Macro template" (AND FORM " for ")
(AND FORM (STRING (CAR FORM))))))
(*PACKAGE* (FIND-PACKAGE "IL-CONVERT"))
(*READTABLE* (IL:\\GTREADTABLE "XCL" T)))
(FLET ((COMPLETION-OF-TEMPLATE-EDIT (SEDIT-CONTEXT STRUCTURE)
(WHEN (SYMBOLP (SECOND STRUCTURE))
;; i.e. not a gap any more...
(PUSH (CDR STRUCTURE)
*USER-ADDED-TEMPLATES*)
(PUSH (CDR STRUCTURE)
*WALKER-TEMPLATES*))))
(SEDIT:SEDIT TEMPLATE `(:NAME ,EDITOR-NAME :TYPE :MACRO-TEMPLATE :COMPLETION-FN
,#'COMPLETION-OF-TEMPLATE-EDIT))))))
```

```
(DEFUN FILE-WARNING-BROWSER-DELETEDFN (BROWSER ITEM WINDOW)
;; Record deletedness in the warning-header itself, so that it dumps out properly.
(SETF (SECOND (IL:FETCH IL:TIDATA IL:OF ITEM))
(IL:FETCH IL:TIDELETED IL:OF ITEM)))
```

```
(DEFUN FILE-WARNING-BROWSER-SELECTEDFN (BROWSER ITEM WINDOW)
(LET* ((ENTRY (IL:FETCH IL:TIDATA IL:OF ITEM))
(WARNING-STRING (CAR ENTRY))
(DEF-ENTRIES (CDDR ENTRY))
(WARNING-DEFS-BROWSER (IL:WINDOWPROP WINDOW 'WARNING-DEFS-BROWSER))
(TBITEMS (MAPCAR #'(LAMBDA (X)
(IL:CREATE IL:TABLEITEM
IL:TIDATA IL:_ X
IL:TIDELETED IL:_ (FOURTH X)))
DEF-ENTRIES)))
(IF (NULL WARNING-DEFS-BROWSER)
(SETQ WARNING-DEFS-BROWSER (MAKE-WARNINGS-BROWSER
:
TBITEMS WARNING-STRING WINDOW 'IL:RIGHT
#' (LAMBDA (TB ITEM WINDOW)
(DECLARE (IGNORE TB))
(LET ((STUFF (IL:|fetch| IL:TIDATA IL:|of| ITEM)))
(PRINC (SECOND STUFF)
WINDOW)
(WRITE-CHAR #\Space WINDOW)
(PRINC (FIRST STUFF)
WINDOW))))
:SELECTEDFN
'WARNING-DEFINITIONS-BROWSER-SELECTEDFN :MAINWINDOWPROP
'WARNING-DEFS-BROWSER))
(IL:WINDOWPROP WARNING-DEFS-BROWSER 'IL:TITLE WARNING-STRING))
(IL:TB.REPLACE.ITEMS (IL:WINDOWPROP WARNING-DEFS-BROWSER 'IL:TABLEBROWSER)
TBITEMS)
(IL:WINDOWPROP WARNING-DEFS-BROWSER 'BROWSER-AND-ITEM (CONS BROWSER ITEM))))))
```

```
(DEFUN FS-WINDOW-ADD'L-FILE-BROWSER (FS-WINDOW)
```

```
(OR (IL:WINDOWPROP FS-WINDOW 'ADD\L-FILE-BROWSER)
  (MAKE-WARNINGS-BROWSER NIL "Additional files" FS-WINDOW 'IL:TOP
    #'(LAMBDA (TB ITEM WINDOW)
      (PRINC (FIRST (IL:|fetch| IL:TIDATA IL:|of| ITEM)
        WINDOW))
      :SELECTEDFN
      'ADD\L-FILE-SELECTEDFN :MAINWINDOWPROP 'ADD\L-FILE-BROWSER)))
```

```
(DEFUN FS-WINDOW-BROWSE-TRANSLATIONS (ITEM WINDOW BUTTONS)
  (LET* ((BROWSER (IL:WINDOWPROP WINDOW 'TRANSLATION-BROWSER))
    (CONTEXT (IL:WINDOWPROP WINDOW 'FILE-CONTEXT))
    (OPTS (FILE-CONTEXT-OPTIONS CONTEXT))
    (PRINT-PACKAGE (PROGV (CAR OPTS)
      (CADR OPTS)
      (FIND-PACKAGE *PACKAGE-FOR-RESULT-FILE*)))
    (PRINT-RDTBL (IL:\GTREADTABLE "LISP" T))
    (TBITEMS (MAPCAR #'(LAMBDA (X)
      (IL:CREATE IL:TABLEITEM
        IL:TIDATA IL:_ X)
      (FILE-CONTEXT-DEFINITIONS CONTEXT))))))
  (IF (NULL BROWSER)
    (IL:WINDOWPROP WINDOW
      'TRANSLATION-BROWSER
      (SETQ BROWSER (MAKE-WARNINGS-BROWSER
        TBITEMS "Translations" WINDOW 'IL:TOP
        #'(LAMBDA (TB ITEM WINDOW)
          (DECLARE (IGNORE TB))
          (LET ((*PRINT-LEVEL* 1)
            (*PRINT-LENGTH* 3)
            (*PACKAGE* PRINT-PACKAGE)
            (*READTABLE* PRINT-RDTBL))
            (PRINC (IL:|fetch| IL:TIDATA IL:|of| ITEM)
              WINDOW)))
          :SELECTEDFN
          'TRANSLATION-BROWSER-SELECTEDFN :MAINWINDOWPROP 'TRANSLATION-BROWSER)))
    (IL:TB.REPLACE.ITEMS (IL:WINDOWPROP BROWSER 'IL:TABLEBROWSER)
      TBITEMS)))
```

```
(DEFUN FS-WINDOW-BROWSE-WARNINGS (FS-WINDOW)
  (LET* ((FILE-CONTEXT (IL:WINDOWPROP FS-WINDOW 'FILE-CONTEXT))
    (BROWSER (IL:WINDOWPROP FS-WINDOW 'WARNINGS-BROWSER))
    (TBITEMS))
    (MAPHASH #'(LAMBDA (KEY STUFF)
      (WHEN (STRINGP KEY) ; only want warning strings, not definition entries.
        (PUSH (IL:CREATE IL:TABLEITEM
          IL:TIDATA IL:_ (CONS KEY STUFF)
          IL:TIDLETED IL:_ (FIRST STUFF))
          TBITEMS)))
      (FILE-CONTEXT-WARNINGS FILE-CONTEXT)))
  (IF (NULL BROWSER)
    (PROGN ;; Create a new warnings browser window...
      (SETQ BROWSER (MAKE-WARNINGS-BROWSER
        TBITEMS
        (CONCATENATE 'STRING "Warnings for " (STRING (FILE-CONTEXT-FILENAME
          FILE-CONTEXT)))
        FS-WINDOW
        'IL:BOTTOM
        #'(LAMBDA (TB ITEM WINDOW)
          (DECLARE (IGNORE TB))
          (LET ((STUFF (IL:|fetch| IL:TIDATA IL:|of| ITEM))
            (PRINC (FIRST STUFF)
              WINDOW)))
          :SELECTEDFN
          'FILE-WARNING-BROWSER-SELECTEDFN :DELETEDFN
          'FILE-WARNING-BROWSER-DELETEDFN :MAINWINDOWPROP 'WARNINGS-BROWSER)))
      ;; Browser is already there: Reload it.
      (IL:TB.REPLACE.ITEMS (IL:WINDOWPROP BROWSER 'IL:TABLEBROWSER)
        TBITEMS)))
```

```
(DEFUN FS-WINDOW-DO-TRANSLATE (ITEM WINDOW BUTTONS)
  (LET ((FILE-CONTEXT (IL:WINDOWPROP WINDOW 'FILE-CONTEXT))
    (BROWSER (IL:WINDOWPROP WINDOW 'WARNINGS-BROWSER)))
    (WHEN (FILE-CONTEXT-WARNINGS FILE-CONTEXT)
      (UNLESS (MENU-CHOOSE '("Do it" T)
        ("Abort" NIL))
        "Discard all warnings and re-translate?")
      (RETURN-FROM FS-WINDOW-DO-TRANSLATE NIL))
      (IL:CLEARCLISPARRAY))
    (WHEN BROWSER (IL:CLOSEW BROWSER))
    (IL:WINDOWPROP *STANDARD-OUTPUT* 'IL:PAGEFULLFN 'IL:NILL)
    (CONVERT-FILE-FOR-CONTEXT FILE-CONTEXT))
```

```
(IL:CLOSEW *STANDARD-OUTPUT*)
(WHEN (FILE-CONTEXT-WARNINGS FILE-CONTEXT)
      (FS-WINDOW-BROWSE-WARNINGS WINDOW))))
```

```
(DEFUN FS-WINDOW-EDIT-OPTIONS (ITEM WINDOW BUTTONS)
  (DECLARE (IGNORE ITEM BUTTONS))
  (LET ((EDWINDOW (IL:FREEMENU *OPTIONS-FREEMENU-DESCRIPTION* "Options")))
    (IL:ATTACHWINDOW EDWINDOW WINDOW 'IL:TOP 'IL:JUSTIFY 'IL:LOCALCLOSE)
    (OPTIONS-EDITOR-REVERT NIL EDWINDOW NIL)))
```

```
(DEFUN FS-WINDOW-ICONFN (WINDOW &OPTIONAL ICON IGNORE)
  (OR ICON (CONCATENATE 'STRING (STRING (FILE-CONTEXT-FILENAME (IL:WINDOWPROP WINDOW 'FILE-CONTEXT)))
                        " translation")))
```

```
(DEFUN FS-WINDOW-SAVE-CONTEXT (ITEM WINDOW BUTTONS)
```

;; 1. Dump definitions. 2. Dump Options. 3. Dump Warnings, encoded as list of (Name Type Position-in-definitions Deleted) (String Deleted
;; Encoding-of-fragment-position) ...)

```
(LET* ((CONTEXT (IL:WINDOWPROP WINDOW 'FILE-CONTEXT))
       (DEFAULT-FILENAME (IL:PACKFILENAME.STRING 'IL:NAME (FILE-CONTEXT-FILENAME CONTEXT)
                                                'IL:EXTENSION "PST"))
       (FILENAME (PROMPT-FOR-FILENAME WINDOW "File name:" DEFAULT-FILENAME)))
```

;; default the type to "FCX"

```
(WITH-OPEN-FILE
  (OUTSTREAM FILENAME :DIRECTION :OUTPUT :IF-EXISTS :NEW-VERSION)
  (LET ((*PRINT-BASE* 10)
        (*PRINT-CIRCLE* NIL)
        (*PRINT-ARRAY* T)
        (*PRINT-PRETTY* NIL)
        (*READTABLE* (IL:\\GTREADTABLE "XCL" T))
        (*PACKAGE* (FIND-PACKAGE "XCL")))
    (IL:**COMMENT**FLG NIL))
  (WRITE-CHAR #\ ( OUTSTREAM)
  (PRIN1 ' :FILENAME OUTSTREAM)
  (PRINT (FILE-CONTEXT-FILENAME CONTEXT)
         OUTSTREAM)
  (PRINT ' :DIRNAME OUTSTREAM)
  (PRINT (FILE-CONTEXT-DIRNAME CONTEXT)
         OUTSTREAM)
  (PRINT ' :DEFINITIONS OUTSTREAM)
  (WRITE-CHAR #\ ( OUTSTREAM)
  (MAPC #' (LAMBDA (D)
            (PRINT D OUTSTREAM)
            (TERPRI OUTSTREAM)))
        (FILE-CONTEXT-DEFINITIONS CONTEXT))
  (WRITE-CHAR #\ ) OUTSTREAM)
  (PRINT ' :WARNINGS OUTSTREAM)
```

;; Dump the warnings as positions in the definitions structure.

```
(MAPHASH #' (LAMBDA (KEY VALUE)
              (IF (STRINGP KEY)
                  (PRINT (LIST* KEY (FIRST VALUE)
                                (MAPCAR #' (LAMBDA (X)
                                             (LIST (FIRST X)
                                                  (SECOND X)
                                                  NIL
                                                  (FOURTH X))))))
                  OUTSTREAM)
              (PRINT (LIST* KEY (LOCATE-SUBEXPRESSION (FIRST VALUE)
                                                       (FILE-CONTEXT-DEFINITIONS CONTEXT))
                             (MAPCAR #' (LAMBDA (WE)
                                         (LIST (WE-STRING WE)
                                               (LOCATE-SUBEXPRESSION (WE-EXPRESSION
                                                                    WE)
                                                                    (FIRST VALUE))
                                               (WE-DELETED WE))))
                             (CDR VALUE)))
                    OUTSTREAM)))
  (FILE-CONTEXT-WARNINGS CONTEXT))
  (PRINT ' :OPTIONS OUTSTREAM)
  (PRINT (FILE-CONTEXT-OPTIONS CONTEXT)
         OUTSTREAM)
  (PRINT ' :FUNCTION-CALLS OUTSTREAM)
  (PRINT (FILE-CONTEXT-FUNCTION-CALLS CONTEXT)
         OUTSTREAM)
  (PRINT ' :FREE-REFERENCES OUTSTREAM)
  (PRINT (FILE-CONTEXT-FREE-REFERENCES CONTEXT)
         OUTSTREAM)
  (TERPRI OUTSTREAM)
  (WRITE-CHAR #\ ) OUTSTREAM))))
```

```
(DEFUN FS-WINDOW-SELECT-NEXT-DEFINITION (ITEM WINDOW BUTTON)
```


:: Delete the earlier item and select the subsequent one.

```
(LET* ((BW (IL:WINDOWPROP WINDOW 'WARNINGS-BROWSER))
      (TB (IL:WINDOWPROP BW 'IL:TABLEBROWSER))
      (LASTITEM (GETF (IL:TB.USERDATA TB)
                    'LAST-ITEM))
      (NUMBER (OR (AND LASTITEM (IL:|fetch| IL:TI# IL:|of| LASTITEM))
                 0))
      (NEXTITEM (IL:TB.NTH.ITEM TB (1+ NUMBER)))
      (NEXTITEMBOTTOM (AND NEXTITEM (IL:TB.BOTTOM.OF.ITEM TB NEXTITEM))))
  (WHEN LASTITEM
    (IL:TB.DELETE.ITEM TB LASTITEM)
    (LET ((FN (GETF (IL:TB.USERDATA TB)
                  'IL:DELETEDFN))
          (WHEN FN (FUNCALL FN TB LASTITEM WINDOW))))
      (UNLESS (NULL NEXTITEM)
        (WHEN (IL:|fetch| IL:TIDELETED IL:|of| NEXTITEM)
          ;; Oops, keep looking.
          (INCF NUMBER) ; current "nextitem" #.
          (LOOP (SETQ NEXTITEM (IL:TB.NTH.ITEM TB (INCF NUMBER)))
                (WHEN (NULL NEXTITEM)
                  ;; Out of items... flush.
                  (RETURN-FROM FS-WINDOW-SELECT-NEXT-DEFINITION NIL))
                (UNLESS (IL:FETCH IL:TIDELETED IL:OF NEXTITEM)
                  (SETQ NEXTITEMBOTTOM (IL:TB.BOTTOM.OF.ITEM TB NEXTITEM))
                  (RETURN))))
          (IL:BLTSHADE IL:BLACKSHADE BW IL:TB.LEFT.MARGIN NEXTITEMBOTTOM (IL:WINDOWPROP BW 'IL:WIDTH)
                    (IL:|fetch| IL:TBITEMHEIGHT IL:|of| TB)
                    'IL:INVERT)
          (LET ((FN (GETF (IL:TB.USERDATA TB)
                        'IL:SELECTEDFN))
                (WHEN FN (FUNCALL FN TB NEXTITEM WINDOW)))
            (SETF (GETF (IL:FETCH IL:TBUSERDATA IL:OF TB)
                      'LAST-ITEM)
                  NEXTITEM)
            (IL:BLTSHADE IL:BLACKSHADE BW IL:TB.LEFT.MARGIN NEXTITEMBOTTOM (IL:WINDOWPROP BW 'IL:WIDTH)
                    (IL:|fetch| IL:TBITEMHEIGHT IL:|of| TB)
                    'IL:INVERT))))))
```

```
(DEFUN FS-WINDOW-WRITE-TRANSLATION (ITEM WINDOW BUTTON)
  (LET* ((CONTEXT (IL:WINDOWPROP WINDOW 'FILE-CONTEXT))
        (FILENAME (FILE-CONTEXT-FILENAME CONTEXT))
        (PROMPT-FILENAME (IL:PACKFILENAME.STRING 'IL:NAME FILENAME 'IL:EXTENSION "LISP" 'IL:VERSION NIL))
        (REAL-FILENAME (PROMPT-FOR-FILENAME WINDOW "File name:" PROMPT-FILENAME)))
    (FS-WINDOW-WRITE-TRANSLATION-INTERNAL CONTEXT REAL-FILENAME)))
```

```
(DEFUN FS-WINDOW-WRITE-TRANSLATION-INTERNAL (CONTEXT FILENAME)
  (LET* ((DEFS (FILE-CONTEXT-DEFINITIONS CONTEXT))
        (*PRINT-BASE* 10)
        (IL:*PRINT-SEMICOLON-COMMENTS* T)
        (*PACKAGE* (FIND-PACKAGE (GET-FILE-CONTEXT-OPTION '*PACKAGE-FOR-RESULT-FILE* CONTEXT)))
        (*READTABLE* (IL:\GTTREADTABLE "XCL" T))
        (*PRINT-PRETTY* T)
        (*PRINT-ARRAY* T)
        (*PRINT-CASE* :DOWNCASE)
        (IL:FILELINELENGTH 79)
        (IL:**COMMENT**FLG NIL))
    (DECLARE (SPECIAL IL:PRETTYPRINTMACROS IL:**COMMENT**FLG IL:*PRINT-SEMICOLON-COMMENTS*
                IL:FILELINELENGTH))
    (WITH-OPEN-FILE (OUTSTREAM FILENAME :IF-EXISTS :NEW-VERSION :DIRECTION :OUTPUT :IF-DOES-NOT-EXIST
                    :CREATE)
      (IL:LINELENGTH 79 OUTSTREAM)
      (PRINT '(LET ((*PACKAGE* *PACKAGE*))
                (IN-PACKAGE :INTERLISP :USE NIL :NICKNAMES '(:IL)))
              OUTSTREAM)
      (PRINT `(IN-PACKAGE ,(PACKAGE-NAME *PACKAGE*)
                        :USE
                        ,(MAPCAR 'PACKAGE-NAME (PACKAGE-USE-LIST *PACKAGE*)))
              OUTSTREAM)
      (TERPRI OUTSTREAM)
      (WHEN (FILE-CONTEXT-EXPORTED-SYMS CONTEXT)
        (LET ((ALIST NIL))
          (DOLIST (S (FILE-CONTEXT-EXPORTED-SYMS CONTEXT))
            (LET ((ASSOC (ASSOC (PACKAGE-NAME (SYMBOL-PACKAGE S))
                              ALIST)))
              (IF ASSOC
                (PUSH S (CDR ASSOC))
                (PUSH (CONS (PACKAGE-NAME (SYMBOL-PACKAGE S))
                          (CONS S NIL))
                      ALIST))))
            (DOLIST (A ALIST)
              (PRINT `(EXPORT (MAPCAR 'INTERN ',(MAPCAR 'STRING (CDR A)))
                            ,(STRING (CAR A)))
```

```

        OUTSTREAM)))
      (TERPRI OUTSTREAM))
    (DOLIST (FORM (EXPURGATE-EXTRANEOUS-PROGNS DEFS))
      (WHEN FORM
        (PRINT FORM OUTSTREAM)
        (TERPRI OUTSTREAM))))))

```

```

(DEFMACRO GET-FILE-CONTEXT-OPTION (OPTION CONTEXT)
  `(IF ,CONTEXT
    (NTH (POSITION ,OPTION (FIRST (FILE-CONTEXT-OPTIONS ,CONTEXT)))
      (SECOND (FILE-CONTEXT-OPTIONS ,CONTEXT)))
    ;; If there's no file context, read out the "default" values.
    (SYMBOL-VALUE ,OPTION)))

```

```

(DEFUN INVERT-CALLER-LIST (CALLS-LIST)
  (MAPCAN #'(LAMBDA (X)
    (LET ((VALUE (MAPCAN #'(LAMBDA (Y)
      (AND (MEMBER (CAR X)
        (CDR Y))
        (LIST (CAR Y))))
      CALLS-LIST)))
      (AND VALUE (CONS (CONS (CAR X)
        VALUE)
        NIL))))
    CALLS-LIST))

```

```

(DEFUN LOCATE-SUBEXPRESSION (SUB MAIN)
  (IF (EQ SUB MAIN)
    NIL
    (LET ((POS (DO ((MTAIL MAIN (CDR MTAIL))
      (N 0 (1+ N)))
      ((ATOM MTAIL)
        (AND MTAIL (EQ SUB MTAIL))))
      (WHEN (EQ SUB (CAR MTAIL))
        (RETURN (LIST N)))
      (WHEN (EQ SUB MTAIL)
        (RETURN N))))))
    (OR POS (DO ((MTAIL MAIN (CDR MTAIL))
      M
      (N 0 (1+ N)))
      ((ATOM MTAIL)
        (AND (CONSP (SETQ M (CAR MTAIL)))
          (LET ((SUB-N (LOCATE-SUBEXPRESSION SUB M)))
            (AND SUB-N (RETURN (CONS N SUB-N))))))
        (INCF N))))))

```

```

(DEFUN MAKE-WARNINGS-BROWSER (TABLEITEMS TITLE MAINWINDOW EDGE PRINTFN &KEY SELECTEDFN DELETEDFN
  RIGHTBUTTONFN MAINWINDOWPROP)
  (LET ((WINDOW (IL:DECODE WINDOW.ARG '(0 . 0)
    300 100 TITLE NIL T)))
    (IL:ATTACHWINDOW WINDOW MAINWINDOW EDGE 'IL:JUSTIFY 'IL:LOCALCLOSE)
    ;; Note side effect of installing TB in window...
    (IL:REPLACE IL:TBUSERDATA IL:OF (IL:TB.MAKE.BROWSER TABLEITEMS WINDOW (LIST 'IL:PRINTFN PRINTFN))
      IL:WITH (LIST 'IL:SELECTEDFN SELECTEDFN 'IL:DELETEDFN DELETEDFN 'IL:RIGHTBUTTONFN RIGHTBUTTONFN))
    (IL:WINDOWPROP WINDOW 'IL:BUTTONEVENTFN 'WB.BUTTONEVENTFN)
    (IL:WINDOWPROP WINDOW 'IL:RIGHTBUTTONFN 'WB.BUTTONEVENTFN)
    ;; Store it in main window...
    (IL:WINDOWPROP MAINWINDOW MAINWINDOWPROP WINDOW)
    (IL:WINDOWPROP WINDOW 'MAINWINDOWPROP MAINWINDOWPROP)
    ;; When the browser is closed, delete it from the main window.
    (IL:WINDOWADDPROP WINDOW 'IL:CLOSEFN 'WB.CLOSEFN T)
    WINDOW))

```

```

(DEFUN MAP-INTO-CONTEXT (FN SEQUENCE)
  (LET ((CONTEXT *FILE-CONTEXT*))
    (DOLIST (ITEM SEQUENCE)
      (LET ((*WARNINGS-MADE* NIL))
        (MULTIPLE-VALUE-BIND (CONVERSION CALLS FREEREF)
          (FUNCCALL FN ITEM)
          (COND
            (CONTEXT (WHEN *WARNINGS-MADE*
              ;; Some warning was made during the conversion. Put the converted definition on the definition warning
              ;; record so we can edit it.
              (LET ((ELT (GETHASH (FIRST *WARNINGS-MADE*)
                (FILE-CONTEXT-WARNINGS CONTEXT))))
                (UNLESS ELT (ERROR "Couldn't find entry for ~s." ELT))
                (SETF (FIRST ELT)

```

```

(CONVERSION)))
;; Put the converted definition on the definitions list.
(LET ((APPENDAGE (IF (EQ (FIRST CONVERSION)
                        'PROGN)
                    (CDR CONVERSION)
                    (CONS CONVERSION NIL))))
      (IF (NULL (FILE-CONTEXT-DEFINITIONS-LAST-CONS CONTEXT))
          (SETF (FILE-CONTEXT-DEFINITIONS CONTEXT)
                APPENDAGE
                (FILE-CONTEXT-DEFINITIONS-LAST-CONS CONTEXT)
                (LAST (FILE-CONTEXT-DEFINITIONS CONTEXT))))
          (SETF (CDR (FILE-CONTEXT-DEFINITIONS-LAST-CONS CONTEXT))
                APPENDAGE
                (FILE-CONTEXT-DEFINITIONS-LAST-CONS CONTEXT)
                (LAST (FILE-CONTEXT-DEFINITIONS-LAST-CONS CONTEXT))))))
      (WHEN (AND CALLS (NOT (NULL (CDR CALLS))))
            (PUSH CALLS (FILE-CONTEXT-FUNCTION-CALLS CONTEXT)))
          (WHEN (AND FREEREF'S (NOT (NULL (CDR FREEREF'S))))
                (PUSH FREEREF'S (FILE-CONTEXT-FREE-REFERENCES CONTEXT))))
      NIL)
(T CONVERSION))))))

```

```

(DEFUN MAPCAN-INTO-CONTEXT (HEAD FN SEQUENCE)
  (MAP-INTO-CONTEXT #'(LAMBDA (S)
                    (CONS HEAD (MAPCAN FN S)))
    (LIST SEQUENCE)))

```

```

(DEFUN MENU-CHOOSE (ITEMS TITLE)
  (IL:MENU (IL:create| IL:MENU
           IL:ITEMS IL:_ ITEMS
           IL:TITLE IL:_ TITLE)))

```

```

(DEFUN NEW-FILE-CONTEXT (NAME DIR-AND-HOST)
  (LET* ((MENU (IL:FREEMENU *FILE-CONTEXT-FREEMENU-DESCRIPTION*))
         (CONTEXT (LET* ((MFE (GET NAME 'IL:MAKEFILE-ENVIRONMENT))
                        (PKGFORM (AND MFE (GETF MFE :PACKAGE)))
                        (PKG (AND PKGFORM (FIND-PACKAGE (EVAL PKGFORM))))
                        (*PACKAGE-FOR-RESULT-FILE* (IF PKG
                                                    (PACKAGE-NAME PKG)
                                                    *PACKAGE-FOR-RESULT-FILE*)))
              (MAKE-FILE-CONTEXT :FILENAME NAME :DIRNAME DIR-AND-HOST :MENU MENU :OPTIONS
                                (LIST *TRANSLATOR-OPTIONS* (MAPCAR 'SYMBOL-VALUE *TRANSLATOR-OPTIONS*))))))
        (IL:FM.CHANGELABEL (IL:FM.GETITEM 'FILENAME NIL MENU)
                           NAME MENU)
        (IL:FM.CHANGELABEL (IL:FM.GETITEM 'DIRNAME NIL MENU)
                           DIR-AND-HOST MENU)
        (PUSH CONTEXT *FILE-CONTEXTS*)
        (IL:WINDOWPROP MENU 'FILE-CONTEXT CONTEXT)
        (IL:WINDOWPROP MENU 'IL:ICONFN 'FS-WINDOW-ICONFN)
        (IL:WINDOWADDPROP MENU 'IL:CLOSEFN #'(LAMBDA (W)
                                                (LET ((FC (IL:WINDOWPROP W 'FILE-CONTEXT)))
                                                    (IL:UNINTERRUPTABLY
                                                     (SETQ *FILE-CONTEXTS* (DELETE (
                                                                                   FILE-CONTEXT-FILENAME
                                                                                   FC)
                                                                                   *FILE-CONTEXTS* :TEST
                                                                                   'EQ :KEY
                                                                                   '
                                                                                   FILE-CONTEXT-FILENAME
                                                                                   )))
                                                    (SETF (FILE-CONTEXT-MENU FC)
                                                          NIL))
                                                    (IL:WINDOWPROP W 'FILE-CONTEXT NIL))))
        (IL:MOVEW MENU (IL:GETBOXPOSITION (IL:WINDOWPROP MENU 'IL:WIDTH)
                                           (IL:WINDOWPROP MENU 'IL:HEIGHT)))
        (IL:OPENW MENU)
        CONTEXT)))

```

```

(DEFUN NTH-SUBEXPRESSION (LOCATOR EXP)
  (DOLIST (N LOCATOR)
    (SETQ EXP (NTH N EXP)))
  EXP)

```

```

(DEFUN OPTIONS-EDITOR-ACCEPT (ITEM WINDOW BUTTONS)
  (DECLARE (IGNORE ITEM BUTTONS))
  (LET ((CONTEXT (IL:WINDOWPROP (IL:WINDOWPROP WINDOW 'IL:MAINWINDOW)
                                'FILE-CONTEXT)))
    ;; Go thru and check out any fields with validation fns.
    (DOLIST (ID *TRANSLATOR-OPTIONS*)
      (LET* ((ITEM (IL:FM.GETITEM ID NIL WINDOW))

```

```
(VALFN (AND ITEM (IL:FM.ITEMPROP ITEM 'VALIDATION-FN)))
(WHEN VALFN
  (LET ((VALUE (ECASE (IL:FM.ITEMPROP ITEM 'TYPE)
                     (IL:STATE (IL:FM.ITEMPROP ITEM 'IL:STATE))
                     (IL:EDIT (IL:FM.ITEMPROP ITEM 'IL:LABEL)))))
    (MULTIPLE-VALUE-BIND (VALID CORRECTED)
      (FUNCALL VALFN VALUE)
      (UNLESS VALID
        (FORMAT (IL:GETPROMPTWINDOW WINDOW 1)
          "Bad value ~a." VALUE)
        (IL:FM.EDITITEM ITEM WINDOW)
        (RETURN-FROM OPTIONS-EDITOR-ACCEPT NIL))
      (ECASE (IL:FM.ITEMPROP ITEM 'TYPE)
        (IL:STATE (IL:FM.ITEMPROP ITEM 'IL:STATE CORRECTED))
        (IL:EDIT (IL:FM.ITEMPROP ITEM 'IL:LABEL CORRECTED)))))))
```

;; Set values now...

```
(DOLIST (ID *TRANSLATOR-OPTIONS*)
  (LET* ((ITEM (OR (IL:FM.GETITEM ID NIL WINDOW)
                  (IL:FM.NWAYPROP WINDOW ID 'IL:STATE)
                  (ERROR "No item or NWAY collection named ~s." ID)))
    (VALUE (ECASE (IL:FM.ITEMPROP ITEM 'TYPE)
                 (IL:NWAY (IL:FM.ITEMPROP ITEM 'IL:ID))
                 (IL:STATE (IL:FM.ITEMPROP ITEM 'IL:STATE))
                 (IL:EDIT (IL:FM.ITEMPROP ITEM 'IL:LABEL)))))
    (IF CONTEXT
      (SETF (NTH (POSITION ID (FIRST (FILE-CONTEXT-OPTIONS CONTEXT)))
                (SECOND (FILE-CONTEXT-OPTIONS CONTEXT)))
        VALUE)
      ;; If no file context, set the default value. The ID is the same as the variable name.
      (SET ID VALUE))))
(IL:CLOSEW WINDOW)
```

```
(DEFUN OPTIONS-EDITOR-BROWSE-TEMPLATES (ITEM WINDOW BUTTON)
  (LET ((ITEMS (MAPCAR #'(LAMBDA (X)
                        (IL:CREATE IL:TABLEITEM
                                  IL:TIDATA IL:_ (CAR X))
                        *USER-ADDED-TEMPLATES*))
    (BW (IL:WINDOWPROP WINDOW 'TEMPLATE-BROWSER))
    (IF BW
      (IL:TB.REPLACE.ITEMS (IL:WINDOWPROP BW 'IL:TABLEBROWSER)
        ITEMS)
      (IL:WINDOWADDPROP (MAKE-WARNINGS-BROWSER ITEMS "CL Macro Templates" WINDOW 'IL:RIGHT
        #'(LAMBDA (BROWSER ITEM WINDOW)
          (PRINC (IL:FETCH IL:TIDATA IL:OF ITEM)
            WINDOW))
        :SELECTEDFN
        'TEMPLATE-BROWSER-SELECTEDFN :MAINWINDOWPROP 'TEMPLATE-BROWSER)
      'IL:CLOSEFN
      'TEMPLATE-BROWSER-CLOSEFN T))))
```

```
(DEFUN OPTIONS-EDITOR-BROWSE-TRANSLATOR-MACROS (ITEM WINDOW BUTTON)
  (LET ((ITEMS (MAPCAR #'(LAMBDA (X)
                        (IL:CREATE IL:TABLEITEM
                                  IL:TIDATA IL:_ X))
                        *USER-ADDED-TRANSLATOR-MACROS*))
    (BW (IL:WINDOWPROP WINDOW 'TRANSMACRO-BROWSER))
    (IF BW
      (IL:TB.REPLACE.ITEMS (IL:WINDOWPROP BW 'IL:TABLEBROWSER)
        ITEMS)
      (IL:WINDOWADDPROP (MAKE-WARNINGS-BROWSER ITEMS "IL Translator Macros" WINDOW 'IL:RIGHT
        #'(LAMBDA (BROWSER ITEM WINDOW)
          (PRINC (IL:FETCH IL:TIDATA IL:OF ITEM)
            WINDOW))
        :SELECTEDFN
        'TRANSMACRO-BROWSER-SELECTEDFN :MAINWINDOWPROP 'TRANSMACRO-BROWSER)
      'IL:CLOSEFN
      'TRANSMACRO-BROWSER-CLOSEFN T))))
```

```
(DEFUN OPTIONS-EDITOR-CLOSEW (ITEM WINDOW BUTTON)
  (DECLARE (IGNORE ITEM BUTTON))
  (IL:CLOSEW WINDOW))
```

```
(DEFUN OPTIONS-EDITOR-NEW-TEMPLATE (ITEM WINDOW BUTTON)
  (ENTER-NEW-MACRO-TEMPLATE NIL))
```

```
(DEFUN OPTIONS-EDITOR-NEW-TRANSLATOR-MACRO (ITEM WINDOW BUTTON)
  ;; Is the completion function really doing this the right way?!
  (LET ((TEMPLATE (LIST 'IL-DEFCONV SEDIT::BASIC-GAP SEDIT::ARGS-GAP SEDIT::BODY-GAP))
    (EDITOR-NAME "Translator Macro"))
```

```
(*PACKAGE* (FIND-PACKAGE "IL-CONVERT"))
(*READTABLE* (IL:\\GTREADTABLE "XCL" T))
(FLET ((COMPLETION-OF-TRANSLATOR-MACRO-EDIT (SEEDIT-CONTEXT STRUCTURE)
      (UNLESS (SOME #'(LAMBDA (M)
                    (TYPEP M 'SEEDIT::GAP))
                STRUCTURE)
          (PUSH (SECOND STRUCTURE)
                *USER-ADDED-TRANSLATOR-MACROS*)
          (EVAL STRUCTURE))))
      (SEEDIT:SEEDIT TEMPLATE `(:NAME ,EDITOR-NAME :TYPE :MACRO-TEMPLATE :COMPLETION-FN
                                ,#'COMPLETION-OF-TRANSLATOR-MACRO-EDIT))))
```

```
(DEFUN OPTIONS-EDITOR-REVERT (ITEM WINDOW BUTTONS)
  (DECLARE (IGNORE ITEM BUTTONS))
  (LET* ((CONTEXT (IL:WINDOWPROP (IL:WINDOWPROP WINDOW 'IL:MAINWINDOW)
                                 'FILE-CONTEXT)))
    (DOLIST (ID *TRANSLATOR-OPTIONS*)
      (LET* ((VALUE (GET-FILE-CONTEXT-OPTION ID CONTEXT))
             (ITEM (OR (IL:FM.GETITEM ID NIL WINDOW)
                      (IL:FM.NWAYPROP WINDOW ID 'IL:INITSTATE)
                      (ERROR "No item or NWAY collection named ~s." ID))))
        (WHEN ITEM
          (ECASE (IL:FM.ITEMPROP ITEM 'TYPE)
                ((IL:NWAY (IL:FM.CHANGESTATE ID (IL:FM.GETITEM VALUE NIL WINDOW)
                                                WINDOW))
                 (IL:STATE
                  (IL:FM.ITEMPROP ITEM 'IL:STATE VALUE)
                  ;; Yeah, yucky code duplication...
                  (ECASE (IL:FM.ITEMPROP ITEM 'IL:CHANGESTATE)
                        (YESNOCHANGESTATE (IL:FM.CHANGELABEL ITEM (IF VALUE
                                                                    "Yes"
                                                                    "No")
                                                                    WINDOW))))
                 (IL:EDIT (IL:FM.CHANGELABEL ITEM VALUE WINDOW))))))))
```

```
(DEFUN PACKAGE-VALIDATIONFN (NAME)
  (LET ((PKG (FIND-PACKAGE NAME)))
    (IF PKG
        (VALUES T (PACKAGE-NAME PKG))
        (VALUES NIL NIL))))
```

```
(DEFUN PARSE-FILENAME-FOR-FILE-CONTEXT (FILENAME)
  (LET* ((PLIST (IL:UNPACKFILENAME.STRING FILENAME))
        (HOST (GETF PLIST 'IL:HOST))
        (DIR (GETF PLIST 'IL:DIRECTORY))
        (NAME (GETF PLIST 'IL:NAME))
        (TYPE (GETF PLIST 'TYPE))
        (VERSION (GETF PLIST 'IL:VERSION))
        (DIRSTRING (CONCATENATE 'STRING (AND HOST "{")
                                HOST
                                (AND HOST "}")
                                (AND DIR "<")
                                DIR
                                (AND DIR ">"))))
        (NAMESTRING (IL:MKATOM (NSTRING-UPCASE (CONCATENATE 'STRING NAME (AND TYPE ".")
                                                            TYPE
                                                            (AND VERSION ";"
                                                            VERSION))))))
    (LIST NAMESTRING DIRSTRING)))
```

```
(DEFUN PROMPT-FOR-FILENAME (WINDOW &OPTIONAL (PROMPT "File name: ")
                                (DEFAULT ""))
  (LET ((PW (IL:GETPROMPTWINDOW WINDOW 1))
        (OLDTTYPROCESS (IL:TTY.PROCESS (IL:THIS.PROCESS)))
        (OLDTTYDISPLAYSTREAM)
        (IL:CLEARW PW)
        (IL:OPENW PW)
        (SETQ OLDTTYDISPLAYSTREAM (IL:TTYDISPLAYSTREAM PW))
        (UNWIND-PROTECT
         (IL:PROMPTFORWARD PROMPT DEFAULT)
         (IL:TTYDISPLAYSTREAM OLDTTYDISPLAYSTREAM)
         (IL:TTY.PROCESS OLDTTYPROCESS)
         (IL:REMOVEPROMPTWINDOW WINDOW))))
```

```
(DEFUN RECORD-WARNING (C)
  ;; This is a handler for the WARNING condition type. We record the warning and then let it be printed.
  (LET* ((STR (WRITE-TO-STRING C :ESCAPE NIL))
        (FC *FILE-CONTEXT*)
        (DN *CURRENT-DEFINITION*)
        (WARNING-RECORD (LIST STR *CURRENT-EXPRESSION* NIL)))
```

```

TEMP)
;; Record the definition name indexed by string...
(LET* ((DEFKEY (CONS DN *CURRENT-DEFINITION-TYPE*))
        (STRINGELT (GETHASH STR (FILE-CONTEXT-WARNINGS FC)))
        (DEFELT (GETHASH DEFKEY (FILE-CONTEXT-WARNINGS FC))))
        (WHEN DN (PUSHNEW DEFKEY *WARNINGS-MADE*))
        (IF (NULL STRINGELT)
            (SETF (GETHASH STR (FILE-CONTEXT-WARNINGS FC))
                  (LIST NIL (LIST DN *CURRENT-DEFINITION-TYPE* NIL NIL)))
            (UNLESS (FIND-IF #'(LAMBDA (X)
                               (AND (EQ DN (FIRST X))
                                     (EQUAL *CURRENT-DEFINITION-TYPE* (SECOND X))))
                    (CDR STRINGELT))
                (PUSH (LIST DN *CURRENT-DEFINITION-TYPE* *CURRENT-EXPRESSION* NIL)
                      (CDDR STRINGELT))))))
        ;; record the expression and warning string indexed by defname.
        (IF (NULL DEFELT)
            (SETF (GETHASH DEFKEY (FILE-CONTEXT-WARNINGS FC))
                  (LIST NIL (LIST STR *CURRENT-EXPRESSION* NIL)))
            (PUSH (LIST STR *CURRENT-EXPRESSION* NIL)
                  (CDR DEFELT))))))
NIL)

(DEFUN TEMPLATE-BROWSER-CLOSEFN (W)
  (LET* ((TB (IL:WINDOWPROP W 'IL:TABLEBROWSER))
         (NDEL (IL:TB.NUMBER.OF.ITEMS TB 'IL:DELETED)))
    (WHEN (PLUSP NDEL)
      (WHEN (MENU-CHOOSE ' ("Yes" T)
                          ("No" NIL))
        "Expunge deleted items?"
        (IL:TB.MAP.DELETED.ITEMS TB #'(LAMBDA (BROWSER ITEM)
                                           (IL:UNINTERRUPTABLY
                                             (SETQ *USER-ADDED-TEMPLATES*
                                                  (DELETE (IL:FETCH IL:TIDATA IL:OF ITEM)
                                                         *USER-ADDED-TEMPLATES* :KEY 'CAR)
                                                  *WALKER-TEMPLATES*
                                                  (DELETE (IL:FETCH IL:TIDATA IL:OF ITEM)
                                                         *WALKER-TEMPLATES* :KEY 'CAR))))))))))

(DEFUN TEMPLATE-BROWSER-SELECTEDFN (BROWSER ITEM WINDOW)
  ;; Edits a dummy template for the macro operator in FORM.
  (LET* ((NAME (IL:FETCH IL:TIDATA IL:OF ITEM))
         (REAL-TEMPLATE (ASSOC NAME *USER-ADDED-TEMPLATES*))
         (VISIBLE-TEMPLATE (CONS 'DEFTEMPLATE REAL-TEMPLATE))
         (EDITOR-NAME (CONCATENATE 'STRING "Macro template" (AND NAME " for ")
                                   (AND NAME (STRING NAME))))
         (FLET ((COMPLETION-OF-TEMPLATE-EDIT (SEDT-CONTEXT STRUCTURE)
              (SETF (CDR REAL-TEMPLATE)
                    (CDDR STRUCTURE))))
            (SEDT:SEDT VISIBLE-TEMPLATE `(:NAME ,EDITOR-NAME :TYPE :MACRO-TEMPLATE :COMPLETION-FN
                                          #'COMPLETION-OF-TEMPLATE-EDIT))))))

(DEFUN TRANSLATION-BROWSER-SELECTEDFN (BROWSER ITEM WINDOW)
  (LET* ((DEF-BODY (IL:FETCH IL:TIDATA IL:OF ITEM))
         (CONTEXT (IL:WINDOWPROP (IL:MAINWINDOW (IL:MAINWINDOW WINDOW))
                                  'FILE-CONTEXT))
         (OPTS (FILE-CONTEXT-OPTIONS CONTEXT))
         (*PACKAGE* (PROGV (CAR OPTS)
                          (CADR OPTS)
                          (FIND-PACKAGE *PACKAGE-FOR-RESULT-FILE*)))
         (*READTABLE* (IL:\\GTREADTABLE "LISP" T)))
    (FLET ((COMPLETION-OF-DEFINITION-EDIT (SEDT-CONTEXT STRUCTURE)
          (UNLESS (EQ STRUCTURE DEF-BODY)
            ;; Root cons changed; bash old one with new contents.
            (SETF (CAR DEF-BODY)
                  (CAR STRUCTURE)
                  (CDR DEF-BODY)
                  (CDR STRUCTURE))))))
      (SEDT:SEDT DEF-BODY `(:TYPE :CONVERSION :COMPLETION-FN #'COMPLETION-OF-DEFINITION-EDIT
                            :ENVIRONMENT ,*DEF-EDITOR-SEDT-ENVIRONMENT* IL:DONTWAIT T)
        NIL)))

(DEFUN TRANSLATOR-EDIT-DEFAULT-OPTIONS (ITEM WINDOW BUTTONS)
  (DECLARE (IGNORE ITEM BUTTONS))
  (LET ((EDWINDOW (IL:FREEMENU *DEFAULT-OPTIONS-FREEMENU-DESCRIPTION* "Default Translator Options")))
    (IL:ATTACHWINDOW EDWINDOW WINDOW 'IL:TOP 'IL:JUSTIFY 'IL:LOCALCLOSE)
    (OPTIONS-EDITOR-REVERT NIL EDWINDOW NIL)))

```

```
(DEFUN TRANSLATOR-NEW-FILE-CONTEXT (ITEM WINDOW BUTTONS)
  (LET* ((STRING (NSTRING-UPCASE (PROMPT-FOR-FILENAME WINDOW)))
         (PLIST (IL:UNPACKFILENAME STRING))
         (FILENAME (GETF PLIST 'IL:NAME))
         (DIR (GETF PLIST 'IL:DIRECTORY))
         (HOST (GETF PLIST 'IL:HOST))
         (DIR-AND-HOST (IL:PACKFILENAME.STRING 'IL:HOST HOST 'IL:DIRECTORY DIR)))
    (TRANSLATOR-NEW-FILE-CONTEXT-INTERNAL FILENAME DIR-AND-HOST)))
```

```
(DEFUN TRANSLATOR-NEW-FILE-CONTEXT-INTERNAL (FILENAME DIR-AND-HOST)
  (LET* ((FILEBITS (AND (GET FILENAME 'IL:FILE)
                        (IL:ADDFILE FILENAME)))
         (FULLNAME (OR (AND DIR-AND-HOST (CONCATENATE 'STRING (STRING DIR-AND-HOST)
                                                       (STRING FILENAME))))
                       (FILEPROP (AND FILEBITS (CDDR FILEBITS))))
         (LOADTYPE (AND FILEPROP (IL:fetch| IL:LOADTYPE IL:of| FILEPROP)))
         (WHEN (OR (NULL FILEBITS)
                  (MEMBER LOADTYPE '(IL:LOADCOMP IL:|Compiled| IL:|compiled| IL:|loadfns|))))
         (UNLESS (MENU-CHOOSE '("Load it 'PROP" T)
                              ("Abort" NIL))
                  (CONCATENATE 'STRING "File " (STRING FILENAME)
                               " is not fully resident")))
         (RETURN-FROM TRANSLATOR-NEW-FILE-CONTEXT-INTERNAL NIL))
    ;; Note, it's STRING not FILENAME here so we get the directory.
    (IL:LOAD FULLNAME 'IL:PROP))
  (NEW-FILE-CONTEXT FILENAME DIR-AND-HOST)))
```

```
(DEFUN TRANSLATOR-NOTE-ADDITIONAL-FILE (FILE &OPTIONAL (CONTEXT *FILE-CONTEXT*))
  (LET* ((BROWSER-WINDOW (FS-WINDOW-ADD'L-FILE-BROWSER *TRANSLATOR-MAIN-MENU*))
         (TB (IL:WINDOWPROP BROWSER-WINDOW 'IL:TABLEBROWSER))
         (PLIST (IL:UNPACKFILENAME FILE))
         (DIR (GETF PLIST 'IL:DIRECTORY))
         (HOST (GETF PLIST 'IL:HOST))
         (NAME (GETF PLIST 'IL:NAME))
         (DIR-AND-HOST (OR (AND (OR DIR HOST)
                               (IL:PACKFILENAME.STRING 'IL:HOST HOST 'IL:DIRECTORY DIR))
                          (FILE-CONTEXT-DIRNAME CONTEXT))))
         (UNLESS (OR (IL:TB.FIND.ITEM TB #'(LAMBDA (IGNORE TI)
                                                (EQ NAME (FIRST (IL:FETCH IL:TIDATA IL:OF TI))))
                    (FIND NAME *FILE-CONTEXTS* :TEST 'EQ :KEY 'FILE-CONTEXT-FILENAME))
                 (LET ((NEWITEM (IL:CREATE IL:TABLEITEM
                                           IL:TIDATA IL:_ (LIST NAME DIR-AND-HOST)
                                           IL:TIUNDELETABLE IL:_ T))
                       (IL:TB.INSERT.ITEM TB NEWITEM)
                       (IL:TB.NORMALIZE.ITEM TB NEWITEM))))))
```

```
(DEFUN TRANSLATOR-READ-SAVED-CONTEXT (ITEM WINDOW BUTTON)
  (LET ((FILENAME (PROMPT-FOR-FILENAME WINDOW)))
    (WITH-OPEN-FILE (INSTREAM FILENAME :IF-DOES-NOT-EXIST NIL)
      (LET* ((*PRINT-BASE* 10)
            (*PRINT-CIRCLE* NIL)
            (*PRINT-ARRAY* T)
            (*PRINT-PRETTY* NIL)
            (*READTABLE* (IL:\GTREADTABLE "XCL" T))
            (*PACKAGE* (FIND-PACKAGE "XCL")))
        (ARGS (READ INSTREAM))
        (CONTEXT (APPLY 'MAKE-FILE-CONTEXT ARGS)))
      ;; Go thru the warnings and re-constitute them...
      (MAPHASH #'(LAMBDA (K V)
                  (WHEN (CONSP K)
                      (LET ((DB (FIRST V)))
                        (WHEN (AND (LISTP DB)
                                   (INTEGERP (FIRST DB)))
                          (SETF (FIRST V)
                                (NTH-SUBEXPRESSION DB (FILE-CONTEXT-DEFINITIONS
                                                       CONTEXT))))
                        (DOLIST (ENTRY (CDR V))
                              (WHEN (AND (LISTP ENTRY)
                                           (INTEGERP (FIRST ENTRY)))
                                  (SETF (WE-EXPRESSION ENTRY)
                                        (NTH-SUBEXPRESSION (WE-EXPRESSION ENTRY)
                                                             DB))))))
                    (FILE-CONTEXT-WARNINGS CONTEXT))))))
```

```
(DEFUN TRANSMACRO-BROWSER-CLOSEFN (W)
  ;; This doesn't delete t.m.s other than IL-DEFCONV ones yet.
  (LET ((TB (IL:WINDOWPROP W 'IL:TABLEBROWSER))
        (WHEN (AND (PLUSP (IL:TB.NUMBER.OF.ITEMS TB 'IL:DELETED))
                    (MENU-CHOOSE '("Yes" T)
                                   ("No" NIL))
```

```

"Expunge deleted items?")
(IL:TB.MAP.DELETED.ITEMS
  TB
  #' (LAMBDA (I)
      (LET ((NAME (IL:FETCH IL:TIDATA IL:OF I)))
          (WHEN (EQ (CAR (IL:GETDEF NAME :FUNCTIONS 'IL:CURRENT ' (IL:NODWIM IL:NOCOPY)))
                    'IL-DEFCONV)
              ;; Actually the above wants to be some kind of MEMBER thing.
              (IL:DELDEF NAME :FUNCTIONS)
              (IL:UNINTERRUPTABLY
                (SETQ *USER-ADDED-TRANSLATOR-MACROS* (DELETE NAME
                                                            *USER-ADDED-TRANSLATOR-MACROS*
                                                            ))))))))

```

```

(DEFUN TRANSMACRO-BROWSER-SELECTEDFN (BROWSER ITEM WINDOW)
  (ED (IL:FETCH IL:TIDATA IL:OF ITEM)
      (LIST :FUNCTIONS :DONTWAIT)))

```

```

(DEFUN UNKNOWN-MACRO-FORM (FORM &AUX (NAME (CAR FORM)))
  (CASE *UNKNOWN-MACRO-ACTION*
    (:UM-SILENT FORM)
    (:UM-WARN
     (LET ((*CURRENT-EXPRESSION* FORM)
           (WARN "Macro form ~s not translated" (FIRST FORM)))
       FORM)
    (:UM-PROMPT (LET ((CHOICE (IL:MENU (IL:CREATE IL:MENU
                                                  IL:ITEMS IL:_ ' ("Enter new template" :ENTER)
                                                  ("Edit this form with SEDIT" :EDIT)
                                                  ("Treat forms with this macro as function
                                                  calls" :FUNCTION)
                                                  ("Don't walk forms with this macro" :QUOTE)
                                                  ("Warn for forms with this macro" :WARN)
                                                  ("Leave this expression alone" :LEAVE))
                                                  IL:TITLE IL:_ (CONCATENATE 'STRING "Unknown macro "
                                                  (STRING NAME)
                                                  " :")))))
                (CASE CHOICE
                  (:QUOTE
                   ;; Enter a template to quote everything...
                   (LET ((NEW (LIST 'REPEAT (LIST NIL))))
                       (PUSH NEW *USER-ADDED-TEMPLATES*)
                       (PUSH NAME *USER-ADDED-TEMPLATES*)
                       (PUSH NEW *WALKER-TEMPLATES*)
                       (PUSH NAME *WALKER-TEMPLATES*)
                       (WALK-TEMPLATE FORM NEW)))
                     (:FUNCTION (LET ((NEW (LIST 'CALL 'REPEAT (LIST 'EVAL))))
                                   (PUSH NEW *USER-ADDED-TEMPLATES*)
                                   (PUSH NAME *USER-ADDED-TEMPLATES*)
                                   (PUSH NEW *WALKER-TEMPLATES*)
                                   (PUSH NAME *WALKER-TEMPLATES*)
                                   (WALK-TEMPLATE FORM NEW)))
                     (:WARN (LET* ((WARNING-STRING (CONCATENATE 'STRING "Unknown macro " (STRING NAME)))
                                   (NEW (LIST 'WARN WARNING-STRING)))
                              (PUSH NEW *USER-ADDED-TEMPLATES*)
                              (PUSH NAME *USER-ADDED-TEMPLATES*)
                              (PUSH NEW *WALKER-TEMPLATES*)
                              (PUSH NAME *WALKER-TEMPLATES*)
                              (WALK-TEMPLATE FORM NEW)))
                   (:LEAVE FORM)
                   (:EDIT (SEDT:SEDT FORM NIL NIL))
                   (:ENTER
                    (ENTER-NEW-MACRO-TEMPLATE NAME)
                    (WALK-FORM-INTERNAL FORM))))))

```

```

(DEFUN WALKER-FIND-PARAMETER-LIST (DEFINITION)
  (LET ((*WALKER-FIND-PARAMETER-LIST* T)
        (*CURRENT-FUNCTION-CALLS* NIL))
    (CATCH 'PARAMETER-LIST
      (WALK-FORM-INTERNAL DEFINITION)
      NIL))

```

```

(DEFUN WARN-FOR-PARM-CHANGES (OLD NEW NAME CONTEXT)
  ;; Perhaps we should go over ALL contexts?
  (LET (CALLERS)
    (DOLIST (F (FILE-CONTEXT-FUNCTION-CALLS CONTEXT))
      (WHEN (MEMBER NAME (CDR F))
        :TEST
        'EQ)
      (PUSH (CAR F)
            CALLERS)))

```


;; For each caller, hunt down each place that the victim is used and make a warning for it.

```
(LABELS ((FIND-IT (DEF CALLER VICTIM CONTEXT)
  (LABELS ((FIND-IT-INTERNAL
    (EXPRS CALLER VICTIM CONTEXT DEF)
    (FLET ((ADD-WARNING-TO (CX CALLER EXPR STRING)
      (LET ((WI (FIND CALLER (FILE-CONTEXT-WARNINGS CX)
        :TEST
        'EQ :KEY 'CAAR)))
        (IF WI
          (PUSH (LIST STRING EXPR NIL)
            (CDR WI))
          (PUSH (CONS (LIST CALLER "Function" DEF NIL)
            (LIST (LIST STRING EXPR NIL)))
            (FILE-CONTEXT-WARNINGS CX))))))
    (DOLIST (E EXPRS)
      (WHEN (LISTP E)
        (WHEN (EQ (CAR E)
          VICTIM)
          (ADD-WARNING-TO CONTEXT CALLER E (CONCATENATE 'STRING
            "Parameters of "
            (STRING VICTIM)
            " changed."))
          DEF))
        (WHEN (CDR E)
          (FIND-IT-INTERNAL (CDR E)
            CALLER VICTIM CONTEXT DEF))))))
    (COND
      ((MEMBER (CAR DEF)
        ' (EVAL-WHEN))
        (DOLIST (D (CDDR DEF))
          (FIND-IT D CALLER VICTIM)))
      ((EQ (SECOND DEF)
        CALLER)
        (FIND-IT-INTERNAL (CDDR DEF)
          CALLER VICTIM CONTEXT DEF))))))
  (DOLIST (C CALLERS)
    (DOLIST (D (FILE-CONTEXT-DEFINITIONS CONTEXT))
      (FIND-IT D C NAME CONTEXT))))))
```

(DEFUN **WARNING-DEFINITIONS-BROWSER-SELECTEDFN** (BROWSER ITEM WINDOW)

```
(LET* ((ENTRY (IL:FETCH IL:TIDATA IL:OF ITEM))
  (DEF-NAME (FIRST ENTRY))
  (DEF-TYPE (SECOND ENTRY))
  (KEY (CONS DEF-NAME DEF-TYPE))
  (CONTEXT (IL:WINDOWPROP (IL:MAINWINDOW (IL:MAINWINDOW WINDOW))
    'FILE-CONTEXT))
  (DEF-ENTRY (GETHASH KEY (FILE-CONTEXT-WARNINGS CONTEXT)))
  (DEF-BODY (FIRST DEF-ENTRY))
  (WARNINGS (REST DEF-ENTRY))
  (OPTS (FILE-CONTEXT-OPTIONS CONTEXT))
  (*PACKAGE* (PROGV (CAR OPTS)
    (CADR OPTS)
    (FIND-PACKAGE *PACKAGE-FOR-RESULT-FILE*)))
  (*READTABLE* (IL:\\GTREADTABLE "LISP" T))
  (EDITOR-NAME (CONCATENATE 'STRING "Converted " (STRING DEF-TYPE)
    " "
    (STRING DEF-NAME))))
  (OLD-PARM-LIST (COPY-LIST (WALKER-FIND-PARAMETER-LIST DEF-BODY)))
  (FLET ((COMPLETION-OF-DEFINITION-EDIT (SEEDIT-CONTEXT STRUCTURE)
    (UNLESS (EQ STRUCTURE DEF-BODY)
      ;; Root cons changed; bash old one with new contents.
      (SETF (CAR DEF-BODY)
        (CAR STRUCTURE)
        (CDR DEF-BODY)
        (CDR STRUCTURE))
      ;; Compare parameter-lists at this point.
      (WHEN OLD-PARM-LIST
        (LET ((NEW-PARM-LIST (WALKER-FIND-PARAMETER-LIST STRUCTURE))
          (COMPARE-PARAMETER-LISTS OLD-PARM-LIST NEW-PARM-LIST DEF-NAME CONTEXT))))))
  )
  (LET* ((WB (DEF-EDITOR-BROWSE-WARNINGS (IL:fetch| SEEDIT::DISPLAY-WINDOW
    IL:of| (SEEDIT:SEEDIT DEF-BODY
      `(:NAME ,EDITOR-NAME :TYPE :CONVERSION
        :COMPLETION-FN
        #'COMPLETION-OF-DEFINITION-EDIT
        :ENVIRONMENT
        , *DEF-EDITOR-SEEDIT-ENVIRONMENT*
        IL:DONTWAIT T)
      NIL))
    (B&I (IL:WINDOWPROP WINDOW 'BROWSER-AND-ITEM)))
    (IL:WINDOWPROP WB 'BROWSER-AND-ITEM (CONS (CONS BROWSER ITEM)
      B&I))))))
```

```

(DEFUN WB.BUTTONEVENTFN (IL:WINDOW)
  (IL:TOTOPW IL:WINDOW)
  (LET (IL:FN)
    (COND
      ((IL:INSIDEP (IL:DSPCLIPPINGREGION NIL IL:WINDOW)
        (IL:LASTMOUSEX IL:WINDOW)
        (IL:LASTMOUSEY IL:WINDOW))
       (IL:TB.DO.UNLESS.BUSY IL:WINDOW (IL:FUNCTION WB.DO.ITEM.SELECTION)))
      ((IL:LASTMOUSESTATE (IL:ONLY IL:RIGHT))
       (IL:DOWINDOWCOM IL:WINDOW))
      ((AND (IL:LASTMOUSESTATE (OR IL:LEFT IL:MIDDLE))
        (IL:SETQ IL:FN (IL:|fetch| (IL:TABLEBROWSER IL:TBITLEEVENTFN) IL:|of| (IL:WINDOWPROP IL:WINDOW
          'IL:TABLEBROWSER))))
       (IL:TB.DO.UNLESS.BUSY IL:WINDOW IL:FN))))))

(DEFUN WB.CLOSEFN (W)
  (IL:WINDOWPROP (IL:WINDOWPROP W 'IL:MAINWINDOW)
    (OR (IL:WINDOWPROP W 'MAINWINDOWPROP)
      'WARNINGS-BROWSER)
    NIL)
  (IL:WINDOWDELPROP W 'IL:CLOSEFN 'WB.CLOSEFN))

(DEFUN WB.DO.ITEM.SELECTION (WINDOW BROWSER)
  (DECLARE (SPECIAL IL:PROMPTWINDOW))
  (LET ((SELECTIONREGION (IL:DSPCLIPPINGREGION NIL WINDOW))
    THISITEM LASTITEM LASTITEMBOTTOM LASTY LASTBUTTON)
    (FLET ((INVERT-LASTITEM NIL (IL:BLTSHADE IL:BLACKSHADE WINDOW IL:TB.LEFT.MARGIN LASTITEMBOTTOM
      (IL:WINDOWPROP WINDOW 'IL:WIDTH)
      (IL:|fetch| IL:TBITEMHEIGHT IL:|of| BROWSER)
      'IL:INVERT)))
      (LOOP (IL:GETMOUSESTATE)
        (COND
          ((NOT (IL:INSIDEP SELECTIONREGION (IL:LASTMOUSEX WINDOW)
            (SETQ LASTY (IL:LASTMOUSEY WINDOW))))
            (WHEN LASTITEM
              ;; Erase old highlight
              (INVERT-LASTITEM)
              (SETQ LASTITEM NIL))
              ;; I would like to just return here and let the next window take over, but current mouse arrangement means I'll never get
              ;; control back unless user lets up on mouse
              (COND
                ((IL:LASTMOUSESTATE IL:UP)
                  (RETURN))
                (T (IL:BLOCK))))
            ((IL:LASTMOUSESTATE IL:UP)
              ;; Button released inside window...
              (WHEN LASTITEM
                ;; Erase old highlight
                (INVERT-LASTITEM))
              (WHEN LASTBUTTON
                ;; Left: do edit. Middle: toggle deleted. Right: menu of things.
                (CASE LASTBUTTON
                  (IL:LEFT (LET ((FN (GETF (IL:FETCH IL:TBUSERDATA IL:OF BROWSER)
                    'IL:SELECTEDFN)))
                    (WHEN FN (FUNCALL FN BROWSER LASTITEM WINDOW))))
                  (IL:MIDDLE
                    (IF (IL:|fetch| IL:TIDELETED IL:|of| LASTITEM)
                      (IL:TB.UNDELETE.ITEM BROWSER LASTITEM)
                      (IL:TB.DELETE.ITEM BROWSER LASTITEM))
                    (LET ((FN (GETF (IL:FETCH IL:TBUSERDATA IL:OF BROWSER)
                      'IL:DELETEDFN)))
                      (WHEN FN (FUNCALL FN BROWSER LASTITEM WINDOW))))
                  (IL:RIGHT (LET ((FN (GETF (IL:|fetch| IL:TBUSERDATA IL:|of| BROWSER)
                    'IL:RIGHTBUTTONFN)))
                    (WHEN FN (FUNCALL FN BROWSER LASTITEM WINDOW))))))
              (RETURN))
            (T (SETQ THISITEM (IL:TB.ITEM.FROM.YCOORD BROWSER LASTY))
              (UNLESS (EQ THISITEM LASTITEM)
                (WHEN LASTITEM
                  ;; Erase old highlight...
                  (INVERT-LASTITEM))
                (SETQ LASTITEM THISITEM LASTITEMBOTTOM (AND THISITEM (IL:TB.BOTTOM.OF.ITEM BROWSER
                  THISITEM)))
                (WHEN THISITEM
                  ;; Highlight new item...
                  (INVERT-LASTITEM))))))

```

```
(COND
  ((IL:LASTMOUSESTATE IL:LEFT)
   (SETQ LASTBUTTON 'IL:LEFT))
  ((IL:LASTMOUSESTATE IL:MIDDLE)
   (SETQ LASTBUTTON 'IL:MIDDLE))
  ((IL:LASTMOUSESTATE IL:RIGHT)
   (SETQ LASTBUTTON 'IL:RIGHT))))))
```

```
(DEFMACRO WITH-FILE-CONTEXT-OPTIONS (FC &BODY BODY)
  `(XCL:HANDLER-BIND ((XCL:WARNING 'RECORD-WARNING))
    (PROGV (FIRST (FILE-CONTEXT-OPTIONS ,FC))
            (SECOND (FILE-CONTEXT-OPTIONS ,FC))
            ,@BODY)))
```

```
(DEFUN YESNOCHANGESTATE (ITEM WINDOW BUTTON)
  (LET ((NEWSTATE (NOT (IL:FM.ITEMPROP ITEM 'IL:STATE))))
    (IL:FM.ITEMPROP ITEM 'IL:STATE NEWSTATE)
    (IL:FM.CHANGLABEL ITEM (IF NEWSTATE
                                "Yes"
                                "No")
                          WINDOW NIL)))
```

```
(DEFVAR *TRANSLATOR-MAIN-MENU* (IL:FREEMENU *TRANSLATOR-CONTROL-FREEMENU-DESCRIPTION*))
```

```
(IL:TOTOPW *TRANSLATOR-MAIN-MENU*)
```

```
(IL:PUTPROPS IL:TRANSLATOR-ASSISTANT IL:COPYRIGHT ("ENVOS Corporation" 1989 1990))
```

FUNCTION INDEX

ADD\`L-FILE-SELECTEDFN	4	NTH-SUBEXPRESSION	11
COMPARE-PARAMETER-LISTS	4	OPTIONS-EDITOR-ACCEPT	11
CONVERT-FILE-FOR-CONTEXT	4	OPTIONS-EDITOR-BROWSE-TEMPLATES	12
CONVERT-ONE-FILECOM-FOR-CONTEXT	4	OPTIONS-EDITOR-BROWSE-TRANSLATOR-MACROS	12
CONVERT-SETFS-FILECOM	4	OPTIONS-EDITOR-CLOSEW	12
CONVERT-UNKNOWN-COM	5	OPTIONS-EDITOR-NEW-TEMPLATE	12
DEF-EDITOR-BROWSE-WARNINGS	5	OPTIONS-EDITOR-NEW-TRANSLATOR-MACRO	12
DEF-EDITOR-CONVERT-STRUCTURE	5	OPTIONS-EDITOR-REVERT	13
DEF-WARNING-BROWSER-DELETEDFN	5	PACKAGE-VALIDATIONFN	13
DEF-WARNING-BROWSER-SELECTEDFN	6	PARSE-FILENAME-FOR-FILE-CONTEXT	13
ENTER-NEW-MACRO-TEMPLATE	6	PROMPT-FOR-FILENAME	13
FILE-WARNING-BROWSER-DELETEDFN	6	RECORD-WARNING	13
FILE-WARNING-BROWSER-SELECTEDFN	6	TEMPLATE-BROWSER-CLOSEFN	14
FS-WINDOW-ADD\`L-FILE-BROWSER	6	TEMPLATE-BROWSER-SELECTEDFN	14
FS-WINDOW-BROWSE-TRANSLATIONS	7	TRANSLATION-BROWSER-SELECTEDFN	14
FS-WINDOW-BROWSE-WARNINGS	7	TRANSLATOR-EDIT-DEFAULT-OPTIONS	14
FS-WINDOW-DO-TRANSLATE	7	TRANSLATOR-NEW-FILE-CONTEXT	15
FS-WINDOW-EDIT-OPTIONS	8	TRANSLATOR-NEW-FILE-CONTEXT-INTERNAL	15
FS-WINDOW-ICONFN	8	TRANSLATOR-NOTE-ADDITIONAL-FILE	15
FS-WINDOW-SAVE-CONTEXT	8	TRANSLATOR-READ-MAKED-CONTEXT	15
FS-WINDOW-SELECT-NEXT-DEFINITION	8	TRANSMACRO-BROWSER-CLOSEFN	15
FS-WINDOW-WRITE-TRANSLATION	9	TRANSMACRO-BROWSER-SELECTEDFN	16
FS-WINDOW-WRITE-TRANSLATION-INTERNAL	9	UNKNOWN-MACRO-FORM	16
INVERT-CALLER-LIST	10	WALKER-FIND-PARAMETER-LIST	16
LOCATE-SUBEXPRESSION	10	WARN-FOR-PARM-CHANGES	16
MAKE-WARNINGS-BROWSER	10	WARNING-DEFINITIONS-BROWSER-SELECTEDFN	17
MAP-INTO-CONTEXT	10	WB.BUTONEVENTFN	18
MAPCAN-INTO-CONTEXT	11	WB.CLOSEFN	18
MENU-CHOOSE	11	WB.DO.ITEM.SELECTION	18
NEW-FILE-CONTEXT	11	YESNOCHANGESTATE	19

VARIABLE INDEX

DEF-EDITOR-ADDED-SEEDIT-COMMANDS	1	*OPTIONS-FREEMENU-DESCRIPTION*	2
DEF-EDITOR-SEEDIT-COMTAB	1	*TRANSLATOR-CONTROL-FREEMENU-DESCRIPTION*	3
DEF-EDITOR-SEEDIT-COMTAB-SPEC	1	*TRANSLATOR-MAIN-MENU*	19
DEF-EDITOR-SEEDIT-ENVIRONMENT	1	*TRANSLATOR-OPTIONS*	3
DEFAULT-OPTIONS-FREEMENU-DESCRIPTION	1	*USER-ADDED-TEMPLATES*	3
FILE-CONTEXT-FREEMENU-DESCRIPTION	3	*USER-ADDED-TRANSLATOR-MACROS*	3
FILE-CONTEXTS	3	*WARN-FOR-COMPLEX-I.S.-FORMS*	3
ITERATION-OPERS	2		

STRUCTURE INDEX

FILE-CONTEXT	3	WARNING-ENTRY	3	WARNING-HEADER	3
--------------	---	---------------	---	----------------	---

MACRO INDEX

GET-FILE-CONTEXT-OPTION	10	WITH-FILE-CONTEXT-OPTIONS	19
-------------------------	----	---------------------------	----